

Processamento de Linguagem Natural

Conceitos, Técnicas e Aplicações em Português

Organizado por:
Helena de Medeiros Caseli
Maria das Graças Volpe Nunes

1ª Edição



<https://brasileiraspln.com/livro-pln/1a-edicao/>

Conteúdo

Sobre este livro	1
Prefácio	3
Como ler este livro	5
I Introdução	7
1 O que é PLN?	8
1.1 Introdução	8
1.2 A língua	11
1.3 Os Paradigmas de PLN	12
1.4 Vale a pena lembrar	13
II Fala	14
2 Texto ou fala?	15
2.1 Histórico e panorama da área	15
2.2 Aspectos teóricos fundamentais	17
2.2.1 A estrutura da língua falada	18
2.2.2 Reconhecimento de fala	27
2.2.3 Síntese de fala	38
2.2.4 Considerações finais	40
3 Recursos para o processamento de fala	42
3.1 Introdução	42
3.2 Recursos para síntese de fala	45
3.3 Recursos para segmentação prosódica	48
3.4 Recursos para reconhecimento automático de fala	51
3.5 Recursos para Reconhecimento de Emoções	53
3.6 Recursos para predição de pontuação no cenário de ASR	56
3.6.1 Ilustrando o uso do ASR Whisper na predição da pontuação em português	57
3.6.2 Descrição do Corpus MuPe e seu <i>dataset</i> de teste	59
3.6.3 <i>Corpora</i> usados nos trabalhos de Predição da Pontuação para ASRs	60
3.7 Considerações finais	63
Agradecimentos	63



III	Palavras	64
4	Sequência de caracteres e palavras	65
4.1	Conceitos básicos da morfologia	67
4.1.1	Morfema	67
4.1.2	<i>Token</i> e <i>Type</i>	70
4.1.3	Lexema, Lexia e Lema	70
4.1.4	Léxico e Gramática	71
4.1.5	Léxico comum e Léxico especializado	72
4.1.6	Palavras funcionais e palavras lexicais	73
4.1.7	Processos de formação das palavras	73
4.1.8	Morfologia e morfossintaxe	74
4.2	O processamento morfológico em PLN	74
4.2.1	Sentencição	75
4.2.2	Tokenização	76
4.2.3	Tokenização em Subpalavras	78
4.2.4	Normalização	79
4.2.5	PoS <i>tagging</i>	80
4.2.6	Anotação de atributos morfológicos	82
4.3	Ferramentas e recursos para o processamento morfológico	83
4.3.1	Ferramentas para o português	83
4.3.2	Recursos para o português	86
4.3.3	PortiLexicon-UD: um recurso para processamento léxico em português	88
4.4	Considerações finais	94
5	Expressões multipalavras	96
5.1	<i>Cenas dos próximos capítulos...</i>	96
IV	Estrutura	98
6	A ordem e a função das palavras em uma sentença	99
6.1	Introdução	99
6.2	Reflexões Iniciais	101
6.3	Noções Básicas de Sintaxe	102
6.4	Tipos de representação	104
6.4.1	Colchetes	104
6.4.2	Árvores	105
6.4.3	Setas	106
6.4.4	Parênteses	106
6.4.5	Indentação	107
6.5	Sintaxe de constituição	107
6.5.1	Possibilidades de interpretação e ambiguidades sintáticas	109
6.6	Sintaxe de dependência	111
6.6.1	Núcleo e dependente	112
6.6.2	A representação da sintaxe de dependência	113
6.6.3	Projetos de anotação multilingue: Universal Dependencies	116
6.7	Qual é melhor: constituição ou dependência?	120
6.8	Fronteiras da sintaxe	121



6.8.1	Sintaxe e Morfologia	121
6.8.2	Sintaxe e Semântica	123
6.8.3	Sintaxe e Pragmática	123
6.8.4	Sintaxe e Discurso	124
6.9	Considerações finais	124
7	Ferramentas e recursos para o processamento sintático	126
7.1	Introdução	126
7.2	Tipos de <i>parsing</i>	126
7.3	Recursos e ferramentas para o português	128
7.3.1	<i>Corpora</i>	128
7.3.2	<i>Parsers</i>	130
7.4	Visualização, anotação e edição de <i>treebanks</i>	135
7.4.1	Árvores de constituição	135
7.4.2	Árvores de dependência	136
7.4.3	Anotação de <i>corpus</i> em múltiplos níveis	137
7.5	Considerações Finais	138
V	Significado	141
8	E o significado?	142
9	Semântica com Técnicas Simbólicas	148
9.1	Bases de Conhecimento Semântico	150
9.1.1	<i>Wordnets</i>	151
9.1.2	FrameNet	155
9.1.3	ConceptNet	160
9.2	Considerações Finais	166
10	Semântica Distribucional	168
10.1	Semântica Vetorial	170
10.1.1	Matriz termo-documento	171
10.1.2	Matriz termo-contexto	172
10.1.3	Calculando a similaridade entre vetores semânticos	173
10.2	Vetores esparsos	174
10.2.1	Atribuindo pesos aos termos da matriz termo-documento com TF-IDF	175
10.2.2	Atribuindo pesos aos termos da matriz termo-contexto com PMI . .	176
10.2.3	Reduzindo a dimensionalidade com LSA	177
10.3	Vetores densos estáticos	180
10.3.1	Word2Vec	181
10.3.2	Fasttext	184
10.3.3	Glove	186
10.4	Considerações Finais	188
VI	Discurso	190
11	Modelos discursivos	191



11.1	Introdução	191
11.2	Modelos de relações discursivas	193
11.2.1	O modelo RST (Mann e Thompson, 1988)	196
11.2.2	CST: Cross-document Structure Theory, 2000	198
11.3	Recursos e aplicações para o português brasileiro	201
11.4	Considerações Finais	202
12	Resolução de Correferência	205
12.1	Introdução	205
12.2	Resolução de Correferência	205
12.2.1	Referentes	206
12.2.2	Relações Semânticas Referenciais	208
12.2.3	Correferência, Anáfora e Catáfora	209
12.2.4	Referências Endofóricas e Exofóricas	211
12.2.5	Correferência, coerência e coesão	211
12.3	Abordagens Computacionais para Resolução de Correferência	211
12.3.1	Modelos Baseados em Regras	212
12.3.2	Modelos Baseados em Aprendizado de Máquina	212
12.3.3	Entity-Mention	215
12.3.4	Mention-Ranking	216
12.3.5	Antecedent-Tree	217
12.3.6	Modelos Voltados à Língua Portuguesa	218
12.4	Avaliação da Tarefa de Resolução de Correferência	220
12.4.1	Métricas de Avaliação	220
12.5	Aplicações	223
13	Pragmática	225
VII	Dados e Modelos	226
14	<i>Dataset e corpus</i>	227
14.1	Introdução	227
14.1.1	<i>Dataset</i> ou <i>corpus</i> anotado?	228
14.1.2	Anotação linguística	230
14.2	<i>Datasets</i> pra quê?	231
14.2.1	Sobre a importância dos dados	234
14.3	Características de um bom <i>dataset</i> linguístico	235
14.4	Por onde começar?	236
14.4.1	Definição do problema ou tarefa	237
14.4.2	Conjunto de etiquetas e instruções: o esquema de anotação	237
14.4.3	Escolha do <i>corpus</i>	239
14.4.4	Codificação	239
14.4.5	Anotação: sabedoria de especialistas ou sabedoria da multidão?	241
14.4.6	Ferramentas de anotação	244
14.4.7	Estratégias de anotação	245
14.4.8	Formas de avaliação	245
14.5	Procedimentos e estratégias de anotação e revisão	245



14.5.1	Palavras, regras e padrões linguísticos na construção de um <i>corpus</i> padrão ouro	246
14.5.2	Revisão de anotação	250
14.5.3	Até onde precisamos rever?	250
14.6	Como avaliar a qualidade do <i>dataset</i> ?	251
14.6.1	Concordância entre-anotadores	252
14.6.2	Avaliação intrínseca	254
14.6.3	Avaliação extrínseca	256
14.7	Em resumo...	259
	Agradecimentos	259
15	Modelos de Linguagem	260
15.1	Relembrando a Hipótese Semântica e Definindo Modelos	260
15.2	Modelos de Linguagem Probabilísticos	262
15.2.1	Estimando as probabilidades a partir de <i>corpora</i>	263
15.2.2	Usando o modelo probabilístico	264
15.3	Modelos de Linguagem Neurais	265
15.3.1	Um Contexto Histórico	265
15.3.2	Modelos de Linguagem Neurais Modernos	266
15.3.3	Treinamento e Ajustes em Modelos de Linguagem Neurais	291
15.3.4	Avaliação de Modelos de Linguagem Neurais	298
15.4	Tendências	300
15.4.1	A Era dos <i>Large Language Models</i> (LLMs)	300
15.4.2	Treinamento Eficiente de Modelos de Linguagem Neurais	305
15.4.3	Estratégias de Treinamento para Agentes de Conversação: alinhamento e feedback humano	307
VIII	Aplicações	310
16	Recuperação de Informação	311
16.1	Introdução	311
16.1.1	Relação com o PLN	311
16.1.2	O Foco da Recuperação de Informação	311
16.1.3	O Conceito de Relevância	312
16.1.4	Principais Livros	312
16.1.5	Organização deste Capítulo	313
16.2	Visão Geral de um Sistema Típico de Recuperação de Informação	313
16.2.1	Pré-Processamento	313
16.2.2	Indexação	316
16.3	Modelos Clássicos de recuperação de informação	317
16.3.1	Modelo Booleano	317
16.3.2	Modelo Vetorial	317
16.3.3	Modelos Probabilísticos	321
16.4	Avaliação da Qualidade de Sistemas de recuperação de informação	323
16.4.1	Métricas	323
16.4.2	Coleções de Teste	327
16.5	Modificação Automática de Consultas	330



16.6 Ferramentas e Bibliotecas	331
16.6.1 Sistemas Comerciais	331
16.6.2 Ferramentas para Pesquisa	332
16.7 Conclusão	332
Agradecimentos	332
17 Extração de Informação	333
17.1 Introdução	333
17.2 Um pouco de história	334
17.3 Conceituação formal: Relação e Entidade	336
17.3.1 Entidade	336
17.3.2 Relação	336
17.4 Extração de Informação (EI)	337
17.4.1 Reconhecimento de Entidades Nomeadas	337
17.4.2 Extração de Relações	338
17.4.3 Extração Conjunta de Entidades e Relações	339
17.4.4 Métodos empregados para EI na literatura	339
17.5 Extração de Informação Aberta	343
17.5.1 Formalização	344
17.5.2 Abordagens	344
17.6 Avaliação	346
17.7 Considerações finais	348
Agradecimentos	348
18 Tradução Automática	349
18.1 Introdução	349
18.2 Abordagens	350
18.2.1 Tradução direta	350
18.2.2 Tradução Automática Baseada em Regras	351
18.2.3 Tradução por interlíngua	352
18.2.4 Tradução Automática Baseada em Exemplos	353
18.2.5 Tradução Automática Estatística	354
18.2.6 Tradução Automática Neural	357
18.3 Avaliação da Tradução Automática	361
18.3.1 O que é Avaliação da Tradução Automática?	361
18.3.2 A importância de uma avaliação replicável	364
18.3.3 Métricas Automáticas	365
18.3.4 Métricas Humanas	367
18.3.5 Avaliação dependente de contexto	371
18.4 O Futuro da Tradução Automática	373
19 Correção automática de redação	375
19.1 Introdução	375
19.1.1 O que é uma redação escolar?	376
19.1.2 O que é avaliado?	377
19.1.3 Alguns modelos brasileiros de correção	378
19.2 Detecção de desvios no texto	380
19.2.1 Tipos de desvios	381



19.2.2	Formalismos de regras	383
19.3	Atribuição de nota	386
19.3.1	Como atribuir nota a redações?	386
19.3.2	Atribuição de nota para redações do Enem	388
19.4	Feedback para o aluno	389
19.4.1	Estatísticas básicas do texto	390
19.4.2	Assistentes de escrita e ferramentas de auxílio à escrita	393
19.4.3	Identificação de pontos fortes e elogiáveis	394
19.5	Correção manual vs(?) correção automática	396
19.5.1	Avanços dos últimos anos	396
19.5.2	Vantagens da correção automática	397
19.5.3	Vantagens da correção manual	398
19.5.4	O exemplo da língua inglesa	398
19.5.5	O que defendemos	399
19.6	Considerações finais	399
20	ChatGPT, MariTalk e outros agentes de conversação	401
20.1	Introdução	401
20.2	Os Jogos de Linguagem	405
20.3	Jogos que os agentes parecem jogar bem	407
20.3.1	Sumarização	407
20.3.2	Criação de história	408
20.3.3	Tradução automática	411
20.3.4	Escrever e-mails	414
20.4	Jogos que os agentes parecem jogar mal	417
20.4.1	Simplificação textual	417
20.4.2	Perguntas sobre conteúdo pouco popular na internet	418
20.4.3	Conte uma piada	421
20.4.4	Inferências: um lance válido no jogo?	421
20.5	Tarefas em que os agentes jogam um jogo perigoso	421
20.6	Conclusões provisórias	426
IX	Domínios	429
21	PLN na Saúde	430
21.1	Introdução	430
21.2	O texto livre em narrativas clínicas	432
21.3	Aplicações de PLN na Saúde	434
21.3.1	Predição	434
21.3.2	Desidentificação	434
21.3.3	Extração de conceitos clínicos	435
21.3.4	Relações temporais	436
21.3.5	Sumarização	437
21.4	Para onde estamos caminhando?	437
22	PLN no Direito	439
22.1	Introdução	439
22.2	O Direito – uma moldura para a significação	440



22.3	Entre as terminologias e as palavras no Direito do Brasil	442
22.4	Um caso concreto: em pequeníssima escala	443
22.5	Outros casos/exemplos: Direito Ambiental	445
22.6	Aplicação: Análise de Sentimentos em Direito: desafios e exemplos	446
22.7	Considerações finais	451
23	PLN em Redes Sociais	453
23.1	Introdução	453
23.2	Redes Sociais	453
23.2.1	Facebook	454
23.2.2	Reddit	454
23.2.3	Youtube	455
23.2.4	Twitter/X	455
23.2.5	Whatsapp	455
23.2.6	Instagram	455
23.3	Áreas de Aplicação	456
23.3.1	Detecção de Discurso de Ódio e Linguagem Ofensiva	456
23.3.2	Análise de Sentimento	460
23.3.3	Detecção de Notícias Falsas	462
23.3.4	Detecção de Ironia/Sarcasmo/Humor	463
23.4	Considerações Finais	464
X	Sociedade	466
24	Questões éticas em IA e PLN	467
24.1	Ética em IA	467
24.2	Ética em PLN	472
24.3	Modelos de língua como fonte de conhecimento?	473
25	E agora, PLN?	475
25.1	Desafios e perspectivas para o PLN-Português	475
25.2	Há limites para o PLN?	477
	Referências	479
	Apêndice 1 (Capítulo 2)	536
	Alguns pressupostos: estatística, probabilidade, teoria da informação	536
	Probabilidade	536
	Variáveis aleatórias	537
	Média e Variância	538
	Covariância e Correlação	538
	Vetores aleatórios e distribuições multivariadas	539
	Algumas distribuições úteis	539
	Teoria da Estimção e Teste de Significância	540
	Teoria da Informação, Entropia e Informação Mútua	541
	Sobre as/os autoras/es	543
	Helena de Medeiros Caseli	543



Conteúdo

Maria das Graças Volpe Nunes	543
Adriana Pagano	543
Aline Aver Vanin	543
Aline Macohin	544
Aline Paes	544
Aline Villavicencio	544
Amanda Rassi	544
Ana Clara Souza Pagano	544
Arnaldo Candido Junior	545
Brenda Salenave Santana	545
Camila Azevedo	545
Carlos Ramisch	545
Cláudia Freitas	545
Claudia Moro	546
Daniela Barreiro Claro	546
Daniela Vianna	546
Edresson Casanova	546
Elisa Terumi Rubel Schneider	546
Eloize Seno	546
Evandro Fonseca	547
Flaviane R. Fernandes Svartman	547
Heliana Mello	547
Jackson Wilke da Cruz Souza	547
Jessica Rodrigues	547
Joaquim Santos	548
Laila Mota	548
Larissa Freitas	548
Leidiana Iza Andrade Freitas	548
Lilian Mie Mukai Cintho	548
Lucelene Lopes	548
Maria José Bocorny Finatto	549
Mariza Ferro	549
Marli Quadros Leite	549
Marlo Souza	549
Paula Christina Figueira Cardoso	549
Priscilla de Abreu Lopes	550
Priscila Osório Côrtes	550
Renata Ramisch	550
Renata Vieira	550
Ricardo Marcacini	550
Roana Rodrigues	550
Sandra Maria Aluísio	551
Sheila Castilho	551
Solange Rezende	551
Tayane Soares	551
Valéria de Paiva	552
Vinícius G. Santos	552
Viviane P. Moreira	552



Conteúdo

Vlória Pinheiro	552
Yohan Bonescki Gumiel	552

 <https://brasileiraspln.com/livro-pln/1a-edicao/>



Sobre este livro

Um dos grandes desafios de escrever um livro sobre uma área tecnológica, como o PLN (Processamento de Linguagem Natural), é fazê-lo de tal forma que ele não fique obsoleto pouco tempo depois de seu lançamento. E esse risco é, de fato, muito grande. Cada vez mais as tecnologias não apenas avançam rapidamente como podem ser totalmente substituídas por outras, tornando-se descartáveis. Ainda assim, há sempre um núcleo de conhecimento que se mantém relevante ao longo do tempo e que é indispensável para quem quer conhecer ou atuar nessa área. Por outro lado, por ser uma área tecnológica, é indispensável que o livro também apresente o que há de mais atual.

Este livro aborda conhecimentos, técnicas e aplicações de PLN em português que são relevantes atualmente ou historicamente. Afinal, para chegar até aqui, um longo caminho de construção de conhecimento foi percorrido – ainda que possa parecer que cada nova tecnologia seja sempre uma nova invenção.

Para contemplar história, estado da arte e não obsolescência, optou-se por criar um livro na forma digital, de acesso gratuito e dinâmico, ou seja, assume-se o compromisso de atualizá-lo sempre que for necessário. Mais do que apenas atualizá-lo com novos conhecimentos, o livro deve ser ampliado, com o tempo, com temas que, nesta primeira edição, ficaram de fora. As possibilidades de expansão são inúmeras, já que há uma demanda crescente por novas aplicações que envolvem PLN.

Nesta primeira edição, procuramos introduzir os principais conceitos e tarefas da área, algumas das aplicações mais relevantes hoje em dia, particularmente em português do Brasil, e algumas das técnicas usadas para o desenvolvimento dessas tarefas e aplicações. Os temas de todos os capítulos têm potencial para serem aprofundados em edições futuras deste livro. Outros temas, ainda que muito centrais à área, ficaram para uma segunda edição. Entre eles, destacamos os formalismos e as técnicas de análise sintática (*parsing*), o tratamento de expressões multpalavras, formalismos de representação semântica e pragmática. Novos capítulos com aplicações em diferentes domínios devem ser inseridos no futuro próximo.

A fim de abranger temas variados nessa área, foi importante contar com um grande número de autores especialistas. Eles incluem experientes professores universitários das principais universidades brasileiras e também do exterior, pesquisadoras brasileiras que atuam no exterior, alunos de pós-graduação em Computação e em Linguística, profissionais de PLN que atuam em empresas. A variedade de autoria e a forma colaborativa de construção de conteúdos, sob diferentes visões (computacional e linguística), dão a este livro uma característica única: uma personalidade. E essa personalidade, que tanto nos orgulha, faz deste livro uma história do PLN escrita a muitas mãos.

Com este livro, as autoras e os autores pretendem oferecer um corpo de conhecimento suficientemente abrangente sobre o PLN em português do Brasil – ainda que não esgote todos os temas da área – que atenda expectativas de leitores com diferentes perfis. O livro pode ser usado como material didático para disciplinas de cursos de Computação ou Linguística, ou áreas afins, servindo como fonte de conhecimento teórico e prático para quem está aprendendo essa ciência. Pode servir de referência a profissionais que fazem uso do PLN para criar soluções tecnológicas, servindo como fonte de técnicas e métodos para



desenvolvimento de sistemas tecnológicos. Também acadêmicos e curiosos, de um modo geral, nele encontrarão uma visão geral da área de PLN, e como ela pode interagir com outras áreas de conhecimento.

Agradecemos o apoio da NAACL RAF 2021, uma iniciativa da NAACL¹ (*North American Chapter of the Association for Computational Linguistics*) para apoiar projetos da comunidade de PLN na América Latina.

¹<https://naacl.org/>



Prefácio

O Processamento de Linguagem Natural (PLN) surgiu praticamente ao mesmo tempo que os computadores, por volta da década de 1940, já que a tradução automática entre línguas foi um dos primeiros problemas submetidos aos primeiros computadores². No Brasil, as pesquisas em PLN começaram timidamente ainda na década de 1970, entre acadêmicos interessados em Inteligência Artificial (IA). Em 1984, foi realizada a primeira edição do Simpósio Brasileiro de Inteligência Artificial (SBIA)³, em Porto Alegre, RS, e boa parte dos trabalhos apresentados nesse evento eram da área de PLN. Naquela época, os sistemas propostos para português eram muito simples, sendo apenas estudos de caso bem elementares, como os que visavam encontrar respostas para perguntas do tipo “Qual?”, “O quê?”, “Onde?” e “Quando?”. Demorou mais uma década até que houvesse uma massa crítica de cientistas brasileiros dedicados ao processamento computacional do português.

No início da década de 1990, destacavam-se pesquisadores de PLN em diversos centros acadêmicos, como cientistas da computação e linguistas da UFRGS, PUCRS e da Unisinos, no RS; linguistas da PUC-Rio, no RJ; cientistas da computação na Unicamp, em SP, e na UFPE, em PE. Também nessa época, em 1993, nascia o NILC (Núcleo Interinstitucional de Linguística Computacional), no ICMC-USP/São Carlos⁴, um grupo pioneiro em PLN, que teve como desafio inicial produzir o primeiro revisor ortográfico e gramatical comercial para o português do Brasil. Para levar a cabo essa iniciativa, foi necessário compor uma equipe de informatas e linguistas que se tornaram agentes multiplicadores do PLN no Brasil.

Em 1993 foi realizado o primeiro evento exclusivo de PLN dedicado ao português ibérico e brasileiro, em Lisboa. Tal evento se tornou o *International Conference on the Computational Processing of Portuguese* (PROPOR)⁵, realizado alternadamente no Brasil e em Portugal, com mais de 15 edições realizadas até essa data (2023). No âmbito nacional, em 2003 foi criado o Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)⁶, que abriga diferentes eventos satélites tratando de temas específicos para o processamento da língua, em particular o Português, com 14 edições realizadas até essa data (2023). Em 2007, foi criada a Comissão Especial de PLN (CEPLN)⁷ da Sociedade Brasileira de Computação. A comunidade cresceu, inicialmente graças à formação acadêmica de profissionais e pesquisadores, posteriormente também em consequência da maior demanda de empresas tecnológicas e de outras naturezas, atraindo profissionais com diferentes formações. Em 2020, foi criado o grupo Brasileiras em PLN (BPLN)⁸, responsável pela edição deste livro.

Já há algum tempo as aplicações de PLN saíram dos laboratórios acadêmicos e entraram definitivamente no nosso cotidiano. O desenvolvimento de diversas áreas da computação de

²Os primeiros computadores também foram massivamente usados para realizar cálculos numéricos complexos, mas a tradução automática foi a primeira aplicação simbólica da computação.

³<http://comissoes.sbc.org.br/ce-ia/pg/historico/?sec=SBIA-1984>

⁴<http://www.nilc.icmc.usp.br/>

⁵<https://sites.google.com/view/ce-pln/eventos/propor>

⁶<https://sites.google.com/view/ce-pln/eventos/stil>

⁷<https://sites.google.com/view/ce-pln/inicio>

⁸<https://brasileiraspln.com/>



forma integrada (hardware, software, tratamento de grandes volumes de dados, aprendizado de máquina e aprendizado profundo) impulsionou o desenvolvimento do processamento da língua para uma nova esfera. Em novembro de 2022, o mundo conheceu o ChatGPT, um *chatbot* capaz de responder a qualquer pergunta ou solicitação (escrevendo códigos de programas, p. ex.), em língua natural, incluindo o português. Além de surpreender pelo seu desempenho linguístico, ele acendeu um sinal de alerta para a comunidade de IA e PLN, bem como para vários setores da sociedade. Várias indagações surgiram: Será que a IA está prestes a atingir a meta de criar sistemas que possam vir a substituir o homem em todas as tarefas? Quais os riscos que as diferentes sociedades correm se o mercado de trabalho for dominado por sistemas inteligentes? E se apenas poucas empresas dominarem o mercado de sistemas inteligentes? E o que pode acontecer com países não desenvolvidos? A conscientização desse cenário fez crescer o interesse pelas atividades da comunidade que desenvolve sistemas de IA, incluindo a de PLN, já que o ChatGPT é uma aplicação típica dessa área.

Durante todas essas décadas, a comunidade brasileira de PLN dedicou-se a construir recursos linguístico-computacionais e também aplicações variadas para o português. No início, eram interfaces simples que entendiam perguntas feitas em linguagem natural. Seguiram-se ferramentas de auxílio à escrita, tradutores, sumarizadores, classificadores de documentos, reconhecedores e sintetizadores de fala, entre outros. Atualmente, muitas aplicações envolvem os conteúdos das redes sociais, sendo que a língua tem sido processada também com o objetivo de conhecer a pessoa que faz uso dela, seja para lhe oferecer produtos e serviços, seja para analisar tendências e personalidades. Entram em cena os reconhecedores de opiniões e sentimentos, e de notícias falsas. Essa nova tendência de descobrir o que está implícito ou pode ser inferido da linguagem trouxe problemas muito desafiadores para a área. Vários domínios de conhecimento e atuação estão fortemente presentes nessa revolução tecnológica: a educação, a saúde, o direito, bancos e financeiras, comunicação, comércio, segurança, governança, entre outros. Tanto por serem afetados por ela como por ajudarem a construir novas soluções.

Se antes os especialistas em PLN tinham a tarefa de explicitar e codificar conhecimento linguístico, hoje seu maior desafio é preparar dados/exemplos linguísticos para servirem de entrada aos algoritmos no treinamento de modelos computacionais capazes de transformar dados em conhecimento ou ações. E isso não é pouco. Para enriquecer os dados brutos de modo que o conhecimento adquirido seja o mais completo e correto possível, ou que a ação gerada seja a mais adequada, é preciso saber escolhê-los, prepará-los e eventualmente anotá-los com informações de várias naturezas: morfológicas, sintáticas, semânticas, extralinguísticas etc. É a essa tarefa que muitos especialistas em PLN de todo o mundo têm se dedicado ultimamente. Outro papel importante está nas avaliações de qualidade dos recursos e produtos construídos, o entendimento do problema, a identificação das limitações e a proposta de melhorias. É crucial que as bases do desenvolvimento dessas tecnologias sejam fortes, bem informadas, consistentes e éticas. A linguagem nos define, e ela faz agora parte dos nossos artefatos, a complexidade deste cenário ainda não é bem compreendida. Com este livro, os autores pretendem contribuir para que o desenvolvimento da tecnologia que envolve a linguagem humana seja bem fundamentado, consciente e ético.



Como ler este livro

O objetivo dos autores deste livro é oferecer aos leitores conhecimento sobre Processamento de Linguagem Natural (PLN), ao mesmo tempo, introdutório e abrangente, de modo a atender às expectativas de leitores com diferentes perfis. O livro pode ser usado como material didático para disciplinas de Inteligência Artificial (IA) e PLN de cursos de computação ou linguística, ou áreas afins, e também pode servir de referência a profissionais que fazem uso do PLN para criar soluções tecnológicas.

A fim de oferecer conhecimento sobre temas variados nessa área, foi importante contar com um grande número de especialistas brasileiros⁹, a quem as organizadoras agradecem profundamente.

O objetivo dos autores deste livro é introduzir conceitos e detalhar os principais métodos e técnicas de PLN, além de recursos e aplicações, com foco no português do Brasil. Para tal, ele está dividido em dez partes que contêm 25 capítulos.

A **Parte 1** apresenta a área de PLN no Brasil, no Prefácio, e introduz os principais conceitos no Capítulo 1. É leitura obrigatória para o leitor não familiarizado com PLN.

A **Parte 2 (Fala)**, ao contrário do restante do livro, que trata de processamento de texto escrito, apresenta a área de processamento de fala: seus principais conceitos no Capítulo 2, e técnicas, recursos e aplicações, no Capítulo 3. O leitor, ainda que interessado apenas na síntese ou reconhecimento de fala, se beneficiará dos demais capítulos deste livro para complementar conceitos comuns, como a anotação e o uso de *corpus*, o processamento da fala transcrita em texto, entre outros.

Na **Parte 3 (Palavras)**, a primeira fase do processamento textual é discutida. O Capítulo 4 dedica-se a desvendar a morfologia, que estuda as palavras isoladamente, suas partes (morfemas), seus processos de derivação e composição, bem como partes importantes do processamento automático como a tokenização e a atribuição das categorias das palavras (*part-of-speech tagging*). O Capítulo 5, que será desenvolvido para a próxima edição deste livro, explorará o mundo das expressões multipalavras, que traz grandes desafios à medida que o conhecimento de seus componentes isolados pouco contribui para a aquisição de seu significado.

Crescendo em complexidade, a **Parte 4 (Estrutura)** considera a ordem das palavras numa sentença, buscando extrair seus papéis na organização sintática parcial ou total da sentença. Com tal conhecimento, o processamento da língua alcança um novo patamar, e a partir dele já é possível realizar várias tarefas de PLN, como *parsing* parcial ou total, e viabilizar várias aplicações. O Capítulo 6 fornece toda a conceitualização de sintaxe, os principais tipos de análise, suas diferenças, vantagens e desvantagens. O Capítulo 7 mostra as diferentes ferramentas computacionais para o processamento sintático, em especial, as que são dedicadas ao português. As Partes 3 e 4 são indispensáveis para o leitor estudante ou profissional que pretende atuar na área de PLN, seja pesquisando ou implementando sistemas.

A **Parte 5 (Significado)** promove um salto significativo para a complexidade do PLN: trata dos conceitos, modelos e técnicas relativos à apreensão do sentido implicado pela

⁹A lista completa das autoras e autores deste livro encontra-se em: Sobre as/os autoras/es



língua escrita. Isso pode ocorrer pelo uso de teorias e modelos simbólicos ou não simbólicos. O Capítulo 8 introduz toda a complexidade da semântica da língua. Os Capítulos 9 e 10 mostram as diferentes abordagens (simbólica e estatística, respectivamente) para o tratamento do sentido. A leitura desta parte é indispensável para quem quer ter conhecimento mais amplo sobre a área de PLN.

Questões discursivas e pragmáticas implicadas pelo texto são tratadas na **Parte 6 (Discurso)** deste livro. No Capítulo 11, o leitor encontra os principais modelos discursivos para PLN; no Capítulo 12, um fenômeno muito frequente, e clássico no PLN, é tratado em detalhes: como resolver as correferências discursivas presentes em textos. O Capítulo 13, a ser desenvolvido para a próxima edição, deve tratar das teorias e dos formalismos que visam processar textos sob a ótica pragmática da linguagem natural.

A **Parte 7 (Dados e Modelos)** explora a construção e a anotação de *datasets* ou *corpus*, seu papel no treinamento de algoritmos de aprendizado de máquina (Capítulo 14), bem como os paradigmas estatístico e neural para a geração de modelos de língua (Capítulo 15). É aqui que são apresentadas as técnicas do estado da arte do PLN, como as redes neurais profundas e os Transformers, as técnicas de *fine-tuning* e *prompt engineering*. Essa parte do livro é essencial para o leitor ficar atualizado sobre o que tem sido usado para a criação das principais aplicações de PLN, assim como de toda Inteligência Artificial.

As **Partes 8 (Aplicações) e 9 (Domínios)** apresentam várias aplicações de PLN. Na primeira delas, constam aplicações clássicas, como recuperação (Capítulo 16) e extração (Capítulo 17) de informações, tradução automática (Capítulo 18), correção textual (Capítulo 19), bem como os agora famosos *chatbots* baseados em modelos de língua (Capítulo 20). ChatGPT e MariTalk, dois *chatbots* que combinam um modelo de língua gerativo com um treinamento por reforço humano são analisados em detalhe. Na Parte 9 as aplicações ilustram o PLN aplicado a dois domínios – saúde (Capítulo 21) e direito (Capítulo 22) – e a um gênero específico de texto, aquele das redes sociais (Capítulo 23).

A **Parte 10 (Sociedade)**, antes de finalizar o livro, discute, no Capítulo 24, algumas discussões éticas que a IA, em geral, e o PLN, em particular, têm provocado, pela forma como têm sido criados e usados recentemente. Finalmente, o último capítulo (Capítulo 25) discorre sobre algumas perspectivas para o PLN do português no cenário de 2023.

Complementam esse livro as referências bibliográficas dos capítulos reunidas em Referências, um Glossário¹⁰ de termos, um apêndice do Capítulo 2, além das informações Sobre as/os autoras/es.

¹⁰Disponível em: <https://brasileiraspln.com/livro-pln/glossario/glossario.html>



Parte I

Introdução



Capítulo 1

O que é PLN?

*Helena de Medeiros Caseli
Maria das Graças Volpe Nunes
Adriana Pagano*

1.1 Introdução

O Processamento de Linguagem Natural (PLN) é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana. O adjetivo “Natural”, na sigla, se refere às línguas faladas pelos humanos, distinguindo-as das demais linguagens (matemáticas, visuais, gestuais, de programação etc.). No decorrer deste livro, os termos “língua”, “linguagem humana” e “linguagem natural” serão usados indistintamente; já “linguagem” pode eventualmente se referir a qualquer tipo de linguagem. Na área da Ciência da Computação, PLN está ligado à área de Inteligência Artificial (IA) e também está intrinsecamente relacionada à Linguística Computacional.

Para deixar mais claro o que entendemos por PLN, vamos esclarecer o que se faz nessa área. De modo geral, em PLN buscam-se soluções para problemas computacionais, ou seja, tarefas, sistemas, aplicações ou programas, que requerem o tratamento computacional de uma língua (português, inglês etc.), seja escrita (texto) ou falada (fala). Línguas como as de sinais também têm sido alvo de estudos da área. Cada modo tem suas especificidades. No caso da fala, as características que a distinguem da língua escrita são relacionadas a questões da produção (síntese) e recepção (reconhecimento) do som. Recursos da fala, como a entonação, o volume, o sotaque, podem tanto dificultar o reconhecimento ou a síntese, como também facilitar o reconhecimento de sentimentos ou intenções do falante. Qualquer que seja o modo, fala, escrita, línguas orais e línguas de sinais compartilham a dificuldade maior em PLN: a apreensão do significado de uma expressão linguística. Isso vai ficar claro no decorrer deste livro.

O PLN se divide em duas grandes subáreas: **Interpretação (ou Compreensão) de Linguagem Natural** – NLU (do inglês, *Natural Language Understanding*), e **Geração de Linguagem Natural** – NLG (do inglês, *Natural Language Generation*)¹.

Situa-se em NLU tudo o que diz respeito ao processamento que visa à análise e à interpretação da língua. Por análise, entende-se a segmentação e classificação dos componentes linguísticos (p. ex. palavras e suas classes morfológicas e gramaticais, seus traços semânticos ou ontológicos etc.). Já interpretação se refere à tentativa de apreender significados construídos pelo ser humano. Numa interação com um *chatbot*, por exemplo,

¹Embora algumas siglas, como NLP (*Natural Language Processing*) e AI (*Artificial Intelligence*) tenham sido traduzidas e sejam amplamente utilizadas em português, as siglas NLU e NLG são utilizadas, em textos em português, em sua grafia inglesa.



a interpretação ocorre quando o sistema processa um texto do usuário para descobrir o que ele – o sistema – deve fazer a seguir: se fornecer uma resposta ou executar uma ação. Logo ficará claro que respostas mais ou menos bem-sucedidas do sistema para o significado tencionado pelo humano podem ser suficientes para muitas aplicações, e que o completo alinhamento entre o significado tencionado pelo humano e aquele interpretado pela máquina não deve ser parte das nossas expectativas.

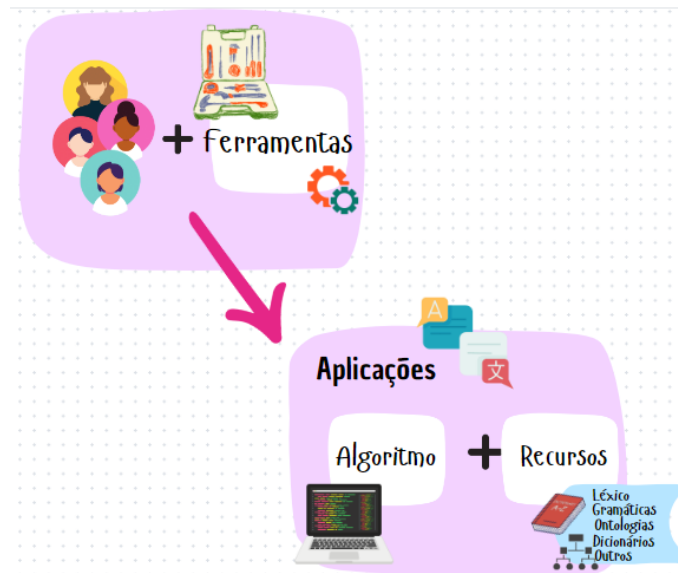
Em NLG, por outro lado, o objetivo é a geração de linguagem natural. Um exemplo de NLG é a geração de respostas ao usuário dos *chatbots*. Para o sistema, isso significa decidir o que responder e como apresentar essa resposta ao usuário. Atualmente, o ChatGPT² é o exemplo de maior sucesso: é capaz de gerar língua de forma tão ou mais fluente quanto muitos humanos.

É importante esclarecer, desde já, alguns conceitos amplamente usados no decorrer deste livro. Eles dizem respeito à classificação de alguns sistemas de PLN quanto ao seu uso.

Esses conceitos são: aplicações, recursos e ferramentas.

Primeiramente, é relevante observar como esses conceitos se relacionam entre si. A Figura 1.1 esquematiza essa dinâmica.

Figura 1.1: Relacionamento entre conceitos



Como vemos na Figura 1.1, em PLN as ferramentas auxiliam na construção de uma aplicação, que pode ser um sistema computacional (desktop, web) ou um aplicativo. As aplicações fornecem um resultado ao usuário tendo uma entrada (*input*) ou saída (*output*) em linguagem natural. Aplicações fazem uso de ferramentas ou conjuntos de ferramentas, conhecidos como “*toolkits*”. Também necessitam recursos, os quais fornecem informações linguísticas necessárias para que as aplicações consigam processar a língua da maneira adequada.

É importante notar que a denominação utilizada – aplicação, recurso ou ferramenta – é imprecisa e depende do uso. Por exemplo, um corretor ortográfico pode ser uma aplicação a ser usada de forma autônoma ou um passo intermediário para uma aplicação de correção de redações; um tradutor automático pode ser uma aplicação em si, com uma

²<https://chat.openai.com/>

interface para colocar um texto de entrada e obter um texto de saída, mas também pode ser usado como ferramenta para traduzir um *corpus* de uma língua para outra, visando a criação de recursos em línguas de comunidades tecnologicamente menos desenvolvidas; um sumário automático pode ser usado para criar resumos para um usuário qualquer, mas também pode ser usado por um buscador da web como passo intermediário para um sistema de recuperação de informação; um dicionário é um recurso, mas também pode ser usado como um aplicativo para consulta; um modelo de língua pode se transformar num *chatbot*, e assim por diante. Os conceitos são caracterizados e exemplificados no Quadro 1.1.

Quadro 1.1. Exemplos de aplicações, recursos e ferramentas

Conceito	Caracterização	Exemplos
Aplicações	processam uma entrada (<i>input</i>) em linguagem natural e a transformam produzindo um determinado resultado	<ul style="list-style-type: none"> - tradutor automático - corretor ortográfico ou gramatical - assistentes virtuais/<i>chatbots</i> - sumário automático - sistemas de recomendação em sites de e-commerce ou entretenimento - sistemas de auxílio à escrita - sistemas de classificação textual - sistemas de recuperação de informação - sistemas de detecção de <i>fake news</i>
Recursos	são fontes de informação linguística para sistemas	<ul style="list-style-type: none"> - léxicos (listas de palavras com informações associadas) da língua em geral ou de terminologia de domínio - dicionários monolíngues ou bilíngues - <i>corpus</i> (<i>datasets</i> linguísticos) anotados manual ou automaticamente (para referência, teste ou treinamento de algoritmos de aprendizado de máquina) - listas de frequências de palavras - (mais estruturados) taxonomias, ontologias, redes de sinônimos e antônimos - em formato matemático: modelos de língua estatísticos (probabilidades) ou neurais (pesos) para informar qual palavra deve ser a próxima num dado contexto
Ferramentas	auxiliam na construção de uma aplicação ou até de outras ferramentas	<ul style="list-style-type: none"> - segmentadores textuais em <i>tokens</i> (tokenizador), sentenças, parágrafos - <i>stemmers</i> (extratores de raiz de uma palavra (e.g. “corr” de “correr”)) - lematizadores (e.g. “correr” de “corri”) - etiquetadores morfossintáticos (PoS <i>taggers</i>) para classe de palavras (verbo, substantivo, adjetivo, artigo, preposição, advérbio etc.) - etiquetadores semânticos, analisadores sintáticos parciais (<i>chunkers</i>) e completos (<i>parsers</i>) - concordanciadores - interfaces de anotação de <i>corpus</i> - componentes em kits de ferramentas (<i>toolkits</i>), como o NLTK³

Neste livro iremos aumentar gradativamente a complexidade do tratamento da língua no PLN, com foco no português brasileiro. Antes de iniciar esta trajetória, a Seção 1.2 apresenta nosso objeto de pesquisa, a língua. Em seguida, a Seção 1.3 introduz os principais paradigmas do PLN, que serão retomados em diversos momentos neste livro. Por fim, a Seção 1.4 destaca os principais pontos apresentados no capítulo.

³<https://www.nltk.org>



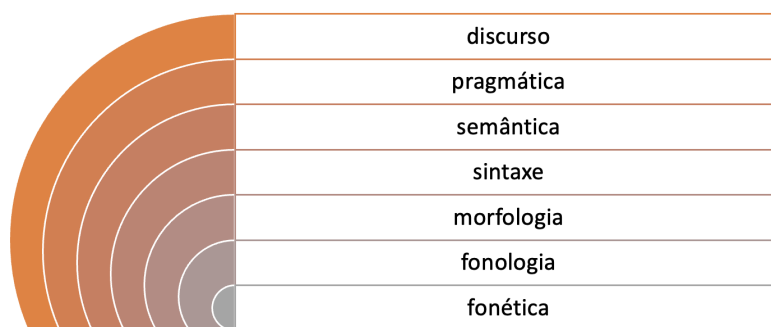
1.2 A língua

A capacidade de usarmos a linguagem para representar nossa realidade e nos comunicar é algo que distingue o ser humano dos outros seres vivos. Poder criar significados, expressar-se e ser compreendida é um dos grandes avanços no desenvolvimento de uma criança. Nos primeiros anos de vida, um bebê vai adquirindo a habilidade de se expressar em sua língua materna. Anos depois, normalmente a criança adquire a capacidade de utilizar símbolos para registrar aquilo que ela deseja por meio da língua escrita. A língua, como um sistema de construção de representações do mundo e comunicação, sobretudo no modo escrito, é o foco deste livro.

Ao longo do livro, nosso foco predominante será a língua escrita⁴, ou seja, sequências de caracteres representados de forma grafológica, os quais constroem significados para nós humanos. Em PLN, chamamos a língua escrita de **texto**, para distingui-la da linguagem oral, que é chamada de **fala**. Portanto, apesar de a linguística reconhecer que existem textos escritos e textos falados, em PLN a palavra **texto** se refere principalmente ao texto escrito. Em relação à língua, neste livro, os exemplos estão em **português brasileiro**, embora muitas das técnicas descritas aqui possam ser aplicadas a outros idiomas.

A linguagem humana organiza-se em diferentes dimensões. A Figura 1.2 mostra uma representação das subáreas que estudam o sistema linguístico.

Figura 1.2: Representação das subáreas de estudo da linguagem



Na Figura 1.2, a língua é representada por meio de círculos concêntricos, sendo cada um deles objeto de estudo de uma subárea dos estudos linguísticos. No núcleo, os sons e sua organização são estudados pela **fonética** e pela **fonologia**. Envolvendo a estrutura sonora, temos o estudo de como os morfemas se organizam para formar palavras, que é objeto de estudo da **morfologia**. Envolvendo a morfologia, temos o estudo de como as palavras se organizam em estruturas para formar sintagmas e orações, objeto de estudo da **sintaxe**. No círculo envolvendo a sintaxe, temos a **semântica**, que estuda o significado de palavras e frases, enquanto a **pragmática** enfoca como as orações são utilizadas na interação para fins comunicativos específicos. Já **discurso** é uma denominação que abrange os estudos com foco no texto como um todo, podendo se referir à análise das relações entre frases ou partes de um texto, ou das etapas na estrutura de um texto.

Cada língua tem suas especificidades que determinam, por exemplo, desde como os caracteres podem ser combinados para compor uma palavra (uma sequência válida que tenha significado naquela língua) até regras que definem a estrutura (sintaxe) dessa língua. No decorrer deste livro, serão abordados os desafios do PLN em cada uma

⁴Com exceção de alguns capítulos que tratam de processamento da língua falada.

dessas subáreas. Contudo, é importante que fique claro que as estratégias computacionais usadas para o processamento da linguagem muitas vezes utilizam conhecimentos de várias subáreas ao mesmo tempo. Por exemplo, no processamento morfosintático realizado por um etiquetador (*tagger*), informações morfológicas e sintáticas são consideradas para se determinar a categoria gramatical (*part-of-speech*, PoS) de uma palavra.

1.3 Os Paradigmas de PLN

Até a década de 1980, o PLN se baseava no que chamamos de **paradigma simbólico**, segundo o qual todo conhecimento sobre a língua é expresso explicitamente em formalismos como léxicos, regras, linguagens lógicas etc., ou seja, formas compreensíveis ao humano. Por exemplo, é possível escrever regras que determinem que, em português, há concordância entre o gênero gramatical atribuído a um substantivo e o gênero atribuído ao adjetivo que o acompanha. Assim, exemplos como “abacaxi maduro” serão considerados corretos de acordo com essa regra, enquanto que outros, como “abacaxi madura”, não.

No início dos anos 1990, as máquinas ganharam mais capacidade de memória e processamento, e diversos algoritmos de aprendizado de máquina foram propostos dando origem ao que chamamos de **paradigma estatístico**. Grandes conjuntos de textos (também chamados de *corpus*) passaram a ser usados como fonte de conhecimento para “ensinar” as máquinas. Por exemplo, fenômenos como a concordância entre substantivo e adjetivo, mencionada anteriormente, passaram a ser aprendidos a partir de exemplos de ocorrência no *corpus* como: “tomate estragado”, “kiwi maduro”, “gergelim preto”. A língua é, então, representada em modelos probabilísticos aprendidos a partir da frequência de ocorrência. Regras explícitas ou implícitas (percursos em árvores, por exemplo) são criadas com base em probabilidades calculadas a partir dos exemplos. Esses modelos são usados para classificar, resumir, traduzir ou gerar novos textos. Uma vez que esses modelos são aprendidos a partir de dados reais, eles têm uma grande chance de serem bons modelos da língua. A tradução automática foi a aplicação de PLN que deu notoriedade a esse paradigma estatístico, que era o mais aplicado até a década de 2010.

O tempo passou e as máquinas continuam ganhando poder de memória e processamento, o que possibilita que grandes quantidades de dados sejam processadas por estruturas (arquiteturas e algoritmos) bastante complexas, como as Redes Neurais Profundas (conhecidas em inglês como *deep learning*). No momento da escrita deste capítulo, o **paradigma neural** é o mais adotado para tarefas de PLN. Da mesma forma que o paradigma estatístico, as redes neurais também se baseiam em grandes volumes de dados para aprender um modelo; contudo, a forma como esse aprendizado é realizado é diferente, uma vez que envolve várias camadas de unidades de processamento para reconhecer padrões recorrentes. Assim, enquanto em outras técnicas de aprendizado de máquina (*machine learning*) tradicional (*shallow* ou baseado em *features*) os algoritmos especificam como o aprendizado deve ocorrer, no *deep learning*, devido à complexidade das arquiteturas compostas por diversas camadas de processamento, não é possível saber exatamente com base em quê o modelo foi aprendido. Além disso, diferentemente do paradigma simbólico, no paradigma neural, o conhecimento da língua é dado por valores numéricos, e não por símbolos ou regras. Dessa forma, o conhecimento linguístico ou a parte do código que tenha produzido um determinado comportamento são praticamente irrecuperáveis, tornando o código opaco, e seu efeito, não previsível (não determinístico).

Nesse sentido, pode-se notar que o PLN tem acompanhado a evolução de paradigmas da IA: simbólico, estatístico e neural. Porém, diante da insuficiência de uma única abordagem,



ganham espaço os **paradigmas híbridos**, que combinam principalmente o simbólico com um dos demais, garantindo, assim, alguma explicitação do conhecimento, conseqüentemente, alguma explicabilidade dos passos seguidos pelos algoritmos.

Além da IA, o PLN tem intersecção com diversos campos de pesquisa e de aplicação no mercado de trabalho como mineração de textos, recuperação de informação e ciência de dados. Na atualidade, todas as aplicações computacionais que processam texto são passíveis de utilizar em maior ou menor grau as técnicas de PLN.

1.4 Vale a pena lembrar

Antes de passarmos para os próximos capítulos deste livro, seguem algumas considerações importantes:

- **Diferentes abordagens podem ser aplicadas no PLN**, desde aquelas associadas ao paradigma mais tradicional (o simbólico) àquelas possibilitadas por paradigmas mais recentes, como o estatístico e o neural.
- **Todas as estratégias automáticas para processamento da língua têm limitações.** Assim, o que define a escolha da melhor estratégia são diversos fatores como: apoio de especialistas (necessário para o paradigma simbólico), poder computacional (um limitante para o paradigma neural) e a disponibilidade de recursos linguísticos em grande quantidade (necessária para as abordagens baseadas em *corpus*).
- **A maioria das estratégias processa caracteres e não unidades linguísticas.** Muitas estratégias geram modelos com base em coocorrência e contexto de ocorrência de palavras e frases, ou seja, são abordagens baseadas em padrões de caracteres. Um modelo neural, por exemplo, não sabe que “casa” pode significar o lugar onde “alguém” mora. Desse modo, podemos dizer que as estratégias usadas na maioria das aplicações do PLN não aprendem a língua, mas apenas aprendem a reproduzir e, às vezes, extrapolar (generalizar) o que aprenderam em um *corpus* de treinamento.
- Muita atenção tem sido dada aos algoritmos de aprendizado de máquina e às arquiteturas neurais, mas **nem tanta atenção assim tem sido dada aos formalismos de representação semântica.** Como vimos, nos diversos anos de pesquisa e desenvolvimento em PLN, as estratégias e abordagens vão e vêm, mas a linguagem natural é muito mais complexa de se aprender e compreender do que uma simples contagem de frequências e coocorrências. Assim, apesar de muito esforço sendo empregado na investigação e evolução de métodos neurais, o conhecimento linguístico e de uso da língua ainda não foi completamente representado/capturado por nenhum dos métodos atuais. Por isso, o processamento completo e adequado da língua só será possível com formalismos de representação híbridos, que incorporam também estruturas semânticas explícitas, e, portanto, mais robustos do que os que são usados hoje por métodos estatísticos e neurais.



Parte II

Fala



Capítulo 2

Texto ou fala?

Camila de Araújo Azevedo
Heliana Ribeiro de Mello
Priscila Osório Côrtes

2.1 Histórico e panorama da área

O processamento da língua falada depende de uma vasta gama de conhecimentos que inclui acústica, fonologia, fonética, linguística geral, semântica, sintaxe, pragmática, estruturas discursivas, entre outras. Para além disso, outros conhecimentos mais comuns à ciência da computação, à engenharia elétrica, à matemática e, até mesmo à psicologia, também são necessários. Neste contexto, este capítulo visa oferecer um panorama da área e das habilidades e métodos mais conhecidos no universo do processamento computacional da língua falada.

Desde os primórdios do surgimento da interação falada na espécie humana até os dias de hoje – e podemos afirmar com tranquilidade, que assim também será no futuro imaginável –, a fala tem sido o principal instrumento para a troca de informações e de coesão social (Rizzolatti; Arbib, 1998). É através da fala¹ que expressamos nossas emoções, a nossa atitude em relação a fatos e eventos, bem como negociamos ideias e ações. A capacidade linguística nos diferencia de outras espécies, mas é a fala, e o que ela nos proporciona, que nos identifica como humanos. Estima-se que a fala tenha surgido na filogênese humana há cerca de 60 mil anos, enquanto a escrita, que é uma tecnologia desenvolvida pelos humanos, surgiu provavelmente há cerca de 10 mil anos. A chamada “dupla articulação” presente na linguagem humana é uma habilidade exclusiva da nossa espécie. Ela se caracteriza por ser a articulação entre unidades significativas (morfemas) e fonemas, que são elementos finitos que se combinam de forma variada, criando infinitas possibilidades de morfemas². A língua falada é hoje expandida para além do domínio da interação face-a-face para meios como a telefonia, a televisão, a interação via computadores. Os aplicativos para interações multimodais imagem/som ganharam uma dimensão inimaginável com a eclosão da pandemia do Sars-Cov-19 em 2020, demonstrando claramente a preferência dos humanos pela interação via fala.

Tal preferência também se reflete na interação homem-máquina e, apesar de ainda estarmos distantes de um mundo em que homens e máquinas interagem majoritariamente através da verbalização oral, já temos aplicações que nos permitem interagir com as

¹Ressaltamos que “fala” neste contexto inclui a comunicação gestual em língua de sinais. As línguas de sinais, como LIBRAS, para o português brasileiro, são línguas naturais, com a mesma riqueza e complexidade que as línguas que se utilizam de sinal sonoro.

²Cf. a discussão sobre Ciências da Fala, em Barbosa, Plínio A. (2020) Ciências da fala. In: *Speech Sciences Entries*. Speech Prosody Studies Group. Disponível em: <https://gepf.falar.org/entries/1>.



máquinas através de comandos orais no contexto doméstico, comercial e computacional.

Em sua fase inicial, o processamento de língua falada em português era bastante limitado devido à falta de recursos computacionais e técnicas apropriadas. As primeiras abordagens eram baseadas em regras gramaticais e modelos acústicos simples. No entanto, com o avanço da tecnologia e o aumento do poder computacional, novas técnicas e abordagens foram desenvolvidas, resultando em avanços significativos nessa área.

A partir da década de 1990, técnicas baseadas em estatística começaram a ganhar popularidade. Esses modelos estatísticos utilizam algoritmos de aprendizado de máquina, como as redes neurais artificiais, para melhorar o desempenho do processamento de língua falada em português. Com a disponibilidade de grandes quantidades de dados de fala e avanços em hardware e software, os sistemas de reconhecimento de fala começaram a se tornar mais precisos e eficientes.

Outro marco importante no processamento de língua falada em português foi a introdução dos sistemas de síntese de fala (Seção 2.2.3). Esses sistemas permitem que um computador gere fala humana a partir de texto escrito em português. Inicialmente, a síntese de fala em português era baseada em técnicas concatenativas, que envolviam a gravação de segmentos de fala de um locutor humano e a concatenação desses segmentos para gerar a fala sintetizada. A concatenação refere-se ao processo de unir ou combinar várias partes ou segmentos de fala para formar uma sequência contínua ou mais longa de palavras ou frases. Com o tempo, surgiram abordagens baseadas em síntese de formantes (na fala, um formante é uma ressonância específica ou pico de intensidade em um espectrograma de som. Os formantes são associados à forma e ao posicionamento da cavidade oral, da faringe e da língua durante a produção de sons da fala, especialmente as vogais) e síntese de fala concatenativa com modelos estatísticos, proporcionando uma qualidade de síntese cada vez melhor.

Avanços mais recentes no processamento da fala em português estão relacionados ao uso de modelos de linguagem neural (Capítulo 15), como os modelos de transformação de sequência a sequência (Seq2Seq) e as redes neurais convolucionais (CNNs) e recorrentes (RNNs). Esses modelos têm oferecido resultados impressionantes em várias tarefas de processamento de língua falada, como reconhecimento automático de fala, tradução automática de fala e resumo automático de áudio.

Além disso, com o advento dos assistentes virtuais e sistemas de processamento de linguagem natural, a interação por meio da fala em português tornou-se cada vez mais comum. Empresas de tecnologia estão investindo em pesquisas e desenvolvimento para melhorar a compreensão e a resposta dos sistemas de processamento de língua falada em português, a fim de proporcionar uma experiência mais natural e intuitiva aos usuários.

Para que se alcancem bons resultados no processamento computacional da fala é preciso que haja *datasets* e *corpora* de fala³ de alta qualidade. Tem havido um esforço considerável da comunidade de pesquisadores para a compilação de dados dessa natureza. Para o português brasileiro, destaca-se o recente *corpus* CORAA ASR v. 1.1 (Corpus de Áudios Anotados)⁴ voltado para tarefas de reconhecimento de fala (Candido Junior et al., 2021), que é apresentado no Capítulo 3.

Os sons da fala podem ser digitalizados e processados usando-se algoritmos tanto para **reconhecimento de fala** (transcrição de formas de onda em texto) quanto para **síntese de fala** (conversão de texto em formas de onda). O processo de digitalização da fala envolve a conversão do sinal analógico das ondas sonoras em um formato digital que pode

³Para saber mais sobre *datasets* e *corpora*, sugere-se a leitura do Capítulo 14.

⁴<https://github.com/nile-nlp/CORAA> e <https://sites.google.com/view/tarsila-c4ai/coraa-versions?pli=1>



ser armazenado e manipulado por um computador. Isso é normalmente feito usando-se um conversor analógico-digital (CAD), que amostra, isto é, faz uma amostragem da onda sonora em intervalos regulares e converte cada amostra em um número binário. Uma vez que o sinal da fala tenha sido digitalizado, ele pode ser processado usando-se várias técnicas, como filtragem, compressão e análise.

Um sistema computacional para a língua falada necessita de capacidades tanto de reconhecimento quanto de síntese de fala. Entretanto, esses dois componentes não são suficientes para a construção de um sistema útil. Um componente de compreensão e diálogo é necessário para a interação com o usuário; o conhecimento de domínio é necessário para guiar a interpretação da fala pelo sistema e permitir que ele determine a ação apropriada. Para todos esses componentes, há uma série de desafios, que incluem robustez, flexibilidade, facilidade de integração e eficiência de engenharia.

2.2 Aspectos teóricos fundamentais

A língua falada é utilizada para diversas funções que se estabelecem entre falantes e ouvintes. A produção e a percepção são ambos elementos importantes na cadeia da fala. A fala se inicia com uma intenção (volição) de comunicação no cérebro do falante, o qual ativa movimentos musculares para a produção de sons. O ouvinte, por sua vez, recebe os sinais sonoros em seu sistema auditivo, processando-os para transformá-los em sinais neurológicos que o cérebro pode compreender. O falante monitora e controla continuamente os órgãos vocais ao receber a sua própria fala como feedback (Moore, 2007).

Considerando os componentes universais da comunicação verbal, a interação falante/ouvinte é tecida a partir de vários elementos distintos. Como dito, o processo de produção da fala começa com a mensagem semântica na mente de uma pessoa a ser transmitida ao ouvinte através da fala. O equivalente computacional ao processo de formulação da mensagem é a semântica da aplicação que cria o conceito a ser expresso. Após a criação da mensagem, o próximo passo é convertê-la em uma sequência de palavras. Cada palavra consiste em uma sequência de fonemas e respectivos alofones (realizações fonéticas correlacionadas do fonema) que correspondem à pronúncia das palavras. Cada frase também contém um padrão prosódico que denota a duração de cada fonema, entonação da frase e volume dos sons. Uma vez que o sistema de linguagem finaliza o mapeamento, o falante executa uma série de sinais neuromusculares. Os comandos neuromusculares realizam o mapeamento articulatório para controlar as cordas vocais, lábios, mandíbula, língua e véu palatino, produzindo assim a sequência sonora como saída final. O processo de compreensão da fala funciona na ordem inversa. Primeiro, o sinal é enviado para a cóclea no ouvido interno, que realiza a análise de frequência como um banco de filtros. Em seguida, um processo de transdução neural converte o sinal espectral em sinais de atividade no nervo auditivo, correspondendo aproximadamente a um componente de extração de recursos. Atualmente, ainda não está claro como a atividade neural é mapeada no sistema de linguagem e como a compreensão da mensagem é alcançada no cérebro.

Os sinais de fala são compostos de padrões sonoros analógicos que servem como base para uma representação discreta e simbólica da linguagem falada – fonemas, sílabas e palavras. A produção e interpretação desses sons são regidas pela sintaxe, semântica e estrutura informacional da língua falada. Neste capítulo, adotamos uma abordagem de baixo para cima para introduzir os conceitos básicos, começando pelos sons e passando pela fonética e fonologia, chegando até as sílabas e palavras.



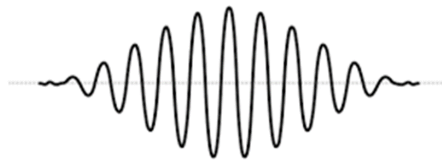
2.2.1 A estrutura da língua falada

Nesta seção, revisamos brevemente os sistemas de produção e percepção de fala humana. Esperamos que, algum dia, a pesquisa em linguagem falada nos permita construir um sistema de computador tão bom quanto o nosso próprio sistema de produção e compreensão de fala.

2.2.1.1 Som

O som é uma onda de pressão longitudinal formada por compressões e rarefações das moléculas de ar, em uma direção paralela àquela da aplicação de energia. Compressões são zonas onde as moléculas de ar foram forçadas pela aplicação de energia a uma configuração mais apertada do que o normal, e rarefações são zonas onde as moléculas de ar estão menos densamente empacotadas. As configurações alternadas de compressão e rarefação de moléculas de ar ao longo do caminho de uma fonte de energia são às vezes descritas pelo gráfico de uma onda senoidal. A forma básica de uma curva senoidal (Figura 2.1) é de uma onda suave, que se repete ao longo de um eixo horizontal. Ela se assemelha a uma série de montanhas e vales, subindo e descendo de forma suave. Neste tipo de representação, as cristas da curva senoidal correspondem a momentos de compressão máxima e os vales correspondem a momentos de rarefação máxima.

Figura 2.1: Curva senoidal



2.2.1.2 Produção de Fala⁵

Aqui revisamos os sistemas básicos de produção de fala humana, que influenciaram a pesquisa em codificação, síntese e reconhecimento de fala.⁶

2.2.1.2.1 Articuladores

A fala é produzida por ondas de pressão de ar que emanam da boca e das narinas de um falante.⁷ Na maioria das línguas do mundo, o inventário de fonemas pode ser dividido em duas classes básicas:

- **Consoantes** – articuladas na presença de constrictões na garganta ou obstruções na boca (língua, dentes, lábios) enquanto falamos;
- **Vogais** – articuladas sem grandes constrictões e obstruções.

Os sons podem ser subdivididos ainda mais em subgrupos com base em certas propriedades articulatórias. Essas propriedades derivam da anatomia de alguns

⁵Recomendamos o PRAAT como software para análise da fala: <https://www.fon.hum.uva.nl/praat/manual/Intro.html>

⁶Toda a seção sobre produção da fala é orientada pelo livro de Huang; Acero; Hon (2001).

⁷Cf. O site Fonética e Fonologia: <https://fonologia.org/>



articuladores importantes e dos locais onde eles tocam as fronteiras do trato vocal humano. Além disso, um grande número de músculos contribui para a posição e o movimento dos articuladores. Nós nos restringimos a apenas uma visão esquemática dos principais articuladores. Os componentes principais do aparelho de produção da fala são os pulmões, traquéia, laringe (órgão de produção de voz), cavidade faríngea (garganta), cavidade oral e nasal. As cavidades faríngea e oral são geralmente referidas como o trato vocal, e a cavidade nasal como o trato nasal. O aparelho de produção de fala humano consiste em:

- **Pulmões:** fonte de ar durante a fala;
- **Cordas vocais (laringe):** quando as pregas vocais são mantidas próximas uma da outra e oscilam uma contra a outra durante um som da fala, o som é categorizado como sonoro. Por exemplo, /b d g/. Quando as pregas são muito soltas ou tensas para vibrar periodicamente, o som é categorizado como surdo. Por exemplo, /p t k/. O local onde as pregas vocais se unem é chamado de glote;
- **Véu palatino (palato mole):** atua como uma válvula, abrindo para permitir a passagem de ar (e, portanto, ressonância) através da cavidade nasal. Sons produzidos com a aba aberta incluem /m/ e /n/;
- **Palato duro:** uma superfície relativamente dura e longa no teto dentro da boca; quando a língua é colocada contra ela, permite a articulação de consoantes, como o λ em alho /a λ u/;
- **Língua:** articulador flexível, afastado do palato para vogais, colocado próximo ou sobre o palato ou outras superfícies duras para articulação de consoantes;
- **Dentes:** outro local de articulação usado para segurar a língua para certas consoantes, como /t d/;
- **Lábios:** podem ser arredondados ou espalhados para afetar a qualidade das vogais, e completamente fechados para interromper o fluxo de ar oral em certas consoantes /p b m/.

2.2.1.2.2 O Mecanismo de Sonorização

A distinção mais fundamental entre os tipos de som na fala é a distinção sonoro/surdo. Sons sonoros, incluindo vogais, têm em sua estrutura temporal e de frequência um padrão regular que sons surdos, como a consoante /s/, não possuem. Sons sonoros geralmente têm mais energia. O que no mecanismo de produção de fala cria essa distinção fundamental? Como já dito na Seção 2.2.1.2.1, quando as pregas vocais vibram durante a articulação do fonema, o fonema é considerado sonoro; caso contrário, é surdo. Vogais são sonoras durante toda a sua duração. Os timbres distintos de vogais são criados usando a língua e os lábios para moldar a principal cavidade de ressonância oral de maneiras diferentes. As pregas vocais vibram em taxas mais lentas ou mais rápidas, desde tão baixas quanto 60 ciclos por segundo (Hz) para um homem de tamanho grande, até 300 Hz ou mais para uma mulher ou criança pequena. A taxa de ciclagem (abertura e fechamento) das pregas vocais na laringe durante a fonação de sons sonoros é chamada de frequência fundamental (f_0). Isso ocorre porque ela estabelece a linha de base periódica para todos os harmônicos de frequência mais alta contribuídos pelas cavidades de ressonância faríngea e oral. A frequência fundamental também contribui mais do que qualquer outro fator único para a percepção de altura (o aumento e queda semelhante à música das tonalidades de voz) na fala.

Uma vez que a onda glotal é periódica, consistindo na frequência fundamental (f_0) e em um número de harmônicos (múltiplos integrais de f_0), ela pode ser analisada como uma soma de ondas senoidais. As ressonâncias do trato vocal (acima da glote) são excitadas



pela energia glotal. Vamos supor, para simplicidade, que o trato vocal seja um tubo reto de área transversal uniforme, fechado na extremidade da glote e aberto nos lábios. Quando a forma do trato vocal muda, as ressonâncias também mudam. Harmônicos próximos às ressonâncias são enfatizados, e, na fala, as ressonâncias das cavidades que são típicas de configurações articulatórias particulares (por exemplo, os diferentes timbres vocálicos) são chamadas de formantes. As vogais em uma forma de onda de fala real podem ser visualizadas a partir de várias perspectivas diferentes, por exemplo, enfatizando uma visão em seção transversal das respostas harmônicas em um único momento ou, por outro lado, uma visão de longo prazo da evolução da trajetória dos formantes ao longo do tempo.

2.2.1.3 Percepção da Fala

Existem dois componentes principais no sistema de percepção auditiva: os órgãos auditivos periféricos (orelhas) e o sistema nervoso auditivo (cérebro). A orelha (ouvido externo) capta um sinal de pressão acústica, processa-o, transformando-o primeiro em um padrão de vibração mecânica na membrana basilar e depois representando o padrão por uma série de pulsos a serem transmitidos pelo nervo auditivo. A informação perceptual é extraída em vários estágios do sistema nervoso auditivo. Nesta seção, focamos principalmente nos órgãos auditivos.

2.2.1.3.1 Fisiologia do Ouvido

O ouvido humano tem três partes: o ouvido externo, o ouvido médio e o ouvido interno. O ouvido externo consiste na parte visível externa e no canal auditivo externo, que forma um tubo ao longo do qual o som viaja. Esse tubo tem cerca de 2,5 cm de comprimento e é coberto pelo tímpano na extremidade distante. Quando variações na pressão do ar alcançam o tímpano do exterior, ele vibra e transmite as vibrações aos ossos adjacentes do seu lado oposto. A vibração do tímpano está na mesma frequência (compressão e rarefação alternadas) que a onda de pressão sonora que chega. O ouvido médio é um espaço ou cavidade cheia de ar com cerca de 1,3 cm de largura e volume de cerca de 6 cm³. O ar viaja pela abertura (quando aberta) que conecta a cavidade com o nariz e a garganta. Há, ainda, a janela oval, que é uma pequena membrana na interface óssea com o ouvido interno (cóclea). Uma vez que as paredes da cóclea são ósseas, a energia é transferida por ação mecânica do estribo para uma impressão na membrana que se estende sobre a janela oval.

A estrutura relevante do ouvido interno para a percepção sonora é a cóclea, que se comunica diretamente com o nervo auditivo, conduzindo uma representação do som para o cérebro. A cóclea é um tubo espiralado de cerca de 3,5 cm de comprimento, que se enrola cerca de 2,6 vezes. A espiral é dividida, principalmente pela membrana basilar que corre longitudinalmente, em duas câmaras preenchidas de líquido. A cóclea pode ser considerada grosseiramente como um banco de filtros, cujas saídas são ordenadas por localização, de modo que uma transformação de frequência local é realizada. Os filtros mais próximos da base da cóclea respondem às frequências mais altas, e aqueles mais próximos do ápice respondem às mais baixas.

Em psicoacústica, faz-se uma distinção básica entre os atributos perceptuais de um som, especialmente de um som de fala, e as propriedades físicas mensuráveis que o caracterizam. Cada um dos atributos perceptuais, conforme listado a seguir, parece ter uma forte correlação com uma propriedade física principal, mas a conexão é complexa, porque outras propriedades físicas do som podem afetar a percepção de maneiras complexas.

O Quadro 2.1 traz a relação entre atributos perceptuais e físicos do som.



Quadro 2.1. Relação entre atributos perceptuais e físicos do som

Quantidade física	Qualidade perceptual
Intensidade	Volume
Frequência fundamental	Tom
Forma espectral	Timbre
Tempo de início/fim	Temporização
Diferença de fase na audição binaural	Localização

Embora sons com níveis de intensidade maiores geralmente soem mais altos, a sensibilidade do ouvido varia com a frequência e a qualidade do som. Uma divergência fundamental entre as qualidades físicas e perceptuais é o fenômeno da percepção de igualdade de intensidade não uniforme de tons de frequências variadas. Em geral, tons de diferentes alturas têm diferentes níveis percebidos de volume. Há uma relativa insensibilidade do ouvido a sons de baixa frequência em níveis de intensidade moderados a baixos. A sensibilidade auditiva atinge um máximo em torno de 4 kHz, que está próximo da primeira frequência de ressonância do canal auditivo externo, e atinge outro pico em torno de 13 kHz, a frequência da segunda ressonância.

A altura está, de fato, mais intimamente relacionada com a frequência fundamental. Quanto maior a frequência fundamental, maior a altura que percebemos. No entanto, a discriminação entre duas alturas depende da frequência da altura inferior. A altura percebida mudará à medida que a intensidade aumentar e a frequência for mantida constante.

Em um exemplo da não identidade de efeitos acústicos e perceptuais, foi observado experimentalmente que, quando o ouvido é exposto a dois ou mais tons diferentes, é comum que um tom possa mascarar os outros. O mascaramento provavelmente é mais bem explicado como um deslocamento ascendente no limiar auditivo do tom mais fraco pelo tom mais alto. Tons puros, sons complexos, bandas estreitas e amplas de ruído mostram diferenças em sua capacidade de mascarar outros sons. Em geral, tons puros, próximos em frequência, se mascaram mais do que tons amplamente separados em frequência. Um tom puro mascara tons de frequência mais alta com mais eficácia do que tons de frequência mais baixa. Quanto maior a intensidade do tom de mascaramento, mais ampla é a faixa de frequências que ele pode mascarar. O mascaramento, no contexto da fala e da audição, pode ter um impacto significativo, causando dificuldade de compreensão e reduzindo a inteligibilidade, além de aumentar o esforço de escuta. O mascaramento pode afetar o reconhecimento automático de fala aumentando a taxa de erros, levando à perda de partes importantes do discurso (perda de contexto) e dificultando a separação de vozes.

A escuta binaural melhora muito nossa capacidade de sentir a direção da fonte de som. A atenção à localização está principalmente focada na discriminação lateral ou de lado a lado. As pistas de tempo e intensidade têm diferentes impactos para frequências baixas e altas, respectivamente. Sons de baixa frequência são lateralizados principalmente com base na diferença interaural de tempo, enquanto sons de alta frequência são localizados principalmente com base na diferença interaural de intensidade.

Finalmente, uma questão perceptual interessante é a questão da qualidade de voz distinta. O discurso de pessoas diferentes soa diferente. Em parte, isso se deve a fatores óbvios, como diferenças na frequência fundamental característica causada, por exemplo, pela maior massa



e comprimento das pregas vocais masculinas adultas em comparação com as femininas. Mas existem efeitos mais sutis também.

Em psicoacústica, o conceito de timbre (de um som ou instrumento) é definido como o atributo da sensação auditiva pelo qual um sujeito pode julgar que dois sons apresentados de maneira semelhante, com a mesma intensidade e altura, são diferentes. Em outras palavras, quando todas as diferenças facilmente mensuráveis são controladas, a percepção restante de diferença é atribuída ao timbre. Isso é mais facilmente ouvido na música, onde a mesma nota na mesma oitava, tocada por igual tempo, por exemplo, em um violino, soa diferente de uma flauta. O timbre de um som depende de muitas variáveis físicas, incluindo a distribuição de energia espectral do som, o envelope temporal, a taxa e profundidade de modulação de amplitude ou frequência e o grau de inarmonia de seus harmônicos.

2.2.1.3.2 Análise de Frequência

Pesquisadores têm realizado trabalhos experimentais psicoacústicos para derivar escalas de frequência que tentam modelar a resposta natural do sistema perceptual humano, uma vez que a cóclea do ouvido interno atua como um analisador de espectro. O complexo mecanismo do ouvido interno e do nervo auditivo implica que os atributos perceptuais de sons em diferentes frequências podem não ser completamente simples ou lineares por natureza. É bem conhecido que a altura musical ocidental é descrita em oitavas e semitons. A altura musical percebida de tons complexos é basicamente proporcional ao logaritmo da frequência. Para tons complexos, a diferença perceptível para frequência é essencialmente constante na escala de oitavas/semitons. As escalas de altura musical são usadas em pesquisas prosódicas (sobre a geração de contorno de entonação da fala).

2.2.1.4 Panorama genérico dos níveis de análise da fala

A fala, diferentemente da escrita, não é uma tecnologia desenvolvida pelos humanos. É algo bem mais complexo e antigo, sendo hoje considerada, por alguns, como uma dotação genética e, por outros, como o produto de diferentes processos cognitivos e corpóreos.

A fala humana pode ser definida genericamente como o processo de expressar pensamentos, ideias e emoções por meio da produção de sons articulados. É uma forma de comunicação específica dos seres humanos e é fundamental para a interação social e o desenvolvimento das sociedades.

A caracterização da fala humana envolve vários aspectos tais como:

- **Produção de sons articulados:** A fala envolve a produção de sons através da coordenação dos órgãos articulatórios, como a língua, os lábios, os dentes e a glote. Esses órgãos são responsáveis por modificar a corrente de ar expirada pelos pulmões para produzir os diferentes sons da fala.
- **Sistema linguístico (linguagem):** A fala é mediada pela linguagem, que é um sistema de símbolos e regras que permite a comunicação entre os indivíduos. A linguagem compreende elementos fonéticos (sons), fonológicos (padrões de som), morfológicos (estrutura das palavras), sintáticos (ordem das palavras), semânticos (significado das palavras) e pragmáticos (uso da linguagem em contextos específicos).
- **Expressão de pensamentos e emoções:** A fala humana permite expressar uma ampla gama de pensamentos, ideias e emoções. Além da transmissão de informações, a fala também é utilizada para expressar sentimentos, intenções, opiniões e experiências pessoais.



- **Comunicação social:** A fala é um meio de interação social fundamental. Por meio da fala, os indivíduos podem se comunicar, compartilhar informações, estabelecer conexões emocionais, resolver problemas e coordenar atividades em grupo.
- **Aquisição:** A habilidade de falar é adquirida ao longo do desenvolvimento humano. As crianças passam por um processo de aprendizado da fala, no qual adquirem as habilidades motoras necessárias para articular os sons e aprendem as regras e estruturas da linguagem de seu ambiente.

É importante ressaltar que a fala humana é altamente diversa e comporta variações entre diferentes idiomas, culturas e indivíduos. Além disso, a fala também pode ser afetada por condições clínicas, como distúrbios da fala e da linguagem.

Diferentemente do que acontece para a escrita, o processamento computacional da fala não parte do encadeamento simbólico de grafemas organizados em itens lexicais e suas supra-estruturas sintáticas. É preciso converter o sinal sonoro em símbolos passíveis de análise por um sistema computacional, ou seja, as ondas sonoras precisam ser convertidas em bits processáveis computacionalmente. Ademais, a fala não pode prescindir de um nível analítico comumente ignorado pelas análises da escrita: a pragmática e, mais especificamente o seu nível prosódico e suas correspondências na estruturação informacional. Neste capítulo não há a possibilidade de explorarmos este assunto com a profundidade que ele merece, portanto recomendamos ao leitor recorrer a leituras específicas para se inteirar sobre isso.

Nas próximas subseções, faremos um apanhado genérico sobre o nível analítico mínimo, o fonético-fonológico.

2.2.1.4.1 Fonética e fonologia

Agora discutiremos as noções de fonética e fonologia básicas necessárias para o processamento da linguagem falada. Fonética refere-se ao estudo dos sons da fala, sua produção, classificação e transcrição. Fonologia é o estudo da distribuição e padrões dos sons da fala em uma língua e das suas regras implícitas.

Ao linguista Ferdinand de Saussure (1857-1913) atribui-se a observação de que a relação entre um sinal e o objeto significado por ele é arbitrária. Assim, um mesmo conceito é arbitrariamente expresso em línguas diferentes: usamos [pe] em português para nos referirmos ao mesmo conceito que em inglês foneticamente seria [fut]. Para a fonética, isso significa que os sons da fala não têm um significado intrínseco e devem ser distribuídos aleatoriamente no léxico.

Os sons são apenas um conjunto de efeitos arbitrários disponibilizados pela anatomia vocal humana. Assim como as impressões digitais, a anatomia vocal de cada falante é única, o que resulta em vocalizações também únicas. No entanto, a comunicação linguística é baseada na comunalidade de formas no nível perceptual. Para permitir a discussão das semelhanças, os pesquisadores identificaram certas características gerais dos sons da fala que são adequadas para a descrição e classificação das palavras nos dicionários. Eles também adotaram vários sistemas de notação para representar o subconjunto de fenômenos fonéticos que são cruciais para o significado.

Na ciência da fala, o termo fonema é usado para denotar qualquer uma das unidades mínimas de som da fala em uma língua que podem servir para distinguir uma palavra de outra. O termo fone é utilizado para denotar a realização acústica de um fonema. Há duas classes de fonemas: vogais e consoantes (Seção 2.2.1.2.1).

As vogais são definidas fonologicamente com base em três características principais: qualidade, altura e tensão.



A qualidade vocálica refere-se à diferença perceptível entre os diferentes sons de vogais. Ela é determinada principalmente pela posição da língua e pelos formatos das cavidades oral e faríngea durante a produção da vogal. Por exemplo, as vogais podem ser classificadas como “abertas” ou “fechadas”, dependendo da posição da mandíbula e da abertura da boca. Por exemplo, /a/ é uma vogal aberta e /i/ é uma vogal fechada.

A altura vocálica se refere à posição vertical da língua em relação ao palato durante a produção da vogal. As vogais podem ser classificadas como “alta”, “média” ou “baixa” com base na posição da língua. Por exemplo, a vogal /i/ em “pique” é considerada alta, enquanto a vogal /a/ em “casa” é considerada baixa.

A tensão vocálica se refere à tensão muscular envolvida na produção da vogal. As vogais podem ser classificadas como “tensas” ou “frouxas”. Vogais tensas são produzidas com maior tensão muscular e duração, enquanto vogais frouxas são produzidas com menos tensão muscular e têm uma duração mais curta. No português brasileiro não se considera que haja essa diferenciação. No português europeu, dependendo do dialeto, seriam encontradas vogais tensas como o /ɔ/ em “corta” ou “porta”, e vogais frouxas como o /i/ em “pia” ou “fria”.

Essas características fonológicas das vogais são usadas para distinguir as palavras em um determinado idioma. As diferenças na qualidade, altura e tensão vocálicas são consideradas contrastivas e podem levar a diferentes significados das palavras. Por exemplo, as palavras “bela”/ 'bɛlɐ/ e “bola”/ 'bɔlɐ/ são distinguidas pela qualidade vocálica dos fonemas /ɛ/ e /ɔ/ respectivamente.

A forma e a posição da língua na cavidade oral não formam uma obstrução significativa do fluxo de ar durante a articulação das vogais. No entanto, variações no posicionamento da língua conferem a cada vogal seu caráter distintivo, alterando a ressonância, assim como diferentes tamanhos e formas de garrafas produzem efeitos acústicos diferentes quando são golpeadas. A energia primária que entra nas cavidades faríngea e oral na produção das vogais vibra na frequência fundamental. As principais ressonâncias das cavidades oral e faríngea para as vogais são chamadas de f1 e f2 - primeiro e segundo formantes, respectivamente. Eles são determinados pelo posicionamento da língua e pela forma do trato oral nas vogais e determinam o timbre ou a qualidade característica da vogal.

As consoantes, por outro lado, são definidas fonologicamente como sons produzidos por meio de um bloqueio ou estreitamento parcial ou completo do trato vocal, que resulta na turbulência do ar que passa através do ponto de obstrução.

Existem diferentes tipos de consoantes, classificadas de acordo com o ponto e modo de articulação, e também com a presença ou ausência de vozeamento (consoantes surdas e sonoras) (Seção 2.2.1.2.1). Por exemplo, as consoantes, quanto ao ponto de articulação, podem ser bilabiais, alveolares, palatais ou velares, além de serem oclusivas, fricativas, aproximantes ou nasais, quanto ao modo de articulação, entre outras classificações possíveis.

Um exemplo de um par de consoantes contrastivas no português seria /p/ e /b/. Ambas são consoantes oclusivas bilabiais, produzidas bloqueando completamente o fluxo de ar nos lábios (para /p/) ou, além disso, vibrando as cordas vocais enquanto bloqueiam o fluxo de ar (para /b/).

2.2.1.4.2 O Alofone: Som e Contexto

As tabelas que representam vogais e consoantes fornecem símbolos abstratos para os fonemas⁸ - principais distinções sonoras. As unidades fonêmicas devem estar correlacionadas

⁸Consultar a tabela da Associação Internacional de Fonética disponível em <https://www.internationalpho>



com distinções de significado potencial. Por exemplo, a mudança criada ao manter a língua alta e à frente (/i/) em comparação à posição diretamente abaixo (frontal) para /e/, no contexto consonantal /m _ w/, corresponde a uma importante distinção de significado no léxico do português: mil /miw/ vs. meu /mew/. Esta distinção de significado, condicionada por um par de sons bastante similares, em um contexto idêntico, justifica a inclusão de /i/ e /e/ como distinções logicamente separadas. No entanto, um dos sons fundamentais que distingue significados é muitas vezes modificado de forma sistemática por seus vizinhos fonéticos. O processo pelo qual sons vizinhos influenciam um ao outro é chamado de coarticulação. As variações na realização fonética de um fonema, resultantes dos processos coarticulatórios, são chamadas de alofones. As diferenças alofônicas são sempre categóricas, ou seja, podem ser entendidas e denotadas por meio de um pequeno número delimitado de símbolos ou diacríticos nos símbolos fonêmicos básicos.

2.2.1.4.3 Taxa de articulação e Coarticulação

Além dos alofones, existem outras variações na fala para as quais não é possível delimitar um pequeno conjunto de categorias estabelecidas de variação. Essas variações são graduais, existindo ao longo de uma escala para cada dimensão relevante, com falantes distribuídos de maneira ampla. Falantes individuais podem variar suas taxas de acordo com o conteúdo e contexto de sua fala, e também pode haver grandes diferenças entre os falantes de uma dada língua. Alguns falantes podem fazer pausas frequentes, enquanto outros podem falar muitas palavras por minuto com quase nenhuma pausa entre enunciados. Nas taxas mais rápidas, é menos provável que os alvos de formantes sejam completamente alcançados. Além disso, alofones individuais podem se fundir ou desaparecer completamente (por exemplo, possibilidades do dialeto mineirês no enunciado “você sabe se esse ônibus passa na Savassi”, passível de realização, em representação ortográfica, como “cêsasessonspasansavas”)

2.2.1.4.4 Sílabas e Palavras

Os fonemas são como tijolos em uma construção. Para contribuir para o significado de uma língua, eles devem ser organizados em extensões coesas mais longas, e as unidades formadas devem ser combinadas em padrões característicos para ter significado, como sílabas e palavras.

A sílaba, uma unidade intermediária, é considerada como interposta entre os fonemas e o nível da palavra. O conceito de sílaba é complexo, com implicações tanto para a produção quanto para a percepção da fala. Aqui trataremos a sílaba como uma unidade perceptual. Em português, as sílabas geralmente são centradas em torno de vogais. Por exemplo, numa palavra como “casa” /ka.za/, há duas sílabas porque há duas vogais. Para dividir completamente uma palavra em sílabas, é necessário fazer julgamentos de afiliação consonantal (tomando as vogais como pico da sílaba). A questão de saber se tais julgamentos devem ser baseados em critérios articulatórios ou perceptuais, e como podem ser rigorosamente aplicados, ainda não está resolvida. Os núcleos das sílabas podem ser considerados picos de sonoridade (seções de alta amplitude). Esses picos de sonoridade têm vizinhanças afiliadas de sonoridade estritamente não crescente. Para a diferenciação dos níveis de sonoridade, pode-se utilizar uma escala de sonoridade, classificando consoantes ao longo de um continuum de oclusivas, africadas, fricativas e aproximantes. Portanto, em uma palavra como “verbal”, a silabificação seria “ver-bal”, mas não “ve-rbal”, porque

neticassociation.org/content/full-ipa-chart e no Apêndice 1.



colocar a aproximante /r/ antes da oclusiva /b/ na segunda sílaba violaria o requisito de sonoridade não crescente em direção à sílaba.

As sílabas são consideradas pelos fonólogos como tendo uma estrutura interna, e vale a pena conhecer os termos atribuídos às partes dessa estrutura. Considere uma sílaba como “trans” /trans/, por exemplo. Ela consiste em um pico vocálico, chamado de núcleo, cercado pelos outros sons em suas posições características. O elemento inicial de uma sílaba é o ataque - preenchido por consoantes. O ataque é um elemento opcional - há sílabas sem ataque, por exemplo, em uma palavra como “as”. A rima consiste da combinação do núcleo com consoantes finais, a coda, se estas estiverem presentes. Em alguns tratamentos, a última consoante em um *cluster* de final de sílaba pertenceria a um apêndice e não à coda. Assim, em “trans”, teríamos /tr/ em ataque e /ans/ em rima; a rima é formada pelo núcleo, que é /a/, e pela coda que é /ns/. A sílaba é às vezes considerada o domínio primário da coarticulação, ou seja, os sons dentro de uma sílaba influenciam mais a realização uns dos outros do que os mesmos sons se estiverem separados por uma fronteira de sílaba.

O conceito de palavra parece intuitivamente óbvio para a maioria dos falantes de línguas indo-europeias. A palavra pode ser definida, de forma geral, como: um item lexical, com um significado aceito em uma determinada comunidade de fala, e que tem a liberdade de combinação sintática permitida pela sua classe (substantivo, verbo etc.).

Na fala, há um problema de segmentação das palavras: elas se fundem, a menos que sejam afetadas por uma disfluência (problema não intencional de produção de fala) ou pela pausa deliberada (silêncio) por alguma razão estrutural ou comunicativa. Isso é surpreendente para muitas pessoas, porque a alfabetização condicionou os falantes/leitores de línguas indo-europeias a esperar um espaço em branco entre as palavras na página impressa. Mas na fala, apenas algumas pausas verdadeiras (o equivalente sonoro de um espaço em branco entre sinais gráficos na escrita) podem estar presentes. Portanto, o que parece para o olho do leitor como “você sabe se esse ônibus passa na Savassi” na escrita, soaria para o ouvido, se simplesmente usarmos letras para representar seus sons correspondentes no dialeto mineirês, como “cêsasessonspasansavas” (Seção 2.2.1.4.3) – não há pausas nesse enunciado. Frequentemente, o que encontramos na fala, são quebras prosódicas, que podem ser de natureza não-terminal – indicando unidades entoacionais em um enunciado e representadas por /, e quebras terminais, indicando a conclusão de um enunciado e representadas por //. Assim, dependendo da constituição informacional, uma sequência de palavras como: “não deu a altura que a Mari marcou lá”, pode ser enunciada com propósitos ilocucionários distintos como as seguintes configurações, dentre outras:

- não deu a altura que a Mari marcou lá // um enunciado, com uma unidade entoacional;
- não // deu a altura que a Mari marcou lá // dois enunciados, com uma unidade entoacional cada;
- não // deu a altura / que a Mari marcou / lá // dois enunciados, um com uma unidade entoacional e o outro com três unidades entoacionais.

Certos fatos sobre a estrutura das palavras e as suas possibilidades de combinação são evidentes para a maioria dos falantes nativos e foram confirmados por décadas de pesquisa linguística. Alguns desses fatos descrevem as relações entre as palavras quando consideradas isoladamente, outros dizem respeito a grupos de palavras relacionadas que parecem intuitivamente similares ao longo de alguma dimensão de forma ou significado - essas propriedades são chamadas de paradigmáticas. As propriedades paradigmáticas das palavras incluem a sua classe gramatical, a sua morfologia flexional e derivacional e



a sua estrutura em compostos. Outras propriedades das palavras dizem respeito ao seu comportamento e distribuição quando combinadas para fins comunicativos em enunciados – essas propriedades são chamadas de sintagmáticas.

2.2.2 Reconhecimento de fala

A tarefa de reconhecimento de fala, também conhecida como ASR (do inglês, *automatic speech recognition*), consiste na transformação do sinal acústico de um trecho de fala em um trecho de texto (Figura 2.2).

Figura 2.2: Conversão do áudio da fala em sua transcrição textual.



Essa tarefa tem diversas aplicações, mas a mais difundida é no uso de assistentes de voz, também conhecidos como assistentes virtuais. Os assistentes, comumente embutidos em celulares, como o próprio nome revela, foram criados para ajudar as pessoas em tarefas corriqueiras, como enviar mensagens, fazer ligações, agendar compromissos etc. Para que a ajuda dos assistentes “valha a pena”, eles devem interagir com o humano da forma mais natural, isto é, por meio da fala. Para que isso aconteça, o assistente precisa, antes de tudo, compreender a fala do humano. A primeira etapa dessa compreensão⁹ envolve o reconhecimento da fala, ou a sua conversão em texto.

No processamento da fala, assim como em diversas aplicações de PLN na atualidade, também concluiu-se ao longo do tempo que os modelos de aprendizado profundo, baseados em dados, são os que geram melhores resultados. Essa abordagem se baseia em grandes quantidades de dados, a partir dos quais a rede neural conseguirá aprender, isto é, identificar padrões e ajustar os pesos dos neurônios. No caso do reconhecimento de fala, os dados são *corpora* de áudio e texto, isto é, para cada trecho de áudio produzido por humanos, em geral uma sentença ou enunciado, deve haver uma transcrição correspondente, para que o modelo consiga associar uma coisa à outra. A seguir, falaremos mais sobre como devem ser esses dados, e sobre aspectos fundamentais do reconhecimento de fala.

2.2.2.1 Coleta de dados

Os dados, que são o ponto de partida para o treinamento de uma rede neural, devem ser os mais representativos possíveis para a língua falada que se deseja processar. O que isso quer dizer? Da mesma forma como acontece com humanos, a rede neural aprende a partir do que é mostrado a ela, e ela aprende melhor o que for mostrado mais vezes. Nesse

⁹A compreensão da fala envolve também a tarefa chamada de NLU, *Natural Language Understanding*, na qual aspectos relacionados ao sentido do enunciado são processados pelo computador e convertidos em algum tipo de ação. No caso dos assistentes de voz, um enunciado como “Escrever um e-mail”, dito pelo usuário, será convertido na ação de escrita do e-mail - provavelmente algum aplicativo de e-mail será aberto ou o assistente perguntará para o usuário quem seria o destinatário ou qual o texto do e-mail etc. Essa tarefa não faz parte do que hoje está no escopo da maioria dos modelos de reconhecimento de fala, mas é a etapa que a segue.

sentido, essa seção aborda alguns pontos muito importantes na coleta dos dados: propósito, público-alvo, variações de fala e contexto.

No caso do reconhecimento de fala, é ideal que se tenha em mente para qual **tipo de produto** o modelo de ASR será usado. Tomando novamente como exemplo os assistentes virtuais, seu objetivo principal é o reconhecimento correto de comandos de voz. Dessa forma, os dados para o treinamento da rede neural deverão conter também¹⁰ comandos de voz, instâncias primordiais da interação de usuários com assistentes. É claro que é possível construir um reconhecedor de fala “geral”, isto é, que não esteja destinado a um tipo específico de aplicação, mas que visa a reconhecer qualquer tipo de fala que for dado como entrada, seja um diálogo com um *chatbot*, seja uma conversa entre amigos. No entanto, a acurácia de um modelo “geral” tenderá a ser bem inferior à de um modelo específico, uma vez que a fala espontânea encontrada em conversas entre amigos possui muitas particularidades que dificultam o reconhecimento, tais como sobreposição de fala, ruídos de ambiente e fala menos articulada.

Os dados também precisam representar o **usuário-alvo**. Com relação a assistentes de voz, os usuários costumam ser pessoas portadoras de celulares, o que hoje em dia significa “praticamente todo mundo”. Mas, pensando bem, talvez nem tanto crianças abaixo de 12 anos ou idosos com mais de 70. Dessa forma, as gravações que compõem o *corpus* de treinamento precisam ser feitas por todo tipo de usuário, mas especialmente por adolescentes e adultos de uma faixa etária entre 12 e 70 anos, em igual proporção de homens e mulheres. Se um modelo for treinado apenas com crianças do gênero feminino, por exemplo, ele será excelente em reconhecer a fala de crianças do gênero feminino, mas provavelmente bem ruim em reconhecer a fala de senhores de 70 anos.

Outro ponto ao qual devemos nos atentar no momento de coleta de dados é a **representatividade dialetal**. Da mesma forma que o modelo precisa ver áudios produzidos tanto por homens quanto por mulheres, adolescentes e idosos, ele também precisa ver áudios de usuários de Caucaia (CE) e de Uruguaiana (RS), por exemplo, localidades nas quais o português falado difere consideravelmente no âmbito fonético, principalmente. Se o modelo for treinado com dados de usuários da mesma variedade dialetal, ele será bom em reconhecer a fala desses usuários, mas não tão bom em reconhecer a fala de usuários de outras regiões. Nesse sentido, vale mencionar que enquanto as variações de fala encontradas nas variantes do português brasileiro e europeu – ou mesmo nos diferentes sotaques e pronúncias dentro do próprio Brasil – têm um grande impacto no PLN da fala, esse impacto no PLN de texto é bem menor.

Finalmente, é preciso também levar em consideração a **forma como a gravação foi feita**. Idealmente, para o produto assistente de voz, as gravações que comporão o *corpus* de treinamento deverão também ter sido feitas utilizando-se o gravador do celular, inclusive com os ruídos de fundo típicos do contexto de uso final da aplicação. As pessoas utilizam o celular na rua, dentro de carros, em casa, em restaurantes, onde há ruídos de conversas, trânsito, música etc., mas muito raramente em estúdios com isolamento acústico perfeito. Portanto, é preciso mostrar à rede neural uma parcela significativa de áudios com esses tipos de ruído¹¹.

Em resumo, os dados do treinamento de uma rede neural precisam ser representativos

¹⁰Digo “também”, porque se os dados contiverem apenas comandos de voz, a aplicação ficará muito restrita. Falaremos disso mais adiante.

¹¹É claro que áudios com ruídos muito intensos atrapalham consideravelmente o reconhecimento de fala e não devem ser considerados válidos ou representativos do uso, e por isso, não devem ser incluídos no treinamento.



da interação ou contexto de uso, tanto no conteúdo e formato do texto, quanto na forma de gravação, e do perfil de usuário que se quer atingir.

Talvez o leitor esteja se perguntando onde é possível encontrar dados tão peculiares. De fato, esse é um grande desafio da tarefa de reconhecimento de fala, senão o maior. Em se tratando do português, assim como faltam recursos para outras tarefas de PLN, faltam também *corpora* de áudio e texto suficientemente grandes que estejam disponíveis de forma gratuita. Há alguns recursos grátis na internet, como o Mozilla Common Voice (sentenças lidas, em sua maioria)¹² e o LibriVox (audiolivros)¹³, mas, infelizmente, eles são insuficientes em termos do número de horas de gravação para se treinar um modelo *end-to-end* do zero. Em geral, o treinamento de uma rede neural para o reconhecimento de fala requer milhares de horas¹⁴. Fica aqui um convite aos recém-chegados à área para investir na coleta de dados para o português brasileiro.

Para lidar com essa questão da disponibilidade de dados, existem algumas técnicas. Uma técnica bastante usada é a de aumento de dados (*data augmentation*)¹⁵. Essa estratégia não é restrita ao reconhecimento de fala, mas, no caso desta tarefa, se refere ao aumento dos dados com base em manipulações dos dados já existentes. Um número de gravações do *corpus* de treinamento pode, por exemplo, sofrer adição de ruídos diversos, como os mencionados anteriormente. Suponhamos que o *corpus* de treinamento seja composto por 100 horas de gravação. Podemos, por exemplo, separar 20% dos áudios e adicionar cinco tipos de ruídos a eles, de modo que teremos ao final 200 áudios diferentes (100 áudios iniciais + 100 gerados por manipulação). Assim, os dados resultantes serão diferentes entre si, mas não haverá o trabalho de se criar novos dados do zero. Há outras técnicas para se melhorar a acurácia de um modelo, das quais falaremos na Seção 2.2.2.5.

2.2.2.2 Pré-processamento do texto: limpeza e formatação

Uma vez coletados os dados de texto e fala para formar o *corpus* paralelo de treinamento, é necessário formatá-los para que possam servir de entrada para a rede neural. Essa seção descreve o processo de limpeza e formatação do texto correspondente à transcrição dos áudios. Idealmente, não deve haver muitos erros de digitação ou grafia nas transcrições, para que a rede não aprenda errado. Em outras palavras, a saída de um reconhecedor não deve conter erros de grafia, por isso não seria bom treinar um modelo com um *corpus* no qual o *token* “tambem” ocorresse um número igual ou superior de vezes que sua versão correta, “também”. Se esse fosse o caso, o modelo aprenderia que o *chunk* acústico [tẽbẽj] ¹⁶ corresponderia a “tambem”, e, por conseguinte, a saída do modelo conteria o *typo* “tambem”. Por isso, é importante fazer um levantamento desse tipo de erro no *corpus* de treinamento, por exemplo, contrastando a lista de palavras do *corpus* com uma lista-referência da língua para a qual a aplicação está sendo desenvolvida¹⁶.

¹²<https://commonvoice.mozilla.org/en/datasets>

¹³https://librivox.org/search?primary_key=52&search_category=language&search_page=1&search_form=get_results

¹⁴O número de horas depende da arquitetura de rede neural utilizada. Estamos considerando aqui o estado da arte, que são modelos *end-to-end*. Modelos híbridos conseguem ser bem treinados com bem menos horas. Falaremos mais disso na seção sobre arquiteturas.

¹⁵Na realidade, há mais de uma técnica de aumento de dados. Uma delas, bastante utilizada em treinamentos para ASR, é a *spec augmentation* (Park et al., 2019), que, apesar de ter “*augmentation*” no nome, não aumenta os dados, mas faz edições nos dados já existentes.

¹⁶Vale lembrar que um sistema de reconhecimento de fala é dependente do idioma, isto é, um modelo bem treinado em dados do português não conseguirá fazer um bom reconhecimento do francês. O mais próximo que se pode chegar de modelos independentes de língua seria por meio de *transfer learning*,



Depois de levantados os erros, é preciso corrigi-los de alguma forma caso sejam muito frequentes. Isso é muito comum em dados coletados na internet ou que não passaram por um processo rigoroso de transcrição e revisão. Outra forma de lidar com esse problema dos *typos*, caso não se queira investir tempo na limpeza dos dados, é implementar um módulo de pós-processamento que corrige grafias incorretas, mas isso pode trazer desvantagens, como um possível aumento na latência (tempo corrente entre a fala do usuário e o reconhecimento do texto, crucial em aplicações como a dos assistentes de voz).

Finalmente, talvez seja necessário normalizar o texto antes do treinamento¹⁷. As técnicas de normalização são as mesmas utilizadas em processamento de texto (Capítulo 4), por isso não vamos repeti-las aqui. Vale apenas dizer que atualmente existem modelos de reconhecimento de fala *end-to-end*, isto é, que têm como entrada o texto não normalizado, minimamente manipulado, e como saída, a transcrição também já normalizada inversamente, da forma exata como deve aparecer para o usuário. No entanto, para se obter uma acurácia boa em modelos *end-to-end*, é necessária uma quantidade muito grande de dados, o que é inviável de se obter para muitos pesquisadores e empresas, por isso não se deve descartar a normalização.

2.2.2.3 Pré-processamento do áudio

Depois da limpeza do texto, é preciso “limpar” os áudios. Áudios distorcidos¹⁸ devem ser removidos e também aqueles cuja duração é muito discrepante da duração da maioria. Mais uma vez, isso só é necessário caso o número de áudios *outliers* seja muito grande. Um caso ou outro não vai atrapalhar a aprendizagem. Por fim, os áudios e a transcrição devem ser segmentados e alinhados de alguma forma, caso já não estejam assim. Essa segmentação e alinhamento são importantes para garantir que a rede possa aprender a partir de dados que sejam os mais específicos e corretos possíveis.

Conforme mencionado anteriormente, o reconhecimento de fala é feito atualmente por meio de redes neurais, mas, qualquer que seja a arquitetura utilizada (veremos as principais na próxima seção), a primeira etapa envolve processamento de sinais. O primeiro passo é sempre a conversão do sinal analógico para digital. A isso se segue a extração de informações do sinal, que serão os elementos de entrada para a rede neural (combinados ao texto)¹⁹.

2.2.2.3.1 Conversão analógico-digital

Como explicado na Seção 2.2.1, o sinal acústico da fala nada mais é que o resultado da vibração das pregas vocais pela passagem do ar. O ar que respiramos passa pelas cordas vocais e causa sua vibração, gerando ondas sonoras, que passam pela faringe e laringe até atingir a cavidade bucal. Nela, as ressonâncias geradas pela vibração das pregas encontram obstáculos e são por eles modificadas e, finalmente, liberadas com a abertura da boca (e pelo nariz, no caso de nasais), quando falamos. Os “obstáculos” mencionados são as diferentes posições que os nossos articuladores assumem²⁰. Dessa forma, o nosso aparato

que é uma técnica de aprendizado de máquina que “aproveita” um treinamento prévio como ponto de partida para um treinamento com outros dados. No *transfer learning*, os pesos da rede neural não são iniciados em 1, mas já ajustados com base no treinamento anterior.

¹⁷Há dois tipos de normalização de texto, a direta, conhecida apenas como “normalização”, e a inversa, conhecida como “ITN (*inverse text normalization*)”, da qual falaremos mais adiante.

¹⁸Nos referimos a arquivos de áudio em que ocorre distorção do sinal e perceptualmente ouve-se um chiado alto que compromete a compreensão da fala.

¹⁹Exceto no modelo *wav2vec* (Baevski et al., 2020), que será descrito mais adiante.

²⁰Em linguística, a vogal [i], por exemplo, costuma ser descrita como “anterior alta não-arredondada”.



vocálico atua como um filtro para as frequências originais emitidas pela glote, e o que ouvimos é o que passou pelo filtro. Essas frequências filtradas são captadas por microfones como ondas analógicas, que precisam ser digitalizadas para serem processadas por um sistema de reconhecimento de fala.

A conversão do sinal envolve dois processos: a **amostragem** e a **quantização**²¹. A amostragem é a seleção das amostras de amplitude do sinal acústico que serão medidas para se representar digitalmente a onda. Deve-se selecionar, no mínimo, duas amostras por ciclo, uma correspondente ao pico, e a outra, ao vale da onda. O número de amostras por segundo corresponde à taxa de amostragem. Qualquer que seja a taxa de amostragem, a máxima frequência registrada em 1 segundo será sempre a metade do número de amostras em 1 segundo, uma vez que uma repetição da onda deverá ter, pelo menos, duas medições para ser minimamente registrada. A taxa de amostragem para gravações de fala deve ser de no mínimo 20 kHz (vinte mil medições por segundo), uma vez que a maioria das informações relevantes para a fala estão abaixo de 10 kHz.

A quantização é a representação desses valores de amplitude em inteiros pelo computador. As representações mais comuns para um sinal acústico são de 8 ou 16 bits. Quanto maior o número de bits que podem ser alocados para representar uma medição de amplitude, melhor será a representação digital da onda, uma vez que mais pontos de amplitude poderão ser armazenados.

2.2.2.3.2 Janelamento

Pelo fato de ser gerado de maneira irregular (vibrações da glote), o sinal de fala é um sinal não-estacionário, isto é, não mantém suas propriedades constantes por mais de 100 ms. No entanto, entre 5 e 100 ms, as propriedades se mantêm relativamente constantes, e o sinal se assemelha a um sinal estacionário²². Por isso, para representar um sinal com duração de vários segundos ou até minutos, utiliza-se o método de janelamento²³. Esse método consiste na fragmentação do sinal em pequenas janelas de tempo de modo que o início da próxima janela ocorra cerca de alguns milissegundos após o início da anterior²⁴. Para que não haja cortes abruptos na representação da amplitude do sinal entre uma janela e outra, costuma-se aplicar a função Hamming em cada janela. Essa função aproxima de zero os valores de amplitude nas extremidades das janelas.

2.2.2.3.3 Extração de informações das frequências do sinal

Uma vez separado em janelas, é preciso extrair as informações das frequências do sinal digital, pois é nas frequências que residem os correlatos dos fones (a informação que nos permite identificar diferentes fones)²⁵. São informações de frequência e pressão que servirão de entrada para a modelagem da fala. Há mais de um método de extração dessas informações, mas o mais comum atualmente é a Transformada Discreta de Fourier (DFT), computado pelo algoritmo FFT (*Fast Fourier Transform*). Esse método é aplicado a cada

Cada um desses adjetivos se refere a um aspecto da articulação do [i]. “Anterior” e “alta” se referem ao posicionamento da língua, e “não-arredondada”, à configuração dos lábios.

²¹Recomendamos a leitura de (Johnson, 2011) para saber mais.

²²Para saber mais, consulte (Rabiner; Juang, 1993).

²³Para representar um sinal estacionário, como o ruído branco, bastaria apenas uma janela, uma vez que esse sinal é constante ao longo do tempo.

²⁴Segundo Fayek (2016), costuma-se usar intervalos de 20 a 40 ms para a janela, e 10 ms de deslocamento (50-60% de sobreposição entre duas janelas consecutivas).

²⁵Lembre-se das aulas de fonética acústica e dos formantes que caracterizam cada vogal. Para saber mais, consulte (Johnson, 2011).



janela, tendo como entrada a amplitude do sinal em um dado intervalo de tempo, e, como saída, informações de frequência e pressão para cada janela.

Depois de extraídas, as informações das frequências do sinal são convertidas para a escala mel (Stevens, 1937), uma escala de frequência baseada na percepção humana do sinal acústico. Nosso ouvido é mais sensível a mudanças sutis de amplitude nas frequências mais baixas e menos sensível a mudanças nas frequências mais altas. Dessa forma, a escala mel agrupa as frequências com base em filtros logaritmicamente diferentes, isto é, as frequências mais baixas possuem mais agrupamentos de menos faixas de frequência, cada, e as frequências mais altas possuem menos agrupamentos com mais faixas de frequência em cada um. Assim, as frequências mais baixas são representadas em mais detalhes do que as mais altas.

As janelas de sinal digitalizado e representado na forma de frequências na escala mel são transformadas em vetores, que servirão de entrada para a rede neural de reconhecimento de fala, como veremos adiante.

2.2.2.4 Modelos de reconhecimento

O problema de reconhecimento de fala é um problema de classificação de sequências. A entrada é um sinal contínuo, o sinal acústico, que deve ser primeiro filtrado para que a fala seja separada do ruído²⁶, e digitalizado. Assim, o sinal é transformado em uma sequência de unidades discretas, como vimos na seção anterior. Essa sequência de unidades será classificada como outra sequência, que será a saída do processo. A sequência de saída é, na maioria dos casos, palavras.

No caso da conversão de fala em texto, a diferença de tamanho entre a sequência de entrada da rede neural, vetores com *features* acústicas, e a de saída, palavras, costuma ser muito grande. Lembre-se de que o áudio foi digitalizado e, com a extração das informações de frequência, vetorizado. Cada vetor corresponde a uma janela de 10 ms, como vimos na Seção 2.2.2.3.2, então, para uma sentença de 10 s, com 5 palavras, teríamos 100 vetores. Para minimizar essa discrepância, realiza-se um *subamostragem*, processo de redução do número de vetores do *input*.

Até alguns anos atrás, empregavam-se modelos estatísticos híbridos para resolver o problema do reconhecimento de fala. As arquiteturas utilizadas continham módulos que eram treinados de maneira independente. Os módulos eram o modelo acústico (AM), o modelo de língua (LM) e um modelo lexical com um dicionário de pronúncias. Os modelos conhecidos como HMM (*Hidden Markov Model*) foram amplamente utilizados com relativo sucesso nas tarefas de ASR. No entanto, essas arquiteturas trabalhavam com modelos de linguagem baseados em n-gramas²⁷ e assumiam independência entre as probabilidades de ocorrência dos fones, e, por isso, não eram eficazes em processar informações de longa distância²⁸. Hoje, as arquiteturas do tipo *encoder-decoder* são as mais utilizadas em ASR.

Os modelos HMM que geravam melhores resultados eram baseados numa arquitetura de máquina de estados finitos, em que cada estado corresponde a uma parte de um fone. Por exemplo, para o fone [a], gerava-se um HMM com três estados: o primeiro representando o início do fone [a], o segundo representando a parte mais estável do fone, e o último, o final do fone. Dessa forma, os modelos eram treinados para todos os fones da língua. Para

²⁶Atualmente, as redes neurais conseguem aprender qual parte de um áudio contém fala.

²⁷N-gramas representam contextos muito restritos, na maioria das vezes.

²⁸Outra desvantagem dos modelos híbridos é o fato de o LM precisar alocar muita memória (100 GB) para que se obtenham bons resultados no modelo. Isso se torna proibitivo em se tratando de ASR embarcados em dispositivos.



tratar o problema mencionado anteriormente de ausência de contexto, treinava-se modelos com grupos de três fones seguidos (trifones). Os melhores modelos eram agrupados no módulo do modelo acústico. A saída do modelo acústico, por sua vez, era interpolada com um dicionário de pronúncias. O último passo era a combinação da saída do módulo lexical com um modelo de língua, que continha n-gramas e suas probabilidades de ocorrência. O Quadro 2.2 demonstra esse processo:

Quadro 2.2. Modelos de reconhecimento

AM + Léxico	LM
ingressos para a próxima seção ->	ingressos para a próxima <u>sessão</u>

Na primeira coluna do Quadro 2.2, temos a saída do modelo acústico e do léxico, que é “ingressos para a próxima seção”. Note-se que a palavra “seção”, um dos homófonos conhecidos do português, não está escrita da maneira correta. Nesse contexto, a grafia correta seria “sessão”. A adição de um modelo de língua, treinado com uma quantidade suficiente de dados, é capaz de acertar a grafia correta para esse contexto, com base nas relações entre as palavras.

Como o treinamento do modelo acústico HMM era baseado nos fones, era necessário balancear os dados de treinamento foneticamente. Isto é, a distribuição dos fones nos dados deveria refletir a sua proporção na língua falada²⁹. A consoante [l], por exemplo, um dos fones mais frequentes do português brasileiro, deveria ocorrer mais vezes nos dados de treinamento do que sua parente [lh], menos comum.

Uma arquitetura parecida com as híbridas, chamada CTC (*Connectionist Temporal Classification*), configura a forma mais simples de executar a tarefa de reconhecimento de fala. Assim como os modelos acústicos dos modelos híbridos, o CTC atribui *labels* (classes, dentre as possíveis letras do alfabeto) a cada *frame* de atributos acústicos e depois elimina as letras duplicadas seguidas uma da outra. O principal problema do CTC é a sua “falta de memória”, isto é, ele considera todas as saídas independentes umas das outras, e computa a saída para um instante *t* apenas com base na entrada desse mesmo instante *t*.

Mais recentemente, começou-se a empregar redes neurais recorrentes na tarefa de ASR. Basicamente, essas redes, chamadas de RNN, tinham a vantagem de armazenar informação desde o início da sequência, ou no nosso caso, da sentença, configurando uma forma de “memória”³⁰. A computação dentro de uma unidade da rede leva em consideração a saída da unidade da etapa anterior bem como a saída do próprio neurônio na etapa atual. As RNN-T (T de *Transducer*) são a combinação do CTC, enquanto modelo acústico, com um predictor que faria as vezes de modelo de língua e reavaliaria a saída do CTC, gerando uma nova saída, levando em consideração o contexto.

Outra opção muito usada são os Transformers com *self-attention*. De forma resumida, diferentemente das RNN, nos Transformers, os vetores de entrada e de saída têm o mesmo tamanho e cada bloco de atenção tem acesso às entradas dos blocos anteriores. Assim, cada entrada é comparada com as demais para que a saída mais provável seja gerada. Os

²⁹Alcain; Solewicz; Moraes (1992) descrevem uma metodologia para gerar listas foneticamente balanceadas. Um procedimento semelhante, baseado no cálculo do qui quadrado, pode ser aplicado ao balanceamento de *corpora*.

³⁰RNN têm a desvantagem dos *vanishing gradients* e precisam ser combinadas com técnicas de *gating*, como LSTM e GRUs. Na prática, RNNs são séries de LSTMs.



Transformers são eficazes em modelar contextos mais distantes, mas menos eficazes em contextos de curta distância.

Atualmente, tanto RNN-T quanto Transformers são técnicas bastante utilizadas em ASR. No entanto, alguns estudos mais recentes apontam outras soluções como ainda melhores. Gulati et al. (2020) mostram resultados competitivos com o uso de *Conformers*, arquitetura que une as redes convolucionais (CNN) com os Transformers (daí o nome “*conformer*”). Na combinação CNN + Transformers, as limitações de ambas arquiteturas são suavizadas, porque o que é deficiente em uma é o ponto forte da outra. Os Transformers são melhores em contextos mais globais, e as CNN, em contextos mais locais.

Nas arquiteturas de *encoder-decoder*, o “*encoding*” pode assumir diferentes unidades, como fones, sílabas ou grafemas. No entanto, os resultados mais competitivos em ASR utilizam *wordpieces* como as menores unidades codificadas. *Wordpieces*, ou *subwords*, são exatamente o que os nomes indicam: partes de palavras (Capítulo 4). Mas não devem ser confundidos com morfemas! Diferentemente dos morfemas, as *wordpieces* não carregam nenhum significado necessariamente³¹. Elas podem ser geradas de maneira empírica por diferentes algoritmos (WordPieceModel, *byte pair encoding* (BPE) e outros) e constituem um vocabulário induzido a partir de dados de texto. A segmentação das palavras da língua em unidades menores é, de certa forma, arbitrária (sua geração envolve etapas “*greedy*”), embora se baseie na frequência com que essas unidades aparecem no *corpus*. Por exemplo, em um *corpus* formado apenas por sentenças com verbos no infinitivo, é de se esperar que um vocabulário induzido a partir dele contenha alguma *wordpiece* que termine em “-ar”, como **tar_** (o “*underscore*” após a *string* representa final de palavra). Dessa forma, caso o modelo se depare com o neologismo “deletar”, considerando que ele não esteve presente no *corpus* de treinamento, o modelo conseguirá gerá-lo concatenando a *wordpiece* “tar_” com outras *wordpieces* (talvez “de_”, de “deixar, derrubar”, “le_” de “ler, levar”, e “tar_”).

A abordagem de *wordpieces* como unidade de modelagem se mostrou melhor do que a de grafemas no que diz respeito especialmente às palavras OOV (*out-of-vocabulary*), como neologismos, nomes próprios, palavras estrangeiras e termos da moda. Nos modelos híbridos, os *frames* acústicos eram mapeados para fones e depois era necessária uma interpolação com um dicionário de pronúncia para gerar as palavras. Nos modelos *end-to-end*, em que se busca eliminar essa última etapa, *wordpieces* têm gerado resultados melhores pelo fato de trazerem em si uma espécie de contexto. Na maioria das línguas, incluindo o português, um grafema isolado pode ser associado a mais de uma pronúncia, como é o caso de “r” (“rato” e “caro”). Ao contrário, o fone [h] de “rato” não ocorrerá na *wordpiece* “_ro”. Os grafemas e o léxico de pronúncia funcionam bem para palavras conhecidas da língua, mas deixam a desejar quando se deparam com palavras que não estão no dicionário.

Mais recentemente, em 2019, uma arquitetura bastante promissora foi proposta pela Facebook AI, o *encoder wav2vec*³². Baseado no *word2vec* (Capítulo 10) do processamento de texto, a ideia do *wav2vec* é obter representações vetoriais diretamente a partir do áudio puro, isto é, eliminando a etapa de extração de atributos acústicos e a necessidade de se treinar com áudios transcritos. Por meio de duas redes convolucionais sucessivas, o modelo transforma áudio digitalizado em vetores e aprende distinguindo trechos reais de áudio de trechos modificados por ele mesmo. O *wav2vec* é uma arquitetura de aprendizado autossupervisionada (*self-supervised learning*) que aprende a predizer trechos de áudio. Esse modelo depois pode ser combinado com outras redes neurais usadas em ASR. A

³¹Apenas no caso de uma *wordpiece* coincidir com um morfema, ela intrinsecamente terá significado, mas isso não é levado em consideração pelo modelo.

³²<https://ai.facebook.com/blog/wav2vec-state-of-the-art-speech-recognition-through-self-supervision/>.



grande vantagem dessa abordagem é que ela resolve o principal problema da tarefa de reconhecimento de fala: a falta de dados de áudio e texto, especialmente para *low resource languages*, para as quais a oferta de dados é baixíssima ou até mesmo inexistente. Mesmo para línguas como o inglês, bem representado em termos de dados para processamento de fala, o wav2vec é bastante eficiente, porque precisa de 100 vezes menos horas de áudio de treinamento do que as arquiteturas *end-to-end* que vimos acima (Baeviski et al., 2020).

2.2.2.5 Etapas adicionais

Devido à escassez de dados de fala anotados disponíveis e à necessidade que os modelos *end-to-end* têm de muitos dados, várias técnicas vêm sendo experimentadas para que seja possível contornar essa questão. Uma técnica bastante conhecida é o *shallow fusion* (Williams et al., 2018). Nessa técnica, um modelo de língua, treinado a partir de *corpora* de textos, é adicionado ao *pipeline* de treinamento. Esse LM externo, como é chamado, é eficaz em completar sequências de palavras, então sua contribuição se dá na reavaliação de dado segmento para um *chunk* mais provável. Suponhamos que o modelo de ASR treinado com áudio e texto tenha gerado a seguinte saída “essas ideias como-as com sebo”. O recálculo da hipótese pelo LM externo provavelmente chegaria em “essas ideias como as concebo”, que é um trecho mais provável de ocorrer, dada a semântica das palavras envolvidas.

Há muitas outras técnicas de aprendizado de máquina que podem ser usadas, e combinadas, para aprimorar o resultado de um sistema de reconhecimento de fala. Há quem recorra à síntese de áudio para resolver o problema da falta de dados, por exemplo.

2.2.2.6 Pós-processamento

Uma última etapa do *pipeline* de um sistema de ASR costuma ser a normalização inversa, mais conhecida como ITN (*Inverse Text Normalization*)³³. O que ocorre nessa etapa é a conversão de *strings* que foram transcritas da forma como foram faladas (domínio falado) em símbolos (domínio escrito). Isso se aplica principalmente a números, unidades de medida, moedas, contas de matemática, números romanos, tudo que envolve uma simbologia diferente de “palavra”. Normalmente, a hipótese gerada pelo ASR não contém caracteres numéricos nem outros símbolos, como “a” ou “&”, então, para que um número de telefone, por exemplo, seja mais legível para o usuário final da aplicação, se faz a normalização inversa. Dessa forma, um número como “noventa e nove nove nove nove nove nove nove nove nove”, formato gerado na hipótese, pode ser convertido para (99) 9999-9999. O mesmo costuma acontecer com quantias monetárias, como “dois milhões e quinhentos mil reais”, que pode ser transformado em “R\$ 2.500.000” ou “R\$ 2500000”, a depender da convenção adotada.

Os módulos de ITN podem ser feitos por meio de regras escritas por especialistas ou podem ser redes neurais. Recentemente, começou-se a migrar para os ITNs neurais, como indica o artigo da Amazon AWS AI de 2021 (Sunkara et al., 2021). Um ITN baseado em regras funciona segundo um modelo de transdutor de estados finitos (FST), semelhante à máquina de estados finitos mencionada anteriormente na explicação dos HMMs.

³³Isso depende bastante de qual aplicação se está desenvolvendo. Em se tratando de aplicações para fins comerciais, é extremamente comum haver a normalização inversa, por questões de experiência do usuário.



2.2.2.7 Métricas de avaliação

A acurácia de um modelo de ASR costuma ser medida em termos de taxa de erro de palavras e de sentenças. As métricas mais utilizadas são a *Word Error Rate* (WER) e a *Sentence Error Rate* (SER). A WER é calculada com base na soma de deleções, substituições e inserções dividida pelo total de palavras da referência e multiplicada por 100. Veja o exemplo abaixo, retirado da base de teste do LibriVox:

Referência: A virtude é comunicável. O vício é contagioso. Os governos fracos fazem fortes os ambiciosos e insurgentes. Atividade sem juízo é mais ruinosa que a preguiça.

Hipótese : A virtude é comunicável. O vício é contagioso. Os governos fracos fazem fortes os ambiciosos e insurgentes. **Atividades** sem juízo é mais **ruidosa** que a preguiça.

Avaliação: S S

O trecho da referência é a transcrição manual do áudio, e o trecho da hipótese é a saída gerada por um sistema de ASR. Os segmentos sublinhados são aqueles cujo reconhecimento automático errou. Enquanto a referência era “atividade”, no singular, a hipótese gerada foi “atividades”, no plural; enquanto a referência era “ruinosa”, a hipótese foi “ruidosa”. Esses são exemplos de erros de substituição e a WER desse trecho é dada por $2/26 * 100 = 7,69\%$, em que 2 é a soma das substituições e 26 é o total de palavras do trecho.

Em geral, calcula-se um valor único de WER, para um dado conjunto de teste, para se avaliar o desempenho de um modelo. Atualmente, os melhores modelos atingem um valor de WER inferior a 5% sem técnicas de *fine-tuning* e *shallow fusion*.

A métrica SER é referente à computação do número de sentenças com pelo menos um erro. Portanto, para um conjunto de teste com 100 sentenças, das quais dez apresentaram um ou mais erros de inserção, deleção ou substituição, a taxa de SER será de 10%. Por ser mais detalhada e dar uma ideia melhor do desempenho de um modelo, a WER costuma ser mais utilizada do que a SER. A SER é indicada para casos em que se queira medir o desempenho de um normalizador inverso, por exemplo, em que o número de *tokens* de uma sentença não normalizada para uma normalizada não nos diz muito. Por exemplo, a sentença “Você me deve cinco reais”, quando normalizada inversamente, gera “Você me deve R\$5,00”, a depender da convenção adotada pelo ITN. Digamos que a saída de um ITN para essa sentença seja “Você me deve R\$ 5,00”. Se computarmos o WER, obteremos $2/4 * 100 = 50\%$. Nesse caso, o WER não nos diria muito sobre a eficácia do ITN. Por isso é mais interessante computarmos a SER e sabermos qual a porcentagem de sentenças do conjunto de teste que apresentaram algum erro de normalização.

Como bem apontou Jurafsky; Martin (2023), talvez fosse interessante criar uma métrica que levasse em consideração a relevância das palavras da sentença, atribuindo um peso maior às palavras mais relevantes, que são, em geral, palavras de conteúdo, como verbos e nomes (Capítulo 4). Por exemplo, uma sentença como “Mande um beijo para a Juliana” reconhecida por um ASR como “Mande um beijo pra Juliana” seria muito menos problemática para todos os efeitos do que uma saída como “Mande um beijo para a Júlia”. Embora o WER da segunda sentença (16,6%) seja menor do que o da primeira (50%), a primeira hipótese é muito mais fiel ao conteúdo da sentença. Em muitas aplicações, o ASR é o primeiro passo de um *pipeline* de PLN que envolve a atribuição da sentença a uma intenção do falante e depois realiza uma ação. Nesse caso, enviar um beijo para a pessoa errada pode ter sérias consequências.



2.2.2.8 Desafios em reconhecimento de fala

Mesmo quando um modelo atinge uma acurácia de quase cem por cento de acerto no reconhecimento das palavras, há ainda alguns erros bastante difíceis de corrigir. Os casos que apresentamos aqui valem para o português brasileiro. É possível que se apliquem a outras línguas em situações parecidas, mas o que será apresentado se baseia nas observações com relação ao português do Brasil. Esses problemas estão relacionados aos artigos “a” e “o”, vogais átonas, na maioria das vezes, quando ocorrem no fim de uma palavra seguidas da mesma vogal também em posição átona, como no Exemplo 2.1.

Exemplo 2.1.

Mande um beijo **para a** Amanda

Quando falamos espontaneamente, ou até mesmo numa fala colaborativa, cujo “interlocutor” é um assistente virtual, situação em que tendemos a falar de um modo mais monitorado e articulado, as vogais em sequência são pronunciadas de forma contínua, numa mesma corrente de ar. Não costumamos fazer pausas (chamadas de *glottal stops*) entre uma vogal e outra nessas situações. No Exemplo 2.1, o [a] final de “para” se junta ao [a] do artigo “a” e ambos podem ser interpretados pelos modelos como sendo apenas um único fone [a], como ilustrado em Exemplo 2.2.

Exemplo 2.2.

Mande um beijo para **a** Amanda

Embora a diferença de duração entre um caso e outro seja de apenas alguns milissegundos, nem sempre o modelo consegue fazer a segmentação correta. Dessa forma, é possível que um modelo reconheça “Mande um beijo **para** Amanda” em vez do esperado. Isso não quer dizer que os modelos nunca irão acertar o trecho “para a”. Como mostrado nas seções anteriores, há outros fatores que não apenas a correspondência grafema-fone em jogo no reconhecimento de fala (por exemplo, a distribuição das palavras na língua dada pelo LM).

Exemplo 2.3.

Quero instalar o WhatsApp

Algo semelhante poderia acontecer com Exemplo 2.3, em que os modelos podem ter dificuldade em reconhecer o artigo “o” pelo fato de a vogal [o] átona ser bastante próxima em qualidade da semivogal de “wa” em “WhatsApp” e de ambas serem produzidas em coarticulação. É possível que uma saída para a transcrição automática dessa sentença fosse “Quero instalar WhatsApp”.

Esses dois exemplos têm outro ponto em comum: ambas as possibilidades são bastante banais e frequentes na língua. Tanto “para” quanto “para a” são formas muito usadas em qualquer contexto. O mesmo vale para “instalar WhatsApp” e “instalar o WhatsApp”. As duas formas são muito comuns. Isso dificulta a resolução do problema por meio de uma interpolação com um modelo de língua, por exemplo, uma vez que as formas com e sem artigo provavelmente serão bem próximas em probabilidade de ocorrência.

Outro caso de semelhança fonética que confunde um modelo de ASR é o par “no/do” (e suas variações). Pelo fato de as duas preposições poderem ocorrer nos mesmos contextos e ainda serem formadas de apenas dois fones muito parecidos, a sua distinção não é trivial



para o modelo. Desse modo, uma sentença como “vou buscar um trabalho **na** escola” pode facilmente ser reconhecida como “vou buscar um trabalho **da** escola”. É claro que isso depende também do quão articulada a fala é e também da qualidade do áudio, e da presença ou ausência de ruído.

Todos os casos relatados nesta seção não constituem, a priori, erros graves de reconhecimento de fala, uma vez que não alteram o significado das sentenças em questão de maneira drástica. Apesar disso, como dito na Seção 2.2.2.7, a principal métrica utilizada na avaliação de um modelo de ASR não faz nenhum tipo de discriminação entre as palavras, e considera todas de igual peso. Embora a princípio um pouco injusta, essa prática se explica pelo fato de que seria necessário algum trabalho etiquetador para identificar as palavras relevantes nas sentenças. Talvez a classificação binária entre palavras de conteúdo versus palavras gramaticais (Capítulo 4) não fosse suficiente para todos os casos. Poderia haver, por exemplo, algum caso em que “na” e “da” trouxessem uma distinção decisiva de significado. Talvez por isso ainda seja mais viável manter todas as palavras com o mesmo status durante a avaliação.

2.2.3 Síntese de fala

Síntese de fala é o processo de conversão de texto ortográfico para áudio. Nos sistemas de conversão texto-fala ocorre um mapeamento de sequências de letras para formas de ondas sonoras.

Comumente utilizado por softwares de acessibilidade, módulos de atendimento automático e assistentes virtuais, os sistemas de conversão texto-fala têm suas unidades acústicas segmentadas e concatenadas conforme informações de transcrição fonética do texto que se deseja sintetizar, transformando então aquela sentença em sinal acústico.

Um TTS (do inglês, *text-to-speech*) pode ser dividido em duas etapas: a primeira, chamada de análise do texto, onde o texto de entrada é normalizado e transcrito da forma ortográfica para a fonológica; e a segunda, síntese do sinal, onde ocorre a concatenação das unidades fonológicas e a inserção da prosódia. Vamos detalhar cada uma destas etapas a seguir.

2.2.3.1 Análise do texto

Na etapa de Análise do texto o objetivo é decodificar o texto de entrada e prepará-lo para ser convertido em áudio. Essa etapa, também conhecida como pré-processamento, pode ser dividida em outras duas tarefas: a normalização, que expande o texto de entrada para a sua forma literal; e a segunda, que converte o texto já expandido para fonemas, ou representações de pronúncia, e o entrega para a etapa seguinte.

2.2.3.1.1 Normalização

Ao receber o texto a ser sintetizado o sistema de TTS, nesse primeiro estágio, a tarefa é normalizar a sentença de entrada. Nesta etapa normalizar significa substituir elementos do texto como números e abreviaturas, por palavras ou sequência de palavras escritas por extenso. Exemplos são apresentados no Quadro 2.3.

Quadro 2.3. Exemplos de normalização



Texto de entrada	Texto normalizado
1990	mil novecentos e noventa
68,3%	sessenta e oito vírgula três por cento
Av.	avenida
km ²	quilômetros quadrados
2:45PM	Duas e quarenta e cinco da tarde

Algumas classes de normalização têm mais problemas do que outras. As siglas, por exemplo, podem ser lidas letra por letra, como “OMS”, ou como uma única palavra, no caso dos acrônimos, como em “USP”, ou ainda serem expandidas como em “SP – São Paulo”. No português ainda temos o caso do gênero gramatical para casos como dos algarismos 1 e 2, que podem ser expandidos como um/uma e dois/duas, a depender da palavra que vem a seguir. Exemplos são apresentados no Quadro 2.4.

Quadro 2.4. Exemplos de normalização para algarismos

Texto de entrada	Texto expandido
1 mesa	uma mesa
1 copo	um copo
2 dias	dois dias

Para a etapa de normalização, a divisão por categoria pode tornar a organização do trabalho mais fácil e/ou intuitiva. Assim, dá-se conta de casos a serem normalizados por categorias ou classes de fenômenos linguísticos de acordo com suas ocorrências na língua. Algumas categorias de normalização no português brasileiro são apresentadas no Quadro 2.5.

Quadro 2.5. Categorias de normalização no português brasileiro

Categoria	Exemplos
Abreviaturas	“Av.” “Ltda.” “Sra.”
Siglas	“SP” “USP” “IR”
Números cardinais	“102” “1.500” “18,20”
Números ordinais	“2 ^o ” “3 ^a ”
Datas	“08/07/2014” “1 ^o de maio de 1.886”
Valores monetários	“R\$ 32,50” “US\$ 150,00” “R\$ 149,6 bilhões”
Links/URLs	“www.google.com”
Porcentagem	“12,8%” “21,3%”
Unidades de medida	“3cm” “8,515 mi Km ² ” “100km/h” 45

A tarefa de normalização do texto pode ser feita com a utilização de duas diferentes técnicas: (1) É possível optar por desenvolvê-la por meio de regras: muitas vezes utilizando-se de expressões regulares, tais regras são descritas de modo a analisar o texto *token* a *token* e buscar padrões compatíveis no texto. Uma vez que um padrão do texto dá



match com uma regra descrita, a regra cuida de substituir o *token* em questão por seu correspondente por extenso. Modelos de TTS mais robustos contam com sistemas como o Kestral de (Ebden; Sproat, 2014) que também é baseado em regras, mas primeiro classifica e analisa cada entrada do texto e depois produz um novo texto usando uma gramática de verbalização. O normalizador desenvolvido com base em regras tem a vantagem de não depender de dados de treinamento anotados, mas as regras podem se tornar complexas e frágeis, além de carecer de escritores especializados para mantê-las.

Há também normalizadores baseados em redes neurais (2) chamados de modelo codificador-decodificador, que demonstram melhor funcionamento se comparados aos normalizadores baseados em regras, mas que demandam grandes conjuntos de dados anotados.

Além das etapas aqui apresentadas, a síntese de fala ainda passa pela etapa de conversão grafema-fonema, treinamento da voz e validação do modelo treinado. A etapa de conversão grafema-fonema para o português brasileiro é comumente realizada com uso de regras descritas de modo a mapear as letras do alfabeto para o som correspondente a ela, de acordo com o contexto em que tal letra aparece. Já os treinamentos do modelo de voz, por muitos anos feitos por meio de métodos estatísticos (*Hidden Markov Models* – HMMs), hoje são comumente realizados com o uso de redes neurais, método conhecido como Tacotron2 integrado à LPCnet. A avaliação de qualidade e acurácia desses modelos é feita por meio de uma medida numérica baseada na opinião pessoal de humanos, o Mean Opinion Score (MOS) é uma classificação de qualidade de voz. O teste consiste em humanos falantes nativos do idioma ouvirem e atribuírem uma nota entre 1 (ruim) e 5 (excelente) para áudios sintetizados a partir do modelo a ser avaliado. A média das notas atribuídas aos áudios sintéticos passam a ser a nota da avaliação do modelo. Ainda muito dependentes da impressão dos avaliadores humanos, a acurácia dos modelos assim treinados ainda não é mensurada numericamente, ou seja, com avaliações automáticas e objetivas, o que torna a validação das tecnologias hoje empregadas na área bastante dependentes das percepções dos avaliadores.

2.2.4 Considerações finais

Neste capítulo, vimos um pouco sobre a história do processamento de fala, sobre as características da língua falada e sobre as principais tarefas da área de processamento de fala, que são o reconhecimento automático e a síntese de fala. Esperamos ter conseguido demonstrar no que o processamento de fala difere do processamento de texto e quais são os seus principais desafios. De maneira semelhante ao que ocorre no processamento de texto, há carência de dados de qualidade para o processamento do português brasileiro em comparação com o cenário do processamento do inglês. Atualmente, os modelos de reconhecimento de fala *end-to-end*, que são o estado da arte, necessitam de uma quantidade muito grande de dados para que seja obtida uma qualidade de ponta. Os modelos de síntese, por sua vez, necessitam de menos horas de fala, porém a qualidade das gravações precisa ser impecável e há a necessidade de se gravar a mesma pessoa, o que aponta para um custo elevado, tanto financeiro quanto de tempo.

Conforme demonstrado na Seção 2.2.2.1, em se tratando de ASR, é necessário considerar variações dialetais, tanto de pronúncia quanto de vocabulário e sintaxe, durante o treinamento dos modelos. Os dados precisam ser suficientemente variados e representativos de cada variedade a fim de que um sistema genérico o bastante para dada língua seja desenvolvido. Isso não ocorre no processamento de texto nas mesmas proporções.



Especialmente quando comparamos o português europeu com o brasileiro no que diz respeito ao reconhecimento, e também à síntese de fala, por serem variedades muito diversas, especialmente foneticamente, seria preciso construir sistemas de ASR separados para processar as duas línguas. No processamento de texto, diferentemente, pelo fato de a língua escrita ser mais conservadora, as duas variedades se aproximam, embora cada uma continue tendo suas peculiaridades de grafia, vocabulário e sintaxe. O impacto da distância entre as variedades se torna mais evidente na síntese de fala, uma vez que um sistema desenvolvido para o português europeu não seria bem aceito por falantes brasileiros residentes no Brasil. Basta pensar no quão estranho seria utilizar um assistente virtual que falasse português europeu. Apesar de todas essas considerações, vemos despontar, nos últimos meses, modelos de reconhecimento e de síntese de fala treinados com várias línguas. Os estudos de Yang et al. (2023), Pratap et al. (2020a) e Saeki et al. (2023) explicam como essas técnicas funcionam. Esse tópico é bastante interessante e será objeto de uma próxima edição deste capítulo.

Além do reconhecimento e da síntese de fala, há várias outras tarefas na área de processamento de fala. Podemos elencar aqui as seguintes: clonagem de voz, detecção de palavras-chave, identificação de falantes, diarização da fala, entre outras. O Capítulo 3 deste livro tratará de recursos para o desenvolvimento dessas e de outras tarefas do processamento de fala e também apresentará uma breve descrição de cada uma.



Capítulo 3

Recursos para o processamento de fala

*Edresson Casanova
Vinícius G. Santos
Flaviane R. Fernandes Svartman
Marli Quadros Leite
Arnaldo Candido Jr.
Ricardo M. Marcacini
Solange O. Rezende
Sandra Maria Aluísio*

 <https://brasileiraspln.com/livro-pln/1a-edicao/>

3.1 Introdução

Até a metade de 2020, o português brasileiro (PB) possuía apenas algumas dezenas de horas de dados de fala públicos ou abertos para pesquisas acadêmicas, disponíveis para treinar modelos para os sistemas mais comuns, que são os reconhecedores automáticos de fala (em inglês, *Automatic Speech Recognition* ou ASR) e os sintetizadores de fala (em inglês, *Text-to-Speech Synthesis* ou TTS). Havia um grande contraste com a língua inglesa, cujos recursos eram **maiores tanto em número de horas quanto em número de locutores** e, assim, mais adequados à aplicação de métodos de aprendizado profundo de máquina, chamados de *deep learning*, em inglês.

Para o treinamento de modelos de reconhecimento de fala, havia aproximadamente 60 horas, divididas em quatro pequenos conjuntos de dados de fala lida (em inglês, *read speech*), isto é, uma fala preparada para ser lida, em contraste com a fala espontânea: (1) o Common Voice Corpus versão 5.1 (da Mozilla)¹ (2) o *dataset* Sid, (3) o VoxForge e (4) o LapsBM 1.4². Para o treinamento de modelos de síntese de fala, havia um conjunto de dados de um único locutor com 10 horas e 28 minutos de fala, chamado TTS-Portuguese Corpus³.

A fala espontânea possui fenômenos que tornam o seu reconhecimento mais complexo do que o da fala lida, como as pausas preenchidas e as disfluências de edição. Exemplos de projetos que tratam da fala lida são o Librivox⁴, que distribui os livros de domínio público em formato de áudio. Estes áudios têm sido usados em vários projetos para criação de recursos para processamento de fala em inglês como o LibriSpeech ASR Corpus⁵ e o LibriTTS⁶, ambos alocados no repositório *Open Speech and Language Resources*. O

¹<https://commonvoice.mozilla.org/pt/>

²O *dataset* Sid, o VoxForge e o LapsBM 1.4 estão disponíveis em: <https://igormq.github.io/datasets/>

³<https://github.com/Edresson/TTS-Portuguese-Corpus>

⁴<https://librivox.org/pages/about-librivox/>

⁵<https://www.openslr.org/12>

⁶<https://www.openslr.org/60/>



LibriSpeech é um grande *corpus* de fala lida em inglês, com 1.000 horas, destinado a pesquisas de reconhecimento automático de fala. O LibriTTS é um *corpus* multilocutor derivado do LibriSpeech para pesquisas em síntese de fala, com 585 horas.

Nesse cenário de escassez de dados públicos de fala em PB para treinamento de sistemas de processamento de fala, foi concebido, em agosto de 2020, o projeto TaRSila⁷ do Center for Artificial Intelligence⁸ da Universidade de São Paulo, financiado pela IBM e FAPESP. O projeto TaRSila visa a aumentar os conjuntos de dados de fala em PB tanto para treinamento de sistemas como também para pesquisas linguísticas nas seguintes **tarefas** do processamento de fala:

1. reconhecimento automático de fala espontânea, que transcreve automaticamente a fala espontânea, por exemplo, diálogos, entrevistas e conversas informais;
2. síntese de fala multilocutor expressiva, que gera vozes de diferentes locutores/falantes de maneira próxima a um falante humano, a partir de um texto;
3. clonagem de voz. A clonagem de voz engloba duas grandes tarefas do processamento de fala: a síntese de fala e a conversão de voz. O objetivo da clonagem é copiar a voz de um locutor e gerar novas amostras de áudio utilizando-se de características da voz do locutor. Existem diferentes métodos de clonagem, sendo os mais interessantes os de síntese de fala multilocutor *zero-shot* e os métodos de conversão de voz *zero-shot* que conseguem clonar a voz de um locutor utilizando apenas alguns segundos de fala;
4. modelagem de tópicos a partir das transcrições dos áudios, que é útil para organizar, resumir e visualizar o conteúdo de vídeos em tópicos similares;
5. predição da pontuação de segmentos de fala e predicação da capitalização, isto é, quais palavras devem ser escritas com a primeira letra maiúscula. Essas tarefas são importantes para facilitar o entendimento humano do texto de uma transcrição automática. Também são importantes para o encadeamento de sistemas, quando o ASR é usado antes de um sistema de tradução automática ou de um sistema de reconhecimento de entidades nomeadas (em inglês, *Name Entity Recognition* ou *NER*). Exemplos de entidades nomeadas são nomes de pessoas, de lugares, e de organizações, entre outras;
6. segmentação prosódica da fala espontânea em unidades prosódicas terminais e não terminais, que transmitem a ideia de conclusão do enunciado e a ideia de não conclusão, respectivamente. Ajuda a analisar o conteúdo de uma sequência falada, dados os vários sentidos possíveis, e facilita a criação de conjuntos de dados para treinamento de ASRs para fala espontânea, por indicar o limite de uma sequência de fala e, conseqüentemente, auxiliar na predição da pontuação (tarefa citada acima);
7. reconhecimento de emoções a partir da fala (em inglês, *Speech Emotion Recognition* ou *SER*). Reconhece o estado emocional de um locutor a partir de sua fala e é útil para muitas aplicações, como o desenvolvimento de ferramentas de diagnóstico para terapeutas, assistentes de voz e ferramentas para análise de comunicações em call centers.

Além das sete tarefas acima em estudo no TaRSila, o livro sobre Processamento de Fala⁹ (Bäckström et al., 2022) apresenta outras tarefas típicas, como o reconhecimento e verificação de locutor, a restauração de fala e a diarização:

⁷<https://sites.google.com/view/tarsila-c4ai/>

⁸<http://c4ai.inova.usp.br/>

⁹<https://speechprocessingbook.aalto.fi/>.



- reconhecimento de locutor e verificação de locutor, que se referem, respectivamente, à identificação do locutor (quem está falando?) e à verificação se o locutor é quem afirma ser;
- a restauração (ou aprimoramento) da fala refere-se à melhoria de uma gravação de um sinal de fala para, por exemplo, remover o ruído de fundo ou o efeito da acústica do ambiente;
- a diarização da fala é o processo de segmentar uma conversa de vários falantes em segmentos contínuos de um único falante.

O livro coloca o reconhecimento de emoções em áudios (citada acima) no grupo das tarefas de análise paralinguística, dado que extrai do sinal de áudio informação não linguística (diferente da pontuação, por exemplo) e não relacionada à identidade de um locutor, como fazem as tarefas de reconhecimento e verificação de locutor.

Neste capítulo, apresentamos os recursos de fala criados nos três primeiros anos do projeto TaRSila para ilustrar várias das tarefas da área de processamento de fala, acima elencadas, que são definidas e exemplificadas em cada seção. Nesse percurso, fazemos um contraste com a língua inglesa que possui mais recursos para cada tarefa, citando os recursos disponibilizados na literatura tanto para o inglês como para o português.

Os vários recursos desenvolvidos no TaRSila têm o prefixo CORAA (CORpus de Áudios Anotados), que é um grande *corpus* multipropósito do português brasileiro no qual os arquivos de áudios estão alinhados com transcrições que foram (ou estão sendo) manualmente validadas para cada tarefa estudada no TaRSila.

O alinhamento de um trecho de áudio com a transcrição correspondente indica o tempo de início e o tempo final do trecho (também chamado de minutagem do áudio) (veja a Figura 3.1), formando pares usados no aprendizado supervisionado de um modelo de reconhecimento de fala. O reconhecedor Whisper¹⁰ da OpenAI, por exemplo, foi treinado dessa forma. O uso de pares áudio-transcrição para treinamento de reconhedores de fala não é a única abordagem para a tarefa, que também pode ser feita por aprendizado não supervisionado, como é o caso, por exemplo, do wav2vec-U¹¹. Essa é uma abordagem na qual o aprendizado dispensa a necessidade de transcrições, e ocorre apenas por meio de áudio. Em todo caso, para a avaliação do desempenho de um reconhecedor, é importante que haja os pares áudio-transcrição.

Figura 3.1: Excerto do inquérito SP_D2_255 do NURC-SP com dois trechos (text) e indicação do tempo de início (xmin) e tempo final (xmax) em segundos no formato TextGrid. A anotação “...” refere-se a uma pausa silenciosa.

```
intervals [1]:
  xmin = 0
  xmax = 0.3575589499895925
  text = "..."/>
intervals [2]:
  xmin = 0.3575589499895925
  xmax = 5.103648343063277
  text = "bem nós gostaríamos de começar esta nossa conversa falando sobre transportes e viagens"
```

Começamos apresentando, na Seção 3.2, o TTS-Portuguese Corpus, *corpus* para treinamento de modelos de síntese de fala, criado e disponibilizado no início de 2020 com a fala de um único locutor. Esse *corpus* permitiu avançar pesquisas sobre síntese de fala, conversão de voz e uma abordagem de aumento de dados para treinar modelos

¹⁰<https://github.com/openai/whisper/>

¹¹<https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec/unsupervised>



de reconhecedores de fala em cenários de baixos recursos de dados. Na Seção 3.3, apresentamos o *corpus* CORAA NURC-SP, que contém 334 horas de fala espontânea e fala preparada de falantes de São Paulo, capital, divididas em uma parte com áudios e transcrições manuais não alinhadas originalmente (47 inquéritos) e outra parte de áudios somente (328 inquéritos). Na Seção 3.4, apresentamos o *corpus* CORAA ASR versão 1.1, composto por quatro *corpora* disponíveis na literatura, que foram validados para a tarefa de ASR, e uma coleção de TeD Talks, totalizando aproximadamente 290 horas. Na Seção 3.5, apresentamos o CORAA SER versão 1.0, composto por aproximadamente 50 minutos de segmentos de áudio rotulados em três classes: neutro, não neutro feminino e não neutro masculino, sendo que a classe neutra representa segmentos de áudio sem estado emocional bem definido e as classes não neutras representam segmentos associados a um dos estados emocionais primários da fala do locutor. Finalmente, na Seção 3.6, apresentamos o *corpus* do Museu da Pessoa (MuPe), com 300 horas de áudios de histórias de vida e transcrições com pontuação, que foi cedido ao TaRSila em um convênio entre o Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP), Universidade Federal de Goiás (UFG) e Museu da Pessoa. O futuro *corpus*, após ser anotado e anonimizado para as várias tarefas em estudo, será denominado CORAA MuPe. Na Seção 3.6 apresentamos o *dataset* de teste do CORAA MuPe, com aproximadamente 17 horas, e que foi usado para a avaliação da tarefa de predição de pontuação do ASR Whisper da OpenAI¹². Finalizamos o capítulo com a apresentação dos recursos futuros que serão criados ou expandidos a partir dos já descritos neste capítulo (Seção 3.7).

3.2 Recursos para síntese de fala

Os sistemas de síntese de fala receberam muita atenção a partir de 2017 devido ao grande avanço proporcionado pela aplicação do *deep learning* (Goodfellow; Bengio; Courville, 2016) nessa área, que permitiu a popularização e aprimoramento de assistentes virtuais, como Apple Siri (Gruber, 2009), Amazon Alexa (Purinton et al., 2017) e Google Home (Dempsey, 2017).

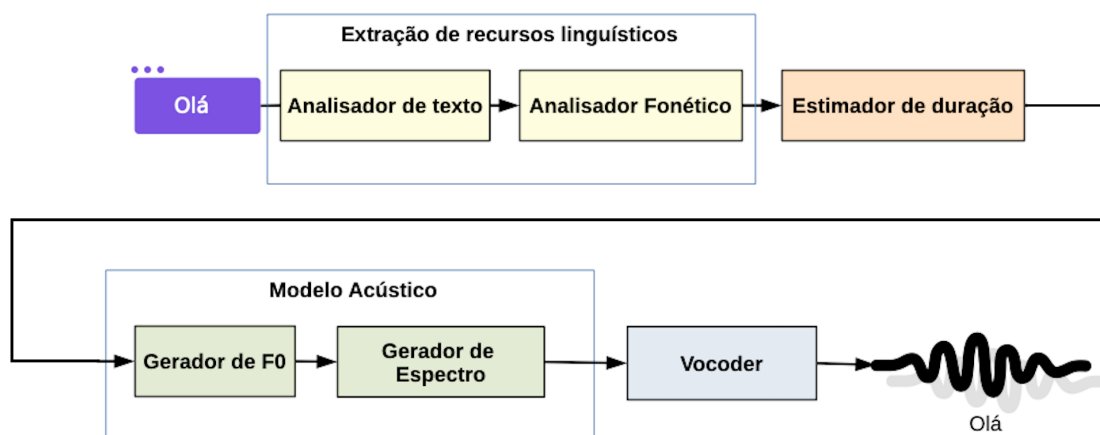
Os sistemas tradicionais de síntese de fala foram muito utilizados até 2010. Entretanto, a partir dessa data, a síntese de fala baseada em redes neurais tornou-se gradualmente o método dominante e alcançou uma qualidade de áudio superior aos sistemas tradicionais (Tan et al., 2021). De acordo com Tachibana; Uenoyama; Aihara (2017), os sistemas tradicionais de síntese de fala não são fáceis de se desenvolver, devido ao fato de serem compostos por muitos módulos específicos, tais como um analisador de texto, um analisador fonético, um estimador de duração, um modelo acústico e um *vocoder*. A Figura 3.2 apresenta um diagrama com os principais componentes de um sistema de síntese de fala tradicional (Casanova; Shulby; Aluísio, 2021). Vários trabalhos na literatura exploraram essa abordagem clássica (Braude; Shimodaira; Youssef, 2013; Charpentier; Stella, 1986; Klatt, 1980; Siddhi; Verghese; Bhavik, 2017; Teixeira; Freitas; Fujisaki, 2003; Tokuda et al., 2000; Wang; Georgila, 2011; Ze; Senior; Schuster, 2013).

O advento do *deep learning* permitiu a integração dos módulos específicos dos sistemas de síntese de fala tradicionais em um único modelo. Apesar dos modelos baseados em *deep learning* serem às vezes criticados devido à dificuldade de interpretá-los, vários sistemas de síntese de fala baseado em *deep learning* (Kim et al., 2020; Kim; Kong; Son, 2021; Kyle;

¹²<https://openai.com/research/whisper>



Figura 3.2: Os principais componentes de um sistema de síntese de fala tradicional.



Fonte: Adaptado de (Casanova; Shulby; Aluísio, 2021, fig. 14.1, p. 184)

Jose; Sotelo, 2017; Ping et al., 2017; Shen et al., 2018; Tachibana; Uenoyama; Aihara, 2017; Valle et al., 2020; Wang et al., 2023; Wang et al., 2017) demonstraram a capacidade de sintetizar fala com uma qualidade muito promissora, superior, inclusive, à dos sistemas tradicionais.

Modelos baseados em *deep learning* requerem uma quantidade maior de dados para treinamento, portanto, idiomas com poucos recursos disponíveis ficam prejudicados. Por esse motivo, a maioria dos modelos de síntese de fala atuais são projetados para o inglês (Kim et al., 2020; Kim; Kong; Son, 2021; Ping et al., 2017; Shen et al., 2018; Valle et al., 2020; Wang et al., 2023), que é um idioma com muitos recursos disponíveis publicamente.

Para o inglês existem vários *corpora* que podem ser utilizados para treinar modelos de síntese de fala baseados em *deep learning*, por exemplo, os *corpora* VCTK (Veaux et al., 2017), LJ Speech (Ito, 2017), LibriTTS (Zen et al., 2019) e LibriTTS-R (Koizumi et al., 2023a).

O *corpus* VCTK (Veaux et al., 2017) é composto por 44 horas de fala de 108 locutores, sendo 61 do sexo feminino e 47 do sexo masculino. Além disso, o *corpus* inclui amostras de 11 variedades linguísticas do inglês, sendo elas: britânico, americano, canadense, neozelandês, sul-africano, australiano, escocês, norte-irlandês, irlandês, indiano e galês. A taxa de amostragem dos áudios presentes nesse *corpus* é de 48 kHz. O *corpus* LJ Speech (Ito, 2017) foi derivado de audiolivros e tem aproximadamente 24 horas de fala de uma locutora profissional em uma taxa amostragem de 24 kHz. LJ Speech é um dos *corpora* abertos mais populares para síntese de fala de um único locutor. O *corpus* LibriTTS (Zen et al., 2019) também foi derivado de audiolivros e possui 585 horas de fala de 2456 locutores, sendo 1185 do sexo feminino e 1271 do sexo masculino. A taxa de amostragem dos áudios presentes nesse *corpus* é de 24 kHz. Por outro lado, o *corpus* LibriTTS-R (Koizumi et al., 2023a) foi criado com a aplicação do modelo de restauração de fala (em inglês, *speech restoration*) Miipher (Koizumi et al., 2023b) no *dataset* LibriTTS. As amostras do LibriTTS-R são idênticas às do LibriTTS, com apenas a qualidade de som melhorada. Os resultados dos experimentos mostraram que os modelos de síntese de fala treinados com o LibriTTS-R apresentaram qualidade significativamente melhor em comparação com os modelos treinados no LibriTTS.

Para a língua portuguesa, até meados em 2019, não havia nenhum *corpus* disponível



publicamente com quantidade de horas e qualidade de áudio suficientes para treinar modelos de síntese de fala baseados em *deep learning*. Embora um *corpus* para síntese de fala em português europeu tenha sido disponibilizado em 2001, pelo fato de ele ter duração de aproximadamente 100 minutos apenas (Teixeira et al., 2001), não é possível utilizá-lo no treinamento de modelos baseados em *deep learning*.

Para suprir essa carência de dados para síntese de fala no português brasileiro, em 2019, a coleta do *corpus* TTS-Portuguese Corpus (Casanova, 2019; Casanova et al., 2022, 2022) foi iniciada (Casanova, 2019). Posteriormente, em 2020, o *corpus* foi tornado público (Casanova et al., 2022) e os detalhes de sua compilação foram publicados em um artigo (Casanova et al., 2022).

Para a construção do TTS-Portuguese Corpus, foram utilizados textos de domínio público. Inicialmente, buscando alcançar um vocabulário amplo, extraíram-se todos os artigos das seções de destaques da Wikipédia (da época em que foi compilado) para todas as áreas do conhecimento. Após essa extração, os artigos foram segmentados em sentenças (considerando-se a pontuação textual). Durante as gravações, o locutor recebeu sentenças desse conjunto, que foram escolhidas de forma aleatória. Além disso, foram utilizados os 20 conjuntos de sentenças foneticamente balanceadas, cada conjunto contendo 10 sentenças, propostas por Seara (1994). Por fim, para aumentar o número de perguntas e introduzir um discurso mais expressivo, foram ainda utilizadas frases do Chatterbot-corpus¹³, um *corpus* criado originalmente para a construção de *chatbots*. Desse modo, o TTS-Portuguese Corpus possui um vocabulário amplo com palavras de diversas áreas. Além disso, também possui uma representação de fala expressiva com o uso de perguntas e respostas de um conjunto de dados de *chatbot*.

A gravação do TTS-Portuguese Corpus foi realizada por um locutor masculino, nativo do português brasileiro, não profissional, em ambiente silencioso, mas sem isolamento acústico devido às dificuldades de acesso a estúdio de gravação. Todos os áudios foram gravados com frequência de amostragem de 48 kHz e resolução de 32 bits. No *corpus*, cada arquivo de áudio possui sua respectiva transcrição textual (a transcrição fonética não foi fornecida). O TTS-Portuguese Corpus consiste em um total de 71358 palavras faladas, 13311 palavras únicas, resultando em 3632 arquivos de áudio e totalizando 10 horas e 28 minutos de fala. Os arquivos de áudio variam em duração de 0.67 a 50.08 segundos (Casanova et al., 2022).

Em paralelo com o TTS-Portuguese Corpus, foram lançados dois conjuntos de dados para reconhecimento automático de fala do português, com boa qualidade. O primeiro, o CETUC (Alencar; Alcaim, 2008), disponibilizado publicamente por Quintanilha; Netto; Biscainho (2020), tem aproximadamente 145 horas de fala de 100 locutores. Nesse *corpus*, cada locutor pronunciou mil sentenças foneticamente balanceadas extraídas de textos jornalísticos; em média, 1,45 horas gravadas por locutor. Já o segundo, o *corpus* Multilingual LibriSpeech (MLS) (Pratap et al., 2020b), é derivado dos audiolivros LibriVox e abrange 8 idiomas, incluindo o português. Para o português, os autores disponibilizaram aproximadamente 130 horas de fala provenientes de 54 locutores, uma média de 2.40 horas de fala por locutor. Embora a qualidade de ambos os *corpora* seja boa, os dois foram disponibilizados com uma taxa de amostragem de 16 kHz e não possuem pontuação em seus textos, dificultando a aplicação desses *corpora* para síntese de fala. Além disso, a quantidade de fala de cada locutor nos dois *corpora* é baixa, o que torna difícil criar um conjunto de dados com um vocabulário grande o suficiente para síntese de fala de um único locutor.

Além disso, mais recentemente, o *corpus* CML-TTS (Oliveira et al., 2023) foi proposto.

¹³<https://github.com/gunthercox/chatbot-corpus/>



O CML-TTS é baseado no *corpus* Multilingual LibriSpeech (MLS) e foi adaptado para treinamento de modelos de síntese de fala. O CML-TTS é composto por audiolivros em sete idiomas: holandês, francês, alemão, italiano, português, polonês e espanhol. Os autores recriaram o *corpus* MLS mantendo a pontuação e os áudios com uma taxa de amostragem de 24 kHz. Amostras que não atendiam às especificações descritas anteriormente foram descartadas. Para o português, após a filtragem, os autores obtiveram aproximadamente 69 horas de fala, provenientes de 31 homens e 17 mulheres.

A Tabela 3.1 apresenta as estatísticas dos recursos disponíveis para síntese de fala do inglês e do português.

Tabela 3.1: Estatísticas dos recursos disponíveis para síntese de fala nas línguas inglesa e portuguesa.

<i>Corpora</i>	Idioma	Duração	Falantes	Disponibilizado
VCTK	inglês	44 horas	108	2017
LJ Speech	inglês	24 horas	1	2017
LibriTTS	inglês	585 horas	2456	2019
LibriTTS-R	inglês	585 horas	2456	2023
TTS-Portuguese Corpus	português	10.4 horas	1	2019
CETUC	português	145 horas	100	2020
MLS	português	130 horas	54	2020
CML-TTS	português	69 horas	48	2023

3.3 Recursos para segmentação prosódica

Existem vários estudos na literatura de processamento de fala com foco na detecção de fronteiras prosódicas nas línguas naturais (Ananthakrishnan; Narayanan, 2008; Huang; Hasegawa-Johnson; Shih, 2008; Jeon; Liu, 2009; e.g. Wightman; Ostendorf, 1991). Para o inglês, entre os recursos frequentemente utilizados em aplicações que consideram fronteiras prosódicas, podemos citar o Santa Barbara Corpus of Spoken American English (Du Bois et al., 2000–2005) e o Boston University Radio Speech Corpus (Ostendorf; Price; Shattuck-Hufnagel, 1995). O primeiro contém ≈ 20 horas de fala espontânea de gêneros variados, transcritas e segmentadas manualmente em unidades entoacionais final e não final (Du Bois et al., 1992). Já o segundo contém 10 horas de notícias de rádio, das quais 3,5 horas estão prosodicamente anotadas de acordo com o sistema ToBI (Beckman; Hirschberg; Shattuck-Hufnagel, 2005).

Para o português brasileiro, trabalhos desenvolvidos no âmbito do projeto C-ORAL–Brasil avançam os estudos para a detecção automática de fronteiras prosódicas na fala espontânea a partir de parâmetros fonético-acústicos e fronteiras identificadas perceptualmente por anotadores treinados (Raso; Teixeira; Barbosa, 2020; Teixeira, 2022; Teixeira; Mittman, 2018; Teixeira; Barbosa; Raso, 2018). Os estudos utilizam excertos de fala monológica masculina (8–24 minutos de áudio e 1339–3697 palavras), provenientes dos *corpora* anotados C-ORAL–Brasil I e II (Mello; Raso; Almeida Ferrari, no prelo; Raso; Mello, 2012a). No âmbito do projeto TaRSila, o CORAA NURC-SP, que vem sendo preparado tanto para viabilizar estudos linguísticos quanto o processamento computacional, contará com ≈ 334 horas de fala transcrita, das quais pelo menos 40 horas serão prosodicamente anotadas.

O CORAA NURC-SP toma como base dados provenientes do projeto acadêmico NURC–Norma Urbana Linguística Culta, que foi iniciado em 1969 com o objetivo de documentar e



estudar a língua portuguesa falada por pessoas com ensino superior completo, denominadas ‘cultas’, de cinco capitais brasileiras: Recife, Salvador, Rio de Janeiro, São Paulo e Porto Alegre. O projeto resultou num grande *corpus* (aprox. 1.570 horas, 2.356 falantes) reunido ao longo dos anos 1970 e 1980 (Castilho, 1990).

Como em todas as capitais, o NURC-São Paulo (NURC-SP)¹⁴ reúne mais de 300 horas de gravação, apresentando falantes com nível superior; nascidos e criados na cidade; filhos de falantes nativos de português; igualmente divididos em homens e mulheres; e distribuídos em três faixas etárias (25–35, 36–55 e 56 anos em diante). As gravações foram realizadas em três situações, gerando diferentes gêneros discursivos: palestras/aulas em contexto formal proferidas por um locutor (EF); diálogos entre documentadores e um participante (DID); e diálogos entre dois participantes mediados por documentadores (D2).

O *corpus* do projeto NURC tem sido amplamente utilizado para estudar vários aspectos da língua falada, tendo se tornado um dos *corpora* mais influentes da linguística brasileira. A maioria dos estudos deriva de transcrições de pequenos *subcorpora* compartilhados por pesquisadores que trabalham em cada capital (Castilho, 1990, 2021a), aqui referidos como *corpus* mínimo. Assim, a contraparte de áudio era normalmente desconsiderada devido à dificuldade de acesso às fitas magnéticas de rolo nas quais as gravações foram feitas. Recentemente, um protocolo para digitalizar, anotar, armazenar e divulgar o material do acervo do NURC-Recife, o NURC Digital (Oliveira Jr., 2016), foi desenvolvido e completamente implementado. Inspirados nesse protocolo, desenvolvemos, no âmbito do projeto TaRSila, um processo para o alinhamento texto-fala do Corpus Mínimo do NURC-SP.

Embora os procedimentos que orientam o processamento do NURC-SP sejam baseados no protocolo do NURC Digital, eles incorporam sistemas de processamento de fala que incluem, por exemplo, um reconhecedor automático de fala atual (Whisper¹⁵), um alinhador forçado áudio-transcrição baseado em síntese de fala (aeneas¹⁶) e alinhadores fonéticos automáticos (Batista; Dias; Neto, 2022; Kruse; Barbosa, 2021) usados em conjunto com um método para a segmentação automática de fala baseada em prosódia (Biron et al., 2021).

A versão CORAA do NURC-SP é composta por 375 inquéritos (aprox. 334 horas de gravação), dos quais alguns já tinham transcrições — mas, até então, não alinhadas ao áudio — e a grande maioria é composta apenas de áudio. No âmbito do TaRSila, o NURC-SP foi dividido em três *subcorpora* de trabalho:

- o Corpus Mínimo (21 gravações + transcrições), que está sendo utilizado para avaliar atualmente os métodos de processamento de todo o acervo;
- o Corpus de Áudios e Transcrições Não Alinhados (26 gravações + transcrições), que está sendo segmentado automaticamente em unidades prosódicas pelo método de Biron et al. (2021), adaptado ao português brasileiro, e validado manualmente; e
- o Corpus de Áudios (328 gravações sem transcrição), que vem sendo transcrito automaticamente pelo ASR Whisper, modelo treinado em 680 mil horas de dados multilíngues coletados da web. Embora outros modelos de reconhecedores automáticos tenham sido avaliados em uma amostra representativa do NURC-SP Corpus Mínimo (Gris et al., 2022), o modelo Whisper foi escolhido, pois, além de realizar uma transcrição de qualidade, consegue colocar pontuações, facilitando a leitura da

¹⁴<https://nurc.fflch.usp.br/>

¹⁵<https://github.com/openai/whisper/>

¹⁶<https://www.readbeyond.it/aeneas/>



transcrição automática. As transcrições estão sendo manualmente validadas para corrigir erros do reconhecedor automático.

Entre esses conjuntos de dados, o Corpus Mínimo é o conjunto que se encontra completamente processado (Santos et al., 2022). Ele está disponível publicamente no repositório do Portulan Clarin¹⁷ e compreende 21 arquivos de áudio e transcrições multiníveis (≈ 18 horas, ≈ 155 mil palavras) alinhadas ao áudio de acordo com unidades prosódicas linguisticamente motivadas abrangendo os três gêneros textuais especificados anteriormente (DID, EF, D2). O conceito de unidade prosódica que utilizamos aqui está fundamentado nos princípios do método de segmentação prosódica do C-ORAL-BRASIL (Raso; Mello, 2012a). Portanto, no fluxo da fala, podemos reconhecer fronteiras de unidades com valores terminais ou não terminais. **Quebras prosódicas terminais** (TB, *terminal break*) marcam sequências terminadas, ou seja, comunicam a conclusão do enunciado, formando a menor unidade pragmaticamente autônoma da fala, enquanto **quebras prosódicas não terminais** (NTB, *non-terminal break*) sinalizam uma unidade prosódica não autônoma e cuja informação não está concluída dentro de um mesmo enunciado. A identificação das quebras prosódicas é baseada principalmente na relevância perceptiva (auditiva) das pistas prosódicas, mas também na inspeção visual da síntese do sinal acústico fornecida pelo Praat (Boersma; Weenink, 2023). As principais pistas para uma quebra prosódica no português brasileiro são a inserção de pausas e mudanças relacionadas à frequência fundamental e à duração (Raso; Teixeira; Barbosa, 2020; Serra, 2009).

As transcrições multiníveis consistem nas seguintes camadas de intervalo anotadas no programa de análise de fala Praat. Veja a Figura 3.3 para uma ilustração:

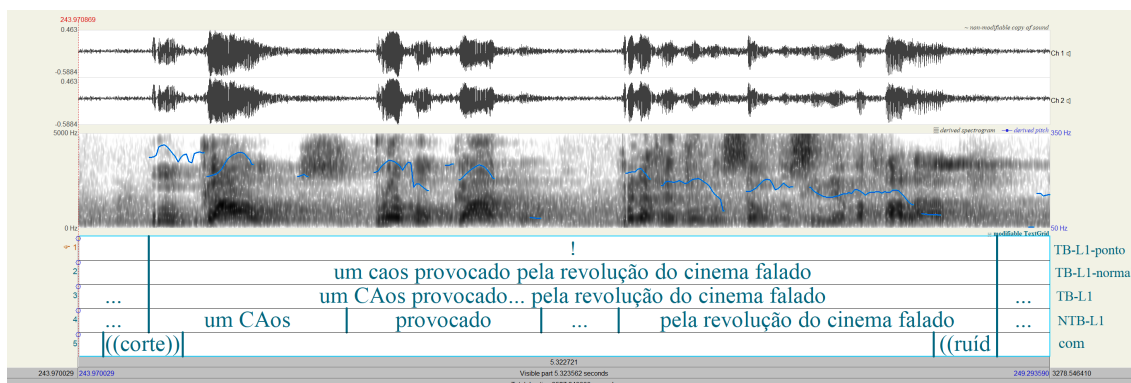
- 2 camadas (TB-, NTB-) nas quais a fala de cada locutor (-L1, -L2) e documentador (-DOC1, -DOC2) é segmentada em unidades prosódicas e transcrita de acordo com normas adaptadas do Projeto NURC.
- 1 camada (LA) para a fala transcrita e segmentada de qualquer locutor aleatório.
- 1 camada para comentários (COM) acerca do áudio e da anotação.
- 1 camada contendo a versão normalizada (-NORMAL) da transcrição de todas as camadas TB e LA.
- 1 camada contendo a pontuação (-PONTO) que finaliza cada TB.

O processamento do Corpus Mínimo do NURC-SP envolveu várias etapas. Em primeiro lugar, os anotadores foram treinados no uso do software Praat e na aplicação das diretrizes de anotação. Paralelamente, foram feitos o alinhamento automático entre o áudio e a transcrição original, usando o aeneas, e a preparação dos arquivos de alinhamento para anotação, que inclui uma revisão ortográfica ampla num editor de texto. Em seguida, foram realizados testes de confiabilidade entre avaliadores para avaliar a segmentação prosódica, com um valor de kappa (Capítulo 14) acima de 0.8 como critério. Na sequência, procedeu-se à anotação, que envolveu: (i) a revisão da transcrição original de acordo com as diretrizes adaptadas do projeto NURC, (ii) a correção do alinhamento automático texto-fala e (iii) a segmentação da fala em unidades prosódicas. Após a conclusão da anotação, os arquivos anotados passaram por uma inspeção realizada por um especialista, em busca de desvios significativos das diretrizes de anotação. Em seguida, a ortografia foi verificada e o texto foi normalizado, a fim de tornar o conjunto de dados adequado para o processamento de

¹⁷Disponível em <https://hdl.handle.net/21.11129/0000-000F-73CA-C> sob a licença CC BY-NC-ND 4.0.



Figura 3.3: Excerto de SP_EF_153 com cinco camadas anotadas no Praat.



linguagem natural. Por fim, foi realizada a anotação da pontuação seguindo as normas ortográficas do português brasileiro.

A relevância de um *corpus* de português brasileiro processado e anotado prosodicamente está no fato de que a delimitação de fronteiras prosódicas melhora o desempenho de sistemas de processamento de línguas naturais (e.g. Chen; Hasegawa-Johnson, 2004; Lin et al., 2016, 2019; Ludusan; Synnaeve; Dupoux, 2015; Yang et al., 2011) e é *input* para a predição de pontuações automáticas. Além disso, é possível usar tal *corpus* como um conjunto de referência para o treinamento de sistemas automáticos de reconhecimento de fala espontânea, detecção de sotaques e *parsing* e, assim, alavancar o desenvolvimento de métodos de processamento de fala do português brasileiro e viabilizar novos estudos linguísticos, dada a sua futura disponibilização integral num portal web que permitirá pesquisas específicas.

3.4 Recursos para reconhecimento automático de fala

A seguir são apresentados grandes *corpora* para a criação de sistemas de reconhecimento de fala voltadas para a língua portuguesa¹⁸. Observa-se que muitos recursos são multilíngues, porém a Tabela 3.2 detalha especificamente as estatísticas para a língua portuguesa. Entre os recursos apresentados, existe uma preponderância um pouco maior da variedade brasileira nos recursos existentes. Porém, a variedade europeia também é contemplada.

Tabela 3.2: Estatísticas aproximadas dos principais recursos disponíveis para ASR na língua portuguesa.

<i>Corpora</i>	Horas	Áudios	Falantes	Licença	Lançamento
CORAA ASR 1.1	289	402.000	1.700	CC BY-NC-ND	2022
Mozilla Common Voice 13.0	197	130.000	3.100	CC-0	2023
MultiLingual LibriSpeech	285	40.000	68	CC BY	2020
MultiLingual TeDx Corpus	164	93.000	-	CC BY-NC-ND	2021
Spotify Podcast Dataset	7.600	123.000	-	Proprietária	2022

CORAA ASR¹⁹ (Candido Junior et al., 2022) é um *corpus* para reconhecimento automático de fala que contém também fala espontânea, um tópico pouco pesquisado em

¹⁸Convém observar que alguns valores são estimativas dos respectivos autores. Muitos projetos estão em atividade, e os valores apresentados devem aumentar com passar do tempo.

¹⁹<https://github.com/nilc-nlp/CORAA>



projetos similares. Esse *corpus* faz parte do *corpus* multi-tarefa CORAA e está inserido no projeto TaRSila. O CORAA ASR é a junção de cinco projetos independentes: (1) ALIP (Gonçalves, 2019); (2) C-ORAL–Brasil I (Raso; Mello, 2012a); (3) NURC-Recife (Oliveira Jr., 2016); (4) SP-2010 (Mendes; Oushiro, 2012); (5) TeDx Talks. Os quatro primeiros projetos foram originalmente criados para análises linguísticas e adaptados para a tarefa de reconhecimento automático de fala. O último é composto de áudios cedidos pela organização TED (The Electronic Development) para a tarefa de reconhecimento e não deve ser confundido com o *corpus* oficial TeDx Talks Brazil, detalhado a seguir, pois existem diferenças entre os áudios disponibilizados. A fala espontânea é mais difícil de ser reconhecida do que a fala preparada, mais comum nos outros projetos, devido à presença mais frequente de fenômenos como pausas preenchidas, hesitações e revisões.

O *corpus* Common Voice²⁰ (Ardila et al., 2019) é um projeto de uso aberto criado pela Fundação Mozilla, responsável pelo navegador Firefox. O projeto é uma resposta à carência de recursos para várias línguas, incluindo o português. No projeto, os usuários podem ao mesmo tempo contribuir para o crescimento da base e acessar áudios de outras pessoas. A proposta de criação de uma grande base colaborativa segue a mesma linha de outros projetos de sucesso em diferentes áreas de aplicação, tais como a Wikipédia e projetos de código aberto de modo geral. Para colaborar com o projeto, os usuários podem doar áudios em suas próprias vozes e revisar doações de outros usuários. O projeto conta com ferramentas para a coleta, a validação e a internacionalização (adequação a diferentes idiomas). A licença de uso permissiva desse projeto permite a exploração do *corpus* inclusive com fins comerciais. Na versão 13, o *subcorpus* para a língua portuguesa conta com 197 horas de áudios e transcrições, das quais 151 foram validadas.

O *corpus* MultiLingual LibriSpeech²¹ (MLS) (Pratap et al., 2020b) foi pensado pelos seus autores tanto para aplicações em síntese quanto em reconhecimento de fala, devido a isso, sendo descrito aqui e na Seção 3.2. Especificamente para a tarefa de reconhecimento, pode ser combinado com outros recursos, visto que possui relativamente poucos falantes (locutores de audiolivros). Cabe aqui comentar que a aplicação de *corpora* para síntese em reconhecimento não é exclusividade do MLS; outros recursos como o *corpus* CETUC (Alencar; Alcaim, 2008) também são relevantes em ASR. Na prática, todos os recursos mencionados na Seção 3.2 podem ser efetivamente usados na tarefa de reconhecimento. Tais recursos são compostos por áudios mais limpos, geralmente em qualidade de estúdio. Por conta disso, modelos construídos unicamente sobre esse tipo de áudio são apropriados apenas para reconhecimento de fala em cenários com pouco ruído. Para contornar essa característica, o projetista pode injetar ruídos nos áudios ou combiná-los com áudios de outros projetos em diferentes níveis de qualidade.

O MultiLingual TeDx Corpus²² (Salesky et al., 2021) foi proposto para permitir pesquisas nas áreas de reconhecimento automático da fala e tradução da fala para texto²³. O recurso é composto por palestras sobre os mais variados assuntos, sendo gerenciado no escopo do projeto TEDx, vinculado ao grupo TED (Technology, Entertainment and Design). No caso da língua portuguesa, também existem traduções das transcrições para as línguas inglesa e espanhola. Além disso, áudios em espanhol e francês também contam com traduções para o português.

²⁰<https://commonvoice.mozilla.org/pt>

²¹<http://www.openslr.org/94/>

²²<http://www.openslr.org/100>

²³Na tradução de fala para texto, o idioma entre o áudio original e a transcrição são diferentes.



O *corpus* Spotify²⁴ (Clifton et al., 2020) foi lançado primeiramente para a língua inglesa. Em 2022, a empresa lançou uma nova versão incorporando o português (Tanaka et al., 2022), oferecendo diversos áudios para a língua portuguesa provenientes principalmente de podcasts disponíveis na plataforma. Ao todo, 76 mil horas de áudios foram disponibilizadas a partir de 123 mil episódios de shows da plataforma. As transcrições foram geradas automaticamente e estão sujeitas a erros de transcrição. Apesar da licença livre para uso acadêmico, até o momento de escrita deste texto, o *corpus* ainda não estava totalmente disponível para uso. Pesquisadores interessados em acessar os áudios devem entrar com um pedido de acesso na página web dos organizadores.

Existem outras bases para tarefa de ASR que também valem a pena ser citadas. Entre elas, o Multilingual Spoken Corpus²⁵ (Mazumder et al., 2021) é uma base de palavras faladas em 50 idiomas e contém um recorte de cerca de 1 segundo dos áudios do Common Voice, totalizando 58 horas de áudio em português. Diferentemente das outras bases discutidas até o momento, os áudios desse *corpus* são compostos de palavras soltas, em vez de enunciados completos. Esse tipo de *corpus* se destina ao treinamento de sistemas de reconhecimento em domínios específicos (por exemplo, teletendimento bancário). Entre as bases menores, pode-se destacar os *corpora* LapsBM, Sidney, VoxForge, três *corpora* que totalizam, aproximadamente, 4, 1 e 1 horas, respectivamente, levantados por Quintanilha; Netto; Biscainho (2020) e disponíveis para download na página do pesquisador²⁶.

Por fim, algumas das bases não são voltadas a ASR, mas a tarefas relacionadas, como tradução de fala para texto. O *corpus* CoVoST (Wang et al., 2020; Wang; Wu; Pino, 2020) é um recorte da base Common Voice, mas com foco em tradução de fala para texto. Na versão 2, cerca de 17 horas são disponibilizadas para o português com as respectivas traduções para o inglês. O *dataset* Vox Populi²⁷ (Wang et al., 2021a) é uma iniciativa da empresa Meta com foco principal no treinamento semi-supervisionado e não-supervisionado de modelos de aprendizado de máquina. A base contém transcrições para algumas línguas, mas o português não é contemplado. Ao todo, 17.500 horas de áudio foram disponibilizadas para o idioma.

3.5 Recursos para Reconhecimento de Emoções

O reconhecimento de emoções a partir da fala é uma área de estudo promissora que visa compreender as emoções expressas vocalmente pelos indivíduos (Akçay; Oğuz, 2020; El Ayadi; Kamel; Karray, 2011; Singh; Goel, 2022). Uma das teorias mais clássicas nesse campo é a Teoria das Emoções Básicas de Ekman (Ekman, 1992), que descreve a existência de seis emoções primárias: alegria, tristeza, raiva, medo, surpresa e aversão.

Recentemente, outras teorias e modelos têm sido propostos, obtendo-se um espectro mais detalhado de emoções. Nesse sentido, o Modelo Circumplexo de Russel (Posner; Russell; Peterson, 2005) oferece uma perspectiva complementar, ao representar as emoções em um espaço bidimensional, com eixos de valência (positivo/negativo) e intensidade (ativa/passiva), conforme apresentado de forma simplificada na Figura 3.4. Reconhecer emoções na fala tem muitas aplicações práticas, como a análise de atendimento ao cliente, apoio na avaliação do estado emocional de indivíduos durante terapias, e o desenvolvimento

²⁴<https://podcastsdataset.byspotify.com/>

²⁵<https://mlcommons.org/en/multilingual-spoken-words/>

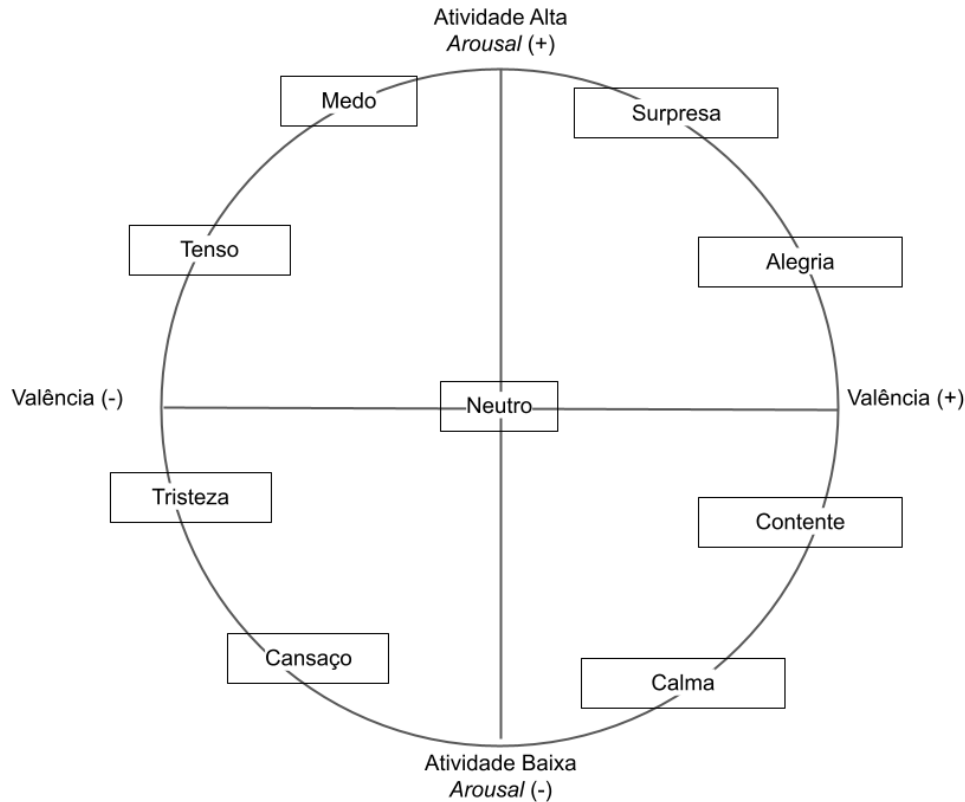
²⁶<https://igormq.github.io/datasets/>

²⁷<https://github.com/facebookresearch/voxpathuli>



de assistentes virtuais mais empáticos, o que ajuda a desenvolver técnicas mais eficientes para interação humano-computador (Wani et al., 2021).

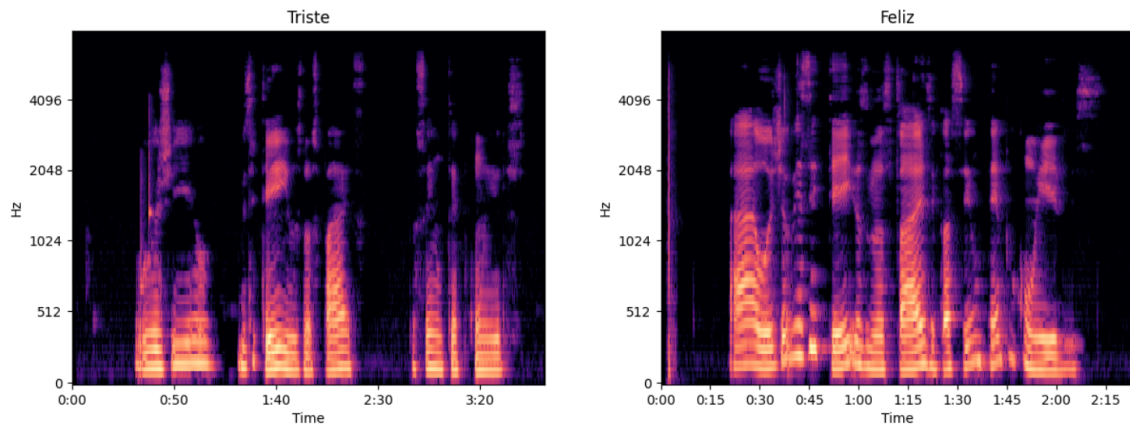
Figura 3.4: Modelo Circumplexo de Russell (simplificado) para representar as emoções em um espaço bidimensional, com eixos de valência (positivo/negativo) e intensidade (ativa/passiva).



Há três grandes desafios em tarefas de reconhecimento de emoções a partir da fala. O primeiro desafio consiste em representar a fala de forma computacionalmente viável, transformando o sinal acústico em representações que contenham características relevantes para identificar as emoções contidas no sinal. Nesse contexto, uma estratégia tradicional é processar o sinal de áudio para identificar características prosódicas (Rao; Koolagudi; Vempada, 2013), como duração e intensidade da fala. Por exemplo, nota-se maior intensidade vocal na emoção “alegria”, enquanto a “tristeza” costuma ter intensidade vocal reduzida. Outra estratégia é o reconhecimento de emoções por meio de espectrogramas (Özseven, 2018), uma representação visual do espectro de frequência de um sinal de áudio ao longo do tempo, obtidas por meio da aplicação da transformada de *Fourier* em janelas de áudio. Na Figura 3.5, há a ilustração de dois espectrogramas que representam o sinal acústico do texto “hoje eu visitei os meus pais e passei um tempo com eles”, falado por uma mesma pessoa. No lado esquerdo, o trecho foi falado com a emoção “triste”. No lado direito, a mesma sentença foi falada com a emoção “alegre”. Os modelos de aprendizado de máquina exploram as características extraídas dos espectrogramas para aprender a diferenciar as categorias de emoção.

O segundo desafio está relacionado à disponibilidade de *corpora* anotados para a tarefa de

Figura 3.5: Exemplo de espectrogramas para o trecho “hoje eu visitei os meus pais e passei um tempo com eles”. (Esquerdo) Espectrograma da fala triste. (Direito) Espectrograma do mesmo trecho, mas com a emoção de alegria.



reconhecimento de emoções. Esses *corpora* são fundamentais para o treinamento de métodos de aprendizado de máquina. Por fim, o terceiro desafio envolve escolha e parametrização do método de aprendizado de máquina visando a geração de modelos eficientes para reconhecimento de emoções.

No âmbito do projeto TaRSila, há uma frente de trabalho denominada SER (*Speech Emotion Recognition*) que visa enfrentar os desafios mencionados anteriormente, com foco específico no reconhecimento de emoções na fala em português. Um diferencial importante deste projeto é o desenvolvimento de abordagens que lidam com fala espontânea, que apresenta desafios adicionais em comparação com a fala preparada. Enquanto a fala preparada envolve cenários planejados ou ensaiados, na qual o indivíduo tem tempo para estruturar suas ideias e escolher suas palavras antes de expressá-las, a fala espontânea ocorre de forma mais imediata, como conversas informais e discussões em grupo, contendo hesitações, pausas, repetições, ruídos e interrupções. Vale ressaltar que a fala espontânea pode expressar emoções de forma mais autêntica, sem ensaios ou autocontrole geralmente presentes na fala preparada.

Uma das etapas cruciais desse projeto foi a preparação do *corpus* CORAA-SER, que consiste em aproximadamente 1 hora de áudio de fala espontânea, anotado com presença ou ausência de emoção, envolvendo homens e mulheres. O *corpus* foi obtido a partir de anotações paralinguísticas de outro *corpus* denominado C-ORAL-BRASIL I, um *corpus* de referência do português brasileiro falado informal (Raso; Mello, 2012b). A primeira versão do CORAA-SER está disponível publicamente²⁸. Com o CORAA-SER, já foi possível explorar diferentes técnicas de representação e métodos de aprendizado de máquina para identificar padrões emocionais na fala espontânea em português. Uma visão geral com os resultados de diferentes trabalhos e grupos de pesquisa foram sumarizados por Marcacini; Candido Junior; Casanova (2022).

O CORAA-SER possui segmentos de áudio rotulados em três categorias: neutro (491 áudios), não-neutro-feminino (89 áudios) e não-neutro-masculino (45 áudios). Também são disponibilizadas duas versões pré-processadas dos áudios:

- Características prosódicas: foram disponibilizadas características físicas da fala, como

²⁸CORAA-SER v1: <https://github.com/rmarcacini/ser-coraa-pt-br>



entonação, ritmos, tom, tempo, intensidade etc. Esse tipo de pré-processamento é tradicionalmente utilizado em métodos tradicionais de reconhecimento de emoções. No total, 56 características prosódicas foram disponibilizadas.

- Características do Wav2Vec: foi utilizado um modelo Wav2Vec (Baevski et al., 2020) pré-treinado para extração de características do áudio. Essas características podem ser usadas para treinar um classificador de reconhecimento de emoções na fala.

Entre os resultados mais recentes, incluindo os resultados obtidos no CORAA-SER, vale destacar o desempenho promissor de modelos estado da arte para reconhecimento de emoções na fala, especialmente baseados em técnicas de *deep learning* e *transfer learning* (Chen; Rudnicky, 2023; Gauy; Finger, 2022; Lope; Graña, 2023; Wagner et al., 2023). No contexto do *deep learning*, arquiteturas como redes neurais convolucionais (CNNs), redes neurais recorrentes (RNNs) e Transformers têm sido amplamente aplicadas, devido à sua capacidade de aprender representações intermediárias a partir dos segmentos de áudios para a tarefa de reconhecimento de emoções. Já *transfer learning* é uma abordagem geralmente usada em conjunto com *deep learning* para o reconhecimento de emoções, permitindo utilizar modelos pré-treinados em grandes *corpora* de áudio. Esses modelos pré-treinados são geralmente usados em tarefas de reconhecimento de fala. A ideia é explorar conhecimento prévio adquirido por esses modelos e especializá-lo para uma nova tarefa, como o reconhecimento de emoções. Essa etapa é denominada de ajuste fino e, em geral, depende de um *corpus* anotado, o que aumenta a importância de projetos como o CORAA-SER do TaRSila.

Para finalizar, vale destacar que a tarefa de reconhecimento de emoções a partir da fala ainda possui muitos desafios relacionados à representação computacional da fala, disponibilidade de *corpora* anotados e escolha de métodos de aprendizado de máquina adequados para esta tarefa. No âmbito do projeto TaRSila, a frente de trabalho SER tem buscado superar esses desafios, com ênfase na fala espontânea em português. A criação do *corpus* CORAA-SER foi um passo relevante nesse processo, pois já permitiu a exploração de algumas técnicas pela comunidade (Marcacini; Candido Junior; Casanova, 2022). As direções futuras e oportunidades de pesquisa neste tema são promissoras. Muitos pesquisadores estão investigando métodos de *transfer learning* para reconhecimento de emoções, baseado em conhecimento prévio de modelos pré-treinados para fala como o Wav2Vec (Baevski et al., 2020) e HuBERT (Hsu et al., 2021). Essas abordagens têm demonstrado um potencial promissor para melhorar a precisão e a eficiência do reconhecimento de emoções em áudio. Também devemos destacar a importância dos trabalhos que ainda exploram características prosódicas, uma vez que relacionar características de duração, intensidade, *pitch* e entonação com diferentes categorias de emoção fornecem maior interpretabilidade no reconhecimento de emoções a partir da fala.

3.6 Recursos para predição de pontuação no cenário de ASR

A saída de sistemas ASR convencionais é uma das principais fontes de dados que requerem capitalização e pontuação, pois é feita de uma sequência de palavras somente. Exemplos de ASR convencionais comerciais são o Google Cloud Speech-to-Text, Microsoft Azure Speech Services, IBM Watson Speech to Text, e SpeechMatics. Quando a saída é um texto escrito para ser lido em voz alta, isto é, um discurso, a tarefa é chamada de restauração da pontuação original, e para a fala conversacional/espontânea a tarefa é chamada de predição da pontuação (Păiş; Tufiş, 2022).



Apresentamos nesta seção, o *dataset* de teste do Corpus CORAA MuPe, balanceado por sexo, com histórias de vida de homens e mulheres, que foi criado para avaliar a tarefa de predição de pontuação no contexto de reconhedores automáticos de fala, usando como reconhecedor o Whisper da OpenAi (Radford et al., 2022).

Também trazemos um resumo dos trabalhos em predição da pontuação em fala preparada e espontânea, descrevendo suas abordagens e *datasets*. Vamos contrastar primeiramente a diferença entre a saída de ASRs convencionais e de ASRs que além de transcreverem automaticamente um segmento de fala espontânea, conseguem fazer a predição da pontuação e capitalização como é o caso do Whisper.

3.6.1 Ilustrando o uso do ASR Whisper na predição da pontuação em português

O trecho do Quadro 3.1 é de uma história de vida do MuPe²⁹ que apresenta seis turnos de uma entrevista (P = pergunta, R = resposta) em que os sinais de pontuação foram removidos da transcrição e as primeiras palavras de cada oração são apresentadas em letras minúsculas. Esse trecho ilustra a saída de um ASR convencional que, embora não tenha erros na transcrição de palavras, ajuda a enfatizar o quanto a ausência de pontuação pode dificultar a compreensão do texto quando se torna longo.

Quadro 3.1. Trecho de uma história de vida do MuPe para ilustrar o formato de saída de um ASR convencional

P – tá e sua mãe ela fazia o que
 R – a minha mãe ela trabalhou mais de 30 anos numa tecelagem aqui em são paulo chamada guilherme jorge que fica lá na vila formosa ela é auxiliar de tecelagem a vida inteira trabalhou nisso e cuidava da casa
 P – mas ela veio de onde onde ela nasceu
 R – a minha mãe nasceu na cidade de no sertão de pernambuco chama lá a cidade chama bodocó pernambuco
 P – e quando que ela veio para são paulo
 R – ela veio para são paulo no final dos anos 40 porque a situação em pernambuco estava no campo estava muito difícil havia seca então ela e três duas irmãs e um irmão vieram pra são paulo início para trabalhar na casa de uma tia dela no bairro da penha que tinha uma pensão aí depois cada um foi como a maioria dos nordestinos chega fica na casa dos familiares e depois vai arrumando emprego aí vai arrumando sua vida ou seja minha mãe e meu pai também vieram pra são paulo porque lá em minas não havia trabalho e ele como tinha essa vontade de trabalhar acredito eu ele antes de completar 18 anos ele fugiu de casa veio para são paulo e ele era o caçula do primeiro casamento da minha vó

A saída do Quadro 3.2 foi gerada pelo ASR Whisper para o mesmo trecho de áudio relacionado a mesma história de vida do MuPe mostrada no Quadro 3.1. As entidades nomeadas aparecem em negrito, para facilitar a análise. Whisper não é um ASR convencional. Ele foi treinado pela empresa de pesquisa em Inteligência Artificial OpenAI usando um grande conjunto de dados multilíngues coletados da web. Ele tem uma nova arquitetura multitarefas, isto é, ele é treinado para prever diversas tarefas de processamento de fala ao mesmo tempo: (i) detecção de atividade de voz, que instrui o modelo a funcionar apenas quando há uma linguagem humana específica e ser robusto ao lidar com ruído/música de fundo; (ii) tradução da fala para o inglês e (iii) reconhecimento

²⁹https://museudapessoa.org/historia-detalle/?id=7136&download_integra_text_pdf/



de fala multilíngue com pontuação. Das várias pontuações inseridas na transcrição, Whisper não é capaz de gerar ponto e vírgula e dois pontos.

Quadro 3.2. História de vida do MuPe gerada pelo ASR Whisper

P – E sua mãe, ela fazia o quê?
R – A minha mãe trabalhou mais de 30 anos numa tecelagem aqui em **São Paulo**, chamada **Guilherme Jorge**, que fica lá na **Vila Formosa**. Era auxiliar de tecelagem e a vida inteira trabalhou nisso e cuidava da casa.
P – mas ela veio de onde, de onde ela nasceu.
R – A minha mãe nasceu na cidade de um sertão de **Pernambuco**, lá a cidade chama-se **Bodocó**, em **Pernambuco**.
P – E quando ela veio para **São Paulo**...
R – Ela veio para **São Paulo** no final dos anos 40, porque a situação em **Pernambuco** estava no campo, estava muito difícil, havia seca, então ela e as duas irmãs e o irmão vieram para **São Paulo**, início para trabalhar na casa de uma tia dela, no bairro da **Penha**, que tinha uma pensão. Aí depois cada um foi, como a maioria dos nordestinos, chega, fica na casa dos familiares, depois vai arrumando emprego, aí vai arrumando sua vida. Ou seja, minha mãe e meu pai também vieram para **São Paulo**, porque lá em **Minas** não havia trabalho. E ele, como tinha essa vontade de trabalhar, acredito eu, antes de completar 18 anos ele fugiu de casa, veio para **São Paulo**. Ele era o caçula do primeiro casamento da minha avó.

Quando a saída do ASR Whisper é comparada com a transcrição manual pontuada e capitalizada (mostrada no Quadro 3.3; entidades nomeadas em negrito), notamos que o Whisper gera orações mais curtas e, portanto, mais sentenças (11) do que a transcrição manual (6). No entanto, a capitalização utilizada é muito semelhante à manual (ver Tabela 3.3). Quanto à capitalização, ela é usada principalmente para entidades nomeadas relacionadas (EN) a cidades, estados e regiões.

Quadro 3.3. História de vida do MuPe gerada por transcrição manual

P – Tá, e sua mãe, ela fazia o que?
R – A minha mãe, ela trabalhou mais de 30 anos numa tecelagem aqui em **São Paulo**, chamada **Guilherme Jorge**, que fica lá na **Vila Formosa**, ela é auxiliar de tecelagem, a vida inteira trabalhou nisso e cuidava da casa.
P – Mas ela veio de onde? Onde ela nasceu?
R – A minha mãe nasceu na cidade de no **Sertão de Pernambuco**, chama lá, a cidade chama **Bodocó**, **Pernambuco**.
P – E quando que ela veio para **São Paulo**?
R – Ela veio para **São Paulo** no final dos anos 40, porque a situação em **Pernambuco** estava no campo, estava muito difícil, havia seca, então, ela e três, duas irmãs e um irmão vieram pra **São Paulo**, início para trabalhar na casa de uma tia dela, no bairro da **Penha**, que tinha uma pensão, aí, depois cada um foi como a maioria dos Nordestinos, chega, fica na casa dos familiares e depois vai arrumando emprego, aí vai arrumando sua vida, ou seja, minha mãe e meu pai também vieram pra **São Paulo**, porque lá em **Minas** não havia trabalho, e ele como tinha essa vontade de trabalhar acredito eu, ele antes de completar 18 anos, ele fugiu de casa, veio para **São Paulo** e ele era o caçula do primeiro casamento da minha vó.



Tabela 3.3: Comparação da pontuação e capitalização da saída do Whisper com uma transcrição manual de um trecho de uma história de vida do MuPe.

	Saída do Whisper	Transcrição Manual
Vírgula	24	31
Reticências	1	0
Ponto Final	9	3
Ponto de Interrogação	1	4
Capitalização no início de sentenças	10	7
Capitalização de EN de Lugares	14	14
Outras Capitalizações	0	1

3.6.2 Descrição do Corpus MuPe e seu *dataset* de teste

O Corpus CORAA MuPe está atualmente em fase de processamento. Ele é um conjunto de 300 horas de histórias de vida que foi cedido ao projeto TaRSila em um convênio de colaboração iniciado em dezembro de 2022 entre o MuPe, o ICMC-USP e a UFG. O objetivo inicial do convênio é o estudo e desenvolvimento de modelos de ASR, de métodos de segmentação automática de transcrição e modelagem de tópicos baseada nas transcrições de vídeos.

O MuPe é um museu virtual que visa contar e preservar as histórias de vida das pessoas e incentiva a participação de pessoas de diferentes idades, sexos, raças e profissões. Fundado em 1991, o MuPe contém atualmente um rico e extenso acervo digital de narrativas de fala espontânea em português, chamadas de histórias de vida que são contadas pelas próprias pessoas ou por terceiros.

As narrativas são gravadas de três formas: (i) na sede do Museu, em estúdio – gravadas em vídeo e coletadas por entrevistadores especializados na metodologia de história de vida, (ii) enviadas via internet pelo Programa Conte sua História ou (iii) via Museu que Anda, programa em que as narrativas de pessoas fora da sede são gravadas em cabines itinerantes. Cada entrevista constitui uma unidade do acervo que é formada pela gravação em áudio ou vídeo da entrevista, a transcrição e edição de cada narrativa, acompanhada de fotos e documentos enviados pelas pessoas que contam suas narrativas de vida.

Depois de gravadas, as histórias de vida coletadas pelo MuPe são transcritas e revisadas. As transcrições possuem anotações de risos, palmas, assobios, fala emocionada, pausas, entre outros, utilizando parênteses. Além disso, as expansões de acrônimos são anotadas usando colchetes. A transcrição é segmentada em enunciados, com pontuação, usando sete sinais de pontuação (ver Tabela 3.5). Os turnos são indicados pelos rótulos P/1 (e P/2) e R seguidos da transcrição do turno, onde P/i (i = 1 ou 2) indica o entrevistador (1 ou 2 entrevistadores) e R o entrevistado. No entanto, como as pausas preenchidas e disfluências de edição (por exemplo, revisões e repetições) comuns na fala espontânea não são anotadas, a transcrição do MuPe pode ser chamada de transcrição textual adaptada.

O *dataset* de teste é composto por 10 narrativas de vida retiradas do projeto Ponto de Cultura da plataforma MuPe. O Corpus MuPe contém 280 narrativas de vida, sendo a maioria delas com conteúdo transcrito completo; algumas poucas apresentam somente um resumo. A Tabela 3.4 mostra as estatísticas do *dataset* de teste, dividido em duas amostras: narrativas masculinas e femininas. O *dataset* de teste do MuPe é composto por 1.349 turnos e totaliza aproximadamente 17 horas. Ele está disponibilizado publicamente³⁰,

³⁰<https://github.com/nlcl-nlp/asr-punctuation-evaluation>



com os links dos áudios e as transcrições manuais anonimizadas, cujos nomes completos dos entrevistados (ou dos familiares) masculinos foram trocados por “João” e os femininos por “Maria”, para não haver grande perda de material original e ainda atender ao requisito do convênio para disponibilização pública. O nome da família foi trocado por “nome da família”.

Tabela 3.4: Estatísticas do *dataset* de teste do MuPe. O tamanho médio do turno e o tamanho médio da sentença são calculados em palavras, sem contar a pontuação. Consideramos como sentença os segmentos terminados em ponto de interrogação, ponto de exclamação e ponto final. O número de *tokens* inclui pontuação.

	Amostra Masculina	Amostra Feminina	Total
Duração do Áudio	8:06:21 h	8:42:13	16:48:34 h
# Turnos	834	515	1.349
Tamanho Médio do Turno	85.26 ± 169.44	138.80 ± 325.46	105.11 ± 240.72
# Sentenças	4.100	4.640	8.740
Tamanho Médio da Sentença	17.16 ± 27.47	14.57 ± 14.64	15.79 ± 21.67
# <i>Tokens</i>	83.953	79.377	163.330

Tabela 3.5: Distribuição das classes de pontuação nas amostras do MuPe.

	Amostra Masculina (#)	Amostra Feminina (#)	Total # (%)
Reticências	257	110	367 (1,56%)
Ponto de Exclamação	45	172	217 (0,92%)
Ponto Final	3.247	4.002	7.249 (30,8%)
Ponto de Interrogação	808	466	1.274 (5,41%)
Vírgula	8.383	6.571	14.954 (63,5%)
Ponto e Vírgula	62	21	83 (0,35%)
Dois Pontos	293	358	651 (2,76%)
Total			23.521

3.6.3 *Corpora* usados nos trabalhos de Predição da Pontuação para ASRs

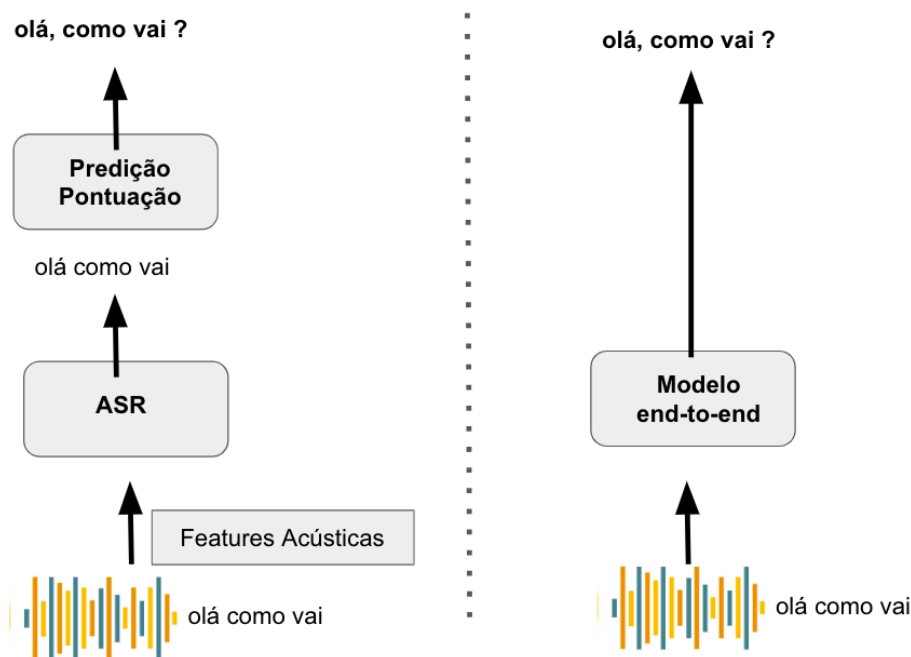
A abordagem dominante na literatura, conhecida como abordagem em cascata, é treinar um modelo de reconhecimento de fala (ou usar um pronto) e um modelo de predição de pontuação separadamente, e, em seguida, colocá-los em cascata, ou seja, inserir marcas de pontuação na transcrição gerada pelo ASR como uma etapa de pós-processamento. Para cada *token* na saída do ASR, as *features* acústicas do áudio são obtidas e são usadas como entrada para o módulo de predição de pontuação (Figura 3.6, esquerda).

Em relação aos recursos, os três tipos de recursos usados para predição e restauração de sinais de pontuação são lexicais, prosódicos e a combinação de recursos prosódicos e lexicais.

Abordagens lexicais recentes na literatura para restauração de pontuação usam redes neurais profundas. As abordagens variam desde o uso de word embeddings pré-treinados, mecanismo de atenção, abordagens baseadas em Transformers treinadas em grandes *corpora* de texto, seja usando apenas o modelo BERT pré-treinado (Devlin et al., 2019) ou realizando uma comparação de modelos diferentes baseados em Transformers. Quanto às *features*



Figura 3.6: Abordagem de um sistema em cascata (esquerda) e abordagem de sistemas *end-to-end* (direita).



Fonte: Adaptado de (Nozaki et al., 2022, fig. 1, p. 1812)

prosódicas elas são baseadas nas durações das pausas, início e duração das palavras que são seguidas por uma pontuação, F0 (pitch) e duração de fonemas, além de *features* mais complexas baseadas no modelo wav2vec³¹.

Os métodos usam as métricas precisão (P), revocação (R) e F1-score (F1) para avaliar o desempenho dos quatro rótulos (Vírgula, Ponto Final, Ponto de Interrogação e O – sem sinal de pontuação), propostos no IWSLT 2011 (*International Workshop on Spoken Language Translation 2011*).

No artigo de Gris et al. (2023), foram revisados cinco trabalhos da literatura para uma comparação com o uso do Whisper no *dataset* de teste do MuPe³². A Tabela 3.6 mostra um resumo dos trabalhos avaliados, com indicação do *corpus* usado na avaliação e seu tamanho em número de enunciados/duração em horas; os conjuntos de teste do IWSLT 2011 são apresentados em número de palavras para transcrições manuais (Referência) e transcrições automáticas (realizadas por um ASR), nesta ordem (Che et al., 2016).

Tabela 3.6: Resumo dos trabalhos apresentados por Gris et al. (2023).

Trabalhos	Línguas(s)	Test Set	Duração/Tam.
Alam et al. (2020)	inglês, bengali	IWSLT 2011 Corpus	12.626/12.822 pal.
Yi and Tao (2019)	inglês	IWSLT 2011 Corpus	12.626/12.822 pal.
Zelasko et al. (2018)	inglês	Fisher Corpus	1.100 enunciados
Sunkara et al. (2020)	inglês	Fisher Corpus	42 h
Nozaki et al. (2022)	inglês, japonês	MuST-C Corpus	2.641 enunciados
Gris et al. (2023)	português	Corpus CORAA MuPe	16:48:34 h

³¹<https://github.com/pytorch/fairseq/blob/master/examples/wav2vec>

³²<https://github.com/nile-nlp/asr-punctuation-evaluation>



Alam; Khan; Alam (2020) avaliaram vários modelos de língua para o inglês (BERT, RoBERTa, ALBERT, DistilBERT) e modelos multilíngues para o bengali (mBERT, XLM-RoBERTa) disponíveis no repositório Hugging Face³³. Avaliaram o desempenho dos quatro rótulos: Vírgula, Ponto, Pergunta e O (sem sinal de pontuação seguido), nos dois conjuntos de teste do IWSLT 2011.

Yi; Tao (2019) propuseram um modelo baseado em auto-atenção usando embeddings de palavra e de fala, respectivamente Glove (Pennington; Socher; Manning, 2014) e Speech2Vec (Chung; Glass, 2018), resolvendo o problema de dependência dos dados de fala alinhados com sua transcrição, pois para muitas línguas há carência destes recursos. Como Alam; Khan; Alam (2020), os autores também avaliaram seu modelo no conjunto de dados do IWSLT 2011, mas os resultados de Alam; Khan; Alam (2020) ainda são melhores do que os de Yi; Tao (2019) (exceto para ponto de interrogação), provavelmente devido ao fato de Alam; Khan; Alam (2020) usar uma técnica de aumento de dados, que melhora o desempenho em dados com ruídos, e um modelo baseado em Transformers.

Zelasko et al. (2018) e Sunkara et al. (2020) avaliaram a previsão de pontuação na fala espontânea. Zelasko et al. (2018) reforçam que o problema da tarefa de previsão de pontuação para fala espontânea é a falta de conjuntos de dados de referência. Eles avaliaram dois modelos de redes neurais profundas: um baseado em *Convolutional Neural Nets* (CNN) e outro baseado em redes *Long Short-Term Memory* Bidirecionais (Bi-LSTM). Os modelos são treinados no Fisher Corpus (Cieri; Miller; Walker, 2004), que inclui anotação de pontuação e capitalização. O conjunto de dados de treinamento consiste em 348 horas de conversação e os conjuntos de desenvolvimento e teste cada um contém cerca de 42 horas.

Sunkara et al. (2020) propõem uma nova estrutura de fusão multimodal de embeddings lexicais e acústicos para previsão de pontuação em fala espontânea chamada arquitetura de aprendizagem semissupervisionada multimodal (MuSe). Embora os resultados de Sunkara et al. (2020) não sejam diretamente comparáveis com os de Zelasko et al. (2018) no Fisher Corpus, pois as divisões dos conjuntos de dados são diferentes, Sunkara et al. (2020) obtiveram melhor desempenho em todas as classes de pontuação.

Nozaki et al. (2022) propuseram um modelo *end-to-end* para reconhecimento de fala com pontuação (Figura 3.6, direita). Eles usaram dois conjuntos de dados de idiomas diferentes: o MuST-C, um *corpus* multilíngue³⁴ (Di Gangi et al., 2019) que foi usado como o conjunto de dados em inglês e o JCALL, um conjunto de dados fechado que consiste de gravações de áudio de conversas, foi usado como o conjunto de dados em japonês.

A OpenAI lançou, em setembro de 2022, o Whisper ASR. Embora também seja um modelo de ASR *end-to-end* semelhante à abordagem de Nozaki et al. (2022), ele tem duas diferenças importantes: é de código aberto e foi treinado em um *dataset* grande e multilíngue. Whisper é um ASR capaz de incluir pontuação e capitalização nas transcrições (Radford et al., 2022), embora seja somente capaz de gerar 5 tipos de pontuações: reticências, ponto final, vírgulas, ponto de interrogação e ponto de exclamação. O conjunto de dados de teste do MuPe possui duas pontuações a mais (ponto e vírgula e dois pontos).

³³<https://huggingface.co/docs/transformers/index>

³⁴MuST-C inclui gravações de áudio de TED Talks em inglês alinhadas no nível da frase com suas transcrições e traduções manuais.



3.7 Considerações finais

Apresentamos neste capítulo os recursos de processamento de fala que foram criados nos três anos iniciais do projeto TaRSila. Dois grandes *corpora* estão ainda em fase de processamento para serem lançados em 2024: (i) o Corpus CORAA NURC-SP e o Corpus CORAA MuPe. Além dos estudos previstos no convênio com o MuPe, uma tarefa futura será a compilação de um *dataset* para modelagem e teste de sistemas TTS, nos moldes do LibriTTS, citado na Seção 3.2, já que as histórias de vida cedidas foram gravadas em estúdio. A diferença entre o LibriTTS e o *corpus* a ser criado a partir do Corpus MuPe está no gênero dos textos: as histórias de vida do MuPe são exemplos da fala espontânea, guiada por entrevista, e o LibriTTS contém fala lida.

Agradecimentos

Em primeiro lugar agradecemos aos bolsistas do projeto TaRSila que foram incansáveis nas revisões das transcrições automáticas, no treinamento e teste dos modelos para vários sistemas de processamento de fala. Este trabalho faz parte de um Acordo de Transferência de Tecnologia entre Museu da Pessoa (MuPe), Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP) e Universidade Federal de Goiás. Este trabalho foi realizado no Centro de Inteligência Artificial (C4AI-USP), com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (bolsa FAPESP nº 2019/07665-4) e da IBM Corporation. Agradecemos também o apoio do Centro de Excelência em Inteligência Artificial (CEIA) financiado pela Fundação do Estado de Goiás (bolsa FAPEG nº 201910267000527), à Fundação de Apoio à Universidade de São Paulo (FUSP) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (bolsa PQ CNPq, processo 304961/2021-3). Este projeto também foi apoiado pelo Ministério da Ciência, Tecnologia e Inovação, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do PPI-SOFTEX, coordenado pela Softex e publicado Residência no TIC 13, DOU 01245.010222/2022-44.



Parte III

Palavras



Capítulo 4

Sequência de caracteres e palavras

Maria José Bocorny Finatto

Helena de Medeiros Caseli

Lucelene Lopes

Amanda Rassi

Neste capítulo, trataremos de algo que parece simples, mas não é: identificar a unidade mínima quando tratamos, computacionalmente, a língua. Essa delimitação não é consensual entre pesquisadores e profissionais de PLN. E, mesmo em linguística, há sempre controvérsias e necessidade de pontos de referência para se definir, por exemplo, o que seja uma palavra ou mesmo uma frase.

As subáreas especializadas dos estudos linguísticos entendem como **unidade mínima de processamento** diferentes elementos conforme seus focos e pontos de vista. A fonologia (Capítulo 2), por exemplo, considera o fonema como a menor unidade sonora e distintiva de uma língua. Se tomarmos o exemplo do que diferencia as palavras “sábua” (mulher inteligente), “sabiá” (pássaro) e “sabia” (verbo “saber”), percebemos que a sílaba tônica é o diferencial, especialmente quando pensamos em sons e fala e não em escrita.

Já a morfologia considera o **morfema** como a menor unidade dotada de significado na língua. Nessa perspectiva, temos os “pedacinhos” de palavras e seus valores, como seria o caso da marca de diminutivo “-inho”, que assinala o masculino e o singular em “menininho”, ou o segmento “-ei” no verbo “comprei”, que marca um modo-tempo (pretérito perfeito do modo indicativo) e também um número-pessoa (primeira pessoa do singular).

Assim, conforme o ponto de vista de quem analisa, uma palavra pode ser feita de sons e de sílabas tônicas e/ou composta de vários segmentos gráficos menores. Podemos, ainda, considerar segmentos ou pedaços mais abrangentes, conforme o critério que utilizamos. Um exemplo nessa linha seria a palavra “guarda-pó”, que pode ser considerada como uma palavra só ou a junção de duas palavras. Outro caso ilustrativo é “escova de dente”, que, para alguns, é a união de três palavras, e, para outros, é uma palavra só, mesmo que não tenha hífen. Além dessas questões, também é controverso tratar das abreviaturas, siglas, interjeições, dos modos de escrita diferenciados nas redes sociais, com internetês, hashtags, emojis, símbolos e outras peculiaridades.

Fazendo um paralelo, podemos entender que, de modo geral, os modelos de PLN trabalham as **palavras** como unidade primária de processamento. Vejamos, por exemplo, o caso da frase no Exemplo 4.1.

Exemplo 4.1.

Jacinta Maria comprou uma cadeira em São Paulo ontem e pagou 25 reais por ela.



Na frase do Exemplo 4.1 são 15 palavras se considerarmos que é palavra toda a sequência de caracteres separada por um espaço em branco. Mas se pode pensar que Jacinta+Maria e São+Paulo são palavras compostas e que, talvez, o número 25 não seja bem uma palavra, não? A resposta será: depende do critério que você usar e da finalidade que tem ou busca com essa referência de unidade e/ou partes.

Ao fazer o processamento computacional de textos escritos, a definição de que tipo de unidade de processamento se quer buscar/estudar parece estar atrelada às necessidades da tarefa ou trabalho pretendidos. Geralmente, considera-se que uma palavra é, simplesmente, uma unidade grafológica delimitada, nas línguas europeias, entre espaços em branco na representação gráfica, ou entre um espaço em branco e um sinal de pontuação¹. Essa é uma definição bastante concreta, e bastante prática. No entanto, ao pensarmos em nossos modelos computacionais e suas aplicações no mundo, é importante nos aprofundarmos um pouco mais na conceituação do que é uma palavra e nas possibilidades de processamento e implicações das decisões tomadas no pré-processamento dos *corpora*.

Segundo Cabré (1999, p. 20), as palavras são as unidades de referência da realidade empregadas pelos falantes. De acordo com essa definição, as palavras compõem a dimensão linguística mais estreitamente ligada ao mundo real. Ainda segundo a mesma autora, o **léxico** consiste no conjunto das palavras de uma língua e dos padrões que possibilitam a criatividade do falante. As palavras e, principalmente, as associações infinitas e imprevisíveis que os seres humanos são capazes de traçar entre elas, constituem a manifestação mais concreta e mais produtiva da língua. Assim, é importante que, ao processar dados textuais para gerar modelos computacionais, nos recordemos sempre de que não estamos simplesmente organizando um conjunto de caracteres ou ordenando uma representação ortográfica formal, mas que estamos trabalhando com recursos linguísticos que representam a experiência humana.

Em seguida, devemos considerar também o conceito de **palavra computacional**, que se refere a uma unidade linguística que foi adaptada ou criada especificamente para facilitar seu processamento por máquinas. Isso pode envolver a manipulação de palavras, frases ou até mesmo caracteres de maneira que seja mais conveniente para algoritmos e sistemas de PLN lidarem com elas.

A necessidade dessas palavras computacionais surge devido às complexidades do processamento de linguagem natural por máquinas. A linguagem humana é rica e ambígua, cheia de nuances e variações que podem ser difíceis de interpretar e analisar automaticamente. Portanto, ao transformar palavras em formas mais padronizadas ou simplificadas, os sistemas de PLN podem executar tarefas como análise gramatical, extração de informações (Capítulo 17) e tradução (Capítulo 18) com maior eficácia. Dependendo do objetivo da tarefa ou aplicação de PLN, é possível definir quais rotinas de pré-processamento são mais produtivas para criar as palavras computacionais, ou seja, pode-se remover ou não acentos ortográficos, espaços em branco em itens como “fim de semana”, ou hífen como em “guarda-chuva”.

No mesmo sentido, dependendo dos propósitos da tarefa, um modelo de aprendizado de máquina pode precisar definir expressões multipalavras (Capítulo 5) (e.g. “nem que a vaca tussa”, “deus me livre” ou “ciência de dados”), hashtags (e.g. “#euamopuzzles”), URLs e outros compostos como palavras (ou unidades lexicais) únicas, como se fossem representadas sem espaços: “nemqueavacatussa”, “deusmelivre”, “ciênciadedados” e “euamopuzzles”.

¹Considerando línguas como o português nas quais o processo de tokenização (apresentado na Seção 4.2) pode ser baseado na presença de espaços em branco delimitadores das palavras. Em outras línguas, como o alemão ou o chinês, por exemplo, esse processo de tokenização pode ser um pouco mais complexo.



Essas e outras estratégias similares podem facilitar o processamento automático, mas devem considerar as necessidades de cada aplicação computacional.

Este capítulo apresenta conceitos (Seção 4.1) relacionados a essas unidades mínimas de processamento e que devem ser considerados quando lidamos com textos. Em seguida, são apresentadas as tarefas relacionadas ao processamento morfológico dos textos (Seção 4.2). E depois são indicadas as ferramentas e os recursos disponíveis para o português, exemplificando com uma aplicação prática de um recurso específico para o português brasileiro (Seção 4.3). Por fim, fazemos uma retomada dos principais tópicos discutidos, apresentando as considerações finais e os planos para uma futura versão deste capítulo (Seção 4.4).

4.1 Conceitos básicos da morfologia

Antes de vermos como identificar e tratar computacionalmente as unidades mínimas de processamento, cabe definir alguns conceitos linguísticos básicos necessários. Nesta seção, definimos os conceitos de Morfema (e todos os seus tipos) (Seção 4.1.1), *Token* e *Type* (Seção 4.1.2), Lexema, Lexia e Lema (Seção 4.1.3), Léxico e Gramática (Seção 4.1.4), Léxico comum e especializado (Seção 4.1.5), além de palavras funcionais e lexicais (Seção 4.1.6). Essa seção traz, ainda, informações sobre os processos de formação das palavras (Seção 4.1.7) e morfologia e morfossintaxe (Seção 4.1.8).

4.1.1 Morfema

O objeto principal de estudo da morfologia é o **morfema**, definido linguisticamente como “unidade mínima significativa”. Segundo essa definição, o morfema é a menor unidade linguística dotada de significado, considerando que há outras unidades linguísticas que também possuem significado, como a palavra, o sintagma, a frase, a oração, o período, o texto etc. Além disso, o **morfema** é considerado como “dotado de significado” por oposição ao fonema, que é o objeto de estudo da fonética e da fonologia, e, na verdade, é a menor unidade de análise linguística, porém não possui significado em si, mas tem a função de estabelecer diferença de significado entre uma palavra outra. Em outras palavras, o **fonema** é apenas uma unidade linguística distintiva (não significativa), pois diferencia as palavras por meio de seus sons (e.g. “faca” e “vaca” são duas palavras diferentes que se distinguem pelo fonema inicial “\f” ou “\v”). O mesmo vale para o trio “sábua”, “sabua” e “sabiá”, lembrando que estamos no território dos sons e não da escrita. Para saber mais sobre o processamento da fala, sugere-se a leitura de Capítulo 2 e Capítulo 3.

Em uma explicação simples, podemos dizer que os morfemas são os pedacinhos que se juntam para formar as palavras. E esses “pedacinhos” podem ser de vários tipos: **desinência**, **raiz**, **radical**, **afixo**, **vogal temática** e **tema**. Por exemplo, podemos juntar o radical “experiment” com a vogal temática “a” com a desinência verbal “ri” e com a desinência verbal “a” para formar a palavra “experimental”. Esses quatro “pedacinhos” são chamados de morfemas e eles possuem significados:

- (i) “experiment” significa o conceito lexical de “prova, ensaio, tentativa”;
- (ii) “a” significa que é um verbo da primeira conjugação;



- (iii) “ri” significa que esse verbo está flexionado no tempo futuro do pretérito do modo indicativo;
- (iv) “a” significa que esse verbo está flexionado na terceira pessoa do singular.

A seguir apresentamos brevemente uma definição e exemplos dos vários tipos de morfemas em português.

4.1.1.1 Desinência

Desinências são os morfemas que geralmente ficam no final da palavra e podem marcar gênero e número (no caso dos substantivos e adjetivos) ou marcar número, pessoa, tempo e modo (no caso dos verbos). Por isso as desinências podem ser classificadas em: nominais ou verbais.

Em português, o substantivo “meninas” é formado pelo radical “menin” + duas **desinências nominais**: “a”, que indica feminino, e “s”, que indica plural. Assim, dizemos que a palavra “meninas” está flexionada no feminino plural. Já o verbo “adotássemos” é formado pelo radical “adot” + a vogal temática “a” + duas **desinências verbais**: “sse”, que indica modo e tempo (pretérito imperfeito do subjuntivo), e “mos”, que indica número e pessoa (primeira pessoa do plural). Assim, dizemos que o verbo “adotássemos” está flexionado na primeira pessoa do plural do pretérito imperfeito do subjuntivo.

4.1.1.2 Raiz e radical

Na linguística teórica, existe uma diferença conceitual entre **raiz** e **radical**. Embora a definição de ambos os conceitos ressalte que são os constituintes da palavra que contêm significado lexical, no caso da **raiz**, ela não inclui afixos derivacionais ou flexionais (e.g. “beb” é a raiz de “beber”, “beberemos”, “bebendo”, “bebida”, “bebidinhas” e tantas outras formas flexionadas). No caso do **radical**, ele não inclui afixos de flexão, mas pode incluir afixos derivacionais (e.g. “beb” é o radical de “beber”, “beberemos” e “bebendo”, mas “bebid” seria o radical correto de “bebida” e de “bebidinhas”). Essa distinção, em termos linguísticos, é muito sutil e geralmente as aplicações de PLN assumem os dois termos como sinônimos.

A raiz ou radical é o morfema nuclear de uma palavra, ou seja, aquele constituinte básico que expressa sua base significativa, que designa o significado lexical da palavra. Portanto, ele é o componente comum a todas as palavras que pertencem à mesma família. Por exemplo, em português, “menino”, “meninas”, “meninada”, “meninice” e outras possuem a mesma raiz ou radical “menin”.

4.1.1.3 Afixo

Os **afixos** são os morfemas lexicais que se juntam com o radical ou com o tema para formar uma nova palavra, neste caso chamada de palavra derivada. A inserção de um afixo ao radical de uma palavra pode mudar-lhe o sentido ou adicionar-lhe uma ideia secundária ou ainda mudar sua classe gramatical.

Em português, os afixos podem ser de três tipos:

1. **prefixos**, quando são inseridos antes do radical (ex: “**des**matar”, “**im**mortal”, “**anti**oxidante”, “**ref**azer”);
2. **infixos**, quando são inseridos no meio de um radical, mas são bem raros; e
3. **sufixos**, quando são anexados ao final do radical (ex: “ativamente”, “imaginação”, “crescimento”).



Apesar de os sufixos e as desinências serem morfemas acoplados ao final da palavra, eles não devem ser confundidos, pois os primeiros criam novas palavras derivadas, a partir de um processo de formação de palavras chamado de derivação. Já a adição de desinências não cria novas palavras, apenas flexiona a palavra existente em uma nova forma flexionada.

4.1.1.4 Vogal temática

Vogal temática é o nome dado às vogais que aparecem imediatamente após o radical da palavra, mas não representam seu gênero. Em português, as vogais temáticas podem ser de dois tipos:

1. **nominais**, que podem ser “a” (ex: “atleta”, “colega”, “dentista”), “e” (ex: “agente”, “recorrente”, “alegre”) ou “o” (ex: “pássaro”, “crocodilo”, “dezembro”); e
2. **verbais**, que indicam as 3 conjugações verbais: “a”, para verbos da primeira conjugação (ex: “andar”, “passear”, “falar”), “e” ou “o”, para verbos da segunda conjugação (ex: “escrever”, “ler”, “fazer”, “pôr”, “repor”, “compor”) ou “i”, para verbos da terceira conjugação (ex: “agir”, “assumir”, “partir”).

As vogais temáticas não devem ser confundidas com as desinências nominais, por exemplo, que indicam gênero em “menino” e “menina”. Neste caso, “o” e “a” são desinências nominais porque esses morfemas marcam os gêneros masculino e feminino, respectivamente. Novamente, cabe mencionar que há uma boa e extensa discussão sobre a natureza e funcionamento das vogais temáticas em linguística. Para quem quiser uma visão aprofundada, vale consultar o trabalho de Santana (2019), uma tese de doutorado sobre vogais temáticas.

4.1.1.5 Tema

Tema é a forma lexical que se cria quando se juntam dois morfemas: o radical e a vogal temática. Por exemplo, a partir da combinação do radical “crianç” com a vogal temática nominal “a”, forma-se o tema “criança”. Embora seja a junção de dois tipos de morfemas, o tema também é considerado como um morfema.

A forma lexical assumida pelo **tema** coincide com as formas de lexema, lexia e lema, que são as formas de entrada dos verbetes em dicionários, e que serão explicadas na Seção 4.1.3.

Ressalte-se, no entanto, que os termos **tema** e **lema**, na Linguística Textual, representam ideias completamente diferentes desses conceitos da morfologia.

4.1.1.6 Considerações até então

Por fim, vale dizer que todos esses tipos de morfemas explicados nas subseções anteriores podem ser agrupados em duas categorias: (i) **morfemas lexicais**, que representam a família semântica de determinada palavra, ou seja, a raiz, o radical e o tema; (ii) **morfemas gramaticais**, que inserem alguma informação à palavra existente, ou seja, as desinências, os afixos e as vogais temáticas.

Há ainda um tipo de morfema (ou fonema, dependendo da abordagem) que não foi explorado aqui porque não é relevante para os estudos de PLN, que são as vogais e as consoantes de ligação. Elas não possuem significado, mas, por vezes, são inseridas entre um radical e uma desinência ou um afixo, por uma motivação fonológica.



4.1.2 *Token e Type*

Token é um termo que significa qualquer sequência de caracteres à qual se atribui um valor. Nas línguas europeias, a sequência consiste em caracteres delimitados por espaços gráficos, sendo que a tokenização é ajustada para separar sinais de pontuação. Mas, na grande maioria das línguas, a tokenização não opera por espaços gráficos. Diante dessa definição, é comum associarmos *token* à palavra escrita. Nesse sentido, a quantidade de palavras + sinais de pontuação de uma sentença equivale à quantidade de *tokens*, por exemplo, a frase do Exemplo 4.2 contém 12 *tokens*, já que, em PLN, os sinais de pontuação (vírgula e ponto final) também são considerados *tokens*².

Exemplo 4.2.

Eu sempre viajo para Campinas, para Salvador e para Belém.

Type, por sua vez, refere-se aos *tokens* únicos encontrados numa frase ou texto. Retomando a frase do Exemplo 4.2, encontramos 10 *types* (“eu”, “sempre”, “viajo”, “para”, “Campinas”, “,”, “Salvador”, “e”, “Belém” e “:”). Nessa sentença, a palavra “para” ocorre três vezes, então ela é contada 3 vezes como *token*, mas apenas 1 vez como *type*.

A proporção *token/type* (divisão da quantidade de *tokens* pela quantidade de *types*) é um importante indicativo da riqueza lexical de um texto; ou seja, ela indica qual a diversidade de palavras existentes em um *corpus*, excluindo suas repetições. Mas, nessa medida, apenas as formas de palavras (as palavras diferentes) e o número total de palavras são contados. Isto é, sinais de pontuação não são considerados.

4.1.3 Lexema, Lexia e Lema

Lexema é sinônimo de unidade lexical, o que implica características de som, forma e significado. Por exemplo, “comprei” é um lexema cuja representação fonética é [kõpr' ej] ; morfologicamente, é um verbo flexionado na primeira pessoa do singular, no pretérito perfeito do modo indicativo. Seu significado é o que encontramos nos dicionários: adquirir (algo, produto, serviço etc.) em troca de pagamento.

Vale assinalar que, nos estudos do léxico do Brasil, temos também o termo técnico **lexia**, que corresponde à realização concreta de um **lexema**. Por exemplo, um **lexema** – que seria uma forma em abstrato, como “árvore” – pode acontecer sob a forma de uma **lexia** como “árvores”. **Lexia** é, nessa perspectiva, uma “forma que um **lexema** assume no discurso. Exemplo: ‘O dia está claro.’ Temos aí quatro lexias” (Biderman, 1978, p. 130). A **lexia** realiza-se no discurso e/ou texto e se distingue do **lexema**, que se situa ao nível do sistema abstrato que é a língua. Em resumo, o **lexema** é uma representação conceitual enquanto a **lexia** é a unidade linguística materializada no discurso.

Como você pode perceber, em linguística, temos vários termos para designar, algumas vezes, uma mesma noção. Por isso, um termo como **palavra** também equivale a **vocabulo**. Se quiser saber mais sobre essas diferentes concepções linguísticas, no âmbito dos estudos do léxico, vale dar uma olhada na parte introdutória do trabalho de Sarmiento (2019).

Por sua vez, **lema** é a representação das propriedades sintático-semânticas de um item lexical. Isso significa que, a partir de um lema, é possível saber quais argumentos a ele se relacionam. Por exemplo, “comprar” é um verbo que seleciona dois argumentos: um

²Para isso, em muitos modelos, é necessário uma etapa a mais de processamento para isolar o sinal de pontuação da palavra anterior, já que graficamente eles costumam estar juntos.



sujeito e um objeto. Esses dois argumentos são necessários para que a estrutura na qual ele está inserido seja gramatical, ou seja, aceita e compreendida pelos falantes. Além disso, é por meio do lema que se pode acessar seu significado: “comprar” remete a uma ação que envolve uma moeda e a obtenção de algo. Nesse sentido, o lema pode ser considerado uma parte do lexema.

A palavra, na forma de lema, é também a forma de entrada dos verbetes em um dicionário, tendo-se em mente também uma categoria de palavra. Por isso, temos “filósofo” como o lema dos substantivos “filósofo”, “filósofos”, “filósofa” e “filósofas” e temos “filosofar” como o lema de “filosofei”, “filosofamos”, “filosofemos” e todas as demais flexões do verbo.

4.1.4 Léxico e Gramática

Se é verdade que não existe língua sem gramática, mais verdade ainda é que sem léxico não há língua. As palavras são a matéria-prima com que construímos nossas ações de linguagem.

(Antunes, 2017, p. 27)

A afirmação de Antunes (2017) na epígrafe diz muito e coloca em relação os elementos que estruturam e fazem funcionar uma língua. Um conjunto de regras sistemáticas servem para definir o que é considerado certo ou normal em uma língua. Por exemplo, segundo a regra de ordem de palavras, usamos, em um discurso normal, não poético, a frase “ele leu o livro” e não “o leu ele livro”. Regras como essa apontam para **gramática**, enquanto o **léxico** corresponde ao conjunto de palavras de uma língua.

Léxicos contêm as palavras de uma língua juntamente com as definições morfossintáticas (Seção 4.1.8) possíveis para cada uma das palavras. Geralmente cada palavra do léxico tem associada a ela uma ou mais triplas com sua categoria gramatical, também chamado de PoS (*Part-of-Speech*), seu lema e suas características morfológicas, também chamadas de *features*.

As categorias gramaticais podem variar segundo os critérios da representação que será adotada, podendo seguir um entre diversos padrões. Porém, para o português é usual definir as categorias gramaticais: **substantivos**, **adjetivos**, **nomes próprios**, **numerais**, **pronomes**, **preposições**, **conjunções**, **advérbios** e **verbos**. Dependendo dos critérios escolhidos, pode-se incluir outras categorias como **artigos** ou **determinantes**. Pode-se também promover divisões, como por exemplo dividir as **conjunções** em **conjunções coordenativas** e **conjunções subordinativas**, ou ainda **verbos** em **verbos auxiliares** e **verbos plenos**. A escolha do conjunto de categorias possíveis é a primeira decisão para a construção de um léxico. Um exemplo de categorias adotadas para o português é apresentado na Seção 4.3.3.

Igualmente, a escolha das características morfológicas que serão consideradas e seus valores possíveis é também uma decisão importante que deve ser tomada. É usual que a definição de categorias de PoS e características morfológicas seja acompanhada de um conjunto de etiquetas (em inglês, *tags*) que serão usadas para representar as informações associadas a cada palavra do léxico. Por exemplo, uma entrada de um léxico para a palavra “elas” pode conter a categoria gramatical pronome, o lema “ele” e características de pronome pessoal na terceira pessoa do plural e gênero feminino. Neste exemplo, trata-se de uma palavra que só possui uma possível tripla PoS, lema e *features*. No entanto, é bastante comum encontrarmos palavras que possuem diversas triplas possíveis de informações associadas, como por exemplo, a palavra “casas” que pode ser:



- um verbo, com o lema “casar”, no presente do indicativo, na segunda pessoa do singular;
- um substantivo, com o lema “casa”, que é do gênero feminino e está no plural.

Cabe salientar que pela própria natureza das línguas, por mais completo que um léxico possa ser, sempre é possível ter palavras da língua ausentes do léxico.

4.1.5 Léxico comum e Léxico especializado

O **léxico comum** corresponde ao conjunto de palavras de uma língua que não têm um “conceito técnico-científico” bem determinado, historicamente construído, atrelado a ela. Em contrapartida há o **léxico especializado**, no qual a palavra assume um significado específico/especial em relação a um sistema de conceitos específico, que geralmente corresponde a uma área de conhecimento, ciência ou especialidade. É esse ambiente “especializado” que definirá se ela pode ser entendida como uma terminologia “técnica” (**termo**) ou uma palavra comum. Um item que a gente lê e diz que é um termo é, por exemplo, “ferritina”, enquanto “caderno” parece um protótipo de palavra comum, do léxico comum. Novamente, pode-se pensar que a categorização ou classificação são referências e que sempre pode haver algo que parece um meio-termo.

E há ainda termos técnicos que passam a funcionar como palavras, na língua comum, e vice-versa. Um exemplo interessante é o caso de “criança”, que no âmbito jurídico corresponde a uma pessoa que tem até doze anos de idade, conforme vemos no Estatuto da Criança e do Adolescente (ECA)³ do Brasil. Isto é, nesse “cenário especializado” do ECA, a palavra “criança” assume status de terminologia, pois corresponde a um conceito específico, oposto a outros. Já a palavra “acetona”, como sinônimo de “removedor de esmalte de unhas”, é algo que fez o caminho inverso, uma vez que passou de termo do léxico especializado a palavra do léxico comum.

Veja como fica o termo “DNA”, que é uma sigla para um termo “técnico” em inglês, nome de um ácido, que é usado em diferentes situações e parece circular entre o ambiente técnico Exemplo 4.4 e o ambiente da linguagem comum Exemplo 4.3, do nosso dia a dia – pois passou a corresponder a um nome de um exame para confirmação de paternidade.

Exemplo 4.3.

No programa de TV, Joana disse que ia jogar na cara do ex-namorado um **DNA**. E disse que ele, depois, ia ver que o filho que ele renegou tem um DNA de gente de bem.

Exemplo 4.4.

O **DNA** (ácido desoxirribonucleico) é um tipo de ácido nucleico que possui papel fundamental na hereditariedade, sendo considerado o portador da mensagem genética.⁴

Vale mencionar, também, que para aplicações que envolvem mais do que um idioma, como a Tradução Automática (Capítulo 18), os léxicos são bilíngues (ou multilíngues) especificando não apenas as palavras que compõem os léxicos dos vários idiomas, mas também o mapeamento (paralelismo) existente entre palavras de um e outro(s) idioma(s).

³https://www.planalto.gov.br/ccivil_03/leis/l8069.htm

⁴Fonte: <https://www.biologianet.com/biologia-celular/dna.htm>



4.1.6 Palavras funcionais e palavras lexicais

As **palavras funcionais/gramaticais** e as **palavras lexicais** são outra dualidade, também complexa, que podemos tentar “resolver” ou melhor, entender, pensando em classificá-las. As palavras funcionais/gramaticais ficam em uma **classe fechada**. Já as palavras lexicais ficam em outro grupo ou tipo, pensando que correspondem a uma **classe aberta**. A classe fechada é assim pensada porque tem um número finito de componentes. A classe aberta, por outro lado, acomoda um número bem maior de componentes, pois é uma classe que tem a ver com a capacidade de as pessoas criarem palavras novas.

Podemos pensar que as preposições do português são as mesmas desde sempre; não criamos muitas. Já os adjetivos e os substantivos não param de nos surpreender, pois parece que há uma inventividade envolvida em nomes e qualificativos, como o adjetivo, que também pode ser substantivo “cloroquiner”. Essa nova palavra surgiu no contexto da Pandemia de Covid-19, em 2020, no Brasil.

Pensar em conjuntos também nos ajuda a entender essa diferença entre funcional/gramatical e lexical. Mas sempre poderemos pensar que uma palavra como “não” é uma palavra lexical, se o critério para classificar for “palavra que tem um sentido” em si mesma. Via de regra, algumas classes de palavras são sempre consideradas como de classe aberta, como os verbos, os adjetivos e os substantivos, enquanto outras classes são sempre definidas como de classe fechada, tais como os artigos (determinantes), as preposições e as conjunções. Outras classes, como os advérbios ou os pronomes, por exemplo, podem ser considerados palavras lexicais ou funcionais, dependendo de suas subclassificações.

4.1.7 Processos de formação das palavras

Existem dois tipos de processos usados para a formação de novas palavras: (i) por derivação e (ii) por composição. São mecanismos linguísticos que permitem criar novas palavras a partir de unidades já existentes na língua.

A **derivação** é um processo pelo qual novas palavras são criadas adicionando afixos (prefixos, sufixos, infixos etc.) à raiz ou radical. Esses afixos podem alterar o significado, a classe gramatical (substantivo, adjetivo, verbo etc.) ou outros aspectos da palavra base. Por exemplo, considere o substantivo “amigo”. Se adicionarmos o sufixo “-ável” a ele, obtemos o adjetivo “amigável”. Nesse caso, o sufixo altera o sentido da palavra e também sua classe gramatical.

Existem cinco tipos de derivação:

1. **prefixal**, quando se adiciona um prefixo ao radical;
2. **sufixal**, quando se adiciona um sufixo ao radical, como no exemplo acima;
3. **parassintética**, quando se adiciona ao mesmo tempo um prefixo e um sufixo ao radical, como no caso de “desmatamento”, que é derivado de “mata”;
4. **imprópria**, quando muda a categoria gramatical da palavra, mas sem alterar sua forma, como no caso de “Ela tem um andar lento”, em que “andar” originalmente é um verbo, mas passa a ser um substantivo nesse contexto; e
5. **regressiva**, quando se suprime uma desinência de um verbo para formar um substantivo, como é o caso de “choro”, que é derivado de “chorar”.

Já a **composição** é um processo em que novas palavras são formadas combinando duas ou mais palavras independentes, ou dois radicais, para criar uma nova palavra com um significado diferente. As palavras compostas podem ser formadas por substantivos,



adjetivos, verbos, advérbios e outras classes gramaticais. Além disso, elas podem ser escritas juntas, separadas por hífen ou até mesmo separadas sem qualquer marcação, dependendo da língua e das convenções ortográficas. São exemplos de palavras formadas por composição: “girassol” (“gira” + “sol”), “planalto” (“plano” + “alto”), “guarda-chuva” (“guarda” + “-” + “chuva”).

Existem 2 tipos de composição:

1. por **justaposição**, em que uma nova palavra é formada a partir da união de dois ou mais radicais, sem apresentar alterações nos seus sons, ou seja, sem alterações fonéticas, como em “cachorro-quente” (“cachorro” + “-” + “quente”), “passatempo” (“passa” + “tempo”), “guarda-chuva” (“guarda” + “-” + “chuva”); e
2. por **aglutinação**, em que as palavras também são formadas pela união de dois ou mais radicais, porém sofrem alterações, como “vinagre” (“vinho” + “acre”), “embora” (“em” + “boa” + “hora”) e “fidalgo” (“filho” + “de” + “algo”).

Ambos os processos de derivação e de composição são fundamentais para a expansão do vocabulário e a expressão de nuances semânticas na linguagem.

4.1.8 Morfologia e morfossintaxe

Por fim, mas não menos importante, é necessário definir o escopo de estudo do que chamamos de morfologia, pois o seu objeto de estudo muitas vezes se intersecta com o objeto de outra área da linguística, a chamada Sintaxe, que será explorada no Capítulo 6. Na fronteira entre a morfologia e a sintaxe, está a morfossintaxe. Na prática, essas três áreas estão intimamente ligadas e relacionadas, mas, para fins didáticos, distinguimos esses termos a partir de seus objetos de estudo.

A **morfologia** é o ramo da linguística que se concentra no estudo dos morfemas, que são os “pedacinhos” significativos que formam as palavras. Assim, a morfologia examina como eles se combinam nos processos de flexão e de formação de palavras. Em PLN, a morfologia cuida também da classificação dos atributos morfológicos (ou *features* morfológicas), tais como os traços de gênero, número, modo, tempo, pessoa, voz, caso, entre outros.

Já a **morfossintaxe** examina como as escolhas morfológicas (como flexões verbais e concordância nominal) afetam a organização das palavras em uma sentença e como essas escolhas influenciam a estrutura sintática. Em outras palavras, ela categoriza as palavras em diferentes classes de palavras (ou categorias gramaticais) a partir da observação de seus atributos morfológicos. Em PLN, as classes de palavras são chamadas de *part-of-speech* ou PoS e a tarefa de atribuição de etiquetas de PoS nos textos será explicada na Seção 4.2.5.

Em resumo, a morfologia lida com a estrutura interna das palavras e os morfemas que as compõem, enquanto a morfossintaxe explora como as escolhas morfológicas afetam a estrutura das frases. Ambas as áreas são consideradas neste capítulo.

4.2 O processamento morfológico em PLN

Após definir conceitos necessários da área de morfologia, demonstraremos como tratar esse nível de análise linguística no Processamento de Linguagem Natural. Para desenvolver praticamente qualquer aplicação de PLN, é necessário realizar fases/etapas que convencionamos chamar de **pré-processamento**. Nesse pré-processamento, algumas tarefas usuais são: segmentação do texto em sentenças (sentencição), separação de palavras



(tokenização), tokenização em subpalavras (vetorização de *subtokens*), normalização de palavras (lematização e radicalização), entre outras.

Além das etapas do pré-processamento, também podem ser realizadas tarefas de processamento do conteúdo dos textos, como a etiquetagem morfosintática das palavras em relação às suas classes gramaticais (tarefa de PoS *tagging*) e a anotação automática de seus atributos morfológicos (tarefa de anotação de *feats* ou *features* morfológicas), que também serão exploradas nesta seção.

4.2.1 Sentenciação

A sentença (ou sentenciamento) é o processo de segmentação do texto em sentenças, ou seja, é o processo de identificação de unidades textuais de processamento onde se definem os limites de cada sentença. A denominação de **detecção de limite de sentença** é frequentemente utilizada como sinônimo da segmentação de sentenças, pois o problema se limita a descobrir onde cada sentença termina (Hapke; Howard; Lane, 2019). Este processo é naturalmente complexo, pois a ambiguidade das línguas torna impossível ter sempre certeza de onde termina uma sentença (Read et al., 2012).

No caso do português escrito, as técnicas usuais se valem da busca de pontuações delimitadoras como “.”, “!” e “?”. Note-se que no processamento de textos falados, ou mesmo em algumas línguas onde a delimitação de sentenças não é feita por pontuação, o processo de segmentação de sentenças se torna ainda mais difícil.

A detecção do limite de sentença no português não tem como desafio identificar as pontuações delimitadoras, pois esse é usualmente um conjunto finito e conhecido (“.”, “!” e “?”, “..”). O desafio é desambiguar essas ocorrências com outros usos dos mesmos caracteres. Um exemplo disto é o caso das abreviações. Por exemplo, na sentença do Exemplo 4.5 temos duas ocorrências do caractere “.”.

Exemplo 4.5.

Fui à clínica do Dr. Nilo.

Na primeira ocorrência, o “.” é utilizado como indicador da abreviação da palavra “Doutor” e, na segunda, como delimitador do fim da sentença. Note-se que a sentença do Exemplo 4.6 é uma sentença perfeitamente aceitável e, neste caso, o “.” está sendo utilizado duplamente como indicador de abreviação e fim de sentença.

Exemplo 4.6.

Fui à clínica do Dr.

Outro caso comum de ambiguidade no uso de pontuações delimitadoras é encontrado em numerais. Em português, utiliza-se também o caractere “.” como delimitador de milhar em um número, enquanto em inglês ele é utilizado como separador de decimais. Algumas vezes ambos os usos aparecem, ainda que erroneamente, em sentenças em português, como no Exemplo 4.7.

Exemplo 4.7.

A venda de 25.000 ações fez o índice de rentabilidade baixar para 0.5%, segundo a BOVESPA.



Um problema semelhante acontece com algumas definições matemáticas dentro de uma sentença, como quando se utiliza o caractere “!” para definir o fatorial de um número, como no Exemplo 4.8.

Exemplo 4.8.

As permutações de cinco elementos podem ser calculadas como $5!$, que é o fatorial de cinco.

Em todos estes casos, torna-se difícil detectar quando o caractere está sendo utilizado com função de fim de sentença ou não.

Por essas razões, o problema de segmentação automática de textos, ainda que explorado desde o início pela área de PLN, é bastante desafiador e ainda está em aberto. Atualmente utilizam-se três tipos de abordagens computacionais para resolvê-lo:

- **Abordagens baseadas em regras**, onde são definidos padrões de fim de sentença através de regras que podem incluir, por exemplo, heurísticas, abreviações usuais, expressões regulares para números e URLs. Este é, em geral, o método implementado em segmentadores de sentenças disponíveis em pacotes como o NLTK⁵ (Bird; Loper, 2004).
- **Abordagens baseadas em aprendizado de máquina supervisionado**, ou seja, modelos computacionais treinados sobre conjuntos anotados (*gold standard*, veja Capítulo 14) onde o desafio é desenvolver um conjunto de treino de tamanho e características relevantes para os textos que se pretende sentenciar.
- **Abordagens baseadas em aprendizado de máquina não supervisionado**, ou seja, modelos computacionais treinados sobre conjuntos não anotados, mas que são suficientemente grandes e representativos para que se possa construir um modelo de linguagem adequado.

O problema de segmentação de sentenças é extremamente importante, pois, por ser uma etapa inicial do pré-processamento, os problemas não resolvidos nessa etapa tendem a prejudicar as etapas posteriores. Em comparação com outras tarefas de PLN, a segmentação de sentenças costuma receber menos atenção do que deveria, tanto no desenvolvimento de pesquisas, quanto na implementação de casos práticos.

Vale esclarecer que a sentencição é uma tarefa de pré-processamento que tem relação com a morfologia porque, apesar de sua unidade de análise ser a sentença, os casos de ambiguidade (e, portanto, segmentação incorreta) têm a ver com a delimitação das palavras, que é o objeto de estudo da morfologia.

Na Seção 4.3.1, indicaremos alguns sentenciadores (ou também chamados *sentencizers*) disponíveis para o português.

4.2.2 Tokenização

A separação em unidades linguísticas mínimas é denominada **tokenização** (em inglês, *tokenization*) e, como já mencionado anteriormente, no caso do português é feita partindo da separação das palavras através de delimitadores. Neste caso, faz-se necessário identificar os limites das palavras através de caracteres delimitadores como espaços em branco ou símbolos de pontuação como “,”, “:”, “;”, “-” e “.”. Novamente, aqui é necessário se atentar

⁵<https://www.nltk.org>



para casos específicos como “,”, “-” e “.” que não devem ser separados dos demais caracteres que vêm antes ou depois. Por exemplo, a sentença do Exemplo 4.9 possui 11 *tokens*: “Além”, “disso”, “,”, “a”, “produção”, “será”, “descontinuada”, “em”, “8,3”, “%” e “.”, sendo que 8,3 deve ser considerado um token único e não 3 tokens separados.

Exemplo 4.9.

Além disso, a produção será descontinuada em 8,3%.

Outra tarefa frequente da tokenização é a separação de **palavras contraídas**, por exemplo, a palavra “da” é separada em dois *tokens*: “de”+“a” e a palavra “nelas” é separada nos *tokens* “em”+“elas”. Essa tarefa é necessária para diversas aplicações e, em muitos casos, é um processo simples, pois a palavra contraída não é ambígua. No entanto, em alguns casos, pode ser necessário um processo de desambiguação, como nas palavras “pelo” (que pode ser “por”+“o” ou o substantivo “pelo”) e “consigo” (que pode ser tokenizada em “com”+“si” ou corresponder à conjugação do verbo “conseguir”). Esse é ainda um dos desafios da tokenização em português. A Figura 4.1 mostra o resultado da tokenização usando uma das ferramentas atualmente disponíveis para o português⁶ para a sentença do Exemplo 4.10.

Exemplo 4.10.

A raça Lulu da Pomerânia solta pelos em abundância pela casa.

Figura 4.1: Exemplo de tokenização com acertos e erros de descontração de palavras

```
<p> <s> A raça Lulu de_ a Pomerânia solta por_ os em abundância por_ a casa ./ </s> </p>
```

Na Figura 4.1 observamos que “da” e “pela” foram corretamente descontraídas como “de_ a” e “por_ a” respectivamente. Já o substantivo “pelos” foi incorretamente separado em “por_ os”. Problemas de descontração indevida, ou da falta dela quando necessária, geram problemas para a análise sintática, como veremos no Capítulo 6.

Além desses casos, há também outros que envolvem a decisão de separar ou não **palavras hifenizadas**. Por exemplo, usualmente não se separa palavras como “sexta-feira”, que correspondem a um substantivo único. No entanto, é usual tokenizar palavras como “sinto-me” em três *tokens*: o verbo “sinto”, o sinal de pontuação “-” e o pronome “me”. Mais uma vez a aplicação pretendida é que definirá o que deve ser feito ou não.

A tokenização, assim como o sentenciamento, é um processo que pode ser resolvido com estratégias baseadas em regras ou que utilizam as mesmas abordagens de aprendizado de máquina supervisionadas e não supervisionadas citadas anteriormente. A complexidade do processo de tokenização é, no entanto, menor que a do processo de sentenciamento, pois o processo pode ser auxiliado pela existência de recursos léxicos que facilitam bastante a tarefa de identificar os limites possíveis da maioria dos *tokens* a serem separados.

Na Seção 4.3.1, apresentaremos os tokenizadores (ou também chamados *tokenizers*) disponíveis para o português.

⁶<https://portulanclarin.net/workbench/lx-tokenizer/>



4.2.3 Tokenização em Subpalavras

Outro conceito relacionado ao de unidade de processamento que se tornou bastante popular nos últimos tempos (principalmente com o surgimento das arquiteturas neurais para processamento da língua) é o de **subpalavra** (em inglês, *subword*). As aplicações recentes de modelos de linguagem baseados em redes neurais tornaram bastante comum a quebra de palavras em porções eventualmente menores, as subpalavras. Esse processo de **tokenização em subpalavras** tem por objetivo reduzir o vocabulário de trabalho de um modelo de linguagem a um tamanho finito, mas que possa ser usado para representar textos onde o número de *types* (quantidade de *tokens* distintos) seja potencialmente infinito. Dessa forma, pode-se utilizar um conjunto de treino que possua um vocabulário finito e que seja capaz de ser aplicado a qualquer texto⁷.

Por exemplo, um modelo de linguagem pode ser projetado para ter um vocabulário de trabalho com 30.000 *types*, ou seja, um índice numérico de 1 a 30.000 único para cada *type* a ser representado de forma que se possa definir um vetor de 30.000 posições, em que cada posição corresponde a um dos *types* considerados. Essa abordagem é necessária para utilizar representações vetoriais como, por exemplo, as representações word2vec (Mikolov et al., 2013a) e GloVe (Pennington; Socher; Manning, 2014) a serem tratadas no Capítulo 10. Apesar dos vocabulários de trabalho serem usualmente grandes, esses vocabulários são insuficientes para representar textos em português associando uma possível palavra da língua a cada *type*, já que o português possui mais de 800.000 palavras, sem contar novas palavras que podem ser criadas, como o exemplo “cloroquiner” citado na Seção 4.1.6.

A abordagem de tokenização em subpalavras consiste em codificar diretamente algumas palavras mais comuns, como “de”, “fazer”, “são” e “feliz”. No entanto, palavras mais raras, como “desfazer” ou “felizmente” podem ficar fora do vocabulário de trabalho (OOV do termo em inglês *out-of-vocabulary*) e portanto serem representadas como combinações de subpalavras, respectivamente: “de” + “s” + “fazer” e “feliz” + “mente”.

Dessa forma, uma subpalavra pode ser uma sequência qualquer de caracteres ou podem ser sequências com algum significado linguístico como prefixos e sufixos (e.g., “mente” comumente designa advérbios, como “felizmente”), mas até letras únicas de forma que sempre seja possível representar qualquer palavra pela composição de subpalavras pertencentes ao vocabulário (e.g., a subpalavra “s”, que é uma letra única, pode ser utilizada para, junto com as subpalavras “de” e “fazer”, ser articulada para representar “de” + “s” + “fazer”).

A escolha do vocabulário de trabalho, ou seja, a escolha das subpalavras que o compõem, pode ser feita utilizando diversas técnicas, mas três algoritmos são frequentemente utilizados: BPE (*Byte-Pair Encoding*) (Sennrich; Haddow; Birch, 2016b), Word-Piece (Schuster; Nakajima, 2012) e Unigram (Kudo, 2018). O primeiro, BPE, é inspirado em técnicas de compreensão de dados e busca representar como subpalavras os *tokens* mais frequentes. O segundo, Word-Piece, é utilizado pelo BERT (Devlin et al., 2019) e busca reduzir o tamanho do vocabulário através da escolha de subpalavras que possam ser utilizadas em um número maior de *tokens*. O terceiro, Unigram, também foca na redução do vocabulário e inicializa o treinamento com um vocabulário grande de caracteres, subpalavras e palavras, e vai reduzindo esse vocabulário mantendo somente os itens mais relevantes até alcançar o

⁷Geralmente, um modelo computacional é treinado em um conjunto de dados e usado em outros conjuntos diferentes. Nesse sentido, palavras que nunca foram vistas no conjunto de dados de treinamento são consideradas palavras desconhecidas (*unknown*) nos novos conjuntos. A existência de uma palavra desconhecida faz com que o modelo não saiba como tratá-la, o que pode impactar (e até mesmo inviabilizar) o processamento.



tamanho de vocabulário desejado.

4.2.4 Normalização

Por sua vez, a **normalização** é a tarefa que converte as palavras para alguma forma padrão. São exemplos de normalização: conversão de versões abreviadas de palavras (e.g., conversão de “vc” para “você”), conversão para caracteres minúsculos (e.g., convertendo “Você” para “você”), lematização (e.g., estabelecendo que “somos” é uma conjugação do verbo “ser”) e radicalização (e.g., estabelecendo que “retrabalho” tem o radical “trabalho”⁸ precedido do prefixo “re”). De acordo com a abordagem utilizada, diferentes tipos de normalização podem ser necessários para tratar de maneira mais eficiente o processamento textual.

Na verdade, o propósito do tratamento computacional tem muita influência nos tipos de normalização que precisam ser feitos. Em alguns casos, certas informações no texto a ser processado podem ser vistas como relevantes, enquanto outras como apenas ruído. Por exemplo, quando temos como propósito identificar o sentido geral de um texto, a conversão de abreviaturas tende a auxiliar a identificação do conteúdo considerando, por exemplo, “PLN” e “Processamento de linguagem natural” como equivalentes. Por outro lado, com o propósito de identificar entidades nomeadas, a conversão para caracteres minúsculos pode dificultar o processamento.

A tarefa de **conversão de abreviações** é usualmente baseada em listas predefinidas de abreviações comuns que podem ser utilizadas em um processo de busca e substituição. No entanto, alguns cuidados usuais devem ser tomados para que somente *tokens* completos e com o sentido apropriado sejam substituídos. Por exemplo, uma abreviação usual em textos de mensagens é a representação “rs” como abreviação de “risos”, mas a substituição desta abreviação, que está correta na sentença do Exemplo 4.11, ficaria completamente errada na sentença do Exemplo 4.12, pois poderia resultar em “Os youtuberis do RS tem muito sotaque.” que substituiria erroneamente o final da palavra “youtubers”.

Exemplo 4.11.

Eu já sabia... rs.

Exemplo 4.12.

Os youtubers do RS tem muito sotaque.

A tarefa de **conversão para caracteres minúsculos** é menos delicada, podendo ser facilmente implementada por simples processamento da representação individual dos caracteres. No entanto, mesmo nesse caso é necessário tomar certas precauções com nomes próprios e outras representações onde existe semântica associada ao uso de maiúsculas e minúsculas. Por exemplo, se fizermos a substituição de maiúsculas por minúsculas da frase do Exemplo 4.12, podemos erroneamente descaracterizar a denominação do estado do Rio Grande do Sul (“RS”), tornando o texto produzido ambíguo com a abreviação “rs”. Apesar disso, o processo de conversão para caracteres minúsculos é uma tarefa usual, pois permite a consulta a recursos linguísticos de uma forma mais eficiente.

⁸Vale lembrar que o conceito de “radical” não inclui a vogal temática. No exemplo de “trabalho”, o radical correto seria “trabalh” e o “o” seria uma vogal temática, portanto a forma “trabalho”, na verdade, é o tema (e não o radical), porém essa distinção é muito específica e desnecessária para o PLN. Assim, a tarefa de radicalização, em PLN, pode considerar ora o radical ou a raiz, ora o tema.



A tarefa de **lematização** envolve, frequentemente, a consulta a recursos linguísticos que possuem a definição de lemas e morfologia das palavras, como, por exemplo, um léxico da língua (Seção 4.3.2). O grande desafio desta tarefa é a desambiguação sintática das palavras que, segundo seu uso, podem ter lemas distintos. Por exemplo, na sentença do Exemplo 4.13 a palavra “casa” na sua primeira ocorrência terá como lema o verbo “casar”, enquanto a segunda ocorrência terá como lema o substantivo “casa”. Nesses casos é importante desambiguar morfossintaticamente as palavras (Lopes et al., 2023).

Exemplo 4.13.

Quem casa, quer casa.

Outra abordagem utilizada é através de padrões, a fim de trazer a palavra para sua forma canônica (e.g., trazer substantivos para o masculino singular e todas as flexões do verbo para sua forma no infinitivo) (Bertaglia; Nunes, 2016). Note-se que essa abordagem requer sofisticções para tratar palavras que não têm comportamento regular. Por exemplo, se a palavra “meninas” pode ser trazida corretamente ao lema substituindo a terminação “as” por “o” resultando no lema “menino”, a palavra “casas” seria erroneamente lematizada para o lema “caso” se utilizássemos o mesmo princípio.

A tarefa de **radicalização** tem o propósito de converter lexemas para seus radicais. Uma particular vantagem deste tipo de tarefa é uniformizar e diminuir o vocabulário, ainda que possam levar à perda de informação. No entanto, em algumas aplicações, a busca dos radicais de uma palavra pode auxiliar, inclusive, no estabelecimento de subpalavras que foi descrito na Seção 4.2.3. Por exemplo, as palavras “certo”, “certidão”, “incerto”, “certamente”, “certificação”, “certo” e “incerteza” possuem o mesmo radical “cert” e, portanto, tornam “cert” um bom candidato a subpalavra. Algoritmos de radicalização (*stemming*, em inglês) podem ser encontrados em bibliotecas usuais da área de PLN, como NLTK (Bird; Loper, 2004), que oferecem opções para várias línguas, inclusive para o português.

Na Seção 4.3.1, apresentaremos algumas ferramentas disponíveis para o português que fazem a normalização dos textos, tais como lematizadores, radicalizadores, *stemmers* e outros.

4.2.5 PoS *tagging*

O PoS *tagging*, também conhecido como etiquetagem morfossintática, é uma técnica fundamental na área de PLN que envolve a atribuição de etiquetas gramaticais a cada palavra em um texto, com base na sua classe gramatical e em suas características morfológicas. Essas etiquetas ajudam a identificar a função sintática e morfológica das palavras em uma sentença, o que é crucial para a posterior análise sintática, mas também é útil para outras aplicações, como tradução automática (Capítulo 18), análise de sentimentos, geração de resumos, entre outras.

As classes de palavras são universais e valem para a grande maioria das línguas naturais, incluindo o português. São elas: substantivos, verbos, adjetivos, advérbios, pronomes, numerais, artigos, conjunções, preposições e interjeições. Porém, as etiquetas de PoS que cada modelo de anotação define podem ser diferentes, assim como pode haver diferentes níveis de granularidade das etiquetas. Nesse sentido, um *tagger* (nome dado a uma ferramenta computacional que realiza a etiquetagem morfossintática) pode usar apenas as etiquetas de granularidade grossa (em inglês, *coarse tags* ou UPoS), que correspondem



às classes de palavras acima, enquanto outros *taggers* podem usar um conjunto de etiquetas de granularidade fina (em inglês, *fine-grained tags* ou XPoS). Por exemplo, os adjetivos costumam ser associados à *coarse tag* ADJ, mas também podem ser etiquetados como JJ (para adjetivos primitivos), JJR (para adjetivos comparativos) ou JJS (para adjetivos superlativos). Horsmann; Zesch (2016) propõem uma abordagem que combina os dois níveis de anotação para aumentar a precisão do *tagger*.

A Figura 4.2 mostra um exemplo de etiquetagem morfossintática realizada pelo *parser* LXUTagger⁹.

Figura 4.2: Exemplo de anotação de PoS com *coarse tags*

DET NOUN ADJ VERB ADV PUNCT
<p> <s> O gato preto correu rapidamente . </s> </p>

Cada palavra é analisada quanto à sua forma e função na sentença. É necessário fazer essa análise dependente de contexto porque existem palavras polissêmicas, ambíguas, homônimas etc. Dentro do contexto, é possível, por exemplo, distinguir o substantivo “ajuda” (e.g. “Ele recebeu *ajuda* para executar o trabalho.”) do verbo “ajuda” (e.g. “Sempre que precisa, ele *ajuda* a comunidade.”).

Várias abordagens são possíveis e usuais para esse tipo de tarefa, desde a anotação manual das etiquetas e posterior treinamento de modelo, até o uso de redes neurais. As mais conhecidas ainda hoje são as baseadas em regras (Brill, 1992), em redes neurais artificiais (Schmid, 1994), as abordagens estocásticas (Hall, 2003) e as híbridas (Altunyurt; Orhan; Güngör, 2006; Zin, 2009).

A seguir, explicamos brevemente algumas dessas abordagens, resumizando o trabalho de Zewdu; Yitagesu (2022), que fizeram recentemente uma revisão sistemática da literatura sobre as técnicas usadas para etiquetagem de PoS. Ressaltamos, no entanto, que, para uma visão mais aprofundada, recomendamos ler o artigo original.

- **Abordagem baseada em regras** – Utiliza regras criadas manualmente para atribuir *tags* às palavras de uma sentença. Essas regras podem ser: (i) construídas por especialistas linguistas e, assim, depender de características linguísticas, como informações lexicais, morfológicas e sintáticas, ou (ii) inferidas via aprendizado de máquina, a partir de uma grande quantidade de dados (Capítulo 14), dispensando regras especializadas. A primeira abordagem (geração manual de regras) é demorada e propensa a erros, enquanto a segunda (com uso de aprendizado de máquina em um *corpus* anotado) é mais eficiente, embora ainda exija *expertise* linguística.
- **Hidden Markov Models (HMM)** – É uma das abordagens mais amplamente utilizadas para PoS *taggers* que usam modelos estocásticos (Kumawat; Jain, 2015; Zin, 2009). Nessa abordagem, o *tagger* passa de um estado a outro por meio de um estado oculto. Esse estado oculto não é diretamente observável, mas a saída dependente do estado oculto é visível. O algoritmo de Viterbi é um método bastante conhecido para identificar a sequência mais provável de *tags* $T=\{t_1, t_2, t_3... t_n\}$ para cada palavra em uma sentença $W=\{w_1, w_2, w_3...w_n\}$ ao usar um modelo oculto de Markov.
- **Aprendizado de máquina baseado em *features*** – Os algoritmos de Aprendizado de Máquina mais comuns usados para PoS *taggers* são as redes neurais, *Naïve Bayes*,

⁹<https://portulanclarin.net/workbench/lx-utagger/>



HMM, *Support Vector Machine* (SVM), *Conditional Random Field* (CRF), Brill e TnT. Para maiores informações sobre como cada um deles é usado para a tarefa de PoS *tagging*, ver Zewdu; Yitagesu (2022).

- **Aprendizado profundo** – É uma abordagem intensiva em dados baseada no uso de redes neurais artificiais em vários níveis. Nessa abordagem é usual realizar um pré-processamento dos dados (Dhumal Deshmukh; Kiwelekar, 2020). A saída do pré-processamento é, então, usada como entrada para a primeira camada da rede neural. A partir dessa entrada pré-processada, a rede neural utiliza-se de um modelo de linguagem pré-treinado para atribuir a etiqueta PoS mais provável a cada palavra. Cabe salientar que o treinamento desse modelo de linguagem se faz através de um conjunto de treino que alimenta a rede neural repetidas vezes recalculando os pesos de cada camada da rede. Os métodos sequenciais de aprendizado profundo mais comuns para etiquetagem de PoS são: FNN, MLP, GRU, CNN, RNN, LSTM e BLSTM. Não pretendemos detalhar todos eles aqui, mas indicamos o trabalho de Zewdu; Yitagesu (2022), em que todos eles são descritos em pormenores.

Após a etapa de PoS *tagging* para cada uma das palavras de uma sentença, elas passam então para outra etapa do processamento que também pertence à área da Morfologia, que é a atribuição das *features* (ou atributos) morfológicos, e que serão explorados nas próximas seções.

4.2.6 Anotação de atributos morfológicos

A anotação de atributos morfológicos, também conhecida como atribuição de *features* morfológicas (ou somente *feats*), é uma importante tarefa que envolve a marcação ou identificação de informações específicas sobre as características gramaticais e morfológicas de palavras em um texto. Esses atributos morfológicos incluem características como número, gênero, modo, tempo, pessoa e outras informações semelhantes.

O objetivo principal dessa anotação é capturar e codificar informações gramaticais relevantes de maneira estruturada, tornando possível para algoritmos de PLN e de AM entenderem e processarem corretamente a estrutura e as relações linguísticas presentes em um texto. Por exemplo, considere a palavra “falávamos”. A anotação de atributos morfológicos, nesse caso, pode envolver a marcação de tempo (pretérito imperfeito), modo (indicativo), número (plural), pessoa (primeira) e forma verbal (finita).

No exemplo da Figura 4.2 – “O gato preto correu rapidamente” –, a anotação das *features* morfológicas seria conforme apresentado na Figura 4.3, gerada pela ferramenta LX-USuite.¹⁰

Figura 4.3: Exemplo de anotação de *features* morfológicas

	Gender=Masc Number=Sing	Gender=Masc Number=Sing	Gender=Masc Number=Sing	Mood=Ind Number=Sing Person=3 Tense=Past		
	O	GATO	PRETO	CORRER	RAPIDAMENTE	.
	DET	NOUN	ADJ	VERB	ADV	PUNCT
<p> <s>	O	gato	preto	correu	rapidamente	. </s> </p>

Cabe a cada *tagger* (ou etiquetador morfológico) definir seu próprio conjunto de *features* ou características morfológicas que devem ser anotadas ou extraídas,

¹⁰<https://portulanclarin.net/workbench/lx-usuite/>



o que é chamado de *tagset* (ou conjunto de etiquetas). Por exemplo, o UDPipe, que será abordado na Seção 4.3.1, identifica, para verbos, todas as *features* citadas anteriormente: para o verbo “comprei”, lemma = “comprar” e *features* = “VERB _ Mood=Ind|Number=Sing|Person=1|Tense=Past|VerbForm=Fin”.

A quantidade de etiquetas, bem como seus nomes, pode variar bastante de um *tagger* para outro, e essa falta de uniformização constitui um grande desafio para o PLN. De acordo com Fonseca; Rosa; Aluísio (2015), o Penn Treebank do inglês, por exemplo, conta com 45 *tags* incluindo sinais de pontuação, enquanto o CLAWS5 possui 62 e o CLAWS7 possui 137 *tags*, sendo que os dois últimos usaram o mesmo *corpus* (British National Corpus).

Se o objetivo da tarefa for treinar um etiquetador morfossintático, a anotação pode ser feita manualmente por especialistas humanos que analisam e selecionam os atributos morfológicos para as palavras em um *corpus* de treinamento (Capítulo 14). Posteriormente, modelos de PLN e AM podem ser treinados usando esses dados anotados para automatizar a anotação de novos textos e auxiliar em várias tarefas e aplicações. Por outro lado, já existem também abordagens computacionais semelhantes às mencionadas na Seção 4.2.5, que também podem ser usadas para etiquetar automaticamente as palavras em relação às suas características morfológicas.

Além das tarefas de processamento aqui apresentadas, vale mencionar que há ainda outras, como a análise sintática automática (tarefa de *parsing*), a segmentação dos constituintes sintáticos dentro da frase (tarefa de *chunking*), a extração ou anotação de entidades nomeadas (tarefa de *Named Entity Recognition*) etc. O que nos interessa neste Capítulo são as tarefas ligadas ao processamento morfológico e morfossintático, portanto nos limitamos às mencionadas nesta seção.

4.3 Ferramentas e recursos para o processamento morfológico

Esta seção apresenta ferramentas computacionais e recursos disponíveis para o português, que processam a língua no nível morfológico e morfossintático. Para cada uma das tarefas de (pré-)processamento, existem ferramentas específicas que podem ser usadas para a análise automática do português. Dividimos esta seção em ferramentas (Seção 4.3.1) e recursos (Seção 4.3.2). Ao final da seção, selecionamos um dos recursos para explicá-lo de forma mais aprofundada, a saber, o PortiLexicon (Seção 4.3.3).

4.3.1 Ferramentas para o português

Existem ferramentas que têm como foco tarefas específicas de PLN. Outras, como NLTK¹¹ e spaCy¹², são módulos mais completos que implementam submódulos e funcionalidades diversas. Apresentamos a seguir ferramentas disponíveis para o processamento de textos, separadas por tarefa.

Tokenizadores

- O NLTK contém o `nltk.tokenize`¹³, que é um submódulo que implementa diferentes classes e funções para a tarefa de tokenização, como: tokenização por espaço em

¹¹<https://www.nltk.org>

¹²<https://spacy.io/>

¹³<https://www.nltk.org/api/nltk.tokenize.html>



branco, baseada em pontuação, considerando expressões multipalavra (Capítulo 5), entre outros. Também são implementadas funcionalidades para a personalização da tokenização, por exemplo, baseada em expressões regulares definidas pela pessoa usuária.

- O **tokenizador do spaCy**¹⁴ (ou *tokenizer*) segmenta o texto em palavras e sinais de pontuação. Isso é feito aplicando regras específicas para cada idioma; normalmente é criado automaticamente quando uma subclasse *Language* é inicializada e lê suas configurações, como pontuação e regras de casos especiais. As exceções do *tokenizer* definem casos especiais como “don’t” em inglês, que precisa ser dividido em dois *tokens*: {ORTH: “do”} e {ORTH: “n’t”, NORM: “not”}. Os prefixos, sufixos e infixos definem principalmente regras de pontuação – por exemplo, quando separar pontos (no final de uma frase) e quando manter o ponto pertencendo ao *token* (como nos casos de “Dr.”, “25.000” e o fatorial “5!”, apresentados na Seção 4.2.1).

Lematizadores

- O **lematizador do spaCy**¹⁵ (ou *lemmatizer*) converte as palavras para suas formas básicas (lemas) usando regras baseadas em *tags* de classe gramatical ou tabelas de pesquisa. Diferentes subclasses de linguagem podem implementar seus próprios componentes lematizadores por meio de abordagens específicas para cada língua. Isso torna mais fácil personalizar como os lemas devem ser atribuídos ao seu *pipeline*.

Stemmers

- O submódulo `nlk.stem` implementa diferentes abordagens de *stemming*, incluindo:
 - O **RSLP**¹⁶ (Removedor de Sufixos da Língua Portuguesa), inicialmente proposto por Orengo; Huyck (2001) como um algoritmo que realiza as seguintes etapas: redução do plural (supressão do “-s”), redução de feminino (supressão do “-a”), redução de advérbio (remoção do sufixo “-mente”), redução dos sufixos de aumentativo (“-ão”) e diminutivo (“-inho”), redução de sufixos nominais (como “-mento”, “-ção”, “-ncia”), redução de desinências verbais, remoção de vogais e de acentos gráficos.
 - O **SnowballStemmer**, baseado no algoritmo proposto por Porter (1980) para o inglês e estendido para a linguagem Snowball¹⁷ de processamento de *strings* para criação de algoritmos de *stemming* para diversas línguas, também está disponível para o português.
 - Uma classe para a personalização de *stemming* baseada em expressões regulares.

Sentenciadores

- **Punkt** é uma ferramenta para a segmentação de sentenças em português, ou seja, é um sentenciador. O NLTK inclui um modelo do Punkt¹⁸, que foi treinado inicialmente a partir do *corpus* Floresta Sintá(c)tica (Afonso et al., 2002), mas pode ser retreinado

¹⁴<https://spacy.io/api/tokenizer>

¹⁵<https://spacy.io/api/lemmatizer>

¹⁶<https://www.inf.ufrgs.br/~viviane/rslp/index.htm>

¹⁷<https://snowballstem.org/>

¹⁸https://www.nltk.org/howto/portuguese_en.html#sentence-segmentation



e avaliado em outros *corpora*, lembrando que é mais rápido carregar um modelo treinado do que treiná-lo novamente.

- No spaCy, a sentencição é feita conforme a anotação de dependência é executada, por exemplo, por uma instância da classe *DependencyParser*. No entanto, caso um *parser* não seja incluído no *pipeline*, o **sentenciador do spaCy**¹⁹ (ou *sentencizer*), que permite a detecção dos limites das sentenças com base em regras, pode ser incluído como componente no *pipeline*. O *sentencizer* é bem simples para permitir lógica de detecção de limite de frase personalizada que não requer análise de dependência, possibilitando a implementação de uma estratégia mais simples e baseada em regras que não requer o carregamento de um modelo estatístico.

Normalizadores

- A **Enelvo**²⁰ é uma biblioteca que pode ser usada para normalizar textos em português (Bertaglia; Nunes, 2017). Conforme apresentado na Seção 4.2.4, a etapa de normalização pode envolver várias microtarefas. No caso da Enelvo, ela corrige abreviações, gírias, erros ortográficos, capitaliza letras no começo das frases, de nomes próprios e acrônimos. Ela também possui uma função própria para remover pontuações e emojis.

Taggers

São chamadas de *taggers* as ferramentas de etiquetagem morfossintática e morfológica, ou seja, as ferramentas computacionais que atribuem automaticamente o PoS e as etiquetas morfológicas para cada palavra em uma frase. Antes de mencionar os *taggers* propriamente ditos para o português, convém citar o trabalho de Gonçalves et al. (2020), no qual os autores avaliaram várias ferramentas de etiquetagem morfossintática para o português, como: FreeLing²¹, NLTK²², OpenNLP²³, NLPyPort²⁴, PolyGlot²⁵, spaCy²⁶, StanfordNLP²⁷, TreeTagger²⁸ e LinguaKit²⁹. Algumas delas fazem apenas o processamento do português europeu, mas a maioria processa as duas variantes do português.

Além dos PoS *taggers* indicados no parágrafo acima, também estão disponíveis para o português:

- O **Porttagger** é um etiquetador morfossintático (PoS *tagger*) multigênero para o português brasileiro treinado com um conjunto de dados variados, composto por notícias jornalísticas (de vários domínios), textos acadêmicos (do domínio de óleo e gás) e de conteúdo gerado por usuário (tweets do domínio do mercado financeiro), considerados *gold standard*, pois foram revisados por humanos. As etiquetas utilizadas são as 17 etiquetas do modelo gramatical Universal Dependencies³⁰ (Marneffe

¹⁹<https://spacy.io/api/sentencizer>

²⁰<https://github.com/thalesbertaglia/enelvo> ou <https://thalesbertaglia.com/enelvo/sobre/>

²¹<https://nlp.lsi.upc.edu/freeling/index.php/>

²²<https://www.nltk.org>

²³<https://opennlp.apache.org/>

²⁴<https://github.com/NLP-CISUC/NLPyPort>

²⁵<https://draquet.github.io/PolyGlot/>

²⁶<https://spacy.io/>

²⁷<https://stanfordnlp.github.io/stanfordnlp/>

²⁸<https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

²⁹<https://github.com/citiususc/Linguakit>

³⁰<https://universaldependencies.org>



et al., 2021). O etiquetador realiza a classificação com base em um modelo de linguagem (o BERTimbau (Souza; Nogueira; Lotufo, 2020)), que foi implementado usando a biblioteca transformers³¹. O Porttagger possui uma acurácia do estado da arte, superando os resultados anteriores para a língua portuguesa. Para os textos jornalísticos, por exemplo, atinge valores acima de 99% de acurácia (Silva; Pardo; Roman, 2023). O sistema está disponível para uso do público em geral³².

- O **tagger do UDPipe**³³ consiste em vários modelos MorphoDiTa³⁴ que anotam PoS *tags* e/ou lemas. Por padrão, esse *tagger* costuma usar apenas um modelo, que gera todas as *tags* disponíveis (*coarse tags*, *fine-grained*, *features* morfológicas e lema). Porém, o desempenho melhora se um modelo apenas etiquetar as palavras (com UPoS, XPoS e *features* morfológicas), enquanto o outro modelo realiza a lematização. Também é possível especificar a quantidade de modelos que se quer utilizar. Vale ressaltar que o conjunto de etiquetas do UDPipe *tagger* é o mesmo das Universal Dependencies (Marneffe et al., 2021) e o formato de anotação dos dados é o mesmo do CONLL-U. Além do *tagger*, o UDPipe também dispõe de tokenizador, lematizador e *parser*, todos configuráveis para uso em modelos disponíveis, também sendo possível treinar novos modelos.
- O **tagger do Stanza**³⁵ também realiza anotação morfossintática e morfológica. No primeiro tipo, ele etiqueta as palavras em dois níveis: de granularidade grossa (UPoS) e de granularidade fina (XPoS). No segundo tipo, ele etiqueta as palavras a partir de *features* morfológicas (*UFeats*). O Stanza é um pacote Python para análise e processamento das línguas naturais que possui, além do *tagger*, várias outras ferramentas de PLN, como tokenizador, lematizador, *parser* de dependência, reconhecedor de entidades nomeadas e outras. O Stanza dispõe de modelos pré-treinados, inclusive com *treebanks* da UD.
- O **tagger do spaCy**³⁶ é um componente do *pipeline* que prediz as etiquetas de PoS a partir de qualquer *tagset*. Isso é muito útil porque permite que se use como dado de treinamento qualquer *corpus* anotado, sem precisar converter as etiquetas de um modelo para outro. A predição é feita a partir de um *pipeline* treinado com modelos estatísticos, que permitem ao spaCy fazer previsões sobre qual *tag* ou etiqueta provavelmente se aplica num dado contexto. Um componente treinado inclui dados binários que são produzidos mostrando a um sistema exemplos suficientes para que ele faça previsões generalizáveis para todo o idioma – por exemplo, uma palavra que vem depois de “o” em português provavelmente é um substantivo. Para a atribuição das *features* morfológicas, o spaCy dispõe de dois módulos: um estatístico, que atribui as *tags* de classe gramatical de granularidade grossa (UPoS); e um baseado em regras, principalmente para o caso de línguas com morfologia mais simples, como é o caso do inglês.

4.3.2 Recursos para o português

Nesta seção, indicaremos alguns recursos lexicais (principalmente *corpora*) anotados com informação morfológica e morfossintática disponíveis para o português. Não pretendemos

³¹<https://github.com/huggingface/transformers>

³²<https://huggingface.co/spaces/Emanuel/porttagger>

³³<https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-train.html>

³⁴<https://ufal.mff.cuni.cz/morphodita>

³⁵<https://stanfordnlp.github.io/stanza/pos.html>

³⁶<https://spacy.io/api/tagger>



aqui explorar todos os recursos disponíveis, mas apenas mencionar alguns exemplos, considerando que o Capítulo 14 aborda *corpus* e *datasets*. Ao final da seção, exploraremos de forma aprofundada apenas um deles, a fim de exemplificar e deixar claros todos os conceitos e tarefas apresentados neste capítulo.

Os *corpora* mais amplamente conhecidos na comunidade do português e que contenham anotação de PoS são o Tycho Brahe (Namiuti, 2004), o Mac-Morpho e o Bosque. O primeiro deles é um *corpus* do português histórico europeu, que compilou textos literários antigos (do século XIV ao XIX). Por serem de um gênero e domínio muito específicos, além de conter palavras e estruturas arcaicas, ele não é um *corpus* muito representativo do português contemporâneo. Os outros dois (Mac-Morpho e Bosque) serão apresentados na sequência desta seção. Além deles, indicaremos também outros *corpora* mais recentes do português que contenham anotação morfossintática. Alguns deles serão retomados no Capítulo 7.

- O **Mac-Morpho**³⁷ (Aluísio et al., 2003) é um *corpus* em português que contém 915.367 *tokens*³⁸ retirados de textos jornalísticos da Folha de São Paulo, já etiquetadas com suas respectivas classes de palavras. É considerado um dos maiores *corpora* do português contemporâneo e foi anotado manualmente com PoS *tags*. Seu *tagset* original conta com 41 etiquetas de PoS, sendo 19 delas somente para marcas de pontuação. Além de ser um *corpus* anotado, ele também foi usado para treinar várias versões de *taggers* usados pela comunidade do português (Fonseca; Rosa; Aluísio, 2015).
- O **Bosque** é uma parte de um *corpus* maior, o **Floresta Sintá(c)tica**³⁹ (Afonso et al., 2002), mas que contém anotação de PoS, de *features* morfológicas e também de *treebanks* (anotação sintática em árvore de dependência). O Bosque possui cerca de 200.000 *tokens* e é composto por textos jornalísticos das duas variantes do português, provenientes do CETENFolha (português brasileiro) e CETENPúblico (português europeu).
- O **Porttinari**⁴⁰ (Pardo et al., 2021) é um compilado de vários *corpora* de diferentes gêneros (jornalísticos, tweets de ações e bolsas de valores, reviews de produtos de e-commerce, resenhas de livros, entre outros). A versão atualmente disponível desse *corpus* é a porção jornalística (Duran et al., 2023) dividida em três *corpora*: (i) Porttinari-base, (ii) Porttinari-check e (iii) Porttinari-automatic. O **Porttinari-base** contém 8.418 sentenças anotadas manualmente com informação morfológica e morfossintática seguindo as diretrizes do Universal Dependencies (UD) (Marneffe et al., 2021). O **Porttinari-check** é um pequeno *corpus* de 1.685 sentenças com características similares ao Porttinari-base, porém anotado automaticamente usando como conjunto de treino o Porttinari-base para ilustrar o contraste com a anotação manual. O **Porttinari-automatic**, é um grande *corpus* com 3.954.189 sentenças que foi anotado automaticamente também usando o mesmo processo.
- O **MorphoBr**⁴¹ (Alencar; Cuconato; Rademaker, 2018) é um léxico construído com o propósito de subsidiar de forma abrangente a análise morfológica do português. Esse léxico combina recursos livres semelhantes, corrigindo milhares de erros e lacunas

³⁷<http://nilc.icmc.usp.br/macmorpho/>

³⁸Dados da última versão (v3), disponível em <http://nilc.icmc.usp.br/macmorpho/>.

³⁹<https://www.linguateca.pt/Floresta/>

⁴⁰<https://sites.google.com/icmc.usp.br/poetisa/resources-and-tools>

⁴¹<https://github.com/LR-POR/MorphoBr>



dos demais recursos, e aumentando de forma significativa o número de entradas, principalmente em relação aos substantivos e adjetivos no diminutivo.

Nos repositórios do NILC⁴², POeTiSA⁴³ e Linguateca⁴⁴ você pode ter acesso a vários outros recursos para o português, tais como *corpora*, léxicos e outros recursos lexicais com anotações morfológica e morfossintática. No Capítulo 14 também serão indicados *corpora* relevantes para o português, mas que não necessariamente contêm informação morfológica ou morfossintática. Para esclarecer de forma bem detalhada como funcionam esses recursos, a anotação de PoS e de *feats*, as etiquetas etc., selecionamos um deles, o PortiLexicon-UD, para ser apresentado em detalhes na próxima seção (Seção 4.3.3), a fim de exemplificar os conceitos abordados ao longo do capítulo.

4.3.3 PortiLexicon-UD: um recurso para processamento léxico em português

Para ilustrar o uso, na prática, de um recurso para processamento lexical, esta seção apresenta o PortiLexicon-UD (Lopes et al., 2022). O PortiLexicon-UD é um recurso que auxilia na tarefa de identificar as unidades de processamento. Ele é um léxico para o português que elenca palavras e suas anotações morfossintáticas. Especificamente, esse léxico utiliza o padrão Universal Dependencies (UD)⁴⁵ (Marneffe et al., 2021) com etiquetas PoS, lema e etiquetas de atributos morfológicos (gênero, número etc.). Na sua versão atual⁴⁶, PortiLexicon-UD possui 1.226.339 entradas.

O conjunto de etiquetas PoS da UD define **17 classes gramaticais** descritas no Quadro 4.1 com alguns exemplos de utilização no português segundo o mapeamento adotado. Todo mapeamento do português para o padrão UD foi feito durante a construção do *corpus* Porttinari-base (Duran et al., 2023). Dessa forma, o mapeamento reflete as decisões tomadas na anotação deste *corpus*.

Quadro 4.1. Etiquetas PoS da UD e sua descrição no português

PoS	Descrição e Exemplos
ADP	adposições , uma classe fechada que corresponde às preposições em português, como “de”, “para” e “com”.
ADJ	adjetivos , uma classe aberta que inclui palavras como “bonitas”, “último” e “vermelha”, mas que inclui também um subconjunto fechado, os números ordinais escritos por extenso como as palavras “primeiro”, “centésima” e “duodécimo”. No entanto, a anotação UD aceita também anotar como ADJ subconjuntos abertos como os números ordinais expressos em dígitos como “20 ^o ” ou “13 ^a ” ou ainda todas formas de verbos no participípio como “cansado” e “cedido”.

⁴²<https://sites.google.com/view/nilc-usp/resources-and-tools?authuser=0>

⁴³<https://sites.google.com/icmc.usp.br/poetisa/the-project>

⁴⁴<https://www.linguateca.pt>

⁴⁵Para um estudo mais aprofundado do projeto Universal Dependencies, ver Capítulo 6.

⁴⁶O PortiLexicon-UD está disponível no endereço: <https://portilexicon.icmc.usp.br/> e lá também pode ter seus arquivos de dados baixados.



ADV	advérbios , uma classe aberta com um subconjunto fechado, os advérbios primitivos (aqueles não formados com o sufixo “-mente”). Exemplos deste subconjunto fechado são as palavras “cedo”, “agora” e “acima”. Exemplos do subconjunto aberto são as palavras “normalmente” e “insanamente”. Também são incluídas formas abreviadas destes advérbios que aparecem em expressões como “social e economicamente” onde o advérbio “social” é usado como uma forma abreviada do advérbio “socialmente”.
AUX	verbos auxiliares e de cópula , uma classe fechada no português que engloba todas as conjugações dos verbos “ser”, “estar”, “haver”, “ir,”ter” e “vir”.
CCONJ	conjunções coordenativas , uma classe fechada que contém, por exemplo, “e”, “mas” e “portanto”.
DET	determinantes , uma classe fechada que inclui artigos como “um” e “o”, além de palavras específicas como “cujo”, mas que também tem uma grande sobreposição com pronomes como “aquele”, “diversos” e “meus”.
INTJ	interjeições , uma classe aberta que inclui, por exemplo, “tchau”, “oi” e “nossa”.
NOUN	substantivos , uma classe naturalmente aberta que inclui, por exemplo, “presidente”, “quartos”, “bandeirinha”, “salões” e “bola”.
NUM	números cardinais , uma classe aberta que possui um subconjunto fechado, os números cardinais escritos por extenso como “duas”, “trinta” e “quinhentos”, mas também um subconjunto aberto com toda sorte de números escritos com dígitos como “51”, “-3.1415” e até datas como “25/12/1974”.
PART	partículas , classe não mapeada na construção do Portinari-base.
PRON	pronomes , uma classe fechada que possui alguma sobreposição com determinantes como as palavras “sua”, “qual” e “ambos”, mas também palavras que não podem ser usadas como determinantes como “eu”, “que” e “aquilo”.
PROPN	nomes próprios , naturalmente uma classe aberta, tipicamente denominação de entidades nomeadas como “Obama” ou “Paris”.
PUNCT	todas as pontuações , uma classe aberta, sendo as mais usuais ponto final, exclamação, interrogação, vírgula, dois pontos e aspas.
SCONJ	conjunções subordinativas , uma classe fechada que contém, por exemplo, “conquanto”, “se” e “segundo”.
SYM	todos os símbolos simples e compostos , uma classe aberta que inclui, por exemplo, “R\$”, “US\$” e “%”.
VERB	verbos , uma classe aberta que inclui todas as conjugações dos verbos plenos em português, como, por exemplo, “canta”, “jogar”, “chorastes” e “teríamos”.
X	tudo que não pertence ao vocabulário da língua , naturalmente uma classe aberta, que inclui palavras estrangeiras como “détente” e “relaxation”, mas ainda onomatopeias como “oinc” e “ão”.

Devido à natureza dos léxicos, as palavras anotadas como pontuações (PUNCT), símbolos (SYM), além de nomes próprios (PROPN) e palavras fora do vocabulário (X) não estão presentes no PortiLexicon-UD. Levando em conta que a classe PART não é utilizada no mapeamento, o léxico possui palavras pertencentes a **12 classes da UD** que são mapeadas no português conforme sumarizado no Quadro 4.2.

Quadro 4.2. Mapeamento da classificação tradicional do português para as etiquetas PoS da UD utilizado no PortiLexicon-UD



Português	PoS UD	Palavras
Advérbios	ADV	Todos os advérbios
Preposições	ADP	Todas as preposições
Conjunções Coordenativas	CCONJ	Todas as conjunções coordenativas
Conjunções Subordinativas	SCONJ	Todas as conjunções subordinativas
Artigos	DET	Todos os artigos
Pronomes	DET	Todos os pronomes exceto pessoais
	PRON	Todos os pronomes
Substantivos	NOUN	Todos os substantivos
Adjetivos	ADJ	Todos os adjetivos
	ADV	Todos os adjetivos que abreviam um advérbio
Interjeições	INTJ	Todas as interjeições
Numerais	NUM	Todos os cardinais exceto formas altas (“bilhão”)
	ADJ	Todos os ordinais
	NOUN	Multiplicáveis (“triplo”) e formas altas (“milhão”)
Verbos	AUX	Conjugações de “ser”, “estar”, “haver,”ir”, “ter” e “vir”
	VERB	Todos os verbos, exceto conjugações de “estar”
	ADJ	Todos os participípios

O PortiLexicon-UD foi construído a partir de um léxico pré-existente, DELAF-PB (Ranchhod; Mota; Baptista, 1999), e suas versões UNITEX-PB (Muniz, 2004) e MorphoBr (Alencar; Cuconato; Rademaker, 2018). Notadamente, foi estendido o vocabulário e diversas situações foram corrigidas e adaptadas conforme descrito no lançamento do léxico (Lopes et al., 2022).

Todas as entradas no PortiLexicon-UD correspondem a uma **tupla**⁴⁷ com quatro informações:

1. Forma da palavra somente com letras minúsculas;
2. Lema da palavra somente com letras minúsculas;
3. Etiqueta PoS da UD;
4. Conjunto de pares, cada um com um atributos morfológico e seu valor associado ou, na inexistência de atributos morfológicos, utiliza-se o símbolo sublinhado ().

Enquanto as palavras das classes ADP, ADV, CCONJ, INTJ e SCONJ usualmente contêm pouca ou nenhuma informação morfológica, outras classes possuem grande variedade de informações. Alguns exemplos de palavras destas cinco classes estão indicadas no Quadro 4.3.

Quadro 4.3. Exemplos de palavras das classes ADP, ADV, CCONJ, INTJ e SCONJ

⁴⁷Tupla é um termo bem usado em computação, e significa uma cadeia de dois ou mais itens. Para um conjunto ordenado de elementos que são números, usamos o termo “vetor”. Mas, quando temos um conjunto ordenado no qual os elementos são outras coisas, como a cadeia de informações sobre uma dada palavra ou entrada de de um dicionário, dizemos “tupla”.



forma	lema	PoS	atributos morfológicos
de	de	ADP	—
pra	para	ADP	Abbr=Yes
cedo	cedo	ADV	—
felizmente	felizmente	ADV	—
mas	mas	CCONJ	—
portanto	portanto	CCONJ	—
nossa	nossa	INTJ	—
oi	oi	INTJ	—
se	se	SCONJ	—
conquanto	conquanto	SCONJ	—

Já as palavras das classes ADJ, NOUN e NUM possuem uma maior variação de atributos morfológicos como pode ser visto no Quadro 4.4. Usualmente palavras destas classes possuem os atributos *Gender* (gênero) e *Number* (número), e ocasionalmente os atributos *VerbForm* (forma verbal) e *NumType* (tipo numérico) indicando a origem das palavras.

Quadro 4.4. Exemplos de palavras das classes ADJ, NOUN e NUM

forma	lema	PoS	atributos morfológicos
bonitas	bonito	ADJ	Gender=Fem Number=Plur
primeira	primeiro	ADJ	Gender=Fem Number=Sing NumType=Ord
cedido	cedido	ADJ	Gender=Masc Number=Sing VerbForm=Part
presidente	presidente	NOUN	Number=Sing
quartos	quarto	NOUN	Gender=Masc Number=Plur
bandeirinha	bandeirinha	NOUN	Number=Sing
salões	salão	NOUN	Gender=Masc Number=Plur
duas	um	NUM	Gender=Fem NumType=Card
trinta	trinta	NUM	NumType=Card
quinhentos	quinhentos	NUM	Gender=Masc NumType=Card

As palavras etiquetadas como PRON e DET possuem uma variação ainda maior de atributos morfológicos, pois levam em consideração aspectos como o tipo de pronome e o caso. O Quadro 4.5 ilustra alguns destes casos.

Quadro 4.5. Exemplos de palavras das classes PRON e DET

forma	lema	PoS	atributos morfológicos
a	o	DET	Definite=Def Gender=Fem Number=Sing PronType=Art
um	um	DET	Definite=Ind Gender=Masc Number=Sing PronType=Art
cujo	cujo	DET	Gender=Masc Number=Sing PronType=Rel



esse	esse	DET	Gender=Masc Number=Sing PronType=Dem
minha	meu	DET	Gender=Fem Number=Sing Person=1 Poss=Yes PronType=Prs
a	o	PRON	Gender=Fem Number=Sing Person=3 PronType=Dem
eu	eu	PRON	Case=Nom Number=Sing Person=1 PronType=Prs
nossas	nosso	PRON	Gender=Fem Number=Plur Person=1 Poss=Yes PronType=Prs
quais	qual	PRON	Number=Plur PronType=Rel
que	que	PRON	PronType=Int

Finalmente, as palavras que pertencem às classes AUX e VERB também possuem uma variação de atributos morfológicos relevante, porém mais padronizada, já que os atributos descrevem as conjugações dos tempos verbais. O Quadro 4.6 apresenta exemplos dos tempos verbais em português.

Quadro 4.6. Exemplos de palavras das classes AUX e VERB

forma	lema	PoS	atributos morfológicos
sido	ser	AUX	Gender=Masc Number=Sing VerbForm=Part
tava	estar	AUX	Abbr=Yes Mood=Ind Number=Sing Person=3 Tense=Imp VerbForm=Fin
vinha	vir	AUX	Mood=Ind Number=Sing Person=3 Tense=Imp VerbForm=Fin
teríamos	ter	AUX	Mood=Cnd Number=Plur Person=1 VerbForm=Fin
cantar	cantar	VERB	VerbForm=Inf
cantarem	cantar	VERB	Number=Plur Person=3 VerbForm=Inf
cantando	cantar	VERB	VerbForm=Ger
cantada	cantar	VERB	Gender=Fem Number=Sing VerbForm=Part
canto	cantar	VERB	Mood=Ind Number=Sing Person=1 Tense=Pres VerbForm=Fin
cantavas	cantar	VERB	Mood=Ind Number=Sing Person=2 Tense=Imp VerbForm=Fin
cantou	cantar	VERB	Mood=Ind Number=Sing Person=3 Tense=Past VerbForm=Fin
cantáreis	cantar	VERB	Mood=Ind Number=Plur Person=2 Tense=Pqp VerbForm=Fin
cantarão	cantar	VERB	Mood=Ind Number=Plur Person=3 Tense=Fut VerbForm=Fin
cantaria	cantar	VERB	Mood=Cnd Number=Sing Person=1 VerbForm=Fin
cantemos	cantar	VERB	Mood=Sub Number=Plur Person=1 Tense=Pres VerbForm=Fin
cantasses	cantar	VERB	Mood=Sub Number=Sing Person=2 Tense=Past VerbForm=Fin
cantares	cantar	VERB	Mood=Sub Number=Sing Person=2 Tense=Fut VerbForm=Fin
canta	cantar	VERB	Mood=Imp Number=Sing Person=2 VerbForm=Fin

Descrito o conteúdo das entradas do PortiLexicon-UD, a Tabela 4.7 apresenta a



distribuição das entradas e palavras distintas por etiqueta PoS da UD. Nesta tabela também estão indicadas: (i) na coluna *palavras*, o total de palavras distintas por classe; (ii) na coluna *amb*, o número de palavras sintaticamente ambíguas, ou seja, palavras que possuem mais do que uma entrada; (iii) na coluna *non-amb*, o total de palavras não ambíguas e; finalmente, (iv) na coluna *entradas*, o número total de entradas.

Tabela 4.7: Número de palavras e entradas no PortiLexicon-UD por etiqueta PoS da UD

PoS UD	palavras	amb	non-amb	entradas
ADJ	124.332	79.165	45.167	124.449
ADP	38	26	12	38
ADV	4.938	1.838	3.100	4.944
AUX	332	277	55	435
CCONJ	25	17	8	25
DET	109	105	4	117
INTJ	41	12	29	41
NOUN	75.154	26.807	48.347	75.377
NUM	53	42	11	54
PRON	157	123	34	180
SCONJ	14	13	1	14
VERB	746.783	65.672	681.111	1.020.665
total	862.325	84.446	777.879	1.226.339

É possível perceber que, por serem classes abertas, os verbos, substantivos e adjetivos correspondem à maior parte do léxico. As palavras funcionais, como preposições, conjunções e pronomes, correspondem a uma parte bem menor do léxico, apesar de serem extremamente importantes na linguagem.

A maior parte das palavras do PortiLexicon-UD é de palavras não sintaticamente ambíguas e, portanto, possuem uma única entrada. No entanto, algumas palavras são particularmente ambíguas, como é o caso da palavra “que”, que possui o maior número de etiquetas PoS associadas a uma palavra do léxico (7), podendo ser utilizada como ADP, ADV, CCONJ, DET, INTJ, PRON e SCONJ. A palavra que possui o maior número de entradas no léxico é a palavra “fora”, que pode ser anotada com 5 etiquetas PoS distintas, mas possui as seguintes onze entradas no PortiLexicon-UD:

1. lema=fora, PoS=ADP, Atr. Morf.=_;
2. lema=fora, PoS=ADV, Atr. Morf.=_;
3. lema=ir, PoS=AUX,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=1|Tense=Pqp| VerbForm=Fin;
4. lema=ir, PoS=AUX,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=3|Tense=Pqp| VerbForm=Fin;
5. lema=ser, PoS=AUX,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=1|Tense=Pqp| VerbForm=Fin;
6. lema=ser, PoS=AUX,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=3|Tense=Pqp| VerbForm=Fin;



7. lema=fora, PoS=NOUN,
 - a. Atr. Morf.=Gender=Masc|Number=Sing;
8. lema=ir, PoS=VERB,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=1|Tense=Pqp| VerbForm=Fin;
9. lema=ir, PoS=VERB,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=3|Tense=Pqp| VerbForm=Fin;
10. lema=ser, PoS=VERB,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=1|Tense=Pqp| VerbForm=Fin;
11. lema=ser, PoS=VERB,
 - a. Atr. Morf.=Mood=Ind|Number=Sing|Person=3|Tense=Pqp| VerbForm=Fin.

É importante salientar que o PortiLexicon-UD, por seguir o princípio de anotação de *tokens* individualmente definido pela UD, não representa palavras compostas, ênclises, mesóclises ou palavras contraídas. Dessa forma, palavras como “segunda-feira”, “contá-lo”, “trazê-lo-ia” ou “desta” não fazem parte do léxico. Enquanto palavras compostas e nomes próprios podem ser incluídos em novas versões, as ênclises, mesóclises e palavras contraídas devem ser objeto de um pré-processamento que transforme esse tipo de palavra em seus componentes para buscá-las no léxico, por exemplo:

- “contá-lo” deve ser desdobrado em 3 *tokens*: “contar”, “-” e “lo”;
- “trazê-lo-ia” deve ser desdobrado em 3 *tokens*: “traria”, “-” e “lo”; e
- “desta” deve ser desdobrado em 2 *tokens*: “de” e “esta”.

Dessa forma, o PortiLexicon-UD oferece um recurso que permite a identificação de unidades léxicas de processamento de maneira eficiente e precisa. Por exemplo, na anotação de *corpus* (seja manual ou automática), o léxico pode fornecer um grande suporte à atribuição de etiquetas morfológicas, mas também para a análise de características morfológicas o léxico pode funcionar como uma referência de classificação para cada palavra do português.

4.4 Considerações finais

Este capítulo abordou o processamento automático do português no nível da **palavra**, que é considerado em PLN como a menor unidade de processamento. Para definir e delimitar essa unidade de processamento (a palavra), no entanto, é necessário considerar as pequenas unidades linguísticas que a constituem, que são os **morfemas**.

Portanto, neste capítulo, trouxemos uma visão geral da linguística e os principais conceitos da morfologia (morfema, afixo, desinência, radical etc.), mas também trouxemos os conceitos da **morfossintaxe** (lexema, lexia, léxico, *token*, *type* etc.), cujo objeto principal de estudo é a classificação das **palavras** em partes do discurso (ou PoS) (Seção 4.1).

As duas áreas, **Morfologia** e **Morfossintaxe**, se imbricam e se complementam, já que, para identificar quais são os traços morfológicos de uma palavra, é necessário saber qual o seu PoS. Por outro lado, muitas vezes, para definir o PoS de uma palavra, recorre-se aos seus traços morfológicos.



Para traçar um paralelo de forma bem simples entre as áreas e seus objetos de estudo, podemos dizer que, na linguística, a menor unidade significativa da língua é o morfema, que é estudado pela Morfologia. Em PLN, a menor unidade de processamento automático é a palavra, que é estudada pela Morfossintaxe. O Quadro 4.7 apresenta um resumo dessas associações.

Quadro 4.7. Resumo dos objetos de estudo das áreas Morfologia e Morfossintaxe

área	unidade de análise	tipos de etiquetas	unidade mínima de análise/ processamento
Morfologia	morfema	<i>feats</i> ou <i>features</i> ou atributos morfológicos	em linguística
Morfossintaxe	palavra	<i>part-of-speech</i> ou PoS (XPoS e UPoS) ou classes gramaticais ou classes de palavras	em PLN

Após definir todos esses conceitos (Seção 4.1) relevantes para a linguística e para o Processamento de Linguagem Natural, demonstramos como se faz o processamento morfológico em PLN, indicando as principais tarefas e etapas de processamento dos textos (Seção 4.2).

Por fim, indicamos algumas ferramentas e recursos disponíveis para o português (Seção 4.3), focando mais especificamente em um deles, a fim de exemplificar a complexidade desse nível de análise e processamento linguístico-computacional. Os recursos apresentados neste capítulo são apenas exemplificativos, mas é importante lembrar que existem vários outros que podem ser mais apropriados para uma aplicação ou outra, dependendo da necessidade e dos objetivos da tarefa.

Pretendemos, em uma próxima versão deste capítulo, apresentar atividades e exercícios práticos relacionados à etiquetagem morfológica e morfossintática. Também pretendemos explicar passo a passo como utilizar cada um dos recursos e ferramentas citadas aqui, a fim de contribuir, de forma mais prática e didática, com a internalização dos conceitos e o *modus operandi* de fazer PLN.

Também está prevista para a próxima versão uma apresentação mais global dos tipos de recursos lexicais, como *thesaurus* (e.g. o TeP⁴⁸), redes semânticas (e.g. WordNetBr⁴⁹, VerbNetBr⁵⁰, PropBankBr⁵¹), léxicos, dicionários comuns, dicionários especializados, ontologias, glossários e tantos outros recursos construídos para o português.

⁴⁸<http://www.nilc.icmc.usp.br/tep2/>

⁴⁹<http://www.nilc.icmc.usp.br/wordnetbr/>

⁵⁰<http://143.107.183.175:21380/portlex/index.php/en/projects/verbnbringl>

⁵¹<http://143.107.183.175:21380/portlex/index.php/en/projects/propbankbringl>



Capítulo 5

Expressões multipalavras

Renata Ramisch
Carlos Ramisch
Aline Villavicencio

5.1 *Cenas dos próximos capítulos...*

O capítulo sobre as expressões multipalavras, que sairá na próxima edição deste livro, vai abordar um *tema indigesto* no universo no Processamento de Linguagem Natural (PLN). É que essas expressões ficam no limite entre a sintaxe e a semântica, e sempre acabam ficando *no meio do fogo cruzado*. De um lado, elas apresentam idiossincrasias e especificidades que não permitem determinadas operações sintáticas comuns a outros conjuntos de palavras. De outro, as definições do sentido das expressões multipalavras (ou MWEs) em geral fogem à regra de que o significado do todo se dá pela soma das partes. Ainda bem, pois imagine se a expressão “engolir o sapo” fosse literal.

Assim, a nossa intenção para este capítulo é, em primeiro lugar, definir quais são os conceitos fundamentais para quem vai *navegar pelas águas turbulentas* do tratamento computacional de MWEs. Entre esses conceitos, começaremos com a discussão sobre os elementos que compõem uma MWE: seriam palavras? Seriam lexemas?

Na sequência, abordaremos os conceitos de MWEs propriamente ditos. A literatura traz inúmeras maneiras de analisar essas expressões, e as definições são tão variadas quanto os campos de estudo que se interessam por esse tema. Em PLN, escolhemos a definição utilizada pelo projeto PARSEME (o qual também é apresentado brevemente no capítulo que está por vir):

Expressões multipalavras são entendidas como sequências (contínuas ou descontínuas) de palavras que (i) contêm pelo menos duas palavras componentes que são lexicalizadas, ou seja, sempre realizadas pelos mesmos lexemas, incluindo uma palavra principal e pelo menos uma outra palavra sintaticamente relacionada, e (ii) exibem algum grau de idiossincrasia lexical, morfológica, sintática e/ou semântica.

Considerando essa definição inicial de MWE, vamos pormenorizar alguns dos seus aspectos importantes, como idiossincrasias, coesão sintática e elementos lexicalizados. Ao mesmo tempo, também buscamos *separar o joio do trigo*, uma vez que, para entender o que é uma MWE, é fundamental que se saiba também o que ela não é. Então, explicaremos por que, na nossa abordagem, os compostos, as colocações e as metáforas não são expressões multipalavras.

Embora esta seja uma tarefa em aberto, apresentaremos, no nosso futuro capítulo, uma tentativa de classificação de MWEs. O grupo de expressões cujo núcleo é um verbo



foi definido nas diretrizes de anotação do PARSEME, e nós basicamente utilizamos as definições desse projeto. No que se refere às demais categorias, consideramos uma série de aspectos práticos e teóricos na nossa proposta de categorização, levando em conta sobretudo a sua função em uma sentença. Traremos, ainda, algumas discussões sobre ambiguidade, variabilidade e arbitrariedade.

Após a etapa de conceituações, citaremos as principais tarefas de PLN que envolvem MWEs, assim como o estado da arte, os métodos e algoritmos que buscam resolver esse problema. Essas tarefas se dividem basicamente em dois grandes grupos: a) a descoberta/deteccção; e b) a identificação de MWEs.

Você verá no capítulo, ainda, uma breve apresentação dos recursos existentes para o português brasileiro, assim como o conhecimento necessário para entender ou resolver alguma tarefa em torno da questão das MWEs. Depois, trataremos dos principais métodos de avaliação, assim como métricas e testes comumente utilizados pela comunidade para avaliar o desempenho de métodos e algoritmos direcionados a tratar ou resolver as tarefas antes mencionadas.

Por fim, consideramos importante *olhar pelo retrovisor* e entender qual foi o percurso para chegar até onde estamos, no tema das MWEs, em termos científicos, na direção da resolução dessa complexa tarefa. Em que ponto estamos hoje? Qual é a posição do português brasileiro em termos de recursos e de pesquisas diante da comunidade internacional? Quais são os desafios que permanecem aos pesquisadores e entusiastas das MWEs para *descascar esse abacaxi* em tempos de modelos de língua tão grandes que *não cabem em si*?

Sem querer *prometer mundos e fundos*, o capítulo que está por vir tentará acrescentar mais um tijolo na construção do conhecimento linguístico em PLN para o português brasileiro. Como você pôde perceber por este resumo, o capítulo contém uma série de exemplos de expressões multpalavras, tanto para ilustrar fenômenos linguísticos quanto para divertir leitores e leitoras. Vale a pena esperar pelas *cenias do nosso próximo capítulo*!



Parte IV

Estrutura



Capítulo 6

A ordem e a função das palavras em uma sentença

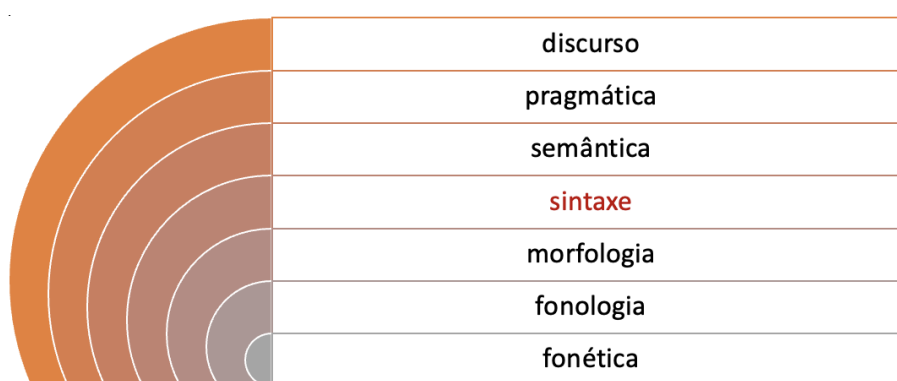
Adriana S Pagano
Amanda Rassi
Ana Clara S Pagano

6.1 Introdução

Nos capítulos anteriores, vimos algumas das unidades de análise que são examinadas nas distintas subáreas dos estudos linguísticos, cada uma das quais com desafios específicos para o PLN. Neste capítulo, nosso foco será uma subárea em particular – a sintaxe –, que estuda como as palavras se organizam nas estruturas que constroem as distintas funções gramaticais no escopo da frase ou sentença.

Retomando nossa representação do estudo da linguagem (Figura 1.2 do Capítulo 1), podemos dizer que a sintaxe é um estrato/camada central do sistema linguístico (Figura 6.1), pois organiza funções no estrato imediatamente inferior – a morfologia – e fornece estruturas de funções que serão importantes nos estratos superiores – semântica e pragmática – para, por exemplo, identificar papéis temáticos cruciais na tarefa de extração de informação, os quais respondem as perguntas “quem?” faz “o quê?”, “para quem?”, “como?”, “quando?”, “onde?” etc.

Figura 6.1: Representação das subáreas de estudo da linguagem com destaque para a sintaxe

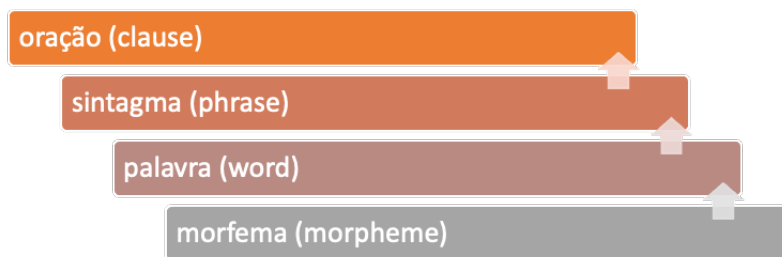


Ainda em relação ao sistema linguístico, além de sua organização em estratos, temos uma segunda forma de organização: a escala de ordens (em inglês, *rank scale*), que organiza



diferentes unidades de análise em níveis hierárquicos. A escala de ordens¹ está representada na Figura 6.2. Os morfemas são as menores unidades constitutivas das palavras, as quais, por sua vez, se organizam em estruturas, chamadas sintagmas, que nos permitem construir funções na oração, tais como sujeito, objetos, adjuntos e complementos.

Figura 6.2: Escala de ordens



A escala de ordens abrange as unidades utilizadas para a descrição da língua, seja esta língua falada ou escrita. A unidade superior de análise – a oração – consiste em um conjunto organizado de palavras que constroem significados sobre algum evento no mundo, que, como dissemos, podemos indagar perguntando “quem?” faz “o quê?”, “para quem?”, “como?”, “quando?”, “onde?” etc.

Na língua escrita, há uma unidade grafológica, geralmente denominada, em português, “sentença”, também conhecida como “frase” ou “período”². A “sentença” segue as convenções prototípicas da língua escrita, isto é, letra maiúscula inicial e sinal de pontuação que indica finalização, podendo ser este um ponto final, um sinal de interrogação ou de exclamação. A distinção entre “sentença” (em inglês, *sentence*) e “oração” (em inglês, *clause*) é muito importante em PLN, uma vez que em uma sentença (aquela que inicia com letra maiúscula e conclui com sinal de pontuação) pode haver mais de uma oração e, portanto, mais de um evento.

Neste capítulo, vamos abordar os níveis da escala de ordens e sua relevância para a análise sintática. Para compreendermos melhor como as palavras funcionam nos distintos tipos de sintagma e os sintagmas na oração, apresentaremos exemplos que esperamos que sejam esclarecedores. Após uma seção de reflexões iniciais, revisaremos conceitos básicos sobre análise sintática, passando por tipos de representação, até chegarmos nos dois tipos de análise sintática mais utilizados em PLN: a sintaxe de constituência e a sintaxe de dependência. Por último, abordaremos as intersecções entre a sintaxe e os demais estratos do sistema linguístico ao final deste capítulo, na Seção 6.8.

Antes de passarmos para nossas reflexões iniciais sobre sintaxe, cabe mencionar que o processo de analisar a estrutura de orações em PLN é denominado “*parsing*”, termo tomado por empréstimo do inglês. O *parsing* sintático toma como base a classe de palavra (em inglês, *part-of-speech*) das distintas palavras que compõem os sintagmas. Como veremos no Capítulo 7, em PLN a análise sintática automática é realizada por meio de softwares denominados *parsers*. Também veremos alguns desafios que a análise em constituintes

¹Cabe aqui um esclarecimento sobre a nomenclatura. Em língua inglesa, utilizamos “*phrase*” para nomear o nível da escala de ordens acima da palavra e abaixo da oração, enquanto “*syntagma*” nomeia a organização das funções numa unidade. Assim, toda “*phrase*” possui uma organização de funções ou “*syntagma*”. Na língua portuguesa, a palavra “sintagma” é utilizada para se referir ao que em inglês é denominado de “*phrase*” e de “*syntagma*”.

²Em estudos de sintaxe em português, “sentença” nomeia a maior unidade de análise sintática, a qual também pode ser nomeada como “frase” ou “período” (Kenedy; Othero, 2018).

sintáticos traz para os *parsers* já existentes, sobretudo em casos nos quais a delimitação de unidades e suas relações entre si na hierarquia da oração admitem mais de uma interpretação possível.

Um outro termo que é importante introduzir neste momento é o conceito de *treebank*, que é utilizado para se nomear um conjunto de sentenças com anotação morfossintática e com representação em diagramas de árvore. Nas seções seguintes, veremos alguns exemplos de diagramas de árvore juntamente com outras formas de representação. No Capítulo 7, veremos alguns dos *treebanks* disponíveis atualmente para PLN em português.

6.2 Reflexões Iniciais

Vamos começar examinando um texto, no Exemplo 6.1. Trata-se de um texto pequeno e simples, reproduzido aqui sem qualquer letra maiúscula e sem pontuação. Até o espaçamento entre palavras foi removido. À medida que for lendo, tente identificar sentido nele:

Exemplo 6.1.

ameninapostouumafoto

Embora a forma como o texto é apresentado acima não seja a mais esperada por nós num texto hoje em dia, ao menos nas línguas de origem europeia contemporâneas, essa forma já foi utilizada na antiguidade, no latim clássico, e é conhecida como *scriptio continua*. De fato, a separação visual entre palavras e sentenças, tal como escrevemos hoje, foi um desenvolvimento histórico que levou à forma como representamos hoje a língua na sua forma escrita. Contudo, apesar do estranhamento que possa causar o texto no Exemplo 6.1, muito provavelmente você não teve muita dificuldade em reinserir o espaçamento entre palavras, chegando a uma versão como a que segue no Exemplo 6.2:

Exemplo 6.2.

a menina postou uma foto

Pensando, ainda, numa versão impressa deste texto, a letra maiúscula inicial e a pontuação final certamente não geram grandes dificuldades e você deve ter chegado à versão no Exemplo 6.3:

Exemplo 6.3.

A menina postou uma foto.

O reconhecimento de cada uma das palavras, separadamente, e da sentença toda como uma unidade, todavia, não é suficiente para podermos explicar por que ou como o texto faz sentido para nós. Entre as palavras individuais e a sentença como um todo, há agrupamentos que funcionam como unidades intermediárias e nos quais a ordem das palavras é condicionada pela estrutura da língua. Assim, na sentença acima, identificamos palavras que agrupamos como pequenos blocos de informação, pois constroem significados e contribuem com o significado do texto como um todo. Esses agrupamentos são, por exemplo, “a menina” e “uma foto”.

Em cada um desses agrupamentos de palavras que naturalmente reconhecemos, há uma ordem em que as palavras se sucedem umas às outras. Assim, uma palavra como



“a” (artigo) ocorre antes de “menina” (substantivo), nunca depois. Quando a ordem das palavras dentro desses agrupamentos é trocada, temos uma forma que consideramos com problemas na sua formulação e que nos causa estranhamento. Por exemplo: “menina a” ou “foto uma”.

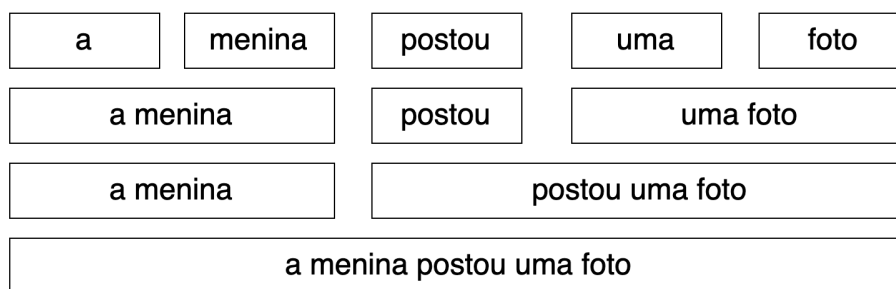
Com estas reflexões iniciais, passamos agora a revisar alguns conceitos básicos de sintaxe.

6.3 Noções Básicas de Sintaxe

O estudo de como as palavras se agrupam e se organizam em uma ordem determinada é o objeto do campo de estudos da sintaxe. É através da sintaxe que reconhecemos as regras que ditam quais agrupamentos de palavras serão aceitos por nós e quais serão considerados problemáticos.

Os agrupamentos operam como pequenos “tijolos” que vão construindo significados, ao serem agrupados em uma unidade maior até construir um significado que é relevante para a oração. Nessa analogia com tijolos que são agrupados, o Exemplo 6.3 pode ser representado como disposto na Figura 6.3:

Figura 6.3: Representação do agrupamento progressivo de unidades menores em maiores



Olhando para a Figura 6.3, temos, no topo, cada uma das palavras do nosso exemplo. À medida que vamos descendo na Figura 6.3, vemos agrupamentos entre as palavras, os quais, progressivamente, culminam na oração. Os agrupamentos mais próximos da oração são aqueles que constroem os significados mais relevantes para a compreensão de informações, como as que respondem às perguntas elencadas no Quadro 6.1:

Quadro 6.1. Perguntas utilizadas para elucidar informações sobre eventos

Pergunta	Resposta
Quem?	a menina
Fez o quê?	postou uma foto
Postou o quê?	uma foto

Em cada um dos agrupamentos de palavras que vamos progressivamente formando até chegar ao agrupamento maior, a oração, as palavras trabalham para realizar uma função em comum; elas são uma única unidade funcionando dentro da oração. Dito de outra forma, cada um desses agrupamentos existe pois exerce uma função dentro da oração. Por exemplo, “a menina” é uma unidade constituída por um artigo, também chamado determinante (“a”), e um substantivo (“menina”). Esses dois elementos operam conjuntamente para constituir



uma função e realizar um significado. Os agrupamentos acima do nível da palavra, como vimos na escala de ordens na Figura 6.2, recebem a denominação **sintagma** (em inglês, *phrase*). Cada um dos sintagmas é uma estrutura que é parte de uma unidade maior e que exerce uma função nela, sendo que esta outra estrutura pode ser um outro sintagma ou a oração.

Existem vários tipos de sintagma: sintagmas nominais, verbais, adverbiais, adjetivais e preposicionais, e cada um possui as suas próprias regras para a organização das palavras que o compõem, de acordo com a função que as palavras exercem dentro do próprio sintagma.

A cada uma das unidades que funciona dentro de um sintagma damos o nome de constituinte (em inglês, *constituent*). São constituintes as palavras individuais bem como seus agrupamentos progressivos em unidades maiores. Nossa sentença “A menina postou uma foto” está constituída por cinco palavras, as quais podem ser agrupadas em constituintes intermediários até dois grandes constituintes: [a menina] e [postou uma foto].

Alguns exemplos de constituintes no nível da palavra são: substantivos, verbos, adjetivos etc.

Retomando o Exemplo 6.3 temos, então, os seguintes sintagmas: “a menina”, que é um sintagma nominal, pois o seu constituinte principal é o substantivo “menina”; e “postou uma foto”, que é um sintagma verbal, pois seu constituinte principal é o verbo “postou”; dentro do sintagma verbal temos ainda um outro sintagma, “uma foto”, que é considerado um sintagma nominal, pois seu constituinte principal é o substantivo “foto”. É importante notar, portanto, que pode haver um ou mais sintagmas dentro de um outro sintagma. Esse constituinte principal que dita qual é o tipo de sintagma é chamado de núcleo, pois ele é o núcleo da estrutura gramatical, ou seja, do sintagma.

Para saber com que tipo de sintagma estamos lidando, precisamos, então, saber que fatores ditam qual é o núcleo desse sintagma. Isso irá depender da função que o sintagma exerce na oração. Pensemos no exemplo: o sintagma nominal “uma foto” exerce a função de objeto dentro do sintagma “postou uma foto”. Objetos são realizados por sintagmas nominais; assim, o seu núcleo é um substantivo.

É importante ressaltar, também, que esses constituintes não se restringem ao nível imediatamente inferior na oração, nem a nenhum outro nível: constituintes são os componentes que integram todas as estruturas da sentença, ou seja, todos os sintagmas, e podem ser compostos por outros constituintes. Daí a relevância da escala de ordens apresentada na Figura 6.2.

No Exemplo 6.3, os sintagmas “a menina”, “postou uma foto” e “uma foto” têm uma função no nível superior da escala de ordens, isto é, na oração. Na oração, essas funções podem ser associadas ao que chamamos, na Semântica, de papéis temáticos.

Papel temático é um conceito utilizado para nomear o tipo de relação que um verbo estabelece com seu sujeito e seus complementos, relação pela qual o verbo lhes atribui uma função semântica, como, por exemplo, Agente, Paciente ou Objeto de uma ação. Os papéis temáticos podem ser mapeados com as funções sintáticas na oração e podem ser indagados por meio de perguntas do tipo “quem?” faz “o quê?”, “para quem?”, “como?”, “quando?”, “onde?” etc., como as exemplificadas no Quadro 6.2. As respostas a cada pergunta nos ajudam a identificar um constituinte com uma função sintática específica na oração e um papel temático. No Exemplo 6.3, a parte da oração que responde a pergunta “quem?” correspondente ao sujeito da oração, que, neste caso, é identificado com o papel temático de Agente da ação realizada. Já a parte que responde a “o quê?” corresponde ao objeto do verbo e ao papel temático Objeto Estativo.



Quadro 6.2. Exemplo de funções sintáticas e papéis temáticos

Pergunta	Resposta	Função sintática	Papel temático
Quem?	a menina	sujeito	Agente
Fez o quê?	postou uma foto	predicado	—
Postou o quê?	uma foto	objeto	Objeto Estativo

Como vimos, então, um sintagma pode ser construído por uma ou mais palavras, sendo uma delas a principal, o núcleo. A estrutura dos constituintes é o que conhecemos como sintagma e o tipo de sintagma é definido pela classe de palavra do seu núcleo ou componente principal. No Quadro 6.3, temos os três constituintes da oração no Exemplo 6.3 e o tipo de sintagma de cada um deles.

Quadro 6.3. Exemplo de tipos de sintagma

Sintagma	Núcleo	Classe de palavra do núcleo	Tipo de sintagma
a menina	menina	substantivo	sintagma nominal
postou uma foto	postou	verbo	sintagma verbal
uma foto	foto	substantivo	sintagma nominal

Palavras e sintagmas podem, assim, ser constituintes, sempre funcionando numa estrutura maior, na escala de ordens, tendo como unidade maior a oração. A análise da estrutura das orações de acordo com seus constituintes e a hierarquia estabelecida entre eles é conhecida como **sintaxe de constituência**. A análise sintática de constituência é um dos dois tipos de análise sintática mais utilizados em PLN, sendo o outro tipo a denominada **sintaxe de dependência**. Neste capítulo, vamos apresentar os dois tipos de análise e discorrer brevemente sobre suas diferenças. Mas, antes, vamos examinar os principais tipos de representação em sintaxe.

6.4 Tipos de representação

Há algumas formas convencionais de se representar as estruturas sintáticas que podem ser utilizadas de acordo com o tipo de análise sintática. Dentre elas, destacamos as seguintes: colchetes, árvores, setas, parênteses e indentação.

6.4.1 Colchetes

Colchetes (em inglês, *brackets*) são uma forma de representação hierárquica por meio da qual indicamos quais palavras estão agrupadas dentro de uma mesma unidade e quais unidades são parte de unidades maiores. Por exemplo, a sentença “A menina postou uma foto.” pode ser representada, de forma progressiva, dos constituintes maiores aos menores, como ilustrado na Figura 6.4.

No topo, temos um único par de colchetes que abrange a sentença toda. Em seguida, separamos, entre colchetes, a oração e seu ponto final. Em seguida, separamos, entre colchetes, os constituintes maiores e, progressivamente, vamos separando, entre colchetes,



Figura 6.4: Representação de hierarquia de constituintes por meio de colchetes

[A menina postou uma foto.]
[A menina postou uma foto] | [.]
[A menina] | [postou uma foto] | [.]
[A] | [menina] | [postou] | [uma foto] | [.]
[A] | [menina] | [postou] | [uma] | [foto] | [.]

constituintes dentro de constituintes, até chegarmos às palavras individuais, que são nossos constituintes mínimos. Barras verticais auxiliam a visualização na Figura 6.4.

Na representação por meio de colchetes, cada par de colchetes indica um **nível de continência**. O par de colchetes mais externo contém a sentença como um todo. Já o par de colchetes mais interno de todos contém as palavras. As unidades menores encontram-se contidas nas maiores, sendo as unidades mínimas cada uma das palavras individualmente. A representação completa pode ser feita numa única linha:

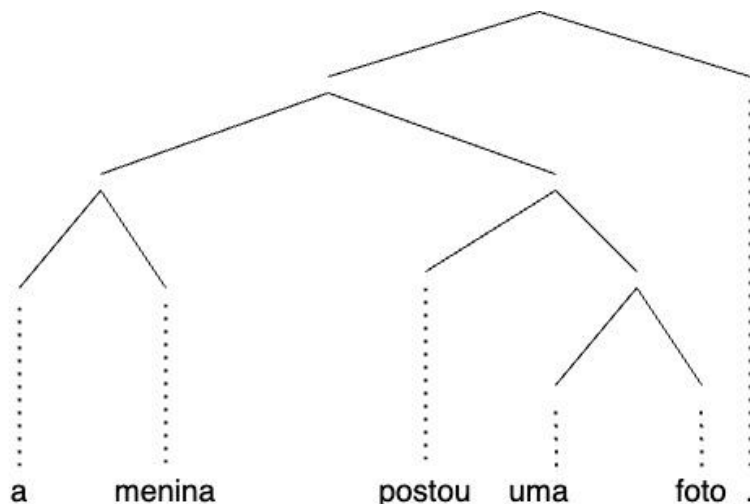
[[[[[A][menina]][[postou][[uma][foto]]]]][.]]]

Este tipo de representação, como veremos neste capítulo, pode ser feito tanto para sintaxe de constituintes como para sintaxe de dependência.

6.4.2 Árvores

Árvores (em inglês, *trees*) são uma forma de representação que fornece uma visualização mais clara da hierarquia dos constituintes. Assim, nosso exemplo anterior tem a seguinte representação arbórea na Figura 6.5:

Figura 6.5: Representação de hierarquia de constituintes por meio de árvore



Na notação com estruturas arbóreas, um diagrama chamado de árvore representa graficamente a estrutura de uma sentença como uma hierarquia, sendo os **nós** (em inglês, *nodes*), elementos discretos (ou elementos finais) em um grafo (ou representação simbólica)



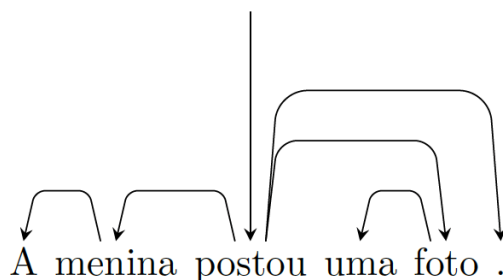
e as **arestas** (em inglês, *edges*) linhas que conectam dois nós e indicam uma relação entre eles.

Este tipo de representação, como veremos neste capítulo, pode ser feito tanto para sintaxe de constituintes como para sintaxe de dependência.

6.4.3 Setas

Setas (em inglês, *arced arrows*) são uma representação na qual setas unidirecionais são desenhadas de um **nó pai** (em inglês, *parent*) para um **nó filho** (em inglês, *child*), como pode ser visto na Figura 6.6. Nela, temos uma representação de relações de dependência entre nós pais e filhos. A seta parte da palavra que governa a relação e chega na palavra dependente. Essa relação será melhor explicada na Seção 6.6.

Figura 6.6: Representação de relações de dependência por meio de setas unidirecionais



A seta central, que chega no verbo “postou” é apenas uma representação conceitual de que esse é o elemento central da sentença. As setas que saem de “postou” e chegam em “menina”, “foto” e “.” indicam que, nessas relações, “postou” é o pai e “menina”, “foto” e “.” são filhos de “postou”. A seta que sai de “menina” e chega em “a” indica que “a” é filho de “menina”, assim como a seta de “foto” para “uma” indica que “foto” é pai de “uma”.

Este tipo de representação, como veremos mais adiante neste capítulo, é prototípico da análise sintática de dependência.

6.4.4 Parênteses

Parênteses (em inglês, *parentheses*) são usados para estabelecer a relação entre pares de palavras. Nesta representação, cada um dos pares de palavras entre as quais se estabelece uma relação unidirecional de dependência é apresentado entre parênteses, sendo a primeira posição a da palavra que governa a relação e a segunda a da palavra dependente.

Na linha 1, temos a relação conceitual de **root**, que é representada pelo elemento vazio, o qual governa o verbo “postou”, considerado o elemento central na sentença. Assim como na representação por setas, temos nas linhas 2, 4 e 6, as relações de dependência em que o verbo “postou” governa seus dependentes “menina”, “foto” e “.”, respectivamente. Na linha 3, vemos a dependência de “a” (dependente) em relação a “menina” (governante), assim como na linha 5, em que “uma” é dependente de “foto”.

Este tipo de representação é utilizado pela análise sintática de dependência.



Figura 6.7: Representação de relações de dependência por meio de parênteses

1	(, postou)
2	(postou, menina)
3	(menina, a)
4	(postou, foto)
5	(foto, uma)
6	(postou, .)

6.4.5 Indentação

Indentação (em inglês, *indentation*) também é usada para explicitar a hierarquia entre as palavras. Este tipo de representação nos permite visualizar a hierarquia nas relações de dependência.

Figura 6.8: Representação de relações de dependência por meio de indentação

1	postou
2	menina
3	a
4	foto
5	uma

Na Figura 6.8, os dependentes ou nós filhos, por exemplo “menina” e “foto”, são indentados em relação ao nó pai “postou”. Os nós “a” e “uma” são indentados em relação aos nós “menina” e “foto”, respectivamente.

Este tipo de representação é utilizado pela análise sintática de dependência.

Como veremos nas seções seguintes, cada formato de representação pode ser mais ou menos adequado para cada tipo de análise sintática. Neste capítulo, vamos explicar as duas abordagens mais comuns de análise sintática que fundamentam os dois tipos de *parsing* mais utilizados em PLN: o *parsing* de constituição e o *parsing* de dependência.

6.5 Sintaxe de constituição

Na sintaxe de constituição, também chamada de sintaxe de constituintes, unidades chamadas constituintes são agrupadas em unidades maiores. Os constituintes podem ser palavras ou distintos tipos de sintagmas, podendo haver sintagmas contidos dentro de sintagmas maiores. Esse fenômeno implica uma hierarquia de unidades, que é capturada e visualizada por meio de representações como as que vimos na seção anterior.

Na sintaxe de constituição, a estrutura hierárquica pode ser representada com o uso de colchetes, que, como vimos anteriormente, indicam quais constituintes inferiores estão contidos nos constituintes hierarquicamente superiores; ou com estruturas arbóreas, nas

quais no topo da árvore encontra-se a sentença completa e, nos níveis inferiores, os constituintes menores. Em ambos os casos, cada constituinte está acompanhado de sua respectiva notação, de acordo com um conjunto específico de etiquetas morfossintáticas (em inglês, *tagset*).

A análise sintática tem como ponto de partida a unidade maior – a sentença – representada, prototipicamente, pela notação **S** (do inglês, *sentence*). Se uma sentença tem pontuação final, o sinal de pontuação recebe a notação **PNT** (do inglês, *punctuation*) e é anotado no mesmo nível hierárquico de **S**. De **S**, derivam constituintes cujas estruturas realizam tipicamente as funções sintáticas no nível da oração: sujeito, geralmente realizada por um constituinte com estrutura de sintagma nominal, que recebe a notação **NP** (do inglês, *noun phrase*); e predicado, geralmente realizada por um constituinte com estrutura de sintagma verbal, com a notação **VP** (do inglês, *verb phrase*), podendo ter objetos com estrutura prototípica de sintagma nominal (**NP**). Há também frases preposicionais, anotadas como **PP** (do inglês, *prepositional phrase*). Nos níveis seguintes, cada constituinte é subdividido em subconstituintes até chegarmos às palavras individuais, as quais recebem a notação de sua classe: **V** para verbo, **N** para substantivo, **DET** para determinantes como artigos e pronomes demonstrativos, **ADJ** para adjetivo, **ADV** para advérbio, e **P** para preposição.

Vejam a representação com colchetes e notação para a sentença: “A menina postou uma foto.”

[**ROOT** [**S** [**S** [**NP** [**DET** **A**] [**N** **menina**]]] [**VP** [**V** **postou**] [**NP** [**DET** **uma**] [**N** **foto**]]]]] [**PNT** .]]]

Como pode ser observado, a cada palavra é atribuída uma classe de palavra de acordo com um conjunto de etiquetas definido. Nesta representação, o par de colchetes mais externos contém a sentença como um todo, incluindo a pontuação, e recebe a etiqueta **ROOT** (em português, raiz) que indica o ponto inicial da análise. Já os pares de colchetes mais internos de todos contêm as palavras individuais com sua respectiva notação para classe de palavra.

Na notação com estruturas arbóreas, a sentença: “A menina postou uma foto.” é representada na forma de árvore³ na Figura 6.9.

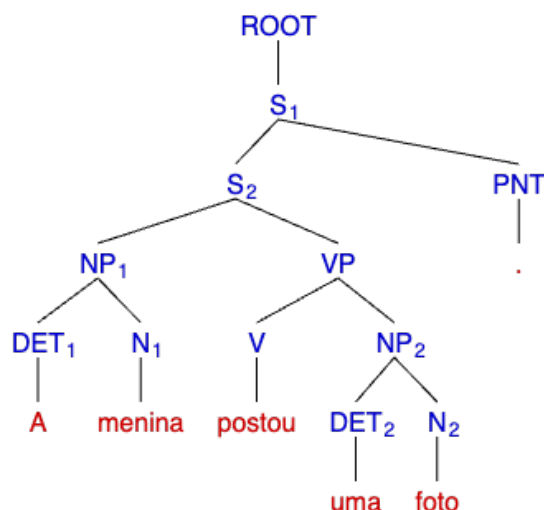
A árvore pode ser lida, de cima para baixo, da seguinte forma: O nó denominado **ROOT** indica a raiz da árvore. O nó **S1** representa a sentença toda como um único constituinte; é o nó que domina todos os outros nós da árvore. No caso de uma sentença grafológica (unidade delimitada por um sinal de pontuação), como é o caso na Figura 6.9, temos uma representação da sentença como um todo (**S1**), abrangendo a sentença (**S2**) e o seu sinal de pontuação (**PNT**). De **S2** derivam constituintes cujas estruturas realizam tipicamente as funções sintáticas no nível da oração: sujeito realizado por um constituinte com estrutura de sintagma nominal (**NP**) e predicado, por um constituinte com estrutura de sintagma verbal (**VP**). Nos níveis seguintes, cada constituinte é subdividido em subconstituintes até chegarmos às palavras individuais.

Na árvore de constituência, as arestas representam segmentos que conectam os nós filhos a um nó pai. Assim, o nó **NP** (“a menina”) possui uma aresta que conecta **DET** (“a”) e **N** (“menina”). As palavras individuais são as folhas da árvore (em inglês, *leaves*) e são o nível inferior na hierarquia da árvore.

³As árvores de constituência utilizadas neste capítulo foram elaboradas com o software *jssyntaxfree*, disponível em <https://github.com/int2str/jssyntaxtree>, com base em anotação das autoras. O conjunto de etiquetas utilizado é o adotado pelo projeto *PortulanClarin*, disponível em <https://portulanclarin.net/>.



Figura 6.9: Representação de sintaxe de constituição em diagrama de árvore



Identificar os constituintes e seu lugar na hierarquia da oração é uma operação fundamental para a compreensão das funções sintáticas, que, como vimos, são a base para as funções semânticas da oração. Uma vantagem da notação com estrutura arbórea é a maior facilidade com a qual podemos visualizar as relações hierárquicas dos constituintes. Contudo, a notação com colchetes é computacionalmente mais fácil de ser processada.

6.5.1 Possibilidades de interpretação e ambiguidades sintáticas

Vejam agora um exemplo de como a análise de constituintes requer uma análise dos papéis temáticos.

Exemplo 6.4.

A menina postou uma foto com o celular.

No Exemplo 6.4, “com o celular” é um sintagma preposicional (**PP**), formado pela preposição “com” e o sintagma nominal “o celular”. Uma característica dos sintagmas preposicionais é que eles podem estar contidos num sintagma nominal ou num sintagma verbal. No Exemplo 6.4, “com o celular” pode estar contido no sintagma nominal “uma foto” ou no sintagma verbal “postou uma foto”. A decisão por considerar uma ou outra análise depende da interpretação do analista, tendo como apoio informações do contexto, ou seja, as informações que podemos depreender de outras partes do texto ou ao ter acesso a imagens, como, por exemplo, na Figura 6.10⁴.

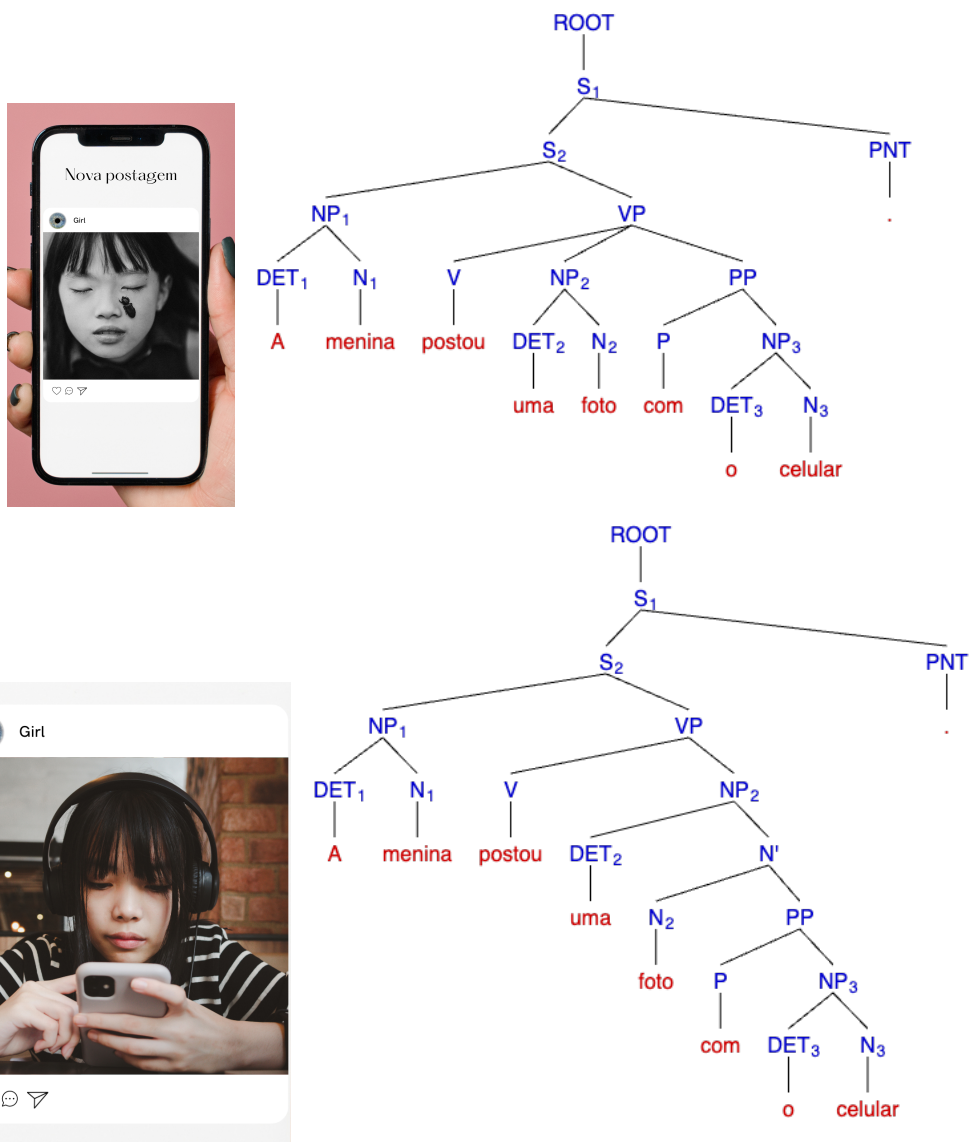
Na Figura 6.10, a imagem de cima pode ser interpretada como uma foto que foi postada com o celular, sendo o celular o instrumento utilizado para a postagem. Nesse caso, o sintagma preposicional “com o celular” estará ligado ao sintagma verbal “postou”, como mostrado na representação da árvore.

Já a imagem de baixo, na Figura 6.10, pode ser interpretada como a menina tendo postado uma foto, na qual ela está segurando o celular. Nesse caso, o sintagma preposicional “com

⁴Imagens geradas com o aplicação <https://gencraft.com/> e modificadas pelas autoras.



Figura 6.10: Duas imagens passíveis de serem rotuladas como “a menina postou uma foto com o celular”



o celular” estará contido no sintagma nominal “uma foto com o celular”, como mostrado na representação da árvore.

Há alguns testes que podem ser realizados para observar se um constituinte, por exemplo, um sintagma preposicional, deve ser interpretado como vinculado a um sintagma verbal ou contido em um sintagma nominal. Se o sintagma preposicional pode ser deslocado de posição numa oração e esse deslocamento é condizente com o significado que interpretamos pelas dicas do contexto, ele estará conectado ao sintagma verbal. Assim, em “A menina postou uma foto com o celular”, podemos mudar a posição do sintagma preposicional “com o celular” e obtermos duas sentenças nas quais “com o celular” se conecta ao sintagma verbal “postou”:

Exemplo 6.5.

Com o celular, a menina postou uma foto.

Uma outra possibilidade é fazer uma paráfrase da oração, como no Exemplo 6.6. Por outro lado, se o sintagma preposicional “com o celular” for interpretado como sendo contido no sintagma nominal “uma foto com o celular”, a paráfrase poderia ser a apresentada no Exemplo 6.7.

Exemplo 6.6.

A menina postou uma foto por meio do celular.

Exemplo 6.7.

A menina postou uma foto na qual ela está segurando o celular.

Se considerarmos as duas interpretações do Exemplo 6.4 sob a perspectiva dos papéis temáticos, podemos afirmar que o sintagma preposicional “com o celular” terá o papel temático de Instrumento quando estiver vinculado ao sintagma verbal “postou”. Se o sintagma preposicional estiver contido dentro do sintagma nominal “uma foto com o celular”, o sintagma como um todo terá o papel temático de Objeto Estativo.

A decisão sobre como segmentar a oração em constituintes exige do analista humano a interpretação de significados que são construídos pelas distintas funções sintáticas, associadas aos papéis temáticos e tendo como apoio o contexto, isto é, informações situacionais (por exemplo, imagens que acompanham a linguagem verbal) ou de sentenças anteriores ou posteriores no próprio texto do qual uma sentença faz parte.

Esta situação é um exemplo representativo de um dos motivos pelos quais o conhecimento linguístico e de uso da língua ainda não têm sido completamente representado/capturado por nenhum dos métodos atuais. A interpretação de significados linguísticos que dependem de informações contextuais é um dos maiores desafios para o PLN atualmente.

Além da análise sintática de constituintes da oração, há uma segunda abordagem, conhecida como sintaxe de dependência. Ela será o objeto da seção seguinte.

6.6 Sintaxe de dependência

O segundo tipo de análise sintática é chamado de sintaxe de dependência e tem suas bases na gramática de dependência de Tesnière (1959).



Diferentemente da abordagem de constituintes, explicada na seção anterior, em que a estrutura de uma sentença é definida por meio de sintagmas contidos em outros sintagmas, a análise de dependência descreve as relações de **dependência** entre palavras. Nessa abordagem, uma palavra é vista como subordinada a outra ou regida por ela, de acordo com relações sintáticas tais como sujeito-verbo; sujeito-objeto; verbo-objeto; coordenação; subordinação etc.

Na sintaxe de dependência, cada palavra é um nó de uma relação com uma outra palavra. Essas relações entre palavras são estabelecidas de forma unidirecional entre uma palavra regente (*head*), que é o nó de onde a relação parte, e uma palavra regida ou dependente, que é o nó ou palavra aonde a relação chega. Essa unidirecionalidade da relação é importante para se estabelecer a hierarquia entre as palavras, pois determina quem é o regente (de onde a relação parte) e quem é o regido (aonde a relação chega), o que será explicado na Seção 6.6.1. Por fim, é importante destacar que uma palavra pode reger várias outras, mas só pode ser regida por uma única.

Se quisermos conectar dois sintagmas, precisamos, diferentemente da análise de constituintes, conectar a palavra que representa o núcleo de um sintagma à palavra que representa o núcleo de outro sintagma. Além disso, também é necessário estabelecer as microrrelações dentro de um mesmo sintagma.

Na sintaxe de dependência, há basicamente dois tipos de relações:

- (i) macrorrelações
- (ii) microrrelações.

Como o próprio nome sugere, as **macrorrelações** estabelecem relações entre os núcleos de diferentes sintagmas e geralmente conectam palavras de classe aberta. Por exemplo, a relação que liga um verbo (núcleo do sintagma verbal) ao seu sujeito (núcleo do sintagma nominal) é chamada de macrorrelação. Já as **microrrelações** conectam elementos mais próximos, podendo ser adjacentes ou estar em uma vizinhança próxima. As microrrelações de dependência geralmente conectam uma palavra de classe aberta a uma palavra de classe fechada, como é o caso de um substantivo (palavra de classe aberta) e seu artigo (palavra de classe fechada).

6.6.1 Núcleo e dependente

Para estabelecer a relação de hierarquia entre duas palavras na sintaxe de Dependência, os termos mais usados em inglês são *head* e *dependent*. Em português, convencionou-se traduzir *dependent* por “dependente”, mas, com relação a “*head*”, os trabalhos de PLN usam nomenclaturas diferentes, tais como “cabeça”, “núcleo”, “dominante” ou o próprio termo em inglês *head*⁵. Também é comum em PLN chamar o núcleo de “pai” e o dependente de “filho”, já que essa nomenclatura permite extrapolar as relações de parentesco e chamar também os nós de “avô” (quando se refere ao núcleo do núcleo), “neto” (quando se refere ao dependente do dependente) e de “irmão” (quando os dependentes possuem o mesmo núcleo)⁶.

Em uma relação de dependência, o **núcleo** é o que rege, que comanda o seu **dependente**. Por exemplo, em um sintagma nominal como “a menina”, o substantivo é o núcleo enquanto o artigo é o dependente, pois é o substantivo que rege o artigo, impondo-lhe os traços morfológicos de gênero (neste caso, feminino) e de número (neste caso, singular).

⁵Entendemos que todos esses termos são sinônimos, mas adotaremos neste capítulo o termo “núcleo”.

⁶Vale ressaltar que estes termos também são usados em inglês: “*parent*”, “*child*”, “*grandparent*”, “*grandchild*” e “*sibling*”.



Definir quem é o núcleo e quem é o dependente nem sempre é uma tarefa fácil. Apesar de existirem convenções de diferentes teorias que definem essa hierarquia, nem sempre essas convenções são consensuais e, às vezes, essas definições são estabelecidas de forma arbitrária ou ad hoc. Por exemplo, preposições são consideradas núcleo em algumas abordagens de dependências (Osborne; Gerdes, 2019) e dependentes em outras (Universal Dependencies).

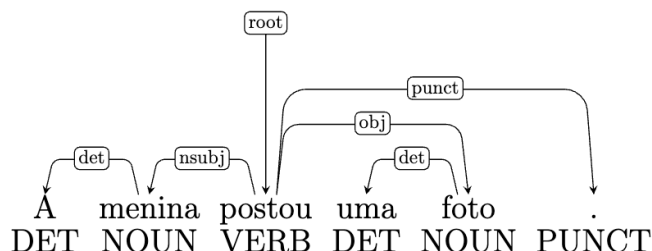
Por fim, vale ressaltar que as palavras podem ser núcleos ou dependentes, a depender da relação que elas estabelecem umas com as outras. Na frase “A menina postou uma foto.”, por exemplo, o substantivo “menina”, será núcleo na relação com seu determinante “a”, mas ao mesmo tempo será dependente na relação com o verbo “postou”. Algumas palavras, principalmente as de classe fechada, são consideradas sempre como dependentes, enquanto outras podem ser ora núcleo, ora dependente, considerando-se a outra palavra com a qual se relacionam.

6.6.2 A representação da sintaxe de dependência

Assim como na sintaxe de constituição, as relações de dependência podem ser representadas de formas distintas, sendo que todas explicitam a hierarquia e a direcionalidade da relação entre duas palavras em uma sentença. Das representações vistas neste capítulo, as duas mais utilizadas são os diagramas com setas e com representação parentética.

Como já foi dito, na representação com seta, as palavras são ligadas umas às outras por setas unidirecionais, ou seja, em um único sentido. Cada seta tem um ponto de partida e um ponto de chegada, o que significa que as relações não são recíprocas e nem uma mera concatenação entre palavras; pelo contrário, essas setas explicitam a dependência de uma palavra em relação à outra. Vale esclarecer que a representação com setas, assim como a representação de dependência com diagrama arbóreo, são ambas chamadas de “árvore de dependência”. A Figura 6.11 mostra um exemplo de representação das relações de dependência com setas.

Figura 6.11: Exemplo de árvore de dependência para “A menina postou uma foto”



Como vemos na Figura 6.11, a cada palavra é atribuída uma etiqueta morfosintática, indicada em caixa alta na parte de baixo da Figura. A etiqueta que aparece acima dos arcos das setas representa o nome da relação que se estabelece entre a palavra núcleo e a palavra dependente. Essa etiqueta é escrita em letras minúsculas. As etiquetas são selecionadas dentro de um conjunto fechado que varia segundo o padrão de anotação. Na Figura 6.11, o conjunto de etiquetas é o proposto pelo projeto Universal Dependencies, sobre o qual falaremos na Seção 6.6.3.

Como vemos na Figura 6.11, toda sentença possui uma única raiz (ou **root**), que é a palavra que não depende de nenhuma outra. Em sentenças que possuem verbos, a raiz geralmente é o próprio verbo.



No exemplo da Figura 6.11, verificamos três macrorrelações entre: i) o verbo (“postou”) e o núcleo do sujeito (“menina”); ii) o verbo (“postou”) e o núcleo do objeto direto (“foto”); e iii) o verbo e o ponto final. Em todas elas, o verbo é o núcleo e as demais palavras são seus dependentes, conforme aponta a direção das setas.

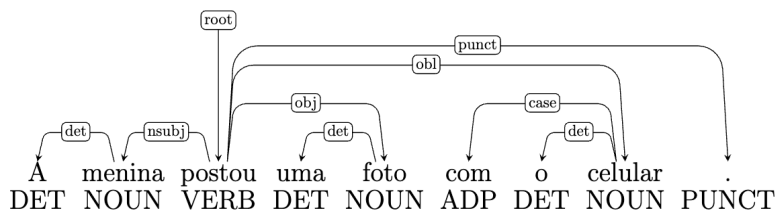
Há também duas microrrelações entre: i) o substantivo (“menina”) e o artigo (“a”); e ii) o substantivo (“foto”) e o artigo (“uma”).

Em uma árvore de dependência, algumas classes de palavras podem ser núcleo; outras, não. Mas cada palavra é dependente de uma outra. Isso significa que, na representação por diagrama com setas, em toda palavra deve chegar alguma (e exclusivamente uma) seta. A única exceção é para a raiz, na qual também chega uma seta, mas é a seta da relação **root**, o que significa que esta é a única palavra que não tem núcleo/pai. Todas as demais palavras, além da raiz, possuem um núcleo, de onde a seta parte.

Conforme sinalizado na Seção 6.6.1, o ponto de partida da seta representa o núcleo, e o ponto de chegada representa o dependente. Algumas relações de dependência permitem qualquer direção de seta, ou seja, pode ir de uma palavra à esquerda para uma palavra à direita ou vice-versa (e.g. relações de **nsubj**, **punct**, **obj** etc.). Outras relações possuem direção obrigatória, como é o caso de **det** ou de **case**, que é sempre da direita para a esquerda em línguas românicas como o português, uma vez que artigos e preposições são antepostos aos substantivos.

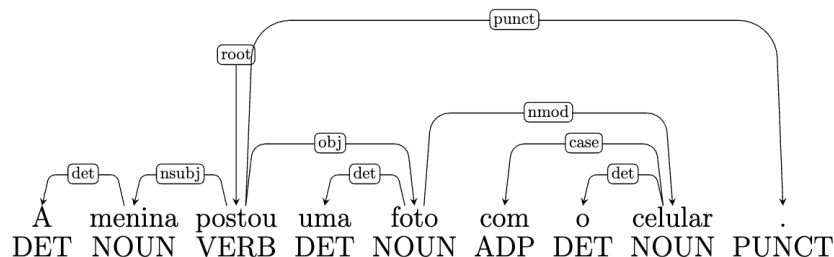
Na Figura 6.12, temos a representação da sentença “A menina postou uma foto com o celular”, aqui anotada em sintaxe de dependência. Nela, “com o celular” é interpretado como Instrumento e como uma função dependente do verbo “postou”.

Figura 6.12: Exemplo de uma árvore de dependência para “A menina postou uma foto com o celular”



Já na Figura 6.13, “com o celular” é interpretado como contido no sintagma nominal e com uma função dependente do substantivo “foto”.

Figura 6.13: Exemplo de uma outra árvore de dependência para “A menina postou uma foto com o celular”



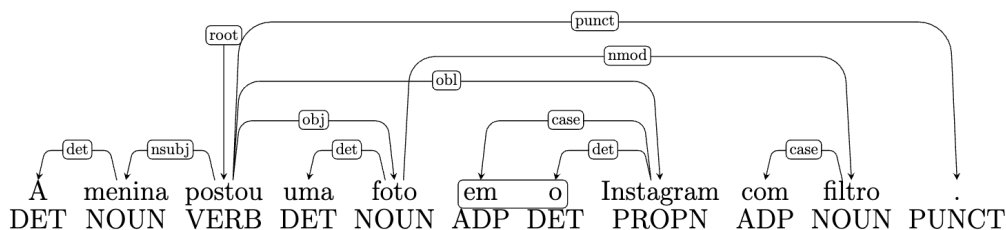
Na sintaxe de dependência, existe uma orientação geral para se evitar cruzamento dos arcos das setas. Na maior parte das vezes, isso é possível e recomendado. Porém, há casos

pontuais em que o cruzamento de arcos é necessário para estabelecer a relação correta entre duas palavras, como ocorre no Exemplo 6.8 da Figura 6.14.

Exemplo 6.8.

A menina postou uma foto no Instagram com filtro.

Figura 6.14: Exemplo de árvore de dependência com cruzamento de arcos



Neste exemplo, foi necessário cruzar os arcos de duas relações porque existe uma relação entre “postou no Instagram” e outra relação entre “foto com filtro”. Para não haver cruzamento de arcos, a sentença teria de ter a seguinte ordem: “A menina postou uma foto com filtro no Instagram.” ou ainda “A menina postou no Instagram uma foto com filtro.”

O segundo tipo de representação mais comum em sintaxe de dependência é aquela feita por meio de parênteses. Na representação parentética, coloca-se o nome da relação de dependência, seguida de parênteses; dentro dos parênteses, coloca-se o núcleo, seguido do dependente e separados por uma vírgula.

Na Figura 6.15, temos a representação parentética da sentença “A menina postou uma foto”.

Figura 6.15: Representação parentética da sentença “A menina postou uma foto”.

1	root(ROOT, postou)
2	nsubj(postou, menina)
3	det(menina, a)
4	obj(postou, foto)
5	det(foto, uma)
6	punct(postou, .)

A representação na Figura 6.15 também mostra a hierarquia entre as duas palavras e a direcionalidade da relação, por meio da ordem das palavras dentro dos parênteses.

A raiz, assim como acontece na representação por diagrama arbóreo, também é representada em uma das linhas, como se fosse “dependente” de um núcleo fictício, chamado **ROOT**, representado por letras maiúsculas, em uma relação chamada **root** representada por letras minúsculas. O elemento **ROOT** representa o elemento vazio.



A vantagem da representação parentética é que ela é mais fácil de ser implementada, já que está no formato de texto e não depende de softwares de anotação. Por outro lado, a desvantagem é que a visualização da estrutura sintática da sentença, por um humano, é mais difícil do que uma representação por árvore. Na Figura 6.16 temos a representação parentética da sentença “A menina postou uma foto no Instagram com filtro”, na qual há cruzamento de arcos. O cruzamento das relações, claramente visível na Figura 6.14, é mais difícil de ser identificado na representação parentética.

Figura 6.16: Representação parentética de sentença com cruzamento de arcos.

1	root(ROOT, postou)	# O elemento ROOT representa o elemento vazio
2	nsubj(postou, menina)	
3	det(menina, a)	
4	obj(postou, foto)	
5	det(foto, uma)	
6	obl(postou, Instagram)	
7	case(Instagram, no)	
8	obl(foto, filtro)	
9	case(filtro, com)	
10	punct(postou, .)	

Destacamos, também, que, na representação parentética, a ordem das linhas não precisa seguir a ordem das palavras, podendo-se começar na primeira linha com a relação de **root** ou de **punct** ou qualquer outra macro ou microrrelação, ou colocar qualquer uma delas na sequência. O importante é que todas as relações macro e micro estejam explícitas. Nesse sentido, verificar se a anotação de todas as relações está completa é mais difícil de ser feito na representação parentética, ao passo que, na representação por diagrama arbóreo, é mais fácil visualizar caso haja palavras nas quais não esteja chegando nenhuma seta.

A sintaxe de dependência foi ganhando espaço no PLN nas últimas duas décadas e é hoje o tipo de *parsing* sintático mais utilizado, sobretudo em tarefas de extração de informação (Capítulo 17). A seguir apresentamos um dos projetos de anotação multilíngue de sintaxe de dependência com reconhecido impacto nacional e internacional. Trata-se do projeto Universal Dependencies, cuja proposta visa maior consistência na anotação de *corpora* nas distintas línguas com base num arcabouço comum que possibilite a comparabilidade entre línguas. Apresentamos, também, um breve histórico de iniciativas de anotação que antecederam a proposta das Universal Dependencies.

6.6.3 Projetos de anotação multilíngue: Universal Dependencies

Na primeira década de 2000, surgiram várias iniciativas para tentar representar as relações sintáticas entre as palavras de uma frase por meio de dependências. Uma delas é a chamada **Stanford Dependencies** (SD), que passou a fazer parte de um dos *parsers* mais utilizados,



o *parser* Stanford, primeiramente para o inglês e posteriormente, para várias outras línguas⁷. Como essa iniciativa despertou interesse na comunidade linguística e de PLN, em 2006 e 2007 foram propostas *shared tasks* no CoNLL-X, nas quais os participantes treinaram e testaram sistemas nos mesmos conjuntos de dados usando anotação de dependências. Em 2007 especificamente, os participantes usaram uma abordagem multilíngue, baseada em *treebanks* de 10 línguas, e adaptada a um domínio. Nivre et al. (2007) descrevem essa tarefa, as diferentes abordagens usadas pelos participantes e os resultados dos experimentos.

Seguindo essa ideia de universalizar as representações de dependência para várias línguas, **Google** também criou seu próprio conjunto de etiquetas, baseando-se na análise de erros que foi feita para aquela *shared task* do CoNLL-X.

Outra iniciativa relevante foi o **Intersect interlíngua** (Zeman, 2008), que criou uma ferramenta para a conversão de etiquetas morfossintáticas entre línguas. Essa abordagem parte do princípio de que algumas línguas são semelhantes entre si, mas nem todas elas possuem recursos suficientes para PLN. A iniciativa propõe converter a anotação morfossintática de uma língua com mais recursos para outra língua que possua menos recursos (Zeman; Resnik, 2008).

Todas essas iniciativas contribuíram para o que foi posteriormente chamado de projeto **Universal Dependency Treebank** (UDT) (McDonald et al., 2013), cuja ideia principal é universalizar as anotações das relações sintáticas e os conjuntos de etiquetas, a fim de propiciar a comparabilidade entre as línguas. Em 2013, foi lançada uma primeira versão envolvendo 6 línguas e depois, em 2014, outra versão envolvendo 11 línguas. A língua portuguesa, com *corpora* de português europeu e brasileiro, está representada nos *treebanks* com anotação em UD disponíveis (Rademaker et al., 2017). Atualmente, os *treebanks* anotados de acordo com o padrão UD abrangem mais de 100 línguas de várias famílias e troncos linguísticos diferentes.

Assim, o projeto das Dependências Universais (em inglês, *Universal Dependencies*), mais conhecido como as UD, é o resultado da combinação de todas essas iniciativas em uma abordagem única e articulada, baseada nas dependências universais de Stanford, uma versão ampliada do conjunto de etiquetas universais propostas pelo Google, um subconjunto revisado do conjunto de *features* do Intersect, e uma versão revisada do formato CoNLL-X (chamado CoNLL-U)⁸. O projeto é a iniciativa mais usada e difundida hoje, motivo que justifica sua descrição em detalhes neste capítulo.

UD é um projeto que visa à anotação sintática sistemática e consistente em diversas línguas, de forma que essas línguas possam ser comparadas em relação à estrutura de suas sentenças. Apesar de buscar a maior padronização possível das anotações para garantir a comparabilidade entre as línguas, essa representação também prevê anotações particulares para línguas específicas, quando necessário. Justamente pelo fato de que algumas línguas possuem especificidades que não podem ser generalizadas para outras, o projeto UD precisa garantir que, apesar disso, as anotações sejam consistentes para que as línguas possam ser comparadas. Portanto, o projeto deve seguir seis premissas básicas⁹ que reproduzimos, em nossa tradução, a seguir:

⁷O *parser* Stanford também dispõe de um conversor que transforma uma árvore de constituição em uma árvore de dependência, o qual funciona para várias línguas, incluindo o português. Mais informações sobre SD e Stanford parser podem ser consultadas em <https://nlp.stanford.edu/software/lex-parser.shtml>.

⁸Tradução nossa do original: “*The new Universal Dependencies is the result of merging all these initiatives into a single coherent framework, based on universal Stanford dependencies, an extended version of the Google universal tagset, a revised subset of the Intersect feature inventory, and a revised version of the CoNLL-X format (called CoNLL-U)*”.

⁹As premissas podem ser consultadas em <https://universaldependencies.org/introduction.html>.



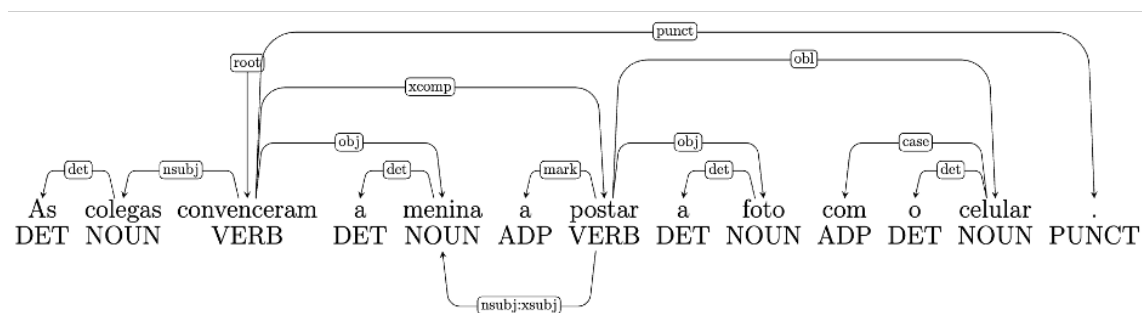
1. A proposta das UD precisa possibilitar uma análise linguística satisfatória para todas as línguas.
2. A proposta das UD precisa ser apropriada para a tipologia linguística, ou seja, fornecer uma base adequada para evidenciar o paralelismo linguístico entre línguas e famílias linguísticas.
3. A proposta das UD precisa ser adequada para uma anotação rápida e consistente pelo anotador humano.
4. A proposta das UD precisa ser compreendida e utilizada com facilidade por uma pessoa sem formação como linguista, seja um aprendiz de línguas ou um engenheiro com demandas básicas de processamento de língua.
5. A proposta das UD precisa ser adequada para o *parsing* de alto desempenho.
6. A proposta das UD precisa apoiar bem as tarefas de Compreensão de Linguagem Natural (Capítulo 1) subsequentes ao *parsing* sintático (extração de relações, compreensão de textos, tradução automática, dentre outras).

Além da representação básica por dependência já explicitada ao longo da Seção 6.6, que é obrigatória para todos os *treebanks*, a UD também prevê um segundo nível de representação de dependência chamada de *enhanced*, no sentido de obter uma representação mais completa e mais enriquecida. Esse segundo nível é uma camada extra de anotação das relações, realizada a fim de dar uma base mais completa para a interpretação semântica. Essa representação *enhanced* não é uma árvore tal como a árvore de dependências básicas, mas uma estrutura que agrega informação à árvore básica, conforme se apresenta para o Exemplo 6.9 na Figura 6.17.

Exemplo 6.9.

As colegas convenceram a menina a postar a foto com o celular.

Figura 6.17: Anotação de sintaxe de dependência incluindo relações básicas e *enhanced*



A Figura 6.17 mostra, na parte de cima da sentença, as relações de dependência básicas, e na parte de baixo, as relações *enhanced*. No Exemplo 6.9, temos dois verbos, cada um dos quais tem seu sujeito. Nas relações básicas, indicamos, por meio da relação *nsubj*, o sujeito (“as colegas”) do verbo principal (“convenceram”). Nas relações *enhanced*, podemos indicar, por meio da relação *nsubj:xsubj*, o sujeito (“a menina”) do segundo verbo (“postar”). Ambos os sujeitos são relevantes para a interpretação de papéis temáticos e, conseqüentemente, para tarefas em PLN como a extração de informação.



6.6.3.1 Tagsets

Conforme introduzido na Seção 6.5, os chamados *tagsets* são conjuntos de etiquetas usadas para a anotação de categorias. Para garantir a comparabilidade linguística, os *treebanks* que aderem às Dependências Universais usam conjuntos de etiquetas, com as mesmas definições dadas para todas elas.

Os *tagsets* usados na UD contemplam a anotação: i) **morfológica**; e ii) **sintática**.

A anotação **morfológica** abrange três informações linguísticas: i) lema da palavra; ii) classe de palavra (em inglês, *part-of-speech* ou PoS) ; e iii) traços ou características morfológicas (em inglês, *features*).

Utilizamos o termo **lema** para referir-nos à forma dicionarizada da palavra. Em nossa sentença, “A menina postou uma foto com o celular”, o verbo “postou” tem como lema “postar”, enquanto o substantivo “menina” tem o lema “menino”.

A **classe de palavra**, como já vimos, é a categoria gramatical na qual essa palavra é classificada. Por exemplo, “postou” pertence à classe dos verbos, “menina” e “celular” pertencem à classe dos substantivos, “uma” e “o” à classe dos determinantes (ou artigos) e “com” à classe das preposições.

O *tagset* de PoS das UDs compreende 17 etiquetas, que são: **ADJ** (adjetivo), **ADP** (adposição, incluindo preposição), **ADV** (advérbio), **AUX** (verbo auxiliar), **CCONJ** (conjunção coordenativa), **DET** (determinante), **INTJ** (interjeição), **NOUN** (substantivo), **NUM** (numeral), **PART** (partícula), **PRON** (pronome), **PROPN** (nome próprio), **PUNCT** (sinal de pontuação), **SCONJ** (conjunção subordinativa), **SYM** (símbolo), **VERB** (verbo) e **X** (outros). Para uma explicação detalhada do *tagset* sugere-se consultar a Seção 4.3.3.

Já para as características morfológicas ou *features*, as UDs contemplam 24 *features*, que são divididas em lexicais e flexionais, e mais de 200 valores para essas *features*. Por exemplo, para a classe substantivo, temos em português as *features* gênero e número. Para gênero, podemos selecionar entre os valores: feminino, masculino e neutro. Para a classe verbo, temos as *features* pessoa, número, tempo verbal, modo, cada uma delas com valores específicos. Assim, o verbo “postou” está conjugado na 3ª pessoa do singular do tempo pretérito perfeito do modo indicativo, e cada uma dessas informações corresponde a um valor dentro de uma *feature* morfológica.

O segundo tipo de anotação que a UD prevê é a **sintática**, por meio da qual são determinadas as relações de dependência entre as palavras de uma sentença. Hoje em dia existem 37 relações de dependência básicas. O Quadro 6.4 mostra as relações de dependência organizadas de acordo com os fundamentos das UDs.

Quadro 6.4. Relações de dependência da UD¹⁰

	Nominals	Clauses	Modifier words	Function words
Core arguments	nsubj obj iobj	csubj ccom xcomp		

¹⁰Referência: <https://universaldependencies.org/u/dep/all.html>.



Non-core dependents	obl	advcl	admod	aux
	vocative		discourse	cop
	expl			mark
	dislocated			
Nominal dependents	nmod	acl	amod	det
	appos			clf
	nummod			case
Coordination	MWE	Loose	Special	Other
conj	fixed	list	orphan	punt
cc	flat	parataxis	goeswith	root
	compound		rearandum	dep

O Quadro 6.4 possui duas partes: na parte superior, estão as principais relações de dependência, distribuídas por tipo (se são relações nominais, verbais, com modificadores ou com palavras funcionais) e ainda classificadas por tipo de argumento. Já a parte inferior do quadro apresenta relações adicionais.

Na parte superior do Quadro 6.4, as relações estão agrupadas segundo duas dimensões:

1. no sentido horizontal, por linha, de acordo com as relações estabelecidas entre uma palavra e seu regente (*head*), e
2. no sentido vertical, por coluna, de acordo com as categorias dos dependentes.

Vemos, assim, no Quadro 6.4, que as relações que uma palavra estabelece com a palavra regente podem ser aquelas equivalentes a funções básicas da predicação em orações (*core arguments*), tais como sujeito da predição (*nsubj*, *csubj*) ou objetos (*obj*, *iobj*) ou complementos (*ccomp*, *xcomp*).

Também podem ser relações que correspondem a funções adicionais àquelas básicas (*non-core dependents of clausal predicates*), tais como adjuntos (*obl*), orações adverbiais (*advcl*), vocativos (*vocative*) etc. Essas relações se aplicam tanto no nível da oração como do sintagma, como é o caso de sintagmas verbais constituídos por um verbo e verbos auxiliares.

Há ainda relações de dependência dentro do sintagma nominal, como é o caso de modificadores adjetivais (*amod*), modificadores adnominais (*nmod*), determinantes (*det*), caso (*case*) etc.

Na parte inferior do Quadro 6.4, temos relações que indicam coordenação de palavras, sintagmas e orações, expressões multipalavra (MWE) (Capítulo 5), relações especiais para distintos fenômenos do discurso, tais como listas, disfluências, elipse, erros ortográficos e sinais de pontuação.

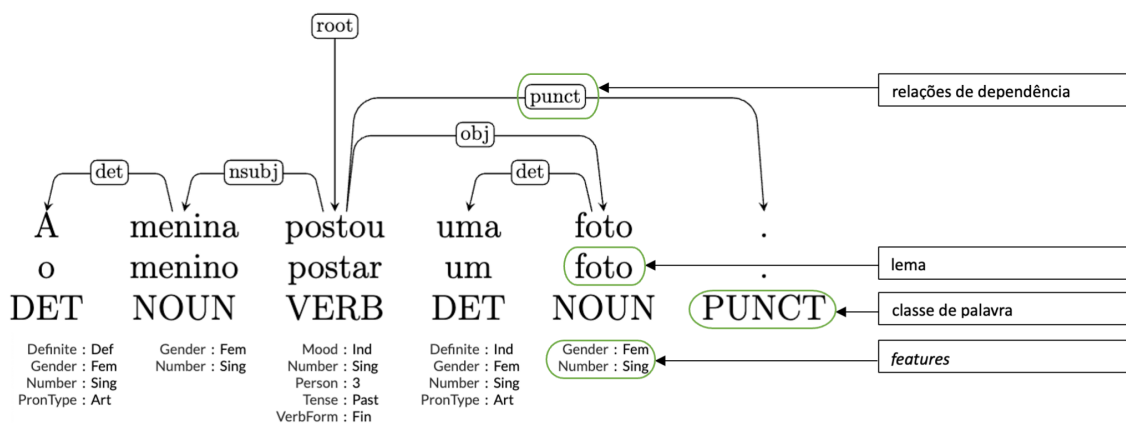
A anotação completa de uma sentença de acordo com as *guidelines* das UD's pode ser observada na Figura 6.18.

6.7 Qual é melhor: constituição ou dependência?

Uma vez apresentados os dois tipos de análise sintática mais utilizados em PLN, cabe perguntarmos quais as vantagens e desvantagens de se adotar cada um deles. A Figura 6.19 apresenta uma síntese dos principais pontos de cada tipo de análise sob a perspectiva de seu potencial em projetos de PLN.



Figura 6.18: Visualização de todas as informações anotadas para uma sentença no projeto UD



6.8 Fronteiras da sintaxe

Conforme apontado na Seção 6.1 deste capítulo, a sintaxe é um dos estratos centrais no sistema linguístico (Figura 6.1). Por estar no centro do sistema, seu estudo possui interseção com vários outros níveis, como a morfologia, a semântica, a pragmática e o discurso. Isso porque a separação em níveis ou estratos é uma forma didática de apresentar o objeto de estudo de cada área; porém, na língua em uso, esses níveis são interdependentes uns dos outros. Portanto, definir os limites entre um nível e outro é uma tarefa complexa. A seguir, exploraremos alguns conceitos e problemas linguísticos que estão na fronteira entre a sintaxe e um outro nível de análise linguística.

6.8.1 Sintaxe e Morfologia

Os traços morfológicos são gramaticalizados de diferentes formas em distintas línguas. Assim, qualquer noção que possa ser expressa morfológicamente em uma língua (por meio de traços morfológicos) pode ser expressa lexicalmente em outras línguas (por meio de perífrases ou paráfrases) (Bender, 1959).

Em português, convencionou-se que os afixos (prefixos, sufixos e infixos), as desinências (nominais e verbais) e outros componentes de palavras são unidades estudadas pela morfologia, enquanto a relação entre as palavras é estudada pela sintaxe.

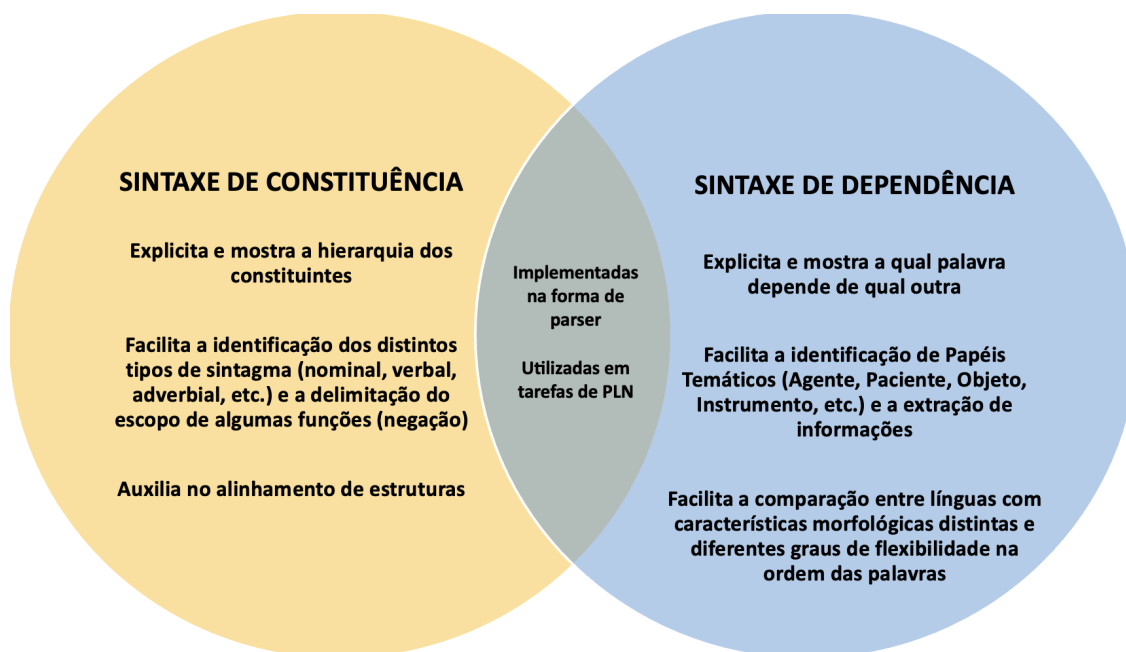
Mesmo tendo-se estabelecido essa convenção, há aspectos linguísticos que estão na fronteira entre a morfologia e a sintaxe. Por exemplo:

A língua portuguesa constrói gênero gramatical (feminino ou masculino) nos substantivos por meio de uma desinência, por exemplo: “médico” e “médica”. Há, porém, substantivos chamados “epícenos”, que designam, com uma mesma forma, os dois gêneros (masculino e feminino), sendo feita a distinção de um gênero ou outro por meio da adição das palavras “macho” e “fêmea”, como, por exemplo: “cobra **macho**” e “cobra **fêmea**”. O gênero dos substantivos em português é, portanto, estudado tanto pela morfologia (quando se utiliza uma desinência) quanto pela sintaxe (quando é necessário usar um sintagma contendo duas palavras).

Há também casos em que a mesma palavra (a mesma grafia) pode ser classificada como adjetivo ou particípio passado de um verbo (e.g. “solto”, “feito”, “motivada”). Isso depende



Figura 6.19: Vantagens e desvantagens da sintaxe de constituição e de dependência



da estrutura na qual a palavra opera. Assim, a distinção entre essas duas classes de palavras é de ordem sintática; contudo, tem impacto na morfologia, uma vez que, dependendo da classe de palavra atribuída, a palavra pode receber *features* de adjetivos ou *features* de verbos. Se em um texto clínico, encontramos a frase “Paciente lúcido e orientado”, interpretamos “lúcido” e “orientado” como adjetivos que qualificam o substantivo “paciente” e constroem significados do estado de saúde do mesmo. Nesse caso, “orientado” é classificado como adjetivo e recebe as *features* de gênero e número. Já em “Paciente orientado a procurar fisioterapia”, “orientado” é classificado como particípio passado do verbo “orientar”, em uma oração passiva sem explicitação do auxiliar “ser” e sem realização de uma função que indica o Papel Temático de Agente da ação, o qual no contexto pode ser interpretado como o profissional da saúde que realizou o atendimento. Neste caso, “orientado” recebe *features* da classe verbo e também da classe adjetivo, uma vez que em português os particípios passados concordam em gênero e número com os substantivos que realizam o sujeito da oração: “O paciente foi orientado”, “A paciente foi orientada”.

O tempo futuro do presente, em português, pode ser construído por meio de uma desinência (por ex. “á” para a terceira pessoa do singular: “soltará”, “fará”, “motivará”) e anotado com a *feature* morfológica de futuro. Também pode ser construído por uma perífrase verbal (e.g. “vai soltar”, “vai fazer”, “vai motivar”). Neste segundo caso, nenhum dos verbos recebe a *feature* morfológica de futuro, já que o significado de futuro se dá pela combinação dos dois verbos, e não por uma desinência. As formas construídas por meio de desinência são chamadas de futuro sintético e constroem a noção de futuro com recursos da morfologia. Já as formas construídas por meio de perífrase são denominadas futuro analítico e constroem a noção de futuro com recursos da sintaxe.

Formas negativas de adjetivos podem ser criadas em português por meio de morfemas (e.g. “capaz” x “incapaz” ou “normal” x “anormal” ou “moral” x “imoral”) ou lexicalmente (i.e “governamental” x “não governamental”). O processo de formação de palavras por meio da adição de prefixos de negação **a-**, **i-**, **im-**, **in-** é estudado dentro da morfologia, ao



passo que a construção da negação a partir da inserção de uma outra palavra é estudada no escopo da sintaxe.

As formas verbais que indicam imperativo em português possuem a mesma morfologia de subjuntivo. A definição entre um modo verbal e outro depende do contexto sintático em que estão inseridas. Por exemplo na sentença “Faça isso imediatamente”, o verbo “faça” é classificado como modo imperativo. Já na sentença “Quero que você faça isso imediatamente”, o verbo “faça” é classificado como subjuntivo. Para definir a *feature* morfológica de modo verbal a ser atribuída a formas como “faça” é preciso antes fazer a análise sintática da sentença.

6.8.2 Sintaxe e Semântica

A análise sintática pode ser complementada pela análise no nível da semântica. Essa complementação pode ser ilustrada com o seguinte exemplo.

Dada uma oração como “João deu um pulo”, uma análise sintática de constituintes nos permite identificar dois constituintes maiores: “João” e “deu um pulo”, este último passível de ser segmentado em constituintes menores contidos nele: “deu” e “um pulo”. Numa análise de sintaxe de dependência, observamos as relações entre as palavras e identificamos “deu” como sendo a palavra raiz (*root*), “João” seu dependente em relação de sujeito (*nsubj*) e “pulo” em relação de objeto (*obj*). Assim, interpretamos que alguém (“João”) realiza uma ação (“dar”) que tem um objeto (“um pulo”). No entanto, todos nós interpretamos “dar um pulo” como expressão de uma ação equivalente a “pular”. Como vimos neste capítulo, os constituintes que exercem a função de objeto geralmente correspondem a um papel temático na semântica. Em uma oração como “João deu um dinheiro”, “dinheiro” é objeto do verbo “dar” e cumpre o papel temático de objeto estativo. Já em “João deu um pulo”, o objeto “pulo” não possui status de papel temático; portanto, consideramos “deu um pulo” como uma única unidade do ponto de vista semântico. Assim, a análise sintática de dependência apresenta a mesma configuração para orações como “João deu um pulo” e “João deu um dinheiro”. Em PLN, casos como este demandam uma análise semântica complementar à análise sintática, pois a identificação de papéis temáticos é relevante em tarefas como a extração de informação.

6.8.3 Sintaxe e Pragmática

A análise sintática pode, também, ser complementada pela análise sob a perspectiva da pragmática. Os exemplos a seguir ilustram essa complementação.

Em português, temos configurações de sentenças com uma palavra indicando negação, as quais constroem, no entanto, significados positivos. Numa sentença exclamativa como “Quantas vezes ela não ligou chorando!”, temos o advérbio “não”, que constrói um significado negativo. Contudo, interpretamos a exclamação como construindo um significado afirmativo: “ela ligou chorando muitas vezes”. Em casos como este, de sentenças com elementos negativos que constroem um significado afirmativo, vemos que a perspectiva pragmática, isto é, a análise da linguagem em uso e dos pressupostos e inferências que fazemos como falantes, é relevante para complementar a análise sintática. Do contrário, interpretaremos a exclamação como a negação do evento: “ela não ligou chorando”.

A perspectiva da pragmática também nos ajuda a interpretar sentenças nas quais não há elemento de negação, mas o significado construído é negativo. Por exemplo, em “Pessoa alguma será condenada”, não temos nenhuma palavra que construa o significado de negação. No entanto, a oração é interpretada como “Nenhuma pessoa será condenada”. Cabe destacar



que, se trocarmos a ordem das palavras neste exemplo, teremos uma interpretação distinta, como é evidente ao contrastarmos “Pessoa alguma será condenada” e “Alguma pessoa será condenada”.

6.8.4 Sintaxe e Discurso

O nível do Discurso e os Modelos Discursivos (Capítulo 11) serão apresentados mais à frente neste livro, mas convém explicitar neste momento uma confusão que se faz comumente entre os níveis sintático e discursivo.

A Gramática Tradicional (GT), nas seções referentes à sintaxe, estuda os conceitos de frase, oração e período, assim como a classificação das orações dentro de um período, por exemplo em oração principal, oração coordenada adversativa ou oração subordinada adverbial consecutiva, dentre várias outras.

Essas mesmas relações são estudadas em PLN no nível do Discurso como relações de contraste, de explicação, de equivalência, dentre várias outras. Isso porque as teorias discursivas, tais como RST (*Rhetorical Structure Theory*) e CST (*Cross-document Structure Theory*) não limitam sua unidade de análise linguística apenas dentro do texto, da sentença, do período ou da frase. São teorias que estudam o discurso independente do tamanho ou dos limites marcados por sinais de pontuação.

Apesar do rol de relações das teorias discursivas não representar uma equivalência exata com os tipos de orações da Gramática Tradicional, podemos identificar paralelos, por exemplo, entre uma relação de *Summary* (da RST) e uma oração coordenada conclusiva (da GT) ou entre uma relação discursiva de *Indirect Speech* (da CST) com uma oração subordinada substantiva objetiva direta).

6.9 Considerações finais

Neste capítulo, introduzimos os conceitos básicos de sintaxe, que é a área da linguística responsável por definir e descrever a ordem e a função das palavras e sintagmas na frase. Assim, foram discutidos conceitos linguísticos de **classe de palavra, constituinte, sintagma, frase, oração, período (ou sentença), escala de ordens, função sintática, papel temático**, assim como conceitos sintáticos voltados para o PLN, tais como *part-of-speech* (PoS), *parser*, *parsing*, *treebanks*, árvore de dependência, núcleo (*head*, *governor*, *parent*), dependente (*child*, *dependent*), entre outros.

Como o escopo deste livro é o processamento de língua natural, explicitamos as duas principais abordagens sintáticas usadas em PLN, que são: a **sintaxe de constituintes** e a **sintaxe de dependência**. Mas é importante ressaltar que existem várias correntes linguísticas teóricas que estudam e descrevem a sintaxe a partir de diferentes pontos de vista. Algumas dessas correntes são: paradigma formal ou estrutural, paradigma funcional ou funcionalista, o sistêmico-funcional, o gerativo ou gerativista, dentre outros.

Tanto a sintaxe de constituintes quanto a de dependência são abordagens amplamente usadas em PLN e, para isso, precisam de uma representação formal. Foram apresentados os tipos mais comuns de representação dessas duas abordagens, a saber: por meio de **colchetes**, de **árvores**, de **setas**, de **parênteses** e de **indentação**.

Ao final do capítulo, apresentamos alguns exemplos de questões e problemas linguísticos que podem ser estudados pelo viés da sintaxe ou de algum outro nível linguístico, já que essas fronteiras não são tão bem definidas na língua em uso. Como o foco do livro está na língua portuguesa, nos limitamos a mencionar brevemente alguns poucos exemplos.



Para um estudo mais aprofundado sobre as fronteiras entre a sintaxe e os demais níveis linguísticos, ver Bender (1959).

O próximo capítulo (Capítulo 7) também é dedicado ao estudo da sintaxe dentro do PLN, porém com um olhar mais computacional voltado para os recursos e ferramentas disponíveis para fazer análise sintática automaticamente, os tipos de *parsing* e as abordagens sintáticas mais comuns em PLN.



Capítulo 7

Ferramentas e recursos para o processamento sintático

Elisa Terumi Rubel Schneider
Adriana S Pagano
Ana Clara S Pagano

7.1 Introdução

A sintaxe é o nível de análise linguística no qual examinamos os padrões de estruturação de sentenças. Isto é, analisamos como as palavras se organizam em unidades que constroem significado dentro da sentença. Para isso, consideramos a classe de cada palavra, sua ordem na sentença e sua relação com as outras palavras. Conforme visto no Capítulo 6, em PLN, a análise computacional realizada no nível sintático é denominada *parsing*, a ferramenta que realiza essa tarefa é denominada *parser* e o recurso criado por meio da análise sintática é chamado *treebank*.

Neste Capítulo, vamos conhecer tipos de *parsing* sob a perspectiva computacional, juntamente com ferramentas e recursos disponíveis para o processamento do português brasileiro.

7.2 Tipos de *parsing*

A tarefa de *parsing* consiste em, dada uma entrada com uma sentença sem nenhuma anotação (*raw*), um modelo faz uma predição da estrutura sintática dessa sentença. Como vimos no Capítulo 6, o objetivo do processamento sintático é identificar as unidades (como palavras, sintagmas e orações) na sentença e estabelecer as relações gramaticais entre elas a fim de extrair algum tipo de informação. Essas relações podem ser analisadas em termos de:

1. **constituência**, ou seja, quais unidades são hierarquicamente inferiores às outras e estão nelas contidas; ou
2. **dependência**, isto é, quais palavras dependem de quais outras e qual é o tipo de relação entre elas, incluindo o papel de cada palavra na sentença (como sujeito, objeto, verbo, adjetivo etc.).

Assim, de acordo com o tipo de análise sintática adotada, há *parsers* de constituição e *parsers* de dependência.

Mas há uma perspectiva adicional sob a qual podemos caracterizar tipos de *parsing* e *parsers*: trata-se do escopo ou profundidade com que a análise sintática é executada.



Nesse sentido, podemos analisar as sentenças de forma exaustiva até obtermos uma análise completa de sua estrutura ou fazer uma análise mais rasa para obtermos uma análise com informações mínimas, mas relevantes para as tarefas em PLN.

O primeiro tipo é denominado *deep* (em português, *profundo*) ou *parsing* completo e o segundo tipo é denominado *shallow* (em português, *superficial*) ou *parsing* parcial. Contudo, cabe uma observação sobre esta terminologia. No uso geral, os termos *parsing* e *parser* acabaram sendo adotados para se referir ao *parsing* completo. Já o *parsing* parcial é conhecido como *chunking* (em português, *cortar*) e a ferramenta como *chunker*, embora *chunking* seja uma dentre várias abordagens para a implementação do *parsing* parcial (Jurafsky; Martin, 2023).

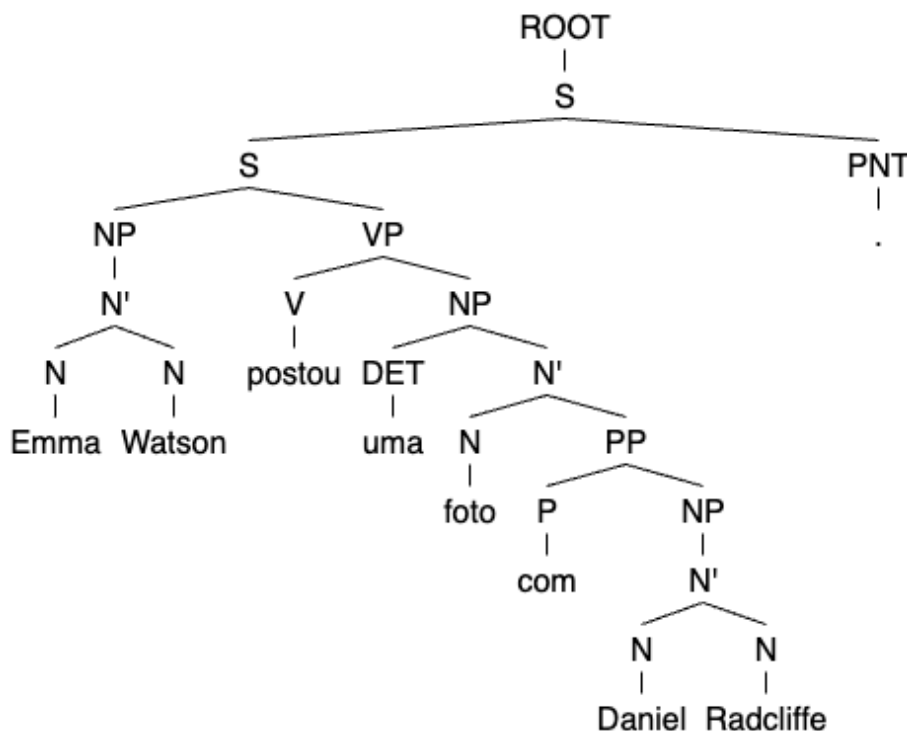
Tanto o *parsing* de constituição como o *parsing* de dependência podem ser executados de forma completa ou parcial. Tomando como exemplo o *parsing* de constituição, uma análise completa ou *deep* extrai todos os agrupamentos e as relações sintáticas em uma sentença. Por exemplo, dada a sentença:

Exemplo 7.1.

Emma Watson postou uma foto com Daniel Radcliffe.

Temos na Figura 7.1 uma representação em diagrama de árvore que mostra a profundidade da análise.

Figura 7.1: Exemplo de saída de *parser*



Já uma análise parcial extrai constituintes delimitados, sem estabelecer a hierarquia entre eles ou de que forma uns estão contidos em outros.

A Figura 7.2 ilustra a análise rasa, não hierárquica do *parsing* parcial para o Exemplo 7.1.

O objetivo do *parsing* parcial é gerar uma representação rasa da estrutura da sentença que possibilite um processamento mais rápido de grandes volumes de texto. É geralmente



Figura 7.2: Exemplo de saída de *parser* parcial

[NP Emma Watson] [VP postou] [NP uma foto] [PP com] [NP Daniel Radcliffe]
--

implementado por meio de tokenização de uma sentença em palavras, identificação da classe de palavra (PoS) e segmentação em pedaços ou *chunks*. O conceito de *chunk* foi proposto por Abney (1992) como uma unidade formada por uma única palavra ou por um conjunto de palavras. Em um *chunk*, há uma palavra de conteúdo circundada por palavras funcionais. A palavra de conteúdo mais explorada em *chunking* é o substantivo, dada a alta correlação de substantivos com entidades.

Assim, a tarefa de *chunking* da sentença do Exemplo 7.1, executada em Python utilizando o modelo de língua portuguesa `pt_core_news_sm`, da biblioteca `spaCy`, gera o resultado disposto na Figura 7.3:

Figura 7.3: Captura de tela de *notebook* em Python com código e resultado de *chunking* de uma sentença em português

```

1  !pip install spacy
2  !python -m spacy download pt_core_news_sm
3  import spacy
4  nlp = spacy.load("pt_core_news_sm")
5  sentence = "Emma Watson postou uma foto com Daniel Radcliffe."
6  doc = nlp(sentence)
7  for chunk in doc.noun_chunks:
8      print(chunk.text)

```

```

Emma Watson
uma foto
Daniel Radcliffe

```

Como vemos na Figura 7.3, o *chunking* reconhece três unidades ou *chunks*, cada uma nucleada por um substantivo. Os três *chunks* são candidatos a entidades, sendo duas delas nomes próprios de pessoas. De fato, como veremos no Capítulo 17, vários modelos de Extração de Informação utilizam análises rasas como a fornecida pelo *chunking*.

7.3 Recursos e ferramentas para o português

Nesta seção, vamos conhecer alguns recursos e ferramentas de PLN para análise sintática do português.

7.3.1 *Corpora*

O primeiro recurso para o processamento linguístico é um *corpus* anotado ou *treebank*, isto é, textos enriquecidos com marcações de classe de palavras (*Part-of-Speech*) e relações sintáticas. Um exemplo de *corpus* em português anotado é o Bosque¹, amplamente utilizado para treinar modelos de análise sintática (Veja Capítulo 14).

¹<https://www.linguateca.pt/Floresta/corpus.html>



O *corpus* Bosque é parte de um *corpus* maior, chamado Floresta Sintá(c)tica², que abrange, além do Bosque, outros *subcorpora*, nomeadamente: Selva, Amazônia e Floresta Virgem. O grande *corpus* foi anotado automaticamente pelo *parser* PALAVRAS (Bick, 2000). O Bosque está integrado por sentenças extraídas dos *corpora* CETENFolha (português brasileiro) e CETEMPúblico (português europeu), ambos constituídos por textos jornalísticos escritos. Uma versão do Bosque³ foi convertida para o formato UD (Universal Dependencies), apresentado no Capítulo 6, e é hoje um dos *treebanks* mais utilizados pela comunidade de PLN no Brasil em modelos de *parsing* de dependência atuais.

Além da Floresta Sintá(c)tica, encontra-se disponível, como recurso para a língua portuguesa, o Corpus Internacional do Português – CINTIL⁴, desenvolvido pela Universidade de Lisboa, que possui 1 milhão de *tokens* de texto jornalístico, com anotação de classe de palavra, lema e expressões multipalavra. Uma versão desse *corpus*, o CINTIL-UDep⁵, é disponibilizada com anotações no padrão UD.

Mais recentemente, o *corpus* PetroGold⁶ foi disponibilizado e, hoje, é um *corpus* passível de ser utilizado em modelos de *parsing* de dependência. PetroGold é um *corpus* de textos acadêmicos no domínio do petróleo, anotado no formato UD e revisado manualmente.

Há diversas iniciativas em andamento, no momento da escrita deste capítulo, para a criação de *corpora* anotados em português brasileiro. No escopo do projeto NLP2, desenvolvido pelo Centro de Inteligência Artificial⁷ (C4A1) da Universidade de São Paulo, com o objetivo de desenvolver recursos, ferramentas e aplicações para levar o português ao estado da arte em PLN, o projeto POeTiSA⁸ desenvolve o *treebank* Porttinari⁹, um *corpus* multi-gênero de textos em português brasileiro anotados de acordo com o padrão UD. Inclui textos jornalísticos do *corpus* da Folha de São Paulo/Kaggle, o *corpus* MAC-MORPHO¹⁰ de textos jornalísticos, o *corpus* DANTE¹¹ (*Dependency-ANalised corpora of TwEets*), integrado por tweets da Bolsa de Valores, B2W-reviews01¹², composto por resenhas e avaliações de consumidores da empresa de comércio eletrônico Americanas e um *corpus* de Resenhas online de livros. A versão Porttinari-base já se encontra disponível¹³.

Uma iniciativa também em andamento é o *corpus* Veredas¹⁴, desenvolvido na Faculdade de Letras da UFMG, que visa à construção de *treebanks* de textos anotados de acordo com o padrão das UD. Inclui amostras de uma variedade de textos em inglês, espanhol e português brasileiro: colunas jornalísticas, fábulas, narrativas, receitas culinárias, questionários médicos e bulas de medicamento. Em parceria com a PUCPR, a Faculdade de Letras da UFMG desenvolveu o *treebank* DepClinBr, um *corpus* de narrativas clínicas anotadas de acordo com o padrão das UD (Oliveira et al., 2022b).

²<https://www.linguateca.pt/Floresta/principal.html>

³https://universaldependencies.org/treebanks/pt_bosque/index.html

⁴<http://cintil.ul.pt/>

⁵https://universaldependencies.org/treebanks/pt_cintil/index.html

⁶https://universaldependencies.org/treebanks/pt_petrogold/index.html

⁷<https://c4ai.inova.usp.br/>

⁸<https://sites.google.com/icmc.usp.br/poetisa/the-project>

⁹<https://sites.google.com/icmc.usp.br/poetisa/resources-and-tools>

¹⁰Mac-Morpho: <http://www.nilc.icmc.usp.br/macmorpho/>

¹¹Brazilian Stock Market Tweets with Emotions|Kaggle: <https://www.kaggle.com/datasets/fernandojvdasilva/stock-tweets-ptbr-emotions>

¹²<https://opencor.gitlab.io/corpora/real19b2wreviews01/>

¹³Corpus Porttinari: <https://sites.google.com/icmc.usp.br/poetisa/porttinari>

¹⁴<http://www.letras.ufmg.br/veredas/>



7.3.2 Parsers

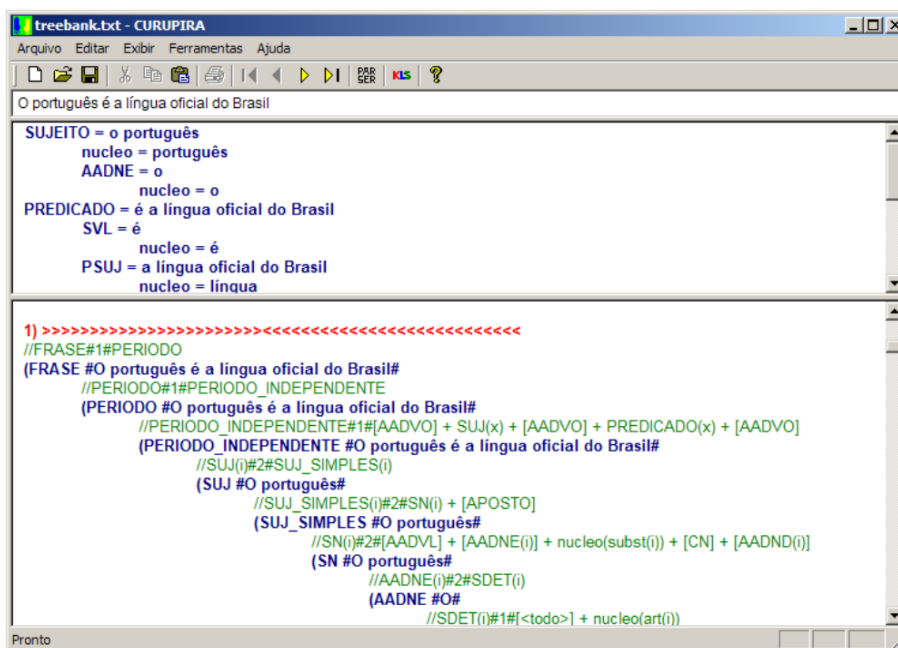
Parsers são ferramentas que podem auxiliar uma aplicação (por exemplo, tradução automática, sumarização de textos, extração de informação, *question-answering*) ou fazer parte de uma ferramenta maior ou conjunto de ferramentas (*toolkit*).

Algumas ferramentas computacionais estão disponíveis para realizar a análise sintática em português. A análise pode ser feita por meio de:

7.3.2.1 Programas e aplicativos

Existem diversos *parsers* desenvolvidos por distintos grupos de pesquisa, fornecidos através de um programa de computador ou aplicativo a ser instalado. No Brasil, podemos citar Curupira¹⁵, Donatus¹⁶ e PassPort¹⁷. Curupira é um analisador robusto de uso geral para o português brasileiro, fornecendo um conjunto das análises sintáticas possíveis para uma frase de entrada. A ferramenta analisa sentenças de cima para baixo, da esquerda para a direita, por meio de uma gramática funcional livre de contexto, restrita e relaxada, para o português brasileiro escrito padrão e um léxico extenso e de ampla cobertura. A Figura 7.4 apresenta a interface gráfica onde é possível ver a obtenção de toda informação da análise realizada pelas regras do *parser*¹⁸.

Figura 7.4: Captura de tela da interface gráfica da ferramenta Curupira



Fonte: (Martins; Nunes; Hasegawa, 2003)

Donatus é um projeto que consiste em ferramentas e gramáticas baseadas em Python e na biblioteca NLTK¹⁹ para análise profunda e anotação sintática de *corpora* do português

¹⁵<http://www.nilc.icmc.usp.br/nilc/tools/curupira.html>

¹⁶<https://sourceforge.net/projects/donatus/>

¹⁷PassPort (A Dependency Parsing Model for Portuguese | SpringerLink): https://link.springer.com/chapter/10.1007/978-3-319-99722-3_48

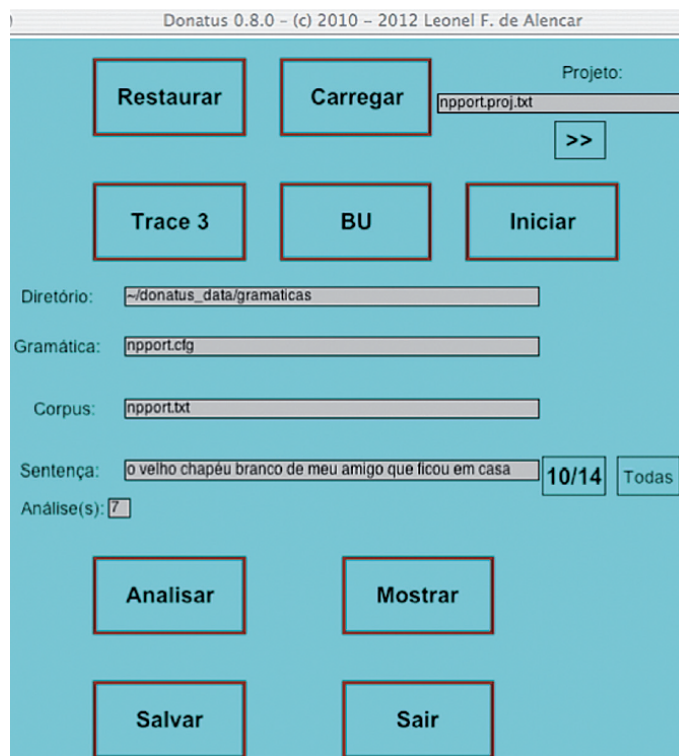
¹⁸No entanto, vale ressaltar que o projeto foi desenvolvido em 2002 - 2004, de acordo com o site, e pode não estar mais disponível para ser obtido.

¹⁹NLTK :: *Natural Language Toolkit* <https://www.nltk.org>



brasileiro. Inclui uma interface gráfica, conforme pode ser visto na Figura 7.5. Está disponível em repositório público, sob licença *GNU General Public License version 3.0 (GPLv3)*.

Figura 7.5: Captura de tela da interface gráfica da ferramenta Donatus



Fonte: (Alencar, 2012)

PassPort é uma ferramenta para análise de dependências de português treinado com o Stanford Parser, utilizando o *corpus* Portuguese Universal Dependency (PT-UD). Infelizmente, a página do projeto não se encontra disponível na data de escrita deste capítulo.

7.3.2.2 Frameworks e Bibliotecas

Devido à popularidade de Python, muitas bibliotecas de PLN foram desenvolvidas na linguagem. Entre as bibliotecas que incluem *parsing* para língua portuguesa, podemos citar spaCy²⁰, Stanza²¹ e NLTK.

spaCy é uma biblioteca PLN que oferece análise linguística eficiente e rápida para várias línguas, incluindo o português. Inclui recursos para tokenização, marcação de parte do discurso (PoS *tagging*), reconhecimento de entidades nomeadas, análise sintática e outros. Através de modelos pré-treinados, o spaCy é capaz de fornecer análises detalhadas, permitindo a extração de informações semânticas de um texto em língua portuguesa.

Stanza²² é outra biblioteca PLN que suporta vários idiomas, incluindo o português,

²⁰<https://spacy.io/>

²¹<https://stanfordnlp.github.io/stanza/index.html>

²²Cabe esclarecer que embora o Stanford forneça um modelo treinado para língua portuguesa, que pode ser utilizado pelas bibliotecas Python, na interface online deste projeto não há suporte para o idioma



desenvolvida pela Universidade Stanford. Fornece uma gama de recursos semelhantes ao spaCy, com suporte a análises mais profundas, como a análise de dependência neural.

NLTK (*Natural Language Toolkit*) também é uma biblioteca em Python que oferece suporte para tarefas de PLN em língua portuguesa, como a análise sintática. NLTK permite o *parse* usando expressões regulares (com *Regex Parser*), análise de dependência com analisador de dependência probabilístico e análise de dependência com analisador de Stanford.

É importante mencionar que, além de Python, outras linguagens de programação também oferecem bibliotecas e *frameworks* para análise sintática.

7.3.2.3 Ferramentas online

A seguir apresentaremos algumas das ferramentas web disponíveis para *parsing* de texto em português, permitindo executar a análise sintática e obter como saída arquivos com distintos formatos, incluindo a visualização das árvores sintáticas. Todas as ferramentas apresentadas são de acesso livre e gratuito.

O Parser LX

O Parser LX, ferramenta integrante do PORTULAN CLARIN²³, é parte integrante de um portal de acesso a infraestrutura de tecnologia linguística no escopo do projeto internacional CLARIN ERIC²⁴.

O Parser LX é disponibilizado tanto para *parsing* de constituição²⁵ como de dependência, este último em duas versões: LX-DepParser²⁶ e LX-UDParser²⁷

A interface é simples e amigável, tendo como entrada uma sentença que o usuário pode digitar em campo próprio ou um arquivo que deverá ser importado.

A Figura 7.6 mostra uma captura de tela da interface do Parser LX de constituição.

Figura 7.6: Captura de tela mostrando a interface do *parser* de constituição LX.



Como vemos na Figura 7.6, o *parser* tem como saída uma visualização na forma de árvore, denominada “amigável” (em inglês, *friendly*), ou uma representação parentética ou entre parênteses/colchetes ou ainda uma representação na forma tabular.

português.

²³<https://portulanclarin.net/>

²⁴<https://www.clarin.eu/>

²⁵<https://portulanclarin.net/workbench/lx-parser/>

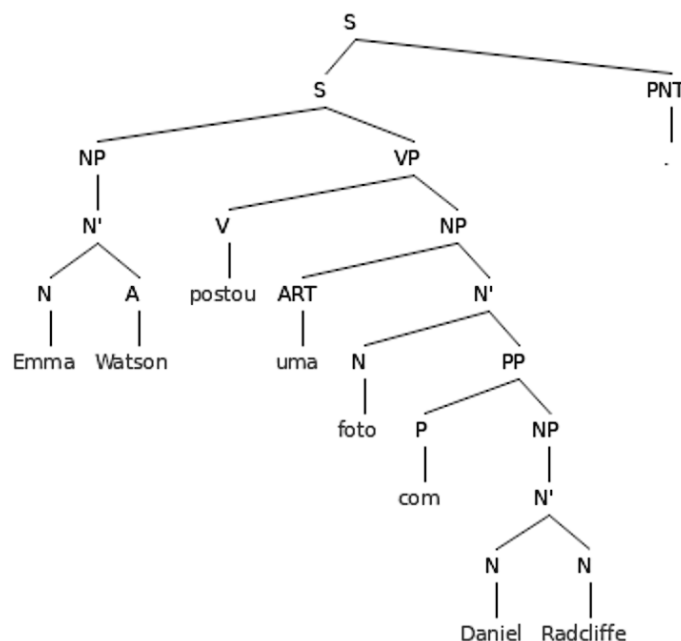
²⁶<https://portulanclarin.net/workbench/lx-depparser/>

²⁷<https://portulanclarin.net/workbench/lx-udparser/>



Para a sentença “Emma Watson postou uma foto com Daniel Radcliffe”, o *parser* gera a árvore de constituição apresentada na Figura 7.7

Figura 7.7: Diagrama de árvore de constituição gerado pelo *parser* LX.



No que diz respeito à sintaxe de dependência, LX possui duas versões, sendo uma delas adaptada ao formato Universal Dependencies, apresentado no Capítulo 6.

A Figura 7.8 mostra uma captura de tela da interface do Parser LX de dependência no formato UD.

Figura 7.8: Captura de tela mostrando a interface do *parser* de dependência LX.

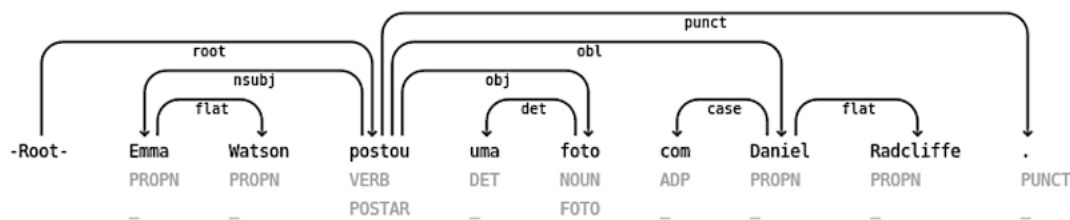


Como a tela mostra, a saída do *parser* pode ser na forma de visualização com setas, denominada “amigável” ou no formato CoNLL-U, que é um formato próprio para *parsing* de dependência, como vimos no Capítulo 6.

Para a sentença “Emma Watson postou uma foto com Daniel Radcliffe”, o *parser* gera a árvore de constituição apresentada na Figura 7.9.



Figura 7.9: Diagrama com setas gerado pelo Parser LX.

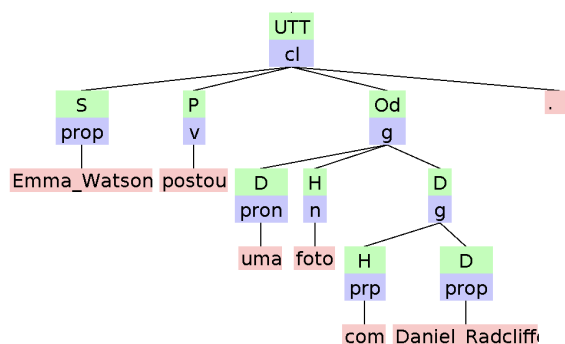


O Parser VISL²⁸

O projeto VISL (do inglês, *Visual Interactive Syntax Learning*) é desenvolvido pelo *Institute of Language and Communication (ISK)* da *University of Southern Denmark*. O projeto disponibiliza recursos (*corpora*) e ferramentas, tais como *parsers* de constituição e dependência. A análise se baseia no *parser* PALAVRAS (Bick, 2000) e no *corpus* Floresta Sintá(c)tica.

Para a sentença “Emma Watson postou uma foto com Daniel Radcliffe”, o *parser* gera a árvore de constituição apresentada na Figura 7.10.

Figura 7.10: Diagrama de árvore de constituição gerado pelo *parser* VISL.



O UDPipe

UDPipe é um conjunto de ferramentas (*toolkit*) e um serviço web que possibilita processar texto por meio de uma *pipeline* que inclui tokenização em palavras, etiquetagem de classe de palavra (PoS), lematização, e *parsing* de dependência. É desenvolvido pelo *Institute of Formal and Applied Linguistics* da *Faculty of Mathematics and Physics* da *Charles University* (República Tcheca). O padrão adotado é das Universal Dependencies e conta com modelos treinados para a maioria dos *treebanks* já anotados no formato UD. Além da interface web, altamente eficiente e amigável, é possível processar texto por meio de um script em Python.

A Figura 7.11 mostra uma captura de tela da interface web do UDPipe, com a seleção do modelo Bosque para a língua portuguesa.

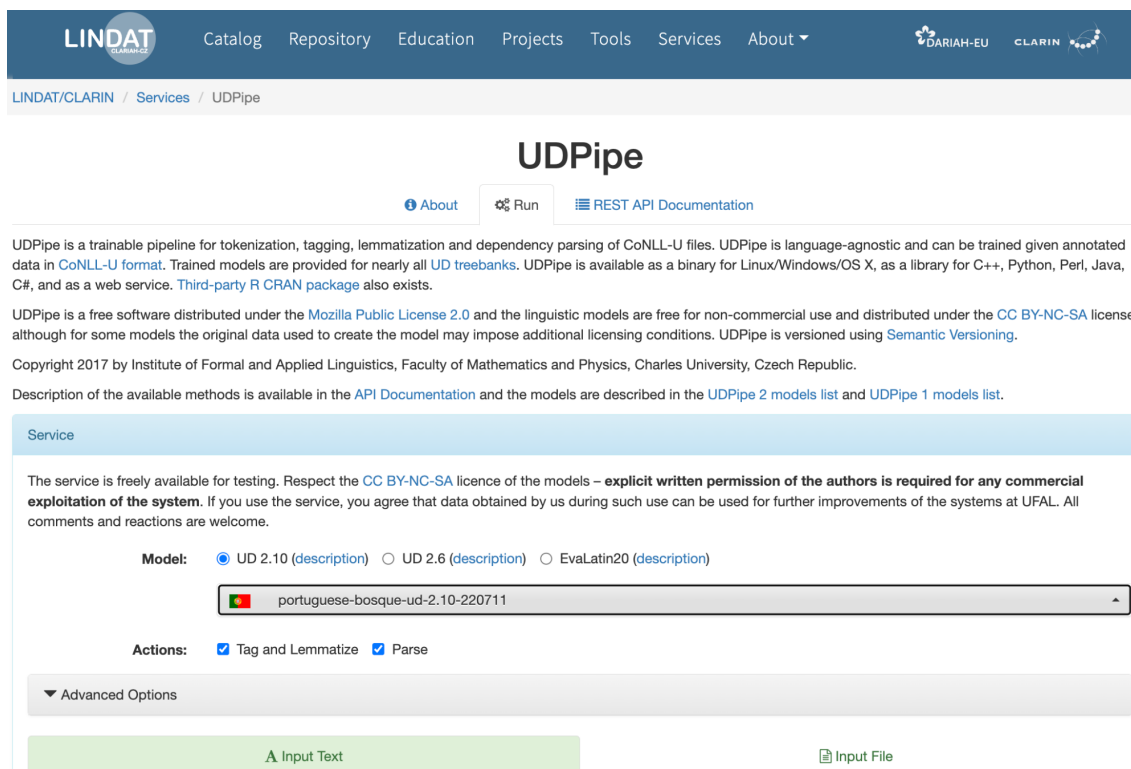
A entrada pode ser texto digitado em um campo próprio ou um arquivo de texto e há três formatos de saída: diagrama arbóreo, formato CoNLL-U e formato tabular.

Para a sentença “Emma Watson postou uma foto com Daniel Radcliffe”, o *parser* gera a árvore de dependência apresentada na Figura 7.12.

²⁸<https://visl.sdu.dk/visl/about/>



Figura 7.11: Captura de tela mostrando a interface do UDPipe.



7.4 Visualização, anotação e edição de *treebanks*

Há diversas ferramentas que permitem tanto a visualização de árvores de constituição e dependência, como a anotação de sentenças e a edição de sentenças já anotadas manualmente ou automaticamente.

7.4.1 Árvores de constituição

Há ferramentas web que oferecem uma visualização gráfica de diagrama de árvores a partir da notação entre colchetes ou parênteses dada como entrada pelo usuário.

Uma dessas ferramentas é *Syntax Tree Generator* (Syntree)²⁹, a qual dada uma entrada com notação em colchetes, gera um diagrama de árvore como mostrado na Figura 7.13.

Ferramentas semelhantes à *Syntax Tree Generator* são:

- Jssyntaxtree (*Dynamic JavaScript version of phpSyntaxTree*)³⁰
- RSyntaxTree (escrita na linguagem de programação Ruby)³¹

²⁹<https://mshang.ca/syntree/>

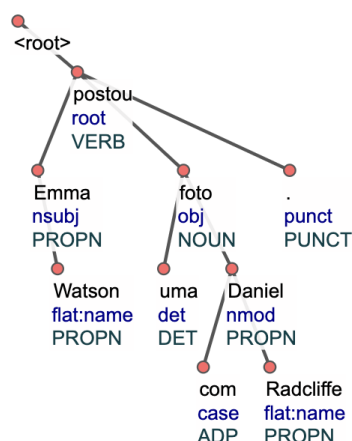
³⁰<https://ironcreek.net/syntaxtree/>

³¹<https://yohasebe.com/rsyntaxtree/>



Figura 7.12: Diagrama de árvore de dependência gerado pelo UDPipe.

Emma Watson postou uma foto com Daniel Radcliffe .



7.4.2 Árvores de dependência

7.4.2.1 Visualização de sentenças anotadas

Há ferramentas web que oferecem uma visualização gráfica de diagrama de árvores a partir do arquivo em formato CoNLL-U dado como entrada pelo usuário. Por exemplo, a ferramenta Grew Web possibilita carregar um arquivo CoNLL-U e visualizar o diagrama de relações de dependência, como mostrado na Figura 7.14.

Uma ferramenta semelhante à Grew Web é CoNLL-U Viewer.³²

7.4.2.2 Buscas em *treebanks*

Há ferramentas web de busca em *treebanks* anotados com relações de dependência. Esse é o caso da ferramenta Grew Match, que possibilita o acesso a 245 *treebanks* e distintos tipos de busca (por exemplo: por palavra, lema, etiqueta de PoS, etiqueta de relação de dependência, ngramas de palavras, lemas e PoS etc.). A Figura 7.15 mostra uma captura de tela com os resultados de uma busca pela palavra “foto” no *treebank* de dissertações e teses em português brasileiro no domínio do petróleo Petrogold.

Uma ferramenta semelhante à Grew Match é TüNDRA (*Tübingen aNnotated Data Retrieval Application*).³³

7.4.2.3 Anotação e edição manual de relações de dependência

Para editar arquivos CoNLL-U previamente anotados de forma manual ou automática, uma das ferramentas mais utilizadas é ArboratorGrew³⁴, que possui uma versão customizada no Brasil pela equipe do ICMC da USP: Arborator-Grew-NILC³⁵. Essa ferramenta permite gerenciar projetos individuais e coletivos de anotação, bem como serve de plataforma instrucional para cursos e treinamentos em anotação de sintaxe de dependência. A

³²https://universaldependencies.org/conllu_viewer.html

³³<https://weblicht.sfs.uni-tuebingen.de/Tundra/>

³⁴<https://arboratorgrew.elizia.net/#/>

³⁵<https://arborator.icmc.usp.br/#/>



Figura 7.13: Visualização de diagrama de árvore de constituição pela ferramenta Syntree.

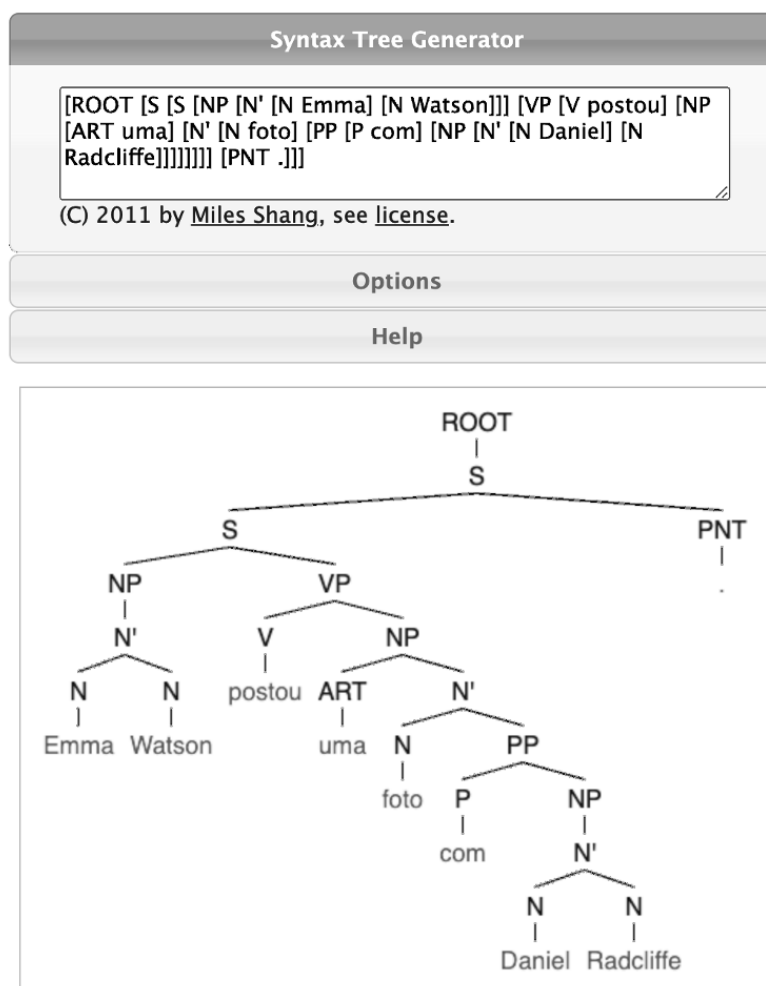


Figura 7.16 mostra a interface da ferramenta no momento de edição de uma das etiquetas de PoS.

A ferramenta Arborator Grew recebe como entrada arquivos CoNLL-U e possibilita a exportação dos arquivos CoNLL-U editados, bem como das imagens dos diagramas de dependência.

Há muitas outras ferramentas de visualização, consulta, anotação e edição de sintaxe de dependência. Nos últimos anos, o projeto Universal Dependencies vem atualizando a lista de ferramentas disponíveis, a qual pode ser consultada no site do projeto³⁶.

7.4.3 Anotação de *corpus* em múltiplos níveis

Há ferramentas que permitem a anotação de sintaxe juntamente com a anotação em outros níveis, como é o caso de entidades nomeadas, relações entre entidades, correferência e outras. Uma das ferramentas mais robustas disponíveis atualmente e com interface amigável e INCEpTION³⁷, desenvolvida pelo *Ubiquitous Knowledge Processing (UKP) Lab*

³⁶<https://universaldependencies.org/tools.html>

³⁷<https://inception-project.github.io/>



Figura 7.14: Visualização de diagrama de árvore de dependência pela ferramenta Grew Web.

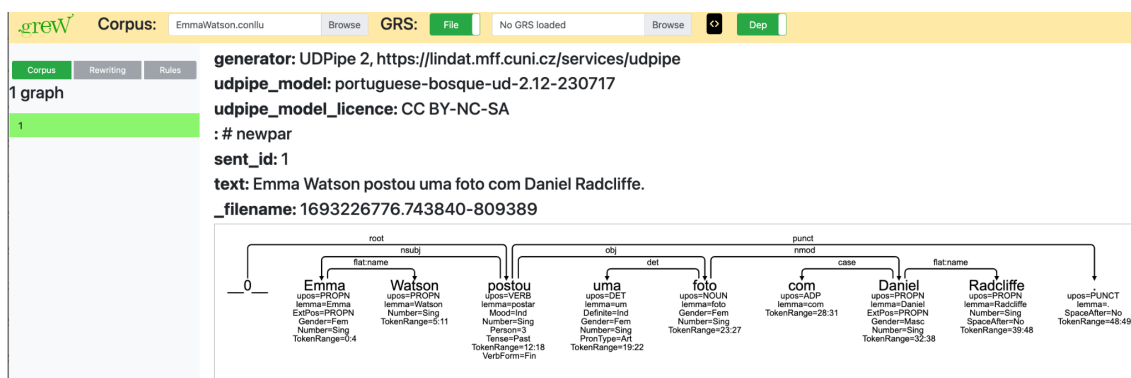
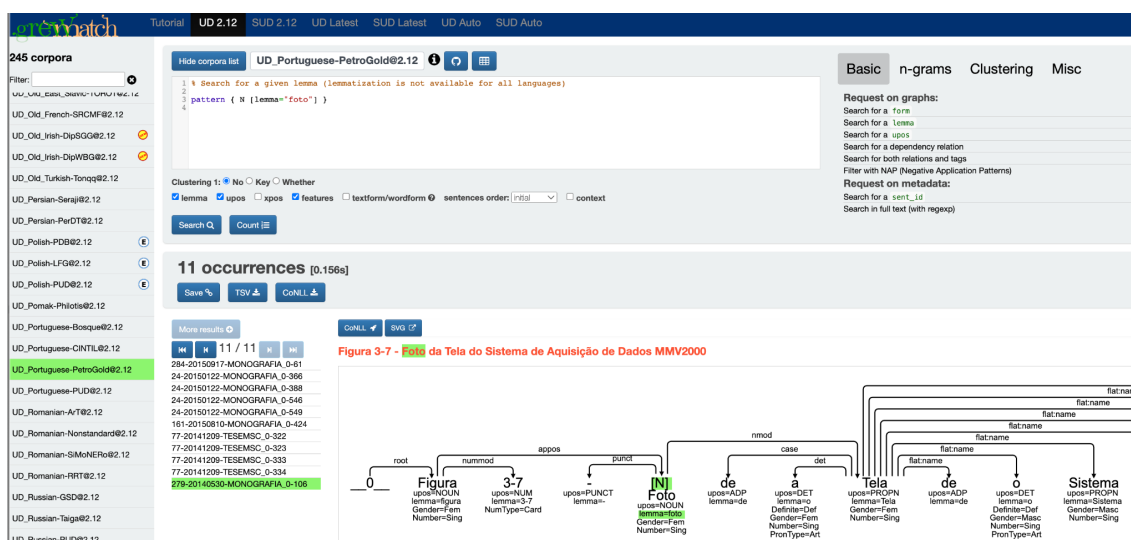


Figura 7.15: Captura de tela da ferramenta Grew Match.



do Department of Computer Science da Technische Universität Darmstadt (Klie et al., 2018).

INCEpTION é apresentada como um ambiente computacional e plataforma de anotação semântica. É uma aplicação web que permite que vários anotadores trabalhem num mesmo projeto, sendo que a instalação é feita na máquina local do anotador, que utiliza a ferramenta por meio de um arquivo executável java e um *localhost*. Possui esquemas prontos de anotação de PoS e relações de dependência e permite anotar várias sentenças e as relações de correferência e outras relações entre elas, como ilustrado na Figura 7.17.

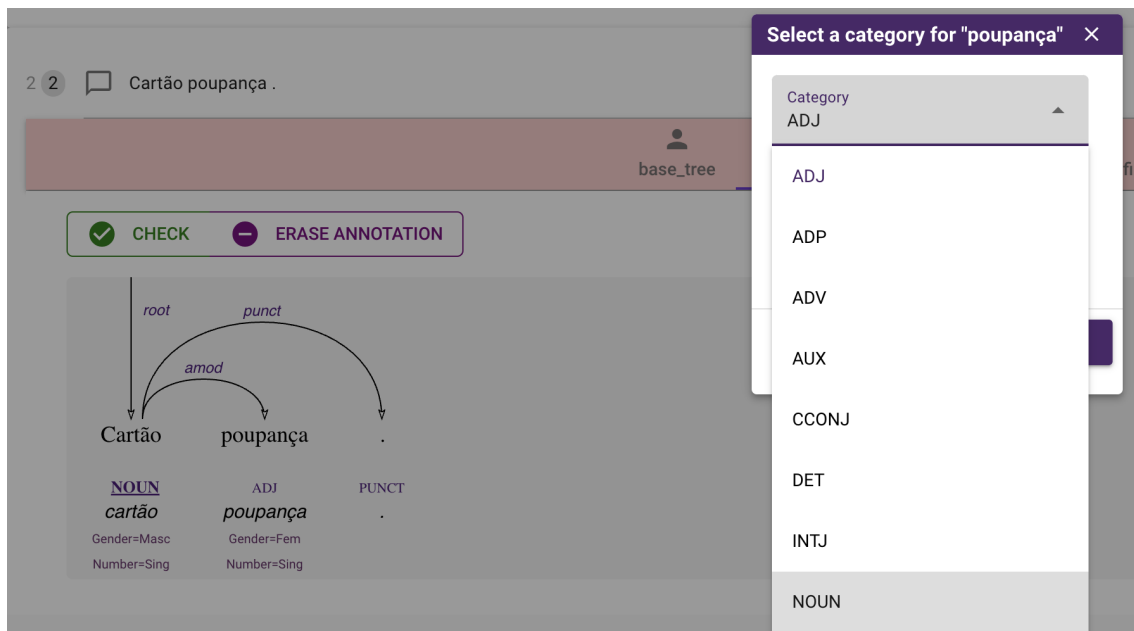
A Figura 7.17 mostra uma captura de tela com anotação em múltiplos níveis na ferramenta.

7.5 Considerações Finais

No cenário de PLN, a análise sintática desempenha um papel importante na compreensão e interpretação de textos. Como vimos, várias bibliotecas e ferramentas foram propostas para trabalhar com a língua portuguesa, oferecendo soluções para a análise sintática de

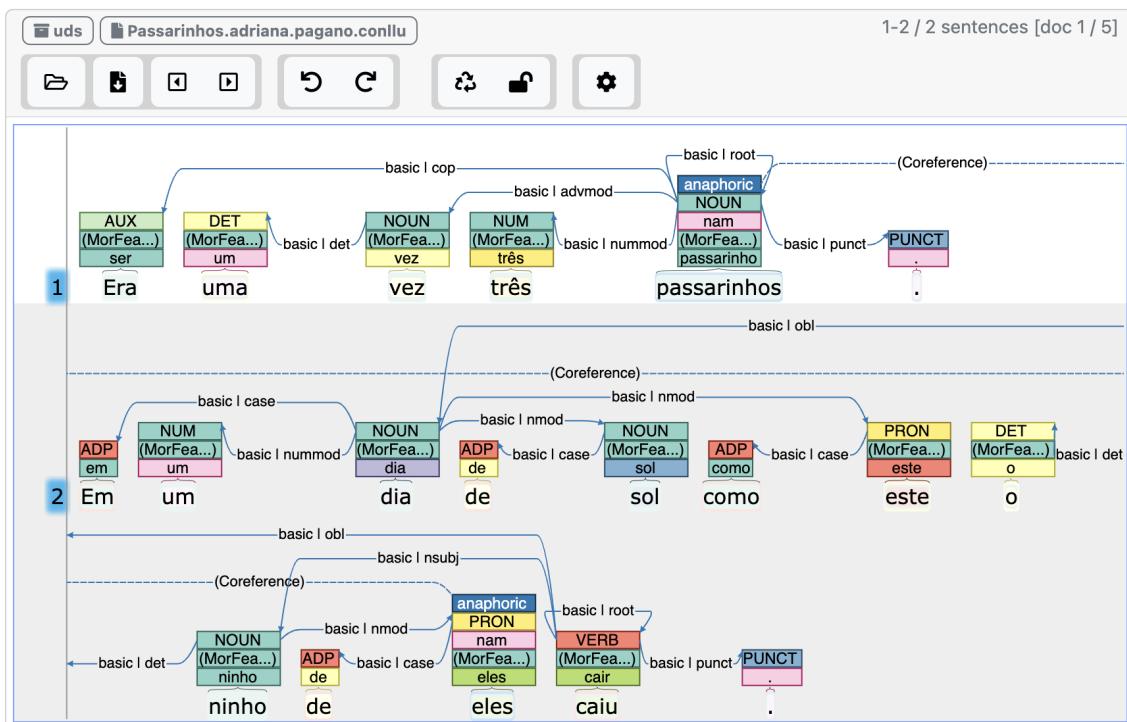


Figura 7.16: Captura de tela no momento de edição de etiqueta de PoS na ferramenta Arborator-Grew-NILC.



sentenças. A interseção entre linguística computacional e programação torna a análise sintática acessível mesmo para aqueles que não são especialistas em PLN, abrindo portas para uma compreensão mais profunda dos textos em língua portuguesa e sua estrutura intrínseca.

Figura 7.17: Captura de tela com anotação em múltiplos níveis na ferramenta INCEpTION.



Parte V

Significado



Capítulo 8

E o significado?

Cláudia Freitas

Semântica lida com o sentido do que é comunicado por meio da linguagem (em oposição ao que é comunicado por imagens ou sons não verbais, por exemplo). Assim, a **semântica estuda o significado de palavras e frases**. Mas a simplicidade relativa ao que é semântica acaba aí. Nos estudos linguísticos, a semântica é conhecida como “um domínio de investigação de limites movediços” e para o qual não há jargões bem estabelecidos (Ilari; Geraldi, 1985, p. 6). A questão “o que é o significado de uma palavra?” (e também o de uma frase) é um dos problemas nucleares da investigação semântica, e sua resposta irá depender da perspectiva teórica adotada. Essa característica é diferente de outras áreas do conhecimento, como a zoologia, por exemplo, em que não há controvérsia sobre o que é um animal.

No PLN, diferentes maneiras de conceber o significado se manifestam em diferentes abordagens para o tratamento do sentido, como veremos nos Capítulos 9 e 10.

Mas... sabemos o que é significado e os dicionários – objetos que contêm o **significado** das palavras – não só existem, como são úteis. Então, por que tanta dificuldade? Por que “limites movediços”?

Para ilustrar essa ideia, vamos fazer uma analogia entre a observação – e descrição – do significado de uma palavra e a observação de uma onda, como narrada em “Palomar na praia”, de um livro de Ítalo Calvino. Trata-se de um capítulo curto, e alguns trechos são transcritos abaixo. A história gira em torno de alguém – Palomar – tentando conhecer algo – uma onda – de forma maximamente objetiva. Palomar, aliás, além de nome do protagonista, é o nome de um observatório astronômico que, durante muito tempo, ostentou o maior telescópio do mundo¹. Vamos à história, apresentada no Quadro 8.1.

Quadro 8.1. Trechos da história “Palomar na praia” de Ítalo Calvino.

O senhor Palomar está de pé na areia e observa uma onda. Não que esteja absorto na contemplação das ondas. Não está absorto, porque sabe bem o que faz: quer observar uma onda e a observa. (...). Em suma, não são “as ondas” que ele pretende observar, mas uma simples onda e pronto: no intuito de evitar as sensações vagas, ele predetermina para cada um de seus atos um objetivo limitado e preciso.

O senhor Palomar vê uma onda apontar na distância, crescer, aproximar-se, mudar de forma e de cor, revolver-se sobre si mesma, quebrar-se, desfazer-se. A essa altura poderia convencer-se de ter levado a cabo a operação a que se havia proposto e ir-se embora. Contudo, isolar uma onda da que se lhe segue de imediato e que parece às vezes suplantá-la ou acrescentar-se a ela e mesmo arrastá-la é algo muito difícil, assim como separá-la da onda que a precede e que parece empurrá-la em direção à praia, quando não dá até mesmo a impressão de voltar-se

¹Devo à Helena Franco Martins a apresentação deste texto como alegoria tanto para as tentativas de apreensão do significado como para a crise relativa ao conhecimento/ciência.



contra ela como se quisesse fechá-la. (...).

Em suma, não se pode observar uma onda sem levar em conta os aspectos complexos que concorrem para formá-la e aqueles também complexos a que essa dá ensejo. Tais aspectos variam continuamente, decorrendo daí que cada onda é diferente de outra onda; mas da mesma maneira é verdade que cada onda é igual a outra onda, mesmo quando não imediatamente contígua ou sucessiva; enfim, são formas e sequências que se repetem, ainda que distribuídas de modo irregular no espaço e no tempo. Como o que o senhor Palomar pretende fazer neste momento é simplesmente ver uma onda, ou seja, colher todos os seus componentes simultâneos sem descurar de nenhum, seu olhar se irá deter sobre o movimento da água que bate na praia a fim de poder registrar os aspectos que a princípio não havia captado (...).

(...) Foi uma dessas línguas baixas de areia que o senhor Palomar escolheu como ponto de observação, porque as ondas nelas batem obliquamente de uma parte e de outra, e ao cavalgarem por cima da superfície semi-submersa vão encontrar-se com as que chegam da outra parte. (...).

O senhor Palomar está procurando agora limitar seu campo de observação; se tem presente um quadrado de, digamos, dez metros de praia por dez metros de mar, pode levantar um inventário de todos os movimentos de ondas que ali se repetem com frequência variada dentro de um dado intervalo de tempo. A dificuldade está em fixar os limites desse quadrado, porque, por exemplo, se ele considera como o lado mais distante de si a linha em relevo de uma onda que avança, essa linha ao aproximar-se dele irá, erguendo-se, ocultar de sua vista tudo o que está atrás; e eis que o espaço tomado para exame se destaca e ao mesmo tempo se comprime. (...)

Contudo, o senhor Palomar não perde o ânimo e a cada momento acredita haver conseguido observar tudo o que poderia ver de seu ponto de observação, mas sempre ocorre alguma coisa que não tinha levado em conta. Prestar atenção em um aspecto faz com que este salte para o primeiro plano, invadindo o quadro, como em certos desenhos diante dos quais basta fecharmos os olhos e ao reabri-los a perspectiva já mudou. (...).

O vento estaria mudando? É pena que a imagem que o senhor Palomar havia conseguido organizar com tanta minúcia agora se desfigure, se fragmente e se perca. Só conseguindo manter presentes todos os aspectos juntos, ele poderia iniciar a segunda fase da operação: estender esse conhecimento a todo o universo.

Bastaria não perder a paciência, coisa que não tarda a acontecer. O senhor Palomar afasta-se ao longo da praia, com os nervos tensos como havia chegado e ainda mais inseguro de tudo. CALVINO, Ítalo. *Palomar*. São Paulo: Companhia das Letras, 1994. p.7-11.

O que vemos, por trás da simplória tarefa de observação de **uma única onda**, é a dificuldade de Palomar diante de um objeto que se transforma continuamente durante a própria atividade de observação. Ainda que Palomar defina, de modo preciso, seu objetivo e seu objeto (“observar uma simples onda e pronto”) e busque uma abrangência descritiva (“Colher todos os seus componentes simultâneos sem descurar de nenhum”), é difícil, na observação, isolar o objeto de suas “adjacências”, reduzir as diferentes instâncias do objeto a uma essência comum (“sempre ocorre alguma coisa que não tinha levado em conta”), controlar a subjetividade, suspender as pressões externas (“bastaria não perder a paciência”), encontrar o ponto de vista superior ou ideal (“Foi uma dessas línguas baixas de areia que o sr. Palomar escolheu como ponto de observação”). Enfim, definitivamente, Palomar não é bem sucedido em sua empreitada, por mais simples que esta parecesse inicialmente.

De volta à semântica, podemos nos imaginar como Palomar na tentativa de observar o significado de uma única palavra. Podemos escolher a palavra “quente”, e teremos uma “sopa quente”, um “dia quente” e uma “cerveja quente”. Estamos tratando da mesma temperatura, do mesmo significado, o que há em comum em todos eles, e que



os define? Podemos escolher outra palavra, “medo”, em “medo de altura”, “medo de perder o emprego”, “medo de barata”, “medo do mar”, “medo de sofrer”. Estamos falando exatamente do mesmo “medo”? Qual o significado exato, preciso, de “medo”? Se estamos diante de uma mesma palavra, e de uma palavra que sabemos usar, não seria esperado que soubéssemos definir, de forma clara e precisa, seu significado? Qual o significado (ou significados) de “tomar”, tomando como exemplos combinações como “tomar um susto”, “tomar um porre”, “tomar cuidado”, “tomar um suco”, “tomar remédios”, “tomar uma decisão”, “tomar conta”, “tomar um tombo”, “tomar ciência”, “tomar porrada”, “tomar dois banhos”, “tomar um susto” etc.

É exatamente este tipo de dificuldade que justifica a existência, nos estudos linguísticos, de duas grandes perspectivas que irão problematizar o que é o significado. São perspectivas concorrentes, e de um modo bastante simplificado podemos chamá-las de **representacionistas ou essencialistas**, por um lado, e de **pragmáticas** (ou, **pragmáticas radicais**), por outro (Martins, 2000, 2004). No PLN, estas visões se manifestam em diferentes maneiras de lidar com o significado: usando técnicas simbólicas (veja Capítulo 9) ou usando representações distribuídas (veja Capítulo 10), respectivamente.

A perspectiva **representacionista/essencialista** é a visão hegemônica, estando presente em boa parte dos estudos linguísticos e no senso comum – e, até recentemente, em boa parte do PLN também.

Nesta visão, palavras seriam como “substitutos” de **entidades extralinguísticas**, entidades externas à linguagem (entidades mentais, reais ou virtuais). As palavras, nessa perspectiva, importam pouco, importando mesmo as ideias (as entidades extralinguísticas) que elas **representam**. Significado e palavra são, assim, entidades distintas, ainda que relacionadas (falamos “do significado das palavras”, por exemplo), e a relação entre elas é hierárquica, com a entidade significado se sobrepondo à entidade palavra (ou à palavra e seus sinônimos), que apenas fornece matéria/forma para “hospedar” o significado.

Ainda de acordo com esta visão, apesar da multiplicidade de usos e contextos que podem existir associados a uma mesma palavra (por exemplo “tomar”, ou “quente”, ou ainda “liberdade”, “violência”, “aprender”, “significado”, “compreensão”) a comunicação é possível porque esses diferentes usos estão **associados** a uma **essência comum** (a entidade extralinguística), e por isso reconhecemos a palavra como sendo a mesma em diferentes situações. A associação entre a palavra (ou a palavra e seus sinônimos) e sua essência/ideia/conteúdo/conceito/significado, por sua vez, é guiada por regras. **Aprender uma língua**, aqui, é aprender a estabelecer a conexão entre a palavra e a entidade extralinguística que ela representa (e diferentes línguas irão variar quanto às palavras usadas para representar estes conceitos/ideias). Este conceito/ideia/significado, que algo é separado da palavra, é um “objeto” extralinguístico (do mundo mental, real, virtual) estável e com contornos bem definidos – mas que, por sua vez, também será descrito por meio de palavras.

Podemos agrupar sob esta visão – apresentada aqui de maneira muito simplificada – uma série de correntes teóricas que, de alguma maneira, compartilham a ideia de que a **estabilidade do significado** (e a compreensão) é o **resultado da representação de algo que lhe é exterior**.

Ainda segundo esta visão, os significados das palavras são, de certo modo, o que o dicionário diz. O fato de dicionários representarem os significados de maneira objetiva, estável e discreta (vemos isso na maneira pela qual as acepções estão claramente separadas e numeradas), faz parecer que os significados das palavras se organizam “naturalmente”



assim². No PLN, reconhecemos esta maneira de lidar com o significado em recursos como *wordnets*, por exemplo, que são bases de dados lexicais que contêm “nomes, verbos, adjetivos e advérbios agrupados em conjuntos de sinônimos cognitivos, **cada um representando um conceito distinto**”³ (grifo meu). Uma apresentação do que se pode fazer partindo desse ponto de vista, e de por que ele continua tendo espaço no PLN, está no Capítulo 9.

Já do ponto de vista pragmático (ou, mais precisamente, pragmático **radical**)⁴, – e fazendo igualmente uma apresentação bastante simplificada – o significado de uma palavra é decorrência de situações concretas (e não o correspondente a uma entidade extralinguística), e situações concretas são variáveis. Nesta visão, os vários usos de uma palavra não se organizam em torno de um núcleo semântico comum (a entidade extralinguística), garantidor da estabilidade do que elas significam. A estabilidade do significado será sempre **provisória**, e o significado dependerá do uso, do contexto, do tempo, do espaço, de quem fala ... A **comunicação se dá no risco** (isto é, pode dar certo ou não, podemos nos entender ou não), e os mal-entendidos existem, estão aí – não são um desvio ou uma falha, são parte do jogo. O que determina se **compreendemos o significado** de um enunciado linguístico é o fato de **a manifestação dessa compreensão** (um comportamento) **ser considerada adequada** no contexto em que é produzida. Por exemplo, a um enunciado como “Está quente aqui”, seriam manifestações legítimas ações como abrir a janela ou respostas como “Não acho” ou “Por que não tira o casaco?”, entre outras. Mas dificilmente aceitaríamos como manifestações de compreensão do enunciado “Está quente aqui” dar uma cambalhota ou uma resposta como “Prefiro melão”⁵. **Aprender uma língua**, aqui, é aprender a tomar parte nas atividades humanas, um aprendizado que nunca se completa.

Assim como uma onda, os limites do significado de uma palavra não têm – aliás, podem não ter, pois não se trata de uma exigência – a precisão ou os limites definidos, necessários à formalização que sempre se buscou fazer. Segundo esta visão, o significado é flexível e maleável, não havendo uma “essência”, algo que perpassa todos os usos, e sobre o qual seja possível se sustentar, se estabilizar. No PLN, esta visão se alinha às representações distribuídas (veja Capítulo 10)⁶.

Sabemos que uma mesma palavra pode aparecer em contextos diferentes – desde contextos completamente distintos, como “banco” e “manga”, até contextos ligeiramente diferentes, como os exemplos de “quente”, “medo” ou “tomar”, que já vimos⁷. Nesse caso, e considerando os modelos de representações distribuídas mais complexos e dinâmicos, cada forma “quente” ou “medo” será representada de uma maneira – e por isso nesses casos falamos de vetores contextuais (*contextual word embeddings*). Nos vetores estáticos, que irão representar de uma única maneira as várias formas “quente” ou “medo”, o alinhamento à visão não-representacionista/não essencialista se mantém, uma vez que não há uma fonte (ou entidade) externa que determina o significado da palavra. Vetores produzidos a partir

²Por trás de dicionários estão lexicógrafos e decisões editoriais.

³<http://wordnet.princeton.edu/>

⁴Dentre as linhas de investigação pragmáticas há as que poderiam ser também enquadradas em um paradigma representacionista. Isto porque algumas correntes da pragmática recomendam a análise das propriedades da prática da comunicação como maneira de fornecer uma explicação do que são as línguas e os significados. Por isso a especificação indicando a “radicalidade” da visão que será apresentada.

⁵Mas mesmo estas poderiam ser aceitas se assim fosse previamente estipulado.

⁶Mas não se alinha à busca do algoritmo capaz de fornecer a representação distribuída “correta”, ou “verdadeira”.

⁷É importante notar que, diferentemente do que supõe o senso comum, os casos de “banco” ou “manga”, apesar de fartamente citados como exemplos de ambiguidade, estão longe de ser prototípicos. Pelo contrário, são raros os casos em que dois sentidos se apresentam tão claramente distintos. O mais comum são casos como “quente” ou “medo”.



de conjuntos de dados diferentes irão levar a representações diferentes⁸.

Durante muito tempo, a semântica computacional esteve ancorada em visões essencialistas-representacionistas (ou simbólicas), como ilustram os capítulos “*Semantics*” de dois compêndios da área de PLN: Jurafsky; Martin (2023) e Mitkov (2003). No entanto, trabalhos de PLN que dialogam claramente com perspectivas não-essencialistas também não são novidade, como Kilgarriff (1997); Kilgarriff (2003) e Brewster; Wilks (2004), por exemplo.

Entre as técnicas simbólicas e as representações distribuídas existem ainda os *datasets* (ou *corpora*) com **anotação semântica** (veja Capítulo 14), uma terceira maneira de lidar com o significado no PLN. Se, por um lado, tais *datasets* se alinham às abordagens probabilísticas, uma vez que podem ser usados como fonte para o aprendizado de máquina (para o **aprendizado do significado**), por outro lado, a atividade de anotação de significado se alinha às abordagens representacionistas. Neste tipo de anotação (também chamada de anotação de *word senses*), cada palavra (ou segmento de texto) é anotada com informação relativa ao significado de acordo com o contexto específico em que aparece no *corpus*. A informação relativa ao significado, por sua vez, vem de **fontes externas** (como dicionários, *wordnets*, *verbnets* e *framenets*) e a tarefa de anotação pode ser descrita como um **trabalho** de desambiguação, pois consistiria em selecionar, dentre os vários sentidos possíveis de uma palavra, aquele usado no contexto da frase. O que a anotação faz, deste modo, é criar uma **representação estável** entre a palavra e o seu significado, no contexto em que está sendo usada. Cada ocorrência de uma palavra poderá estar associada a um significado diferente (e aqui vemos uma aproximação com abordagens pragmáticas), desde que este significado esteja presente no inventário de significados usado na anotação (aqui vemos uma aproximação com abordagens representacionistas). Para as pessoas responsáveis pela anotação, a principal dificuldade está na escolha do sentido adequado conforme o contexto, uma vez que os sentidos frequentemente se sobrepõem, como as ondas observadas por Palomar.

Por exemplo, tomando a palavra “trabalho” destacada no parágrafo anterior, a tarefa consiste em escolher, dentre opções listadas no quadro abaixo, retiradas do dicionário Caldas-Aulete online⁹, aquela adequada ao contexto (se a anotação usasse o inventário de uma *wordnet* como fonte, o inventário de significados poderia ser diferente¹⁰).

Quadro 8.2. Acepções da palavra *trabalho* conforme dicionário

1. Emprego da força física ou intelectual para realizar alguma coisa
2. Aplicação dessas forças como ocupação profissional: *Seu trabalho é de gari.*
3. Local onde isso se realiza: *Mora longe do trabalho.*
4. Esmero, cuidado que se emprega na confecção ou elaboração de uma obra
5. A confecção, elaboração ou composição de uma obra
6. Obra realizada: *Essa cômoda é um belo trabalho de marcenaria.*
7. Grande esforço; TRABALHÃO; TRABALHEIRA
8. Exercício para treino: *A professora passou muito trabalho para casa.*
9. Ação contínua de uma força da natureza e seu efeito: *O trabalho do vento resulta na erosão eólica.*

⁸Mas mesmo representações distribuídas podem ser associadas a visões representacionistas, quando se assume que tais representações são úteis apenas enquanto não encontramos a forma (ou a representação) correta de uma palavra.

⁹<https://www.aulete.com.br/trabalho>

¹⁰Aqui é possível consultar a OpenWordNet-PT para os significados de “trabalho”: https://www.openwordnet-pt.org/search?search_field=all&term=trabalho



10. Med. Fenômeno orgânico que se opera no interior dos tecidos (trabalho inflamatório; trabalho de cicatrização)
11. Resultado do funcionamento de uma máquina, um aparelho etc.: o trabalho de uma pá mecânica.
12. Obrigação ou responsabilidade; DEVER; ENCARGO: *Seu trabalho é protegê-lo do assédio da imprensa.*
13. Econ. Conjunto das atividades humanas empregado na produção de bens: *O capital e o trabalho são os pilares da economia.*
14. Tarefa a ser realizada: *Contratou-o para um trabalho temporário.*

Fonte: (Freitas, 2022)

Parece que as acepções 1, 2 e 14 são aceitáveis no contexto da frase, o que já é um problema se precisamos escolher apenas um sentido, e por isso não é exagero dizer que as pessoas responsáveis pela anotação se sentem como Palomar na tentativa de isolar uma onda.

Corpora anotados com este tipo de informação são escassos, e um dos motivos é justamente a dificuldade de isolar o significado/conteúdo/essência das palavras enquanto estão sendo efetivamente usadas.

O estudo de Baker; Fellbaum; Passonneau (2017) tentou entender por que, com este tipo de anotação, era tão difícil conseguir uma boa concordância entre anotadores (veja Capítulo 14), isto é, era tão difícil que as pessoas concordassem quanto à escolha do significado utilizado. Afinal, a tarefa é simples: associar cada palavra ao seu significado, e, se sabemos a nossa língua, sabemos o significado das palavras que usamos. No estudo, diferentes pessoas deveriam anotar as mesmas palavras, nas mesmas frases, considerando o mesmo inventário de sentidos. Os resultados indicaram uma variação bem maior que o previsto¹¹. Vamos lembrar que, na **anotação**, os significados precisam ser vistos como **unidades discretas e de conteúdo estável** – uma necessidade de ordem prática que se alinha harmoniosamente com visões representacionistas, mas que não encontra respaldo em visões pragmáticas.

O fato de representações distribuídas terem levado a resultados positivos no PLN não deve ser visto como argumento contrário às técnicas simbólicas. São maneiras diferentes de lidar com o sentido das palavras. Como tirar o melhor proveito destas diferentes visões e abordagens, no PLN, é uma das questões que se coloca. O que temos visto é a limitação de cada uma delas, tomada individualmente. Se consideramos o significado como uma entidade estável, como lidar com as mudanças, que inclusive podem ser capturadas pelos dicionários (dicionários, recentemente, mudaram a definição da palavra “família”¹²)? Por outro lado, se consideramos a instabilidade e a dependência dos dados, como evitar vieses indesejados, como a associação entre os sentidos, por exemplo, de “paraguai” e “de baixa qualidade”, quando dizemos “uísque paraguai”?

Os próximos capítulos aprofundam cada uma dessas maneiras de trabalhar com o significado no PLN.

¹¹A seção “Anotações Semânticas” de Freitas (2022) traz um levantamento dos principais estudos sobre anotação semântica e seus desafios, bem como uma apresentação linguística da alternativa oferecida pelas representações distribuídas para o tratamento do significado.

¹²Dicionários mudam definição de família <https://www.metropoles.com/vida-e-estilo/comportamento/dicionario-houaiss-muda-significado-da-palavra-familia>



Capítulo 9

Semântica com Técnicas Simbólicas

Eloize Rossi Marques Seno
Valéria de Paiva
Vlândia Pinheiro

Métodos Simbólicos em Processamento de Linguagem Natural (PLN) envolvem a utilização de regras e representações formais explícitas para processar e entender textos em linguagem natural. Esses métodos especializam-se na manipulação de símbolos e dados estruturados, como gramáticas, ontologias e bases de conhecimento. Especificamente para o entendimento de textos em linguagem natural usando técnicas simbólicas, existem analisadores semânticos (ou *parsers* semânticos) e bases de conhecimento semântico, que visam fornecer uma representação semântica dos textos. A partir desta representação, motores de inferência são capazes de realizar raciocínio para que aplicações possam, por exemplo, extrair informações, sumarizar textos, e responder perguntas com base nos textos.

A Figura 9.1 apresenta uma arquitetura tradicional para sistemas de entendimento de textos em linguagem natural (*Natural Language Understanding* – NLU). A partir do texto de entrada, uma camada de processamento sintático realiza uma série de análises no texto, tais como detecção de língua, separação de sentenças, tokenização, análise morfológica e sintática (Capítulo 4). Na fronteira entre o processamento sintático e a análise semântica, outros processamentos linguísticos são necessários, como reconhecimento de entidades nomeadas, identificação de expressões multipalavras etc. Em seguida, o texto analisado (sintaticamente) é enviado ao analisador semântico (*parser*) que gera uma representação lógica do texto. A representação lógica e a(s) base(s) de conhecimento, no que lhes concerne, são entradas para o motor de inferência. Nesse processo, termos do texto de entrada são associados aos elementos da base de conhecimento e o motor de inferência gera respostas a perguntas (*queries*) para uma aplicação final.

É pertinente fazer uma observação neste ponto para uma definição de base de conhecimento. Uma “base de conhecimento” refere-se a um repositório centralizado, processável por máquina, que contém informações, dados, regras e procedimentos que são usados para capturar, representar e armazenar conhecimento geral ou de um domínio específico. Tais bases de conhecimento são fontes de conhecimento de mundo e suportam diversas tarefas e aplicações em PLN. Uma base de conhecimento pode ser estruturada de diversas maneiras, incluindo bancos de dados relacionais, linguagem para ontologias (e.g. a OWL¹), formalismos para troca de dados entre sistemas (e.g. o formato JSON²), redes semânticas ou sistemas baseados em regras, dependendo da sua finalidade e da natureza do conhecimento armazenado.

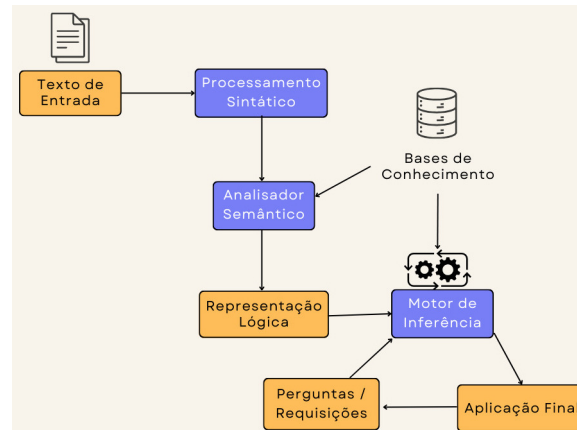
Tradicionalmente, sistemas lógicos são usados para representação formal dos textos e seus motores de inferência servem para gerar conclusões a partir dos textos. Podemos citar

¹<https://www.w3.org/OWL/>

²<https://www.json.org/json-pt.html>



Figura 9.1: Arquitetura de Sistemas de Entendimento de Linguagem Natural (NLU).



Fonte: Adaptada de (Ovchinnikova, 2012, p. 9)

os sistemas lógicos mais usados em PLN: variações de Lógica Descritiva (*Description Logic – DL*) (Baader et al., 2003), da Lógica de Primeira Ordem (Blackburn; Bos, 2005; Eijck; Unger, 2010), vários tipos de Programação em Lógica (PROLOG) (Dahl, 1994) e Lógicas Intensionais (Shapiro, 2000).

Uma característica importante dos sistemas lógicos usados para a semântica de linguagem natural é que eles dependem fortemente da forma lógica do texto ou argumento. No entanto, muitas conclusões e respostas fornecidas ao se ler um texto são justificadas pela contribuição semântica dos conceitos relacionados, não definida a priori, mas somente enquanto usados em um contexto particular. Por exemplo, considere a inferência que conclui que “Alguém foi assassinado” a partir da premissa que “Alguém foi executado”. A contribuição semântica do conceito “executar” (no sentido de “assassinar”) é que torna esta inferência plausível, e não a forma da sentença. Da mesma forma, a inferência “um relâmpago é visto agora” para “um trovão será ouvido em breve” é autorizada pelo conteúdo dos termos “trovão” e “relâmpago”. Para se realizar inferências desta natureza, alguns filósofos como Sellars (Sellars, 1953) e Brandom (Brandom, 2001) propõem abordagens para expressão do significado que suportam análises semânticas não somente sobre a forma das sentenças, mas são capazes de, com base no domínio dos conteúdos dos termos articulados nas sentenças e textos, descobrir como estes [os conteúdos dos termos] contribuem conjuntamente para o significado das sentenças e para realização de inferências.

Este capítulo tem como objetivo examinar, além dos *frameworks* semânticos, tais como AMR (*Abstract Meaning Representation*) (Banarescu et al., 2013) e DELPH-IN (Copestake et al., 2005), os tipos de bases de conhecimento mais utilizados em PLN. Nesta versão inicial do capítulo, são apresentadas as bases (também chamadas de recursos léxico-semânticos) WordNet de Princeton (Fellbaum, 1998) e FrameNet (Baker; Fillmore; Lowe, 1998), e suas versões em português: OpenWordNet-PT (De Paiva; Rademaker; Melo, 2012) e FrameNet Brasil (FN-BR) (Torrent; Ellsworth, 2013); bem como bases de conhecimento voltadas ao senso comum, tais como a ConceptNet (Speer; Chin; Havasi, 2016) e iniciativas para o português (OMCS-BR (Anacleto et al., 2006) e a InferenceNet-BR (Pinheiro et al., 2010)). Existem outros tipos de bases de conhecimento na área do PLN, tais como dicionários e ontologias diversas, por exemplo, WikiData (Vrandečić; Krötzsch, 2014), YAGO (Suchanek; Kasneci; Weikum, 2007) ou BabelNet (Navigli; Ponzetto, 2012). No entanto, nossa descrição aqui visa apenas a uma primeira exposição dos distintos paradigmas de expressão de



conhecimento semântico. Tendo em vista cada base de conhecimento descrita, analisamos um exemplo de texto motivador. Por fim, apresentamos as considerações finais deste capítulo.

9.1 Bases de Conhecimento Semântico

Na área da Inteligência Artificial (IA), o interesse por bases de conhecimento computáveis ou processáveis por máquina surgiu na década de 60 com as primeiras redes semânticas e representações baseadas em *frames*, propostas por Minsky (Minsky, 1975) e Fillmore (Fillmore et al., 1976), respectivamente.

A comunidade de PLN foi rapidamente atraída por tais representações de conhecimento de mundo, pois pareciam prover a solução para problemas de semântica de linguagem natural. As mais antigas abordagens em PLN que utilizaram redes semânticas e *frames* remontam aos trabalhos de Bates et al. (1982) e Bobrow et al. (1977), conforme citado em (Ovchinnikova, 2012).

Neste capítulo, examinaremos dois tipos de bases de conhecimento: (1) recursos léxico-semânticos e (2) bases de conhecimento de senso comum.

1. Recursos Léxico-Semânticos

Um conjunto de palavras existentes em uma determinada língua é chamado de léxico da língua e cada elemento do léxico é chamado de item lexical (Capítulo 4). Estes itens, quando organizados e agrupados de forma a facilitar o uso em processos computacionais, formam uma **base de conhecimento lexical** ou um **recurso léxico-semântico**. Uma *wordnet*³ é um exemplo canônico desse tipo de base, onde a organização dos itens lexicais se dá através de relações semânticas, predominantemente, de hierarquia (hiperonímia/hiponímia), de inclusão (holonímia/meronímia), de equivalência (sinonímia) ou de oposição (antonímia). Essas bases incluem normalmente informação sobre os possíveis sentidos das palavras (por exemplo, um sentido de “manga” é a fruta tropical, mas “manga” também pode ter o sentido de parte de uma camisa), as relações entre sentidos (carro e pneu como merônimo-holônimo; quente e frio como antônimos), e definições e frases que exemplificam a sua utilização. Os recursos da família das *wordnets* são bases muito usadas em PLN e com uma história de sucesso dada a cobertura, variedade de relações e organização do conteúdo, além da facilidade de incorporação em aplicações e ferramentas que precisam entender textos em linguagem natural, através de *toolkits* como o NLTK⁴ e spaCy⁵. Dentre as *wordnets*, temos a original e a mais proeminente – a WordNet de Princeton ou PWN⁶ (Fellbaum, 1998).

2. Bases de Conhecimento de Senso Comum

Para a comunidade de Inteligência Artificial, a expressão “conhecimento de senso comum” se refere aos fatos e conhecimentos informais possuídos pela maioria das pessoas, frutos da experiência da vida diária e baseados na generalização de eventos ou interpretações particulares, sem comprovação formal. Consiste em conhecimentos espaciais, físicos, sociais, temporais e psicológicos (Liu; Singh, 2004). As bases de conhecimento de senso comum expressam relações semânticas entre fragmentos de

³<http://globalwordnet.org/>

⁴<https://www.nltk.org/>

⁵<https://spacy.io/universe>

⁶<http://wordnet.princeton.edu/>



textos, tais como: relações funcionais, causais, afetivas, temporais, motivacionais, estruturais etc. Por exemplo, “bicicleta” é usada para “andar mais rápido que a pé”, ou “cozinhar” é motivada por “fome”. Nessa classe, se enquadram a larga base de senso comum ConceptNet⁷ (Speer; Chin; Havasi, 2016), originalmente gerada de conteúdo coletado de forma colaborativa na internet, e suas variações e congêneres.

Antes de iniciar a descrição das bases de conhecimento, introduziremos um exemplo de texto, Exemplo 9.1⁸, em português brasileiro, para ser analisado conforme os insumos de cada base de conhecimento. Após o exemplo, indicamos algumas conclusões e respostas resultantes de inferências que pessoas, inseridas na cultura brasileira e proficientes no português, fariam ao ler o texto.

Exemplo 9.1.

Um assalto do tipo saidinha bancária, ocorrido na tarde desta terça-feira, terminou com uma mulher de 42 anos baleada pelos assaltantes. O roubo ocorreu na rua Professor Costa Mendes.

Conclusões:

- O assalto teve uso de arma de fogo.
- A vítima estava em uma agência bancária.
- A motivação do crime foi conseguir dinheiro.

Nas próximas subseções, são descritas algumas das bases de conhecimento mais representativas para o PLN – Wordnet, FrameNet e a ConceptNet, e suas bases congêneres para o português. No final de cada subseção, discorremos sobre como essas bases contribuem para análise semântica do Exemplo 9.1. A escolha dessas três bases seguiu critérios de abrangência de suas entradas e representatividade para tarefas de PLN. A WordNet de Princeton é, consensualmente, o recurso léxico-semântico mais utilizado em PLN para dar suporte a tarefas como desambiguação de sentido de palavras, perguntas e respostas, e análise semântica. FrameNet é uma das bases mais relevantes para a tarefa de anotação de papéis semânticos (*Semantic Role Labeling – SRL*), pois atribui papéis semânticos não somente a verbos, mas também a termos das demais classes gramaticais. ConceptNet é a base de conhecimento de senso comum com mais entradas tanto para o inglês quanto para o português.

9.1.1 Wordnets

WordNet, desenvolvida por George A. Miller, Christiane Fellbaum e colaboradores, é considerada uma base de conhecimento léxico-semântica que organiza os itens lexicais (palavras ou expressões) em *synsets* (que vem de *synonym sets*, ou conjuntos de palavras sinônimas). A primeira *wordnet* foi desenvolvida para o inglês por George Miller, na Universidade de Princeton, um projeto que se iniciou em 1985, e é ordinariamente chamada de WordNet de Princeton (ou Princeton WordNet, na sigla PWN)⁹ e é descrita por Fellbaum (1998).

⁷<https://conceptnet.io/>

⁸Esse texto foi adaptado de notícia publicada em jornal digital, disponível em <https://diariodonordeste.verdesmares.com.br/seguranca/mulher-e-baleada-em-saidinha-bancaria-no-montese-1.933137?page=10>.

⁹<http://wordnet.princeton.edu/>



Wordnets são redes de palavras amplamente utilizadas em PLN para dar suporte a tarefas como desambiguação de sentido de palavras, perguntas e respostas, e análise semântica em geral. A unidade básica da WordNet são os *synsets* que representam conjuntos de palavras sinônimas. Cada *synset* expressa um conceito em particular. Os *synsets* têm uma glosa, semelhante a uma definição num dicionário e podem conter ainda frases que ilustram o emprego de alguma das suas palavras. A WordNet está dividida em quatro redes semânticas, uma para cada classe aberta de palavras: substantivo, verbo, adjetivo e advérbio.

Como exemplo, a Figura 9.2 apresenta os *synsets* da palavra “murder” (verbo “assassinar”, em português) da PWN. Ao todo são três *synsets*, um na classe Noun (substantivo) e dois na classe Verb (verbo). O primeiro *synset* da palavra “murder” (na classe Verb) tem como tropônimos diretos os verbos “burke”, “execute” e hiperônimo direto o *synset* “kill”.

Figura 9.2: Synsets da palavra “murder” na WordNet de Princeton.

The screenshot shows the WordNet Search interface. At the top, it says "WordNet Search - 3.1" with links to "WordNet home page", "Glossary", and "Help". Below this is a search bar with "murder" entered and a "Search WordNet" button. There are also "Display Options" and a "Change" button. A key explains that "S:" shows synset relations and "W:" shows word relations. The results are categorized by part of speech: Noun and Verb. Under Noun, there is one entry for "murder" with its gloss. Under Verb, there are several entries, including "murder" with its gloss and a list of related terms like "burke", "execute", "kill", "mangle", and "mutilate".

A PWN é a base léxico-semântica mais utilizada em PLN, com interfaces locais (APIs) em alguns dos maiores sistemas de programação (.NET/C#, dBase, Java, MySQL, OCaml, OSX, Perl, PHP, Prolog, Python, REST, SQL, Windows, XML)¹⁰, mais de 20 mil citações

¹⁰<https://wordnet.princeton.edu/related-projects>



no Google Scholar e dezenas de projetos que a utilizam. Apesar de ser tão utilizada, a WordNet de Princeton parou de evoluir em 2012, por falta de recursos financeiros. A última edição oficial de PWN foi a versão 3.1, lançada em 2011. Em 2019 um consórcio de pesquisadores, incluindo Christiane Fellbaum (a coordenadora da PWN), resolveu transformar a PWN em um recurso moderno, hospedado em GitHub, de tal forma que possa ser sempre atualizado (McCrae et al., 2019), mas a maior parte das aplicações continua usando PWN 3.0 ou 3.1.

Como todas as bases com conhecimento semântico, WordNet não é um projeto acabado e tampouco completo. Algumas lacunas decorrem da divisão e independência entre as redes semânticas, o que dificulta a expressão de relações estruturais (entidade-atributo), de relações semânticas que acontecem entre classes de palavras (verbos e substantivos, por exemplo) em uma particular situação ou contexto; ou informações sintagmáticas: relações que ocorrem entre os termos de um proferimento (entre verbo e substantivo, entre substantivo e adjetivo etc.). No que se refere aos tipos de relações expressas na PWN, essa dispõe de relações causais entre *synsets*, por exemplo, “*snore*” implies “*sleep*”, mas não numa taxa de cobertura suficiente em relação ao conjunto dos *synsets*. Outra limitação é que recursos como a PWN são mais adequados para substantivos concretos do que para conceitos abstratos como “medo”, “felicidade” etc. Enquanto substantivos concretos como “gato”, “felino”, “mamífero”, “animal” etc. são mais facilmente organizados em taxonomias, tal processo é menos consensual quando aplicado às emoções ou a verbos. Um quarto criticismo diz respeito às expressões multipalavras (MWEs – Capítulo 5). Essas existem em PWN, mas não na quantidade suficiente para a modelagem adequada da língua. De acordo com Sag et al. (2002, p. 2), o número de MWEs em PWN precisaria ser maior do que é. Um quinto criticismo diz respeito ao nível de granularidade das distinções de significado na PWN. Essas distinções são muito refinadas, o que faz com que as medidas de concordância entre anotadores sejam baixas.

A partir da WordNet de Princeton, várias *wordnets* foram propostas para diversas línguas, entre elas o português, conforme será descrito na seção a seguir.

9.1.1.1 *Wordnets* para o português

Vários recursos léxico-semânticos foram criados para o português nos últimos anos. Alguns deles são listados na página da Linguateca¹¹. O NILC¹² tem uma coleção de recursos listados no portal PortLex¹³, entre os quais se encontram, entre outros, VerbNet.Br (Scarton; Aluisio, 2012) e PropBank.Br (Duran; Aluísio, 2012).

Há várias versões de *wordnets* para o português, como Wordnet.BR (Dias-da-Silva, 2005), Onto.PT (Gonçalo Oliveira, 2014), PULO (Simões; Guinovart, 2014) e OpenWordNet-PT¹⁴ (De Paiva; Rademaker; Melo, 2012). Essas *wordnets* são discutidas detalhadamente em (De Paiva et al., 2016; Gonçalo Oliveira, 2014), portanto, aqui simplesmente reiteramos a mensagem principal dessas comparações.

Apesar de existirem várias alternativas de *wordnets* para o português, todas são menores e menos desenvolvidas do que a PWN. PWN é um recurso relativamente grande com 16MB, incluindo 155.327 palavras organizadas em 175.979 *synsets* num total de 207.016 pares de palavra-significado. A OpenWordNet-PT (OWN-PT) (De Paiva; Rademaker; Melo, 2012),

¹¹<https://www.linguateca.pt/>

¹²<https://sites.google.com/view/nilc-usp/>

¹³<http://143.107.183.175:21380/portlex/index.php/en/>

¹⁴<https://www.openwordnet-pt.org/>



alinhada à PWN, conta com 47.702 *synsets* (somente 27% da PWN), dos quais 32.855 correspondem a substantivos, 5.060 a verbos, 8.753 a adjetivos e 1.034 a advérbios. O número de projetos usando OWN-PT é muito limitado, possivelmente porque, construída de forma semi-automática, usando aprendizado de máquina no conjunto de wikipédias multilinguais (Melo; Weikum, 2009) e manualmente melhorando os dados obtidos.

Como exemplo, a Figura 9.3 apresenta os *synsets* da palavra “assassinar” na OWN-PT. Ao todo são quatro *synsets*, um na classe *Noun* (substantivo) e três na classe Verbo. O terceiro *synset* (02482425-v) refere-se a “matar intencionalmente e com premeditação” (glosa) e possui como hiperônimo direto o *synset* “matar” (01323958-v).

Figura 9.3: *Synsets* da palavra “assassinar” na OpenWordNet-PT (OWN-PT).

1. [02483000-v](#) assassinate | **assassinar**
 - (murder; especially of socially prominent persons; "Anwar Sadat was assassinated because many people did not like t
2. [01325128-v](#) dispatch | **assassinar**
 - (kill without delay; "the traitor was dispatched by the conspirators")
3. [02482425-v](#) bump_off, slay, remove, off, polish_off, murder, hit, dispatch | **assassinar, matar, despachar**
 - (kill intentionally and with premeditation; "The mafia boss ordered his enemies murdered")
4. [00768701-n](#) felony | **crime**
 - (a serious crime (such as murder or arson))

1

Outras *wordnets* são ainda menores (PULO (Simões; Guinovart, 2014)), ou menos acuradas, pois, construídas numa abordagem mais dinâmica (ONTO.PT (Gonçalo Oliveira, 2014)), podem mudar completamente de uma versão para a seguinte.

Algumas decisões de projeto de uma *wordnet*, assim como de outras bases de conhecimento, parecem claras e já são consenso na comunidade do PLN. *Wordnets* devem ser recursos abertos, grátis e fáceis de utilizar. Devem ter versões adequadas a usuários humanos e a agentes computacionais, isto é, devem ter interfaces de busca para usuários e interfaces ou bibliotecas para usos computacionais. Tais recursos linguísticos precisam ser mantidos e melhorados, pois nenhum é perfeito e as linguagens naturais são sistemas vivos, dinâmicos e em constante e contínua evolução.

Porém, outras decisões permanecem em aberto: uma alternativa só para o português brasileiro e outra para o português de Portugal? Ou uma alternativa para ambas variantes do português? Alternativas multilinguais tais como Open Multilingual WordNet (OMW)¹⁵ (Bond; Foster, 2013) ou somente em português? Somente alternativas alinhadas a PWN ou o alinhamento¹⁶ não é necessário? Somente as relações semânticas de PWN ou outras também? As entidades nomeadas devem ser incluídas no recurso ou não? Qual deve ser o registro do recurso? Deve incluir gírias e palavras de baixo-calão ou não?

9.1.1.2 Análise do Exemplo Motivador usando a WordNet

Usaremos a OWN-PT para analisar o exemplo motivador Exemplo 9.1 apresentado no início da Seção 9.1. No Exemplo 9.2, foram sublinhadas algumas palavras que foram associadas a *synsets* na OWN-PT. Para realizar esta associação, é necessário definir o sentido ou significado da palavra usada no texto. Esta tarefa em PLN denominamos de

¹⁵<https://omwn.org/>

¹⁶O alinhamento entre *wordnets* refere-se ao processo de mapeamento ou ligação de *synsets* entre diferentes *wordnets* de línguas distintas. Por exemplo, o alinhamento entre PWN e OWN-PT seria o processo de identificar que o *synset* em inglês para “car” é equivalente ao *synset* em português para “carro”.



Desambiguação do Sentido de Palavras (*Word Sense Disambiguation* – WSD). Após o exemplo, são listadas algumas afirmações (especificamente de **hiperomínia**) entre o *synset* da palavra usada no texto e outro *synset*.

Exemplo 9.2.

Um assalto do tipo saidinha bancária, ocorrido na tarde desta terça-feira, terminou com uma mulher de 42 anos baleada pelos assaltantes. O roubo ocorreu na rua Professor Costa Mendes.

Usando a OWN-PT, definimos os seguintes *synsets* para as palavras sublinhadas em Exemplo 9.2: “assalto” (00783063-n), “terminar” (02610845-v). Não foi encontrado nenhum *synset* para o termo “balear”. A seguir, algumas afirmações de hiperonímia entre esses *synsets* e outros:

- “roubo” (00781685-n) é hiperônimo de “assalto”;
- “cessar” (02609764-v) é hiperônimo de “terminar”;

Algumas dificuldades com a análise do exemplo, à luz da OWN-PT, foram:

- A desambiguação não é simples nem para um ser humano proficiente na linguagem natural (no caso, o português) e experiente com anotação de sentidos em *wordnets*. A diversidade e granularidade de *synsets* encontrados para uma palavra dificulta a definição do significado. Por exemplo, para a palavra “assalto” tem-se 07 (sete) *synsets* e todos parecem adequados para definir o sentido da palavra no Exemplo 9.2;
- Para algumas palavras, não foram encontrados *synsets* na OWN-PT (e.g. “baleada”).

A partir da associação de uma palavra a um *synset*, um *parser* semântico pode, por exemplo, expandir o texto com tais informações semânticas, servindo como entrada para sistemas de entendimento de linguagem natural.

Como dito anteriormente, as bases não são sempre corretas e, definitivamente, não são completas. A língua muda, evolue o tempo todo e os significados das palavras seguem essa evolução. Nesse sentido, outros recursos léxico-semânticos são propostos e visam preencher lacunas na semântica das linguagens naturais. Na próxima subseção, detalharemos o recurso léxico-semântico FrameNet. Essa base se tornou relevante para a tarefa de Anotação de Papéis Semânticos (*Semantic Role Labeling* – SRL) pela abrangência e por incluir os papéis semânticos associados a substantivos e adjetivos.

9.1.2 FrameNet

FrameNet (Baker; Fillmore; Lowe, 1998), da Universidade de Berkeley¹⁷, é um recurso com conhecimento léxico e semântico baseado na semântica de *frames* (Fillmore et al., 1976) e na teoria de *frames* de (Minsky, 1975). Um *frame* é uma estrutura hierárquica conceitual que define uma situação, objeto ou evento por meio de seus participantes e relacionamentos. FrameNet faz parte da classe de recursos léxico-semânticos que suportam a tarefa de Anotação de Papéis Semânticos (*Semantic Role Labeling* - SRL), pois provê uma base de relações semânticas entre predicados e argumentos. Por exemplo, no evento de cometimento de crime, definido pelo *frame* `Committing_crime`, são definidas as seguintes

¹⁷<https://framenet.icsi.berkeley.edu/>



relações entre os verbos “cometer” ou “perpetrar” e os argumentos “criminoso”, “crime”, “explicação”, “frequência”, “instrumento”, “maneira”, dentre outros. Essas relações são denominadas de papéis semânticos, pois expressam funções que os diferentes constituintes de uma sentença desempenham em relação ao verbo ou predicado da sentença. FrameNet difere-se de outros recursos para SRL, como PropBank (Palmer; Gildea; Kingsbury, 2005) e VerbNet (Kipper; Dang; Palmer, 2000), na medida em que associa papéis semânticos não somente a verbos, mas também a substantivos, a adjetivos, a advérbios, e até a proposições.

A Figura 9.4 apresenta um recorte da definição e componentes do *frame* `Committing_crime`¹⁸.

Figura 9.4: Descrição do *frame* `Committing_crime` na FrameNet de Berkeley.

FrameNet Data

Committing_crime

Definition:

A **Perpetrator** (generally intentionally) commits a **Crime**, i.e. does something not permitted by the laws of society.
They PERPETRATED a felony by substituting a lie for negotiations.
The suspect had allegedly COMMITTED the crime to gain the attention of a female celebrity.

FEs:

Core:

<p>Perpetrator [Perp] Semantic Type: Sentient</p>	<p>The individual that commits a Crime. How can he COMMIT treason against the King of England in a foreign country , if he is not English? He PERPETRATED a crime against mother nature.</p>
--	---

Core Unexpressed:

<p>Crime [Cr]</p>	<p>An act, generally intentional, that has been formally forbidden by law. How can he COMMIT treason against the King of England in a foreign country , if he is not English? He PERPETRATED a crime against mother nature.</p>
--------------------------	--

Como se pode observar na Figura 9.4, o *frame* é formado por vários componentes, descritos a seguir:

- **Elementos de frames (Frame Elements – FE)**, que definem os papéis semânticos envolvidos no *frame*. A sentença “*He committed the murder coldly and deliberately*” (em português, “Ele cometeu o assassinato fria e deliberadamente”) evoca o *frame* `Committing_crime` através do verbo “*commit*”, cujos argumentos “*He*”, “*the murder*” e “*coldly and deliberately*” expressam os seguintes papéis semânticos: “*perpetrator*”, “*crime*” e “*manner*”, conforme abaixo:

– [*PERPETRATOR He*] committed [*CRIME the murder*] [*MANNER coldly and deliberately.*], onde:

* *PERPETRATOR* – o indivíduo que cometeu um crime;

¹⁸A descrição completa do *frame* está disponível em <https://framenet.icsi.berkeley.edu/frameIndex>.



- * *CRIME* – um ato, geralmente intencional, que é formalmente proibido pela lei;
 - * *MANNER* – uma descrição da forma e dos efeitos secundários do crime, assim como descrições gerais comparando eventos, podendo também indicar características salientes do criminoso que afetam a ação (presunçosamente, friamente, deliberadamente, ansiosamente, cuidadosamente).
- **Unidades Lexicais (*Lexical Unit* – LU)**, que são as palavras relacionadas no *frame*. Cada palavra polissêmica¹⁹ com significados distintos pertence a um *frame* diferente. Por exemplo, a palavra “*commit*” possui quatro entradas (sentidos) no léxico, conforme os quatro *frames* dos quais participa: *Imposing_Obligation*, *Institutionalization*, *Commitment*, *Committing_crime*. O léxico da FrameNet contém, para cada unidade léxica, além do termo e de uma definição em linguagem natural, as realizações sintáticas possíveis dos elementos de *frames* relacionados à unidade léxica. Por exemplo, na sentença “*He committed the murder coldly and deliberately*” (em português, “Ele cometeu o assassinato fria e deliberadamente”), o elemento de *frame* ou papel semântico “crime” (*the murder*) possui a realização sintática de sintagma nominal (NP – *Noun Phrase*);
 - **Entradas Lexicais (*Lexical Entry* – LE)**, que são unidades lexicais evocadoras de *frame*, ou seja, que chamam ou ativam *frames*. No *frame* *Committing_crime*, as entradas lexicais são os verbos *commit.v* e *perpetrate.v* e os substantivos *commission.n* e *crime.n*. As entradas lexicais mais comuns são verbos, porém alguns *frames* são ativados por substantivos e adjetivos. Por exemplo, a sentença “... *the reduction of debt levels to \$665 million from \$2.6 billion.*” (em português, “... a redução dos níveis de dívida para 665 milhões de dólares, de 2,6 mil milhões de dólares.”) tem-se um exemplo de uso do *frame* *Cause_change_of_scalar_position*, evocado pelo substantivo “*reduction*”;
 - **Corpus da FrameNet**, um conjunto de sentenças anotadas que exemplificam os componentes da FrameNet. O *corpus* da FrameNet é uma parte crucial, pois representa um recurso valioso para o desenvolvimento e teste de sistemas de PLN que requerem uma compreensão da semântica dos textos, especialmente, dos papéis envolvidos no evento. O conjunto total dos textos anotados contém, atualmente, 202.978 textos, divididos em:
 - Conjunto de Anotações Completas (*Full Text Annotation Sets*), que contém 28.446 anotações semânticas detalhadas para textos inteiros, e não apenas para frases isoladas. Essas anotações incluem informações sobre os elementos do *frame* (papéis semânticos da FrameNet), as unidades e entradas lexicais (léxico da FrameNet), e suas realizações sintáticas;
 - Conjunto de Anotações Lexicográficas (*Lexicographic Annotation Sets*), que contém 174.532 anotações para as palavras em uma língua, incluindo informações sobre os sentidos das palavras, os *frames* que esses sentidos evocam e os papéis semânticos (elementos de *frame*) associados a cada sentido. Em (Ruppenhofer et al., 2006, pp. 67-88), tem-se o detalhamento das camadas de anotação do *corpus* da FrameNet;
 - **Tipos semânticos**, que são associados às unidades lexicais, aos elementos do *frame*

¹⁹Uma palavra polissêmica é uma palavra que possui vários significados ou sentidos relacionados entre si, dependendo do contexto em que é usada. Por exemplo, a palavra “banco” pode se referir a uma instituição financeira, um local para sentar, ou a uma elevação de areia no mar.



ou ao *frame* como um todo. Em (Ruppenhofer et al., 2006, pp. 111-120), tem-se a definição destes marcadores semânticos. Por exemplo, o elemento de *frame* *Perpetrator* do *frame* *Committing_Crime* é marcado como sendo do tipo *sentient* (que percebe pelos sentidos, que recebe impressões).

Além da definição individual de cada *frame*, a FrameNet possui relações semânticas entre *frames*, denominadas relações *frame-to-frame*. Alguns exemplos são: *Inherits_from* (herdade), *Is_Inherited_by* (é herdado por), *Is_Used_by* (é usado por). Em (Ruppenhofer et al., 2006, pp. 104-111), tem-se a descrição das relações *frame-to-frame** suportadas pela FrameNet.

Atualmente, a FrameNet contém 1224 *frames*, 10.478 elementos de *frames* (papéis semânticos), e 13.687 unidades lexicais²⁰.

FrameNet fornece uma nova perspectiva para um recurso léxico-semântico. O significado de palavras ou unidades lexicais é dado no contexto das situações em que podem participar (*frames*), por meio dos papéis que podem assumir. FrameNet não poderia substituir completamente a WordNet porque falta à primeira muitas das relações semânticas úteis como meronímia e hiperonímia. Embora haja uma interseção entre essas bases, elas se distinguem em boa parte. Enquanto a WordNet foca em relações entre *synsets* organizando uma hierarquia e taxonomia do mundo, a FrameNet foca nas relações que ocorrem em eventos.

Alguns projetos visam relacionar as entradas lexicais dessas duas bases. É o caso do projeto SemLink²¹, cujo objetivo é vincular diferentes recursos léxico-semânticos por meio de um conjunto de mapeamentos. Estes mapeamentos permitirão combinar as diferentes informações fornecidas por esses diferentes recursos lexicais para tarefas como inferência em linguagem natural (*Natural Language Inference* – NLI). Os recursos mapeados pelo SemLink são WordNet, FrameNet, VerbNet e PropBank.

9.1.2.1 Framenets para o português

FrameNet Brasil (FN-Br)²² (Salomão, 2009), iniciativa de pesquisa lexicográfica, em desenvolvimento na Universidade Federal de Juiz de Fora (UFJF) desde 2008, tem o objetivo de construir e evoluir, para o português, a contraparte linguística da rede semântica original FrameNet. Atualmente, a base da FN-Br é a base mais robusta e representativa do paradigma da Semântica de *Frames* para o português. Foi construída através da tradução automática dos *frames* existentes na FrameNet original, e posterior adaptação para o português brasileiro. Este processo de adaptação envolveu traduzir e ajustar a descrição e os elementos dos *frames* para garantir que eles sejam relevantes e aplicáveis ao contexto brasileiro. Além da adaptação dos *frames* originais da FrameNet, no âmbito de alguns projetos, como o COPA 2014 (Torrent et al., 2014) e FLAME²³, relativos aos domínios de esporte e turismo, respectivamente, foram criados novos *frames* para representar conceitos e situações específicos da cultura e do português brasileiro. O *corpus* FN-Br é constituído pela combinação de mais de 16 *corpora*, todos caracterizados por permitir acesso público e

²⁰Mais detalhes sobre os números atuais da FrameNet podem ser acessados em https://framenet.icsi.berkeley.edu/current_status.

²¹<https://verbs.colorado.edu/semLink/>. A versão atual do SemLink é a versão 2.0 e pode ser acessada pelo GitHub <https://github.com/cu-clear/semLink>.

²²<https://www2.ufjf.br/framenetbr/>

²³<https://www2.ufjf.br/framenetbr/projetos/>



que representam usos do português europeu e do português brasileiro. Em 2009, de acordo com (Salomão, 2009), os *corpora* totalizavam pouco mais de 280 milhões de palavras.

A Figura 9.5 apresenta um recorte da definição e componentes do *frame* *Cometer_crime* da FN-Br²⁴, adaptado do *frame* *Committing_crime* da FrameNet de Berkeley (vide Figura 9.4).

Figura 9.5: Descrição do *frame* *Cometer_crime* na FrameNet Brasil (FN-Br).

Cometer_crime

Definição	
Um Criminoso (geralmente intencionalmente) comete um Crime , ou seja, faz algo que não é permitido pelas leis da sociedade.	
Elementos de Frame Nucleares	
FE Core:	
Criminoso semantic_type: @sentient	O indivíduo que comete o Crime .
FE Core-Unexpressed:	
Crime	Um ato cometido, geralmente intencional, que é formalmente proibido por lei.
Elementos de Frame Não-Nucleares	
Explicação semantic_type: @state_of_affairs	Um estado de coisas ao qual o Criminoso está respondendo em seu Crime .
Finalidade semantic_type: @state_of_affairs	A ação que o Criminoso estava tentando realizar no Crime .
Frequência	A frequência com a qual um Crime é cometido.

Outras iniciativas culminaram na geração de bases de *frames* em português, todas de menor tamanho que a FN-Br e para domínios ainda mais específicos.

A base FrameFOR (Barreira; Pinheiro; Furtado, 2017) é uma base com 113 *frames* em português brasileiro, adaptados da FrameNet original, contendo os papéis semânticos, unidades e entradas lexicais relacionados aos tipos de crimes mais investigados na Perícia Forense do Estado do Ceará, no Brasil (PEFOCE) – formação de quadrilha, tráfico de drogas, sequestro, corrupção, receptação, contrabando, pedofilia, estupro, agressão, tortura, falsificação, ameaça, porte ilegal de arma, estelionato, e extorsão.

O estudo de (Bertoldi, 2011) analisou os limites da criação automática de léxicos computacionais segundo o paradigma FrameNet, comparando as unidades lexicais evocadoras, os papéis semânticos e a estrutura do *frame* *Criminal_process*, em inglês e português. Esse estudo contrastivo mostrou que os *frames* do domínio jurídico são socialmente orientados e que a criação automática de léxicos em áreas cultural e socialmente orientadas tende a apresentar divergências. Em (Bick, 2009) tem-se a proposta de PFN-PT, um sistema para a anotação semântica automática do português, consistindo numa nova *framenet* contendo cerca de 13.000 padrões sintáticos, cobrindo 7.300 lemas verbais com 10.700 sentidos.

²⁴Ver descrição completa em <https://webtool.framenetbr.ufjf.br/index.php/webtool/report/frame/main>



Todos estes projetos, ainda que de menor porte, possuem relatos de sucesso em aplicações de PLN como extração de informação, anotação de papéis semânticos, reconhecimento de entidades nomeadas, evidenciando a importância de abordar as peculiaridades linguísticas com perspectivas contextualizadas e culturalmente relevantes.

9.1.2.2 Análise do Exemplo Motivador usando a FrameNet

Nesta seção, usaremos a FrameNet de Berkeley para analisar o exemplo motivador definido no início da Seção 9.1. No Exemplo 9.3 são destacados o *frame* associado, as unidades lexicais (elemento evocador) que evocaram o *frame* e os papéis semânticos identificados no texto.

Exemplo 9.3.

Um assalto do tipo saidinha bancária, ocorrido na tarde desta terça-feira, terminou com uma mulher de 42 anos baleada pelos assaltantes. O roubo ocorreu na rua Professor Costa Mendes.

- *FRAME: Robbery* (definição do *frame*: situação em que um perpetrador prejudica uma vítima tirando algo (bens) dela ... O assalto pode ser feito de uma maneira específica (por exemplo, à força) e através de um meio específico (por exemplo, ameaçando a vítima).)
- UNIDADE LÉXICA: *robbery* (“assalto” e “roubo”, em português)
- PAPEL SEMÂNTICO *place*: “rua Professor Costa Mendes”
- PAPEL SEMÂNTICO *time*: “na tarde desta terça-feira”
- PAPEL SEMÂNTICO *perpetrator*: “pelos assaltantes”

Uma dificuldade com a análise desse exemplo, à luz da FrameNet, foi na identificação do papel semântico “vítima” (“uma mulher de 42 anos”), pois o complemento da sentença que a contém “...com uma mulher de 42 anos baleada ...” possui a estrutura sintática Prep.Det.N (preposição + determinante + substantivo) e não é compatível com nenhuma realização sintática do elemento de *frame* *victim*.

9.1.3 ConceptNet

ConceptNet²⁵ (Speer; Chin; Havasi, 2016) é uma base de conhecimento de senso comum que expressa relações rotuladas e ponderadas entre palavras ou fragmentos de textos em linguagem natural, através de um Grafo de Conhecimento (*Knowledge Graph*) contendo *edges* ou afirmações. Alguns exemplos de afirmações expressas na ConceptNet são:

- Uma rede é usada para pescar peixe (*A net is used for catching fish.*);
- “Folhas” é uma forma da palavra “folha” (*“Leaves” is a form of the word “leaf”*);
- A palavra “cold” em inglês é “*studeny*” em tcheco (*The word “cold” in English is “studeny” in Czech*);
- O alimento é usado para comer (*Food is used for eating*);
- Bicicleta é usada para chegar a algum lugar rápido (*Bicycle is used for getting somewhere fast*);
- Cozinhar é motivada por você está com fome (*Cook is motivated by being hungry*).

²⁵<https://conceptnet.io/>



Sua versão original (Havasi; Speer; Alonso, 2007; Liu; Singh, 2004) foi criada pela equipe do MediaLab do *Massachusetts Institute of Technology (MIT)* em 1999, a partir de conhecimentos extraídos do projeto de construção coletiva (*crowdsourcing*) Open Mind Common Sense (OMCS) (Singh et al., 2002). O OMCS surgiu com o objetivo de coletar, pela Internet e de colaboradores voluntários, sentenças que expressavam fatos da vida comum. Por exemplo, a sentença “*The Effect of [falling off a bike] is [you get hurt]*” foi coletada de voluntários, quando solicitados a preencher os espaços do *template* “*The Effect of [.....] is [.....]*”. A alternativa adotada pela equipe da ConceptNet foi construir a rede semântica (nós conceituais interligados pelas relações semânticas), a partir de um processo automático sobre o *corpus* OMCS, o qual extraiu as relações semânticas e seus argumentos.

A motivação do projeto que mantém a ConceptNet é expressar os fatos que as pessoas sabem comumente sobre o mundo — conhecimento de senso comum — através de afirmações que relacionam conceitos. Este tipo de conhecimento é importante porque, quando as pessoas se comunicam, seus proferimentos acontecem sobre suposições implícitas e básicas, as quais suportam e explicam boa parte dos raciocínios necessários para um bom nível de entendimento e, conseqüentemente, uma boa comunicação. Por exemplo, quando alguém fala “Eu comprei doces”, está implícito que usou dinheiro, ou quando fala “Fui a um casamento”, provavelmente tinha uma noiva, um noivo, uma festa com bolo e champagne, e o interlocutor está autorizado a perguntar “A noiva estava bonita?” etc.

Atualmente, a ConceptNet²⁶ evoluiu como um projeto colaborativo com diversas fontes:

- *Open Mind Common Sense* (OMCS) (Singh et al., 2002) e projetos irmãos em outras línguas (Anacleto et al., 2006);
- Informações extraídas da análise do Wikcionário²⁷, em vários idiomas, com um analisador personalizado (“Wikiparsec”);
- “*Games with a Purpose*”, que são jogos projetados para coletar conhecimento comum (Ahn; Kedia; Blum, 2006; Kuo et al., 2009);
- *Open Multilingual WordNet* (Bond; Foster, 2013), uma representação de dados vinculados a WordNet de Princeton e seus projetos paralelos em vários idiomas;
- JMDict (Breen, 2004), um dicionário japonês multilíngue;
- OpenCyc, uma hierarquia de hiperônimos fornecida pelo Cyc (Lenat; Guha, 1989), um sistema que representa o conhecimento do senso comum na lógica de predicados;
- Um subconjunto de DBPedia (Auer et al., 2007), uma rede de fatos extraídos de *infoboxes* da Wikipédia.

A unidade de conhecimento da ConceptNet é uma afirmação ou *edge*²⁸ que é uma relação particular entre termos ou frases em uma linguagem natural, de uma fonte específica. Sucintamente, cada *edge* é uma tripla com um primeiro argumento (nó inicial), um rótulo da relação e um segundo argumento (nó final). Por exemplo, a afirmação “*Bicycle is used to get somewhere fast*” pode ser expressa como (*Bicycle, is used to, get somewhere fast*). Cada *edge* é representada em uma estrutura de dados com os seguintes atributos:

- *URI* – identificador único para a afirmação que está sendo expressa;

²⁶Sua última versão é a ConceptNet 5.8 e a documentação completa está disponível em <https://github.com/commonsense/conceptnet5/wiki>

²⁷Disponível em https://pt.wiktionary.org/wiki/Wikcion%C3%A1rio:P%C3%A1gina_principal

²⁸<https://github.com/commonsense/conceptnet5/wiki/Edges>



- *REL* – o URI da relação expressa no *edge*. Atualmente, existem 34 relações em *edges* da ConceptNet 5 - *RelatedTo, IsA, is Used For; Motivated by, Desires* etc.²⁹;
- *START* – o URI do primeiro argumento da afirmação;
- *END* – o URI do segundo argumento da afirmação;
- *WEIGHT* – a força da afirmação. Um peso típico é 1, mas pode ser maior a depender do número de vezes que a afirmação foi recuperada ou generalizada a partir das fontes;
- *SOURCES* – as fontes que, quando combinadas, dizem que esta afirmação deveria ser verdadeira;
- *LICENSE* – o URI *Creative Commons* para a licença que rege esses dados;
- *DATASET* – o URI que representa o conjunto de dados de uma fonte específica que criou a afirmação;
- *SURFACE TEXT*³⁰ – o texto original em linguagem natural que expressou esta afirmação. Os conceitos do início e fim serão marcadas entre colchetes duplos. Um exemplo é “[[Bicycle]] is used for [[get somewhere fast]]”.

A ConceptNet contém mais de 21 milhões de *edges* e quase 10 milhões de nós com palavras ou fragmentos de textos. A base cobre em torno de 78 linguagens naturais³¹ com pelo menos 10.000 entradas no vocabulário. As 10 (dez) principais linguagens são: inglês, francês, italiano, alemão, espanhol, russo, português, japonês, holandês e chinês. A base de afirmações na linguagem inglesa possui um vocabulário com 1,8 milhão de nós, e no português contém um vocabulário com 473 mil nós.

O *framework* computacional da ConceptNet contém uma hierarquia de URIs que identificam os principais componentes dessa base de conhecimento: afirmações (ou *edges*), termos (palavras ou frases em uma linguagem particular), relações (por exemplo, *IsA*), *datasets*, fontes de dados. Também possui uma API REST³² pela qual você pode obter os componentes no formato JSON. Cada *edge*, termo, relação, *dataset* e fonte da ConceptNet possui uma URI que os identificam e sua definição completa em JSON pode ser acessada via API. A Figura 9.6 apresenta um recorte da definição do termo “*murder*”, contendo a lista de *edges* a partir desse termo.

Ao acessar, via API, a definição deste termo, tem-se acesso à sua definição em JSON, conforme ilustrado na Figura 9.7, com a URI deste conceito (“/c/en/murder”) e a lista de *edges*.

ConceptNet se tornou um *hub* de conteúdo semântico, pois provê *link* para outras bases de dados, com um *toolkit* e uma API que suportam inferências práticas de senso comum sobre textos, tais como descoberta de contexto (que habilita a extração da vizinhança contextual de um conceito, e.g., “tirar a roupa”, “ir dormir”, e “deitar-se” são vizinhos do conceito “ir para a cama”), cadeia de inferências (que habilita encontrar caminhos na rede semântica a partir de um conceito, e.g., “comprar comida” - “ter comida” - “comer comida” - “sentir-se cheio” - “sentir-se com sono”) e analogia conceitual (que envolve encontrar conceitos que são estruturalmente similares, e.g., “funeral” e “casamento”, “sofá” e “cama”). Todos esses exemplos foram extraídos de (Ovchinnikova, 2012).

²⁹A lista completa das relações expressas na ConceptNet está disponível em <https://github.com/commonsense/conceptnet5/wiki/Relations>

³⁰Pode ser nulo porque nem todas as afirmações foram derivadas de entrada em linguagem natural

³¹As bases de afirmações da ConceptNet para todas as linguagens naturais suportadas pode ser consultada em <https://github.com/commonsense/conceptnet5/wiki/Languages>

³²Acessível em api.conceptnet.io. A documentação completa da API da Conceptnet pode ser acessada em <https://github.com/commonsense/conceptnet5/wiki/API>



Figura 9.6: Descrição do termo “murder” na ConceptNet 5.8.



9.1.3.1 Bases de conhecimento de senso comum para o português

Como visto, a Conceptnet 5.8 possui uma cobertura de 473 mil termos ou frases no português, representando assim a mais extensa base de conhecimento de senso comum para essa língua.

O projeto *Open Mind Common Sense – Brasil* (OMCS-Br) foi um projeto do Laboratório de Interação Avançada (LIA) da Universidade Federal de São Carlos – UFSCar, em colaboração com o MediaLab do MIT, para a coleta de conhecimento de senso comum em português (Anacleto et al., 2006). Este projeto em 2010 contava com 160.000 afirmações de senso comum de seus colaboradores. O projeto foi descontinuado, mas diversas aplicações e estudos foram desenvolvidos a partir desta base. Dentre eles, podemos citar, uma ferramenta que utiliza a base de conhecimento de senso comum para auxiliar a interação humana (de alunos e professores) com ferramentas educacionais (Anacleto et al., 2007).

A base InferenceNet-BR (Pinheiro et al., 2010) adapta a ConceptNet (Liu; Singh, 2004) adicionando uma camada que define o papel da afirmação em uma inferência – se como premissa (ou pré-condição) ou como conclusão (ou pós-condição). Além da tradução dos termos e suas afirmações (relações com outros conceitos), o projeto da InferenceNet-BR evoluiu a base com novo conhecimento semântico específico para o domínio de segurança pública em português.

A InferenceNet-BR compõe-se de duas bases de conhecimento:

1. **Base Conceitual** – essa base contém o conjunto de termos (palavras ou frases em linguagem natural) relacionados em uma rede semântica, representada por meio de quádruplas (ARG1, REL, ARG2, PESO, TIPO_INF) que definem as afirmações ou *edges*, onde:

- ARG1 – identificador do termo inicial da relação;
- ARG2 – identificador do termo final da relação;
- REL – identificador da relação semântica de um total de 17 relações, por exemplo, “CapazDe”; “PartDe”; “ÉUm”; “EfeitoDe” etc.;



Figura 9.7: Definição em JSON do termo “murder” na ConceptNet 5.8.

```
"@id": "/c/en/murder",
"edges": [
  {
    "@id": "/a/[r/IsA,lc/en/murder,/c/en/crime/]",
    "@type": "Edge",
    "dataset": "/d/conceptnet/4/en",
    "end": {
      "@id": "/c/en/crime",
      "@type": "Node",
      "label": "a crime",
      "language": "en",
      "term": "/c/en/crime"
    },
    "license": "cc:by/4.0",
    "rel": {
      "@id": "/r/IsA",
      "@type": "Relation",
      "label": "IsA"
    },
    "sources": [
      {
        "@id":
"/and/[s/activity/omcs/omcs1_possibly_free_text/_/s/contributor/omcs/highplacespam1]",
        "@type": "Source",
        "activity": "/s/activity/omcs/omcs1_possibly_free_text",
        "contributor": "/s/contributor/omcs/highplacespam1"
      },
      ...
    ]
  },
  {
    "start": {
      "@id": "/c/en/murder",
      "@type": "Node",
      "label": "a murder",
      "language": "en",
      "term": "/c/en/murder"
    },
    "surfaceText": "[[a murder]] is [[a crime]]",
    "weight": 8.0
  }
],
```

- PESO – força da afirmação. Um peso típico é 1, mas pode ser maior a depender do número de vezes que a afirmação foi recuperada ou generalizada a partir das fontes (Conceptnet original e *corpus* de textos de domínio);
- TIPO_INF – tipo da relação inferencial - premissa (pré-condição - PRE) ou conclusão (pós-condição - POS).

2. **Base de Sentenças-Padrão** - essa base contém a estrutura sintática das sentenças-padrão e suas relações com termos da Base Conceitual. Seja a sentença-padrão $sp_1 = X$ “ser_assassinado_por” Y. Temos que X está relacionado com o termo “vítima” e Y está relacionado com o termo “criminoso”, através da relação “ÉUm”, e com o tipo inferencial **pós-condição**. Ou seja, na sentença “Maria foi assassinada por seu amante” podemos concluir que Maria é a vítima e seu amante é o criminoso. A rede semântica dessa base de sentenças-padrão é representada, portanto, por meio de quádruplas (SP, REL, ARG, PESO, TIPO_INF) que definem as afirmações ou *edges*, onde:

- SP – identificador da sentença-padrão geralmente da forma “(X, sintagma verbal, Y)”;
- ARG – identificador do termo final da relação;



- REL – identificador da relação semântica de um total de 17 relações, por exemplo, “CapazDe”; “PartDe”; “ÉUm”; “EfeitoDe” etc.;
- PESO – força da afirmação. Um peso típico é 1, mas pode ser maior a depender do número de vezes que a afirmação foi recuperada ou generalizada a partir das fontes (Conceptnet original e *corpus* de textos de domínio);
- TIPO INF – tipo da relação inferencial - premissa (pré-condição - PRE) ou conclusão (pós-condição - POS).

A Figura 9.8 ilustra uma parte da rede de relacionamento inferencial da palavra “crime” na base InferenceNet-BR, com as seguintes relações:

- (capazDe, “crime”, “ter vítima”, “Pre”);
- (capazDeReceberAcao, “crime”, “evitar por polícia”, “Pre”);
- (primeiroSubEventoDe, “crime”, “escolher a vítima”, “Pre”);
- (usadoPara, “crime”, “vingança”, “Pre”);
- (capazDeReceberAcao, “crime”, “cometer com arma”, “Pre”);
- (capazDe, “crime”, “envolver violência”, “Pre”);
- (éUm, “crime”, “violação da lei”, “Pre”);
- (motivacaoDe, “crime”, “vingança”, “Pre”);
- (efeitoDe, “crime”, “culpa”, “Pos”);
- (efeitoDesejavelDe, “crime”, “julgamento”, “Pos”).

9.1.3.2 Análise do Exemplo Motivador usando a ConceptNet

Nesta seção, usaremos a base da ConceptNet para o português e para o inglês para analisar o Exemplo 9.1. No Exemplo 9.4 são destacadas algumas afirmações de senso comum associadas aos termos mencionados no texto (termos sublinhados).

Exemplo 9.4.

Um assalto do tipo saidinha bancária, ocorrido na tarde desta terça-feira, terminou com uma mulher de 42 anos baleada pelos assaltantes. O roubo ocorreu na rua Professor Costa Mendes.

Afirmações (*edges*) do termo *assalto*:

- “terror” está localizada em “assalto” (da base para o português);
- “pessoas com medo” está localizada “assalto” (da base para o português);
- “*heist*” is a type of “robbery” (roubo é um tipo de assalto);
- “*revolver*” is a thing used for “robbery” (revólver é uma coisa usada para assaltar)

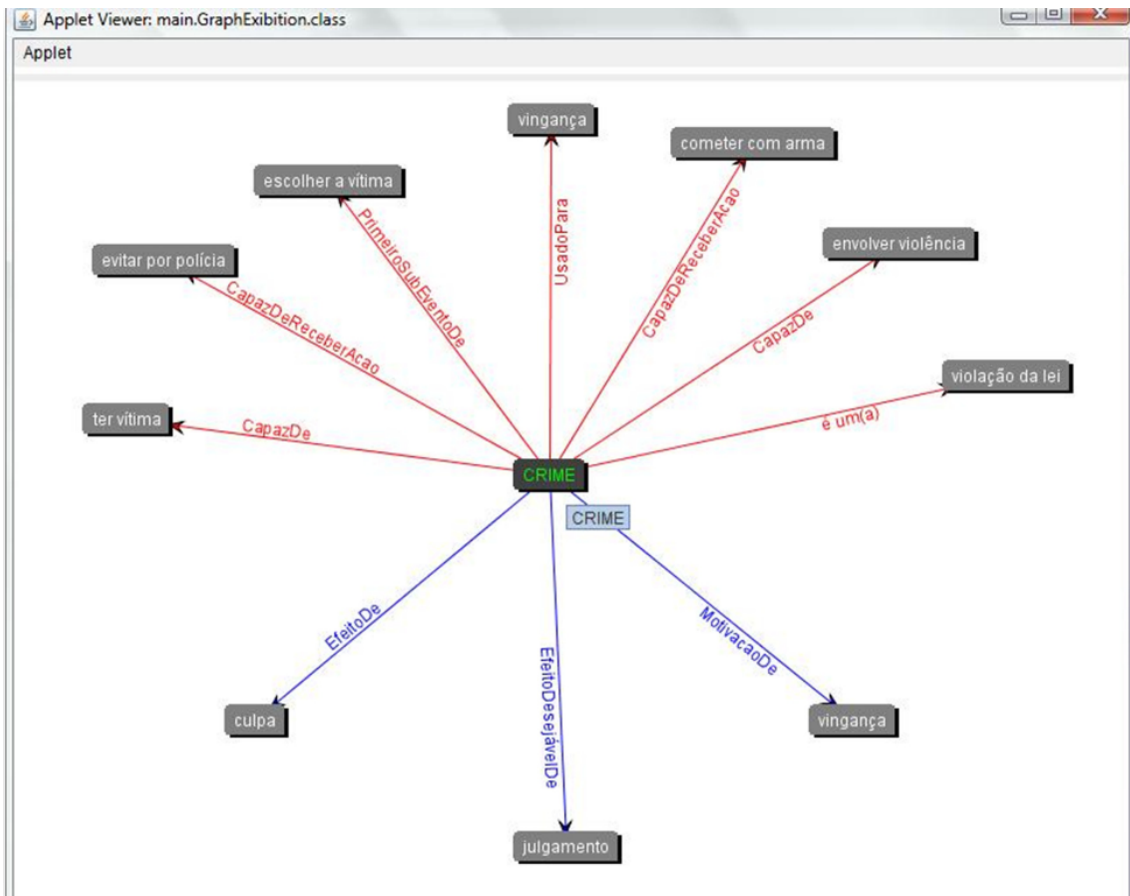
Afirmações (*edges*) do termo *balear* (“shoot”, em inglês):

- “fight the enemy” is subevent of “shoot” (lutar contra o inimigo é subevento de balear);
- “a gun” is a thing used for “shoot” (uma arma é uma coisa usada para balear);
- “shoot” is a way of “kill” (baleiar é uma forma de matar);
- “shoot” is a type of “sprout” (é um tipo de broto, como broto de feijão)

Algumas dificuldades com a análise do exemplo, à luz da ConceptNet, foram:



Figura 9.8: Rede de relações inferenciais do termo “crime” (pré-condições (PRE) são as arestas na cor vermelha, e pós-condições (POS) são as arestas na cor azul).



- A desambiguação do sentido da palavra “baleiar” no exemplo Exemplo 9.4 não é trivial. Na base da ConceptNet não há separação das redes semânticas por classe gramatical e de palavras polissêmicas. Por exemplo, as afirmações de “baleiar” no sentido de atirar e em outros sentidos estão associadas a um mesmo termo, assim como afirmações de “shoot” como verbo e como substantivo;
- As afirmações em português são incipientes na base da ConceptNet, pelo menos para os termos analisados nesse exemplo.

9.2 Considerações Finais

Ao longo deste capítulo exploramos três bases de conhecimento comuns na área de PLN: WordNet, FrameNet e ConceptNet. WordNet, com sua rica estrutura hierárquica, destaca-se por mapear relações semânticas entre conjuntos de sinônimos (*synsets*), fornecendo uma compreensão sobre sinônimos, antônimos, hiperônimos e muito mais. Esse recurso se tornou uma das ferramentas mais utilizadas em aplicações de PLN, desde a análise de sentimentos até a desambiguação de sentidos.

FrameNet, por sua vez, adota uma abordagem baseada em *frames* para capturar significados em dados contextos ou situações, fazendo a ponte entre elementos lexicais e seus respectivos papéis semânticos. Através dessa base, podemos entender mais profundamente



como as palavras interagem dentro de estruturas semânticas e pragmáticas mais amplas. Esta abordagem, focada nos papéis semânticos, permite uma análise mais rica do texto, tornando-a particularmente útil para tarefas de anotação de papéis semânticos e análise de discurso. Em contrapartida, ela não possui um critério claro de completamento, ou seja, não sabemos quando vamos ter (ou se já temos) todos os *frames* necessários.

Por fim, a base ConceptNet destaca-se pelo seu caráter colaborativo e multidimensional. Integra conhecimentos de senso comum e conhecimento léxico-semântico de várias fontes e idiomas, oferecendo uma visão ampla das relações entre conceitos e contextos. Seu formato de rede semântica ajuda a capturar a complexidade e interconexão do conhecimento humano de uma maneira holística. ConceptNet tem sido usada em aplicações de extração de informação e reconhecimento de implicação textual. Em contrapartida, problemas de consistência da base parecem importantes. E tal como no caso de FrameNet, não temos um critério explícito de quando teremos uma cobertura suficiente.

Em resumo, cada uma destas bases de conhecimento representa um recurso para o PLN com perspectivas únicas, mas qual delas se aplica melhor e em quais casos? Recursos léxicos-semânticos parecem não serem suficientes para expressar conhecimento de mundo. De outro lado, bases de conhecimento de senso comum são mais flexíveis e expressivas, porém menos formais. Acreditamos que uma abordagem híbrida, em que tais bases de conhecimento sejam usadas de forma combinada, é mais promissora para o PLN. Nas próximas versões deste capítulo introduziremos os *parsers* semânticos que, em conjunto com as bases de conhecimento, constituem poderoso *framework* para sistemas de entendimento de linguagem natural.



Capítulo 10

Semântica Distribucional

Eloize Rossi Marques Seno

Daniela Claro

Laila Mota

Jessica Rodrigues

É relativamente fácil para nós, seres humanos, visualizarmos um texto e, a partir de uma simples leitura, extrairmos dele determinados tipos de informação. Por exemplo, ao ler o texto “Ser feliz sem motivo é a forma mais autêntica de felicidade.” podemos reconhecer o sentido das palavras e o significado do texto formado pela junção de todas as palavras. Diferente dos humanos, os algoritmos computacionais não conseguem processar símbolos/palavras. Ao invés disso, eles requerem uma representação numérica de um documento ou texto a ser processado, para que consigam realizar suas operações.

A semântica distribucional tem sido atualmente a principal abordagem de representação do significado lexical adotada nas mais diversas tarefas do processamento de linguagem natural. Nessa abordagem, os itens lexicais (palavras) são representados por meio de vetores de valores reais, conhecidos por **vetores semânticos**, que codificam o significado das palavras a partir de sua distribuição em textos.

A semântica distribucional é ancorada na **Hipótese Distribucional** (Firth, 1957a; Harris, 1954) que preconiza que palavras que têm um contexto linguístico semelhante tendem a ter significado similar ou aproximado. É o caso, por exemplo, de palavras como “ensino” e “educação” que costumam aparecer no mesmo contexto de palavras como “aluno”, “escola” e “professor”, sugerindo que existe uma similaridade entre as duas palavras em certos contextos. Vejamos, por exemplo, a ocorrência dessas palavras nas sentenças a seguir¹:

1. MEC deve começar a ouvir alunos sobre novo ensino médio em 8 de maio.
2. Governo prorroga inscrições para concurso público de professores na rede estadual de ensino.
3. Alunos e profissionais da educação terão aulas de comportamento seguro.
4. Aluno é colocado em ensino remoto após intimidar e tentar derrubar professor em SP.

Assim, na semântica distribucional as palavras são caracterizadas pelo contexto em que elas aparecem. Por se basearem em distribuição, os vetores semânticos podem ser aprendidos automaticamente a partir de textos, sem que haja supervisão de um humano (utilizando-se textos não rotulados, portanto). Os modelos que aprendem esse tipo de

¹Sentenças retiradas de títulos de notícias retornadas pelo Google, em 18/04/2023, a partir dos termos de busca “ensino” e “educação” combinados com “aluno”, “escola” e “professor”.

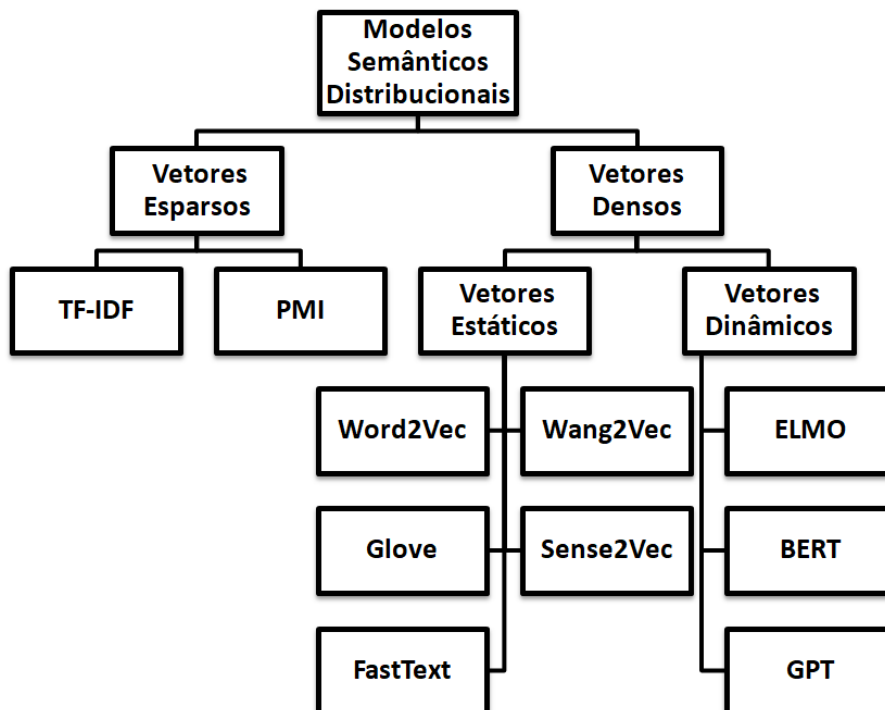


representação são denominados de **Modelos Semânticos Distribucionais - MSD** (*Distributional Semantic Models – DSM*, no inglês).

Os MSD são, frequentemente, classificados como **vetores esparsos** e **vetores densos**. Por exemplo, o modelo TF-IDF, amplamente adotado em tarefas que envolvem a comparação de similaridade entre documentos, como a detecção de plágio, a inferência textual e a recuperação de informações, é um exemplo clássico de vetor esparsos. Nesse modelo, o significado de uma palavra é representado por meio de uma função simples calculada com base na frequência da palavra em uma coleção de documentos, conforme será visto Seção 10.2.1. Como muitas palavras nunca ocorrem em alguns documentos, frequentemente, esse modelo leva a vetores muito grandes e esparsos, ou seja, com muitos zeros. Por outro lado, os modelos da família Word2Vec (Mikolov et al., 2013b) são considerados vetores densos (não esparsos), onde as entradas são números reais que representam propriedades semânticas úteis (conforme será abordado na Seção 10.3.1, ao invés de contagens quase zero).

A Figura 10.1 ilustra a estrutura geral dos Modelos Semânticos Distribucionais. Neste capítulo serão abordados apenas os **vetores esparsos** e os **vetores densos estáticos**. Mais especificamente, iniciaremos apresentando os modelos esparsos TF-IDF e PMI (Seção 10.2) e depois apresentaremos os principais vetores densos estáticos como o Word2Vec, o GloVe e o FastText (Seção 10.3). Os vetores densos dinâmicos, por sua vez, como ELMO, BERT e GPT serão abordados no Capítulo 15. Antes, porém, de apresentar os MSD, introduziremos alguns conceitos fundamentais da semântica vetorial e apresentaremos a **similaridade do cosseno** (Seção 10.1), uma maneira padrão de usar os vetores semânticos para calcular a similaridade entre palavras, sentenças e documentos, que é uma ferramenta fundamental em aplicações práticas como a sumarização automática e a recuperação de informações.

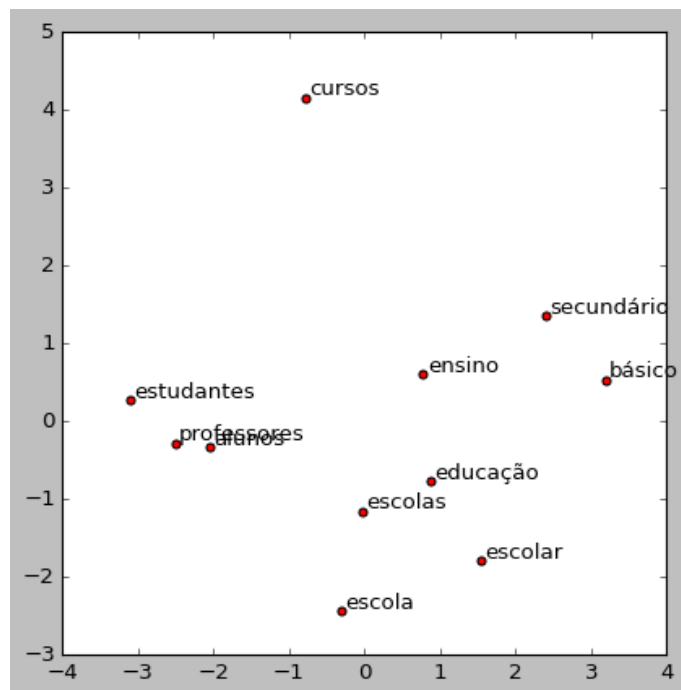
Figura 10.1: Ilustração dos Modelos Semânticos Distribucionais.



10.1 Semântica Vetorial

As primeiras investigações no campo da semântica vetorial, também conhecida por **métodos distribucionais**, tiveram início na década de 1950, impulsionadas pela convicção de que o significado de uma palavra pode ser definido a partir da sua distribuição nos contextos linguísticos em que ela ocorre, compartilhada por linguistas como (Joos, 1950), (Harris, 1954) e (Firth, 1957a), e pela proposta de (Osgood; Suci; Tenenbaum, 1957) de usar um ponto no espaço multidimensional para representar a conotação de uma palavra. Portanto, os métodos distribucionais se definem como uma representação vetorial que retrata o significado de uma palavra a partir da distribuição das palavras que formam o seu contexto (Jurafsky; Martin, 2023). Por exemplo, a Figura 10.2 representa o espaço vetorial semântico da palavra “ensino”². Palavras como “educação”, “estudantes”, “professores” e “alunos” são alguns exemplos de palavras que compartilham esse mesmo espaço semântico.

Figura 10.2: Espaço vetorial semântico da palavra “ensino” (representação gerada com o modelo GloVe (Seção 10.3.3), disponibilizado pelo NILC).



A representação vetorial semântica, ou simplesmente **vetores semânticos**, é um padrão de representação muito usual em PLN, que pode retratar vários aspectos do significado das palavras, como a similaridade (ex. “comércio” e “negócio”); a orientação de sentimento ou polaridade (ex. “fenomenal”, que conota uma avaliação positiva, e “estúpido”, que conota uma avaliação negativa); a associação entre palavras (ex. “futebol” e “bola”, que são claramente relacionados, uma vez que futebol se joga com uma bola), entre outros aspectos.

A ideia dos vetores semânticos é, então, representar cada palavra como um ponto em um **espaço vetorial multidimensional**, construído a partir da distribuição de suas palavras vizinhas.

²<http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>



Espaços vetoriais são objetos de estudo da Álgebra Linear e são bem caracterizados pela sua dimensão, que, grosseiramente falando, representa o número de direções independentes no espaço. Um espaço vetorial é formado por uma coleção de objetos chamados **vetores**. Em um Modelo Semântico Distribucional é possível representar palavras, sentenças e até documentos completos como vetores em um espaço multidimensional.

Geralmente, os vetores semânticos são representados por meio de uma **matriz de coocorrência** (ou distribuição de coocorrência), que retrata a frequência de coocorrência das palavras. As representações matriciais mais comuns são a **matriz termo-documento**, onde cada dimensão (vetor) da matriz representa um documento, e a **matriz termo-contexto**, onde cada dimensão representa uma palavra (Jurafsky; Martin, 2023). As subseções a seguir abordam essas duas formas básicas de representação.

10.1.1 Matriz termo-documento

Na matriz **termo-documento**, o espaço vetorial é formado por uma coleção de documentos³ que representam pontos ou **vetores** nesse espaço. Cada vetor tem dimensão $|V|$, onde $|V|$ representa o tamanho do vocabulário, que contém as palavras distintas (sem repetições) de todos os documentos da coleção. Assim, cada palavra do vocabulário é uma linha na matriz e cada coluna representa um documento da coleção. Cada célula da matriz, por sua vez, representa a frequência de uma palavra em particular em um documento em particular, ou seja, quantas vezes aquela palavra ocorreu naquele documento.

A matriz termo-documento foi definida por (Salton; Allan, 1994) como parte do Modelo de Espaço Vetorial – MEV proposto pelos autores para a recuperação de documentos na web. No MEV, um documento é representado como um vetor de frequência de palavras (ou termos), uma coluna, como no exemplo da Tabela 1. Os três documentos representados na tabela, por meio de colunas, representam pontos em um espaço tridimensional. A matriz foi computada a partir de três textos de divulgação científica extraídos da revista online Pesquisa FAPESP⁴, da seção de Tecnologia, edições 308 e 310 de 2021. Os textos se referem aos seguintes temas: energias renováveis⁵ (coluna 1), risco de escassez de energia fornecida pelas hidrelétricas brasileiras⁶ (coluna 2) e veículos elétricos movidos a etanol⁷ (coluna 3). A matriz da Tabela 10.1 representa apenas um subconjunto das palavras que ocorreram nesses textos. A contagem (frequência) das palavras em cada texto foi realizada considerando a sua forma lematizada.

Tabela 10.1: Matriz termo-documento para 5 palavras extraídas dos textos da revista online Pesquisa FAPESP. As células representam a frequência de uma palavra em cada texto.

	energias renováveis	escassez de energia	veículos elétricos
energia	32	36	9
eólico	15	15	1
renovável	6	8	4
etanol	1	1	22
hidrogênio	0	0	16

³O termo “documento” é usado nesta seção de forma genérica, podendo se referir a sentenças, parágrafos ou documentos completos.

⁴<https://revistapesquisa.fapesp.br/>

⁵<https://revistapesquisa.fapesp.br/a-forca-das-renovaveis/>

⁶<https://revistapesquisa.fapesp.br/sob-o-risco-da-escassez/>

⁷<https://revistapesquisa.fapesp.br/eletricos-movidos-a-etanol/>



Em aplicações reais, os vocabulários têm milhares de palavras e o número de documentos pode ser enorme (imagine todas as páginas da web). Isso frequentemente resulta em vetores muito grandes, levando a matrizes esparsas, já que muitas palavras nunca aparecem em outros documentos. Para lidar com um grande número de dimensões, uma técnica comumente usada é a Análise Semântica Latente (em inglês, *Latent Semantic Analysis* – LSA), que reduz a dimensionalidade do espaço vetorial através da Decomposição em Valores Singulares (em inglês, *Singular Value Decomposition* – SVD), conforme veremos na Seção 10.2.3.

10.1.2 Matriz termo-contexto

Os vetores semânticos também podem ser usados para representar o significado de **palavras** e não apenas de documentos. Para tanto, ao invés de usarmos uma matriz de **termo-documento** (como visto na Seção 10.1.1), usamos uma matriz de **termo-contexto**, também conhecida por matriz **palavra-palavra** ou matriz **termo-termo**.

Na matriz de **termo-contexto**, o espaço vetorial é formado por uma coleção de **palavras** que representam vetores nesse espaço. A matriz de coocorrência tem dimensionalidade $|V| \times |V|$ e cada célula representa o número de vezes que a palavra da linha (alvo) e a palavra da coluna (contexto) coocorrem em algum contexto em um *corpus* de treinamento. O contexto pode ser, por exemplo, um documento, no qual a célula representa o número de vezes que as duas palavras aparecem no mesmo documento. Entretanto, é mais comum usar contextos menores limitados a um número de palavras à esquerda e à direita da palavra-alvo (linha), por exemplo, três palavras à esquerda e três palavras à direita. Dessa forma, cada célula representa o número de ocorrência da palavra da coluna (contexto) em uma janela de três palavras em torno da palavra-alvo.

A Tabela 10.2 apresenta um subconjunto simplificado da matriz de coocorrência termo-contexto computada a partir dos textos da revista Pesquisa FAPESP descritos na Seção 10.1.1. As linhas da tabela representam cada palavra-alvo e cada célula indica o número de vezes que a palavra-alvo correspondente coocorreu em cada contexto (colunas). Para a criação da matriz foi considerada uma janela de contexto de tamanho 5, isto é, cinco palavras à esquerda e cinco palavras à direita da palavra-alvo. Para exemplificação, o vetor da palavra-alvo “energia” está destacado em negrito.

Tabela 10.2: Subconjunto simplificado da matriz de coocorrência termo-contexto computada a partir de três textos da revista Pesquisa FAPESP.

	elétrica	geração	combustível
etanol	0	1	10
veículo	0	0	4
energia	20	5	2
país	5	2	0

Os vetores que representam palavras são frequentemente denominados de *embeddings*, embora muitas vezes esse termo seja usado de maneira mais restrita para se referir apenas aos **vetores densos** como é o caso do Word2Vec, Glove e Fasttext (Seção 10.3), e não aos vetores esparsos como o TF-IDF e o PMI (Seção 10.2).



10.1.3 Calculando a similaridade entre vetores semânticos

Uma tarefa bastante comum do processamento de linguagem natural consiste em calcular a similaridade entre vetores de documentos ou vetores de palavras, seja para estabelecer uma métrica de semelhança entre dois textos ou para se ter uma medida de equivalência entre duas palavras. Para tanto, faz-se necessário o emprego de alguma medida de similaridade entre vetores.

A **medida do Cosseno**, também conhecida por **distância do Cosseno**, é sem dúvida uma das mais clássicas da área de PLN. Essa medida calcula a distância entre dois vetores no espaço vetorial a partir do valor do cosseno do ângulo compreendido entre eles. Se o ângulo compreendido for zero (ambos os vetores apontam para o mesmo lugar), a medida resultará no valor 1. Para um ângulo diferente de zero, o valor resultante será inferior a 1. Para vetores ortogonais⁸, o valor será zero. Se os vetores apontarem em direções contrárias, o valor será -1. Logo, a medida do Cosseno encontra-se no intervalo fechado entre [-1, 1]. Contudo, como os valores de frequência de termos (palavras) são positivos, o cosseno desses vetores encontra-se no intervalo entre [0, 1], sendo que quanto mais próximo de 1 for o valor, maior é a similaridade entre os vetores.

A **distância do Cosseno** é dada pelo produto escalar entre dois vetores, também chamado de produto interno na Álgebra Linear. Sejam x e y dois vetores semânticos n -dimensionais, ambos representando documentos ou ambos representando palavras, o produto escalar entre x e y é definido pela Equação 10.1:

$$\langle x, y \rangle = x \cdot y = \|x\| \|y\| \cos \theta$$

$$\cos \theta = \frac{x \cdot y}{\|x\| \|y\|}$$

onde $\langle x, y \rangle$ representa a somatória dos produtos das coordenadas correspondentes dos vetores x e y , calculada com base na Equação 10.1, e $\|v\|$ representa o comprimento do vetor v definido pela Equação 10.2:

$$x \cdot y = \sum_{i=1}^n x_i y_i = x_1 y_1 + x_2 y_2 + x_3 y_3 + \dots + x_n y_n \quad (10.1)$$

$$\|v\| = \sqrt{\sum_{i=1}^n v_i^2} \quad (10.2)$$

Em suma, a distância do Cosseno entre dois vetores x e y é definida pela Equação 10.3:

$$\cos = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Para fins de exemplificação, consideremos os dois textos do Exemplo 10.1:

Exemplo 10.1.

1. Eu adorei o filme. Foi incrível.
2. Eu detestei o filme. Foi horrível.

Podemos transformar esses textos em vetores que representam a frequência de ocorrência de cada palavra. Para o conjunto de palavras {eu, adorei, detestei, filme, incrível, horrível}, temos os vetores semânticos a e b , representando o texto 1 e o texto 2, respectivamente:

⁸Dois vetores são ortogonais se o Produto Escalar entre eles é nulo (zero), conforme veremos na Equação 10.1.



$$a = [1, 1, 0, 1, 1, 0]$$

$$b = [1, 0, 1, 1, 0, 1]$$

Inicialmente, vamos calcular a somatória dos produtos entre as coordenadas correspondentes de a e b , conforme a Equação 10.1:

$$a \cdot b = 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 = 2$$

Em seguida, calculamos o comprimento de cada vetor com base na Equação 10.2:

$$\|a\| = \sqrt{1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 0^2} = \sqrt{4} = 2$$

$$\|b\| = \sqrt{1^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2} = \sqrt{4} = 2$$

Por fim, substituímos os valores já calculados na Equação 10.3, para obtermos o cosseno do ângulo entre os vetores a e b :

$$\text{Cos}(a, b) = \frac{2}{4} = 0,5$$

A similaridade entre os vetores a e b é diretamente proporcional ao cosseno do ângulo entre os dois vetores, ou seja, 0,5.

A medida do Cosseno é menos sensível à frequência de ocorrência das palavras em um *corpus* do que outras medidas de similaridade, como a Distância Euclidiana. Isso significa que as palavras menos frequentes não terão um peso desproporcional no cálculo da similaridade entre os vetores. Essa é principal razão que faz com que essa medida seja tão frequente na Semântica Distribucional e muito usada para calcular a similaridade entre vetores de palavras.

10.2 Vetores esparsos

Vimos nas Seções 10.1.1 e 10.1.2 que as matrizes termo-documento e termo-contexto associam a frequência de ocorrência de cada termo ao documento ou contexto em que ocorrem. No entanto, a frequência simples de um termo (isto é, o número de vezes que ele ocorre) é pouco discriminativa, já que algumas palavras (como “porque”, “durante”, “após”, “sobre” etc.) são bastante comuns e não caracterizam nenhum documento ou contexto em particular. Abordagens mais avançadas como é o caso das medidas TF-IDF (do inglês, *Term Frequency-Inverse Document Frequency*) e PMI (do inglês, *Pointwise Mutual Information*), costumam ser mais eficazes do que a simples frequência de um termo para discriminar o conteúdo de um documento ou um contexto. Como muitos termos nunca ocorrem em alguns documentos de uma coleção ou nunca aparecem em certos contextos, frequentemente, essas medidas levam a vetores com muitas dimensões e esparsos, ou seja, com muitos valores nulos (zeros). Por essa razão, as matrizes que se utilizam dessas medidas para atribuir valores aos termos são comumente chamadas de **vetores esparsos**. As medidas TF-IDF e PMI serão detalhadas nas Seções 10.2.1 e 10.2.2, respectivamente. Em seguida, na Seção 10.2.3, apresentamos o LSA (do inglês, *Latent Semantic Analysis*), um modelo muito adotado em PLN com o objetivo de reduzir a dimensionalidade de um espaço multidimensional criado com o uso do TF-IDF ou PMI.



10.2.1 Atribuindo pesos aos termos da matriz termo-documento com TF-IDF

A medida TF-IDF representa uma alternativa mais eficiente do que a contagem de termos para atribuir valores aos termos de uma matriz termo-documento. Ela atribui um peso para cada termo de um documento multiplicando a frequência do termo no documento (TF) pelo inverso da frequência do termo em todos os documentos de um *corpus* ou coleção (IDF). Dessa maneira, um termo que ocorre muitas vezes em um documento, mas não em muitos documentos da coleção, terá um peso mais alto, enquanto um termo que ocorre em muitos documentos terá um peso mais baixo.

A frequência de um termo (TF) mede a sua importância em um documento. Ela é calculada com base no número de ocorrências de um termo t em um documento d , dividido pelo total de termos do documento d (conforme a Equação 10.3). Essa medida é importante porque, em geral, as palavras que aparecem com mais frequência em um documento são mais relevantes para discriminar o seu conteúdo. Porém, TF somente não é suficiente para identificar as palavras mais importantes de um documento, pois algumas palavras podem ser muito frequentes em muitos documentos e, portanto, não auxiliam na discriminação do seu conteúdo. A frequência inversa no documento (IDF) é, então, fundamental, para atribuir um peso maior às palavras que são frequentes mas ocorrem em poucos documentos de uma coleção.

A frequência inversa no documento (IDF) mede, portanto, a importância relativa de uma palavra em uma coleção de documentos. Ela é calculada dividindo-se o número total de documentos da coleção pelo número de documentos que contêm a palavra em questão e tomando o logaritmo desse resultado. Seja N o número de documentos de uma coleção, o IDF de um termo t é definido pela Equação 10.4. Em outras palavras, o IDF mede a raridade de uma palavra em um conjunto de documentos. Essa medida é importante porque palavras que aparecem em poucos documentos têm um maior poder de discriminação do conteúdo de um documento.

$$tf(t, d) = \frac{\text{numero_ocorrencias}(t, d)}{\text{total_termos}(d)} \quad (10.3)$$

$$idf(t) = \log \left(\frac{N}{\text{total_documentos}(t)} \right) \quad (10.4)$$

Assim, o TF-IDF de um termo t de um documento d é dado pelo produto entre o valor TF (definido pela Equação 10.3) e o valor de IDF (definido pela Equação 10.4), conforme a Equação 10.5:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) * \text{idf}(t) \quad (10.5)$$

Quanto menor o número de documentos que contêm determinado termo, maior será o TF-IDF daquele termo. Em suma, termos que aparecem com frequência em muitos documentos recebem um peso menor do que os termos mais específicos de um determinado documento. TF-IDF é uma medida bastante versátil e amplamente utilizada em várias tarefas que envolvem o processamento de textos. Alguns exemplos de aplicação mais comuns que podem ser citados são:

- Recuperação de informação: a matriz TF-IDF é usada para classificar e recuperar documentos relevantes com base em termos de pesquisa específicos. Os termos mais importantes, com os valores mais altos de TF-IDF, são usados para classificar a relevância de cada documento.



- Análise de sentimentos: assim como na recuperação de informação, a medida TF-IDF é usada para identificar as palavras mais importantes em um texto e, posteriormente, a polaridade do sentimento associado ao texto (como positiva, negativa ou neutra) é determinada com base nessas palavras.
- Agrupamento de documentos: a matriz TF-IDF também pode ser usada para agrupar documentos que compartilham termos em comum. Os documentos que têm um alto valor de TF-IDF para os mesmos termos são agrupados juntos.
- Sumarização de textos: na sumarização automática a medida TF-IDF pode ser usada para identificar as palavras-chave ou os tópicos mais relevantes de um texto, para gerar o seu resumo.

10.2.2 Atribuindo pesos aos termos da matriz termo-contexto com PMI

Uma forma mais eficaz de pesar os termos de uma matriz termo-contexto, comparada à simples contagem de coocorrência de termos, é usar a medida PMI (do inglês, *Pointwise Mutual Information*). PMI é uma medida estatística que auxilia na identificação de palavras associadas. Dito de outra forma, ela mede qual é a probabilidade que dois termos ocorram juntos um do outro em relação à probabilidade de cada termo ocorrer de forma independente. Por exemplo, o termo “inteligência artificial” tem um significado específico quando as palavras “inteligência” e “artificial” aparecem juntas em um texto. Quando ocorrem isoladamente, essas duas palavras constroem outros significados.

Formalmente, o PMI entre um termo alvo x e um termo contexto y é definido pela Equação 10.6.

$$pmi(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)} = \log_2 \frac{p(x|y)}{p(x)} = \log_2 \frac{p(y|x)}{p(y)} \quad (10.6)$$

onde:

$$p(t) = \frac{\text{frequencia}(t)}{\text{total_termos}}$$

A medida PMI leva em consideração, tanto a probabilidade conjunta dos termos (numerador da Equação 10.6), quanto a distribuição geral de cada termo no *corpus* em que estão sendo analisados (denominador da equação). Vale lembrar que a probabilidade de dois eventos independentes ocorrerem é dada pelo produto das probabilidades dos dois eventos⁹.

A medida PMI é simétrica, ou seja, $pmi(x, y) = pmi(y, x)$. Assim, essa medida não considera a ordem de ocorrência das palavras, ou seja, “inteligência artificial” e “artificial inteligência” terão a mesma probabilidade.

Os valores de PMI podem ser negativos ou positivos. O valor de PMI será 0 quando x e y forem independentes, indicando que não há qualquer associação entre os dois termos. Embora o valor possa ser negativo ou positivo, o resultado esperado é sempre positivo. Valores negativos indicam que os termos estão ocorrendo com menos frequência do que esperaríamos que eles ocorressem ao acaso. A medida PMI é máxima quando x e y estão perfeitamente associados (isto é, $p(x|y)$ ou $p(y|x) = 1$). Ou seja, quando os termos têm alta probabilidade conjunta e cada termo tem uma baixa probabilidade de ocorrer isoladamente, como, por exemplo, o termo “fiada” que provavelmente ocorre com maior frequência com o termo “conversa” (na expressão “conversa fiada”) do que isoladamente.

⁹Os dois últimos termos da Equação 10.6 são considerados equivalentes pelo Teorema de Bayes.



O fato de a medida PMI poder assumir valores positivos ou negativos e sem limites definidos torna mais difícil a interpretação dos resultados. Uma maneira de contornar esse problema é substituir os valores negativos por 0, aplicando uma variação de PMI, denominada PPMI (*Positive Pointwise Mutual Information*), definida pela Equação 10.7:

$$ppmi(x, y) = \max\left(\log_2 \frac{p(x, y)}{p(x)p(y)}, 0\right) \quad (10.7)$$

A medida PPMI é motivada pela observação de que valores negativos de PMI tendem a não ser confiáveis, a menos que tenhamos um *corpus* muito grande e expressivo. Além disso, evita que tenhamos que lidar com valores negativos tendendo ao infinito ($-\infty$) para termos que nunca ocorrem juntos (isto é, $p(x, y) = 0$). Essa variante da medida PMI é mais robusta e precisa para a construção de modelos semânticos distribucionais, uma vez que ela remove os valores negativos de PMI que podem prejudicar a precisão dos modelos. Em geral, quanto maior for o *corpus*, mais confiável será a medida de associação.

Embora PPMI seja uma medida útil para quantificar a força da relação entre dois termos em um *corpus*, ela pode ser muito sensível à frequência dos termos, podendo superestimar a associação entre palavras raras e subestimar a associação entre palavras comuns. Outra limitação das medidas PMI e PPMI é que, por depender somente da coocorrência de palavras no *corpus*, não levam em consideração a posição das palavras no texto. Isso pode ser um problema em situações em que a ordem das palavras é importante como é o caso das expressões multipalavras.

Medidas de associação de termos como PMI e PPMI são úteis em uma variedade de aplicações de processamento de linguagem natural, conforme exemplificamos a seguir:

- Construção de modelos de linguagem: são úteis para identificar palavras que coocorrem com frequência em *corpora*, informação relevante que pode ser incorporada em modelos de linguagem, como os modelos Word2Vec e Glove.
- Agrupamento de tópicos: são úteis para o agrupamento de documentos por tópicos com base na identificação de palavras que são altamente correlacionadas a determinado tópico.
- Recomendação de conteúdo/produto: são importantes para recomendar produtos com base nas preferências do usuário, auxiliando na identificação de palavras frequentemente associadas a conteúdos/produtos específicos. Essas associações de palavras são posteriormente usadas para fazer recomendações personalizadas.
- Tradução automática: são úteis para a identificação de palavras que ocorrem frequentemente em um mesmo contexto de tradução.
- Análise de sentimento: são úteis para analisar a opinião ou sentimento expresso em um texto, a partir da identificação de palavras-chave que são altamente correlacionadas com um determinado sentimento, como “bom” ou “ruim”.

10.2.3 Reduzindo a dimensionalidade com LSA

Os modelos TF-IDF e PMI/PPMI apresentados nas Seções 10.2.1 e 10.2.2 frequentemente geram matrizes esparsas contendo muitas células com valor nulo (zero), dado que muitos termos têm baixa frequência em muitos documentos (no caso da matriz termo-documento) ou não ocorrem em muitos contextos em um *corpus* (no caso da matriz termo-contexto). A esparsidade de uma matriz pode afetar significativamente o desempenho dos algoritmos que a processam, levando a um aumento no tempo de processamento, devido à necessidade de acessar todos os valores armazenados, inclusive os valores nulos. Por conter muitos zeros,



a matriz esparsa ainda demanda muita memória para poder armazenar todos os valores, mesmo com *corpora* relativamente pequenos.

Para lidar com esse problema decorrente do grande número de dimensões, é comum o uso de técnicas que permitam reduzir a dimensionalidade de uma matriz. A Análise de Semântica Latente (em inglês, *Latent Semantic Analysis* – LSA) ou, ainda, Indexação de Semântica Latente (em inglês, *Latent Semantic Indexing* – LSI), como é chamada na área de recuperação de informação, pode ser aplicada para reduzir a dimensionalidade de um espaço multidimensional. O objetivo principal do LSA é reduzir a dimensão da matriz original, diminuindo a importância de valores singulares menores. Isso ajuda a eliminar ruídos e a capturar as relações semânticas subjacentes entre as palavras.

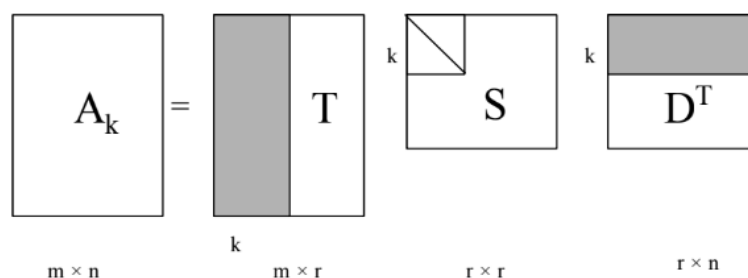
O LSA pode ser aplicado tanto em matrizes de termo-documento como em matrizes de termo-contexto (ou palavra-palavra). A ideia por trás desse modelo é que existe alguma estrutura semântica latente subjacente nos dados, que é ocultada em certa medida pela incerteza na escolha das palavras. Para estimar essa estrutura latente, o LSA utiliza uma técnica de análise matricial, da Álgebra Linear, chamada Decomposição em Valores Singulares (em inglês, *Singular Value Decomposition* – SVD) (Golub; Reinsch, 1970), que permite determinar o conjunto de padrões que descreve um documento ou contexto. A SVD encontra os principais eixos de variância no espaço vetorial. Assim, ao decompor uma matriz termo-documento em valores singulares, por exemplo, é possível identificar o conjunto de termos padrão que descreve qualquer documento daquela coleção. Na decomposição, somente as k dimensões mais importantes da matriz são mantidas. Como resultado, tem-se um espaço semântico no qual termos e documentos semelhantes são colocados próximos uns dos outros.

De maneira geral, decompor uma matriz A de ordem $m \times n$ (para m diferente de n) significa fatorar essa matriz em três matrizes fatores, tal que o produto dessas matrizes permite recompor a matriz A original. Formalmente, a Decomposição em Valores Singulares de uma matriz $m \times n$ de valores reais A é igual ao produto de três matrizes fatores T , S e D^T , conforme segue:

$$A = TSD^T$$

onde T e D são colunas (vetores) ortonormais e S é a matriz diagonal de valores singulares. As colunas de T e D representam os vetores singulares à esquerda (vetores de palavras) e à direita (vetores de documentos ou contextos) de A , respectivamente. A matriz A_k $m \times n$ é construída a partir dos k maiores valores singulares (Figura 10.3).

Figura 10.3: Decomposição em Valores Singulares



Na Figura 10.3, m representa o número de palavras (termos), n é o número de documentos (ou contextos), k é a dimensão desejada do espaço de conceitos reduzido, ou seja, representa

o total de valores singulares mais significativos a serem mantidos na matriz reduzida e r é o ranqueamento da matriz A ($A(\min(m, n))$).

Durante a construção da matriz termo-documento ou termo-contexto, os valores nas células da matriz representam a frequência com que uma palavra ocorre em um determinado documento (ou contexto). Essa coocorrência de palavras gera padrões de associação. Dessa forma, na matriz T resultante da decomposição SVD, as palavras que possuem semelhanças semânticas são definidas através dos padrões de coocorrência entre palavras nos documentos (ou contextos) originais. Mais especificamente, a matriz T contém vetores de palavras que representam as relações semânticas latentes entre as palavras no espaço de conceitos. Cada vetor é uma combinação linear ponderada desses conceitos latentes. As palavras que compartilham padrões semânticos semelhantes terão vetores de palavras semelhantes no espaço de conceitos. A similaridade semântica entre palavras pode ser calculada utilizando a **distância do Cosseno** (Seção 10.1.3) entre os vetores de palavras no espaço de conceitos. Palavras que são semanticamente similares terão vetores que apontam em direções semelhantes e, portanto, terão valores de similaridade próximos de 1. Essa abordagem permite capturar relações semânticas latentes entre as palavras, mesmo que elas não tenham aparecido juntas frequentemente nos documentos originais. Isso é especialmente útil para encontrar sinônimos, identificar tópicos subjacentes e lidar com a chamada “maldição da dimensionalidade” em dados textuais.

A matriz S é obtida com base no produto entre a matriz original A e a sua matriz transposta (A^T), resultando em uma matriz de dimensão $n \times n$ (se a matriz original for $m \times n$). Em seguida, calculam-se os autovalores e autovetores da matriz resultantes do produto $A * A^T$. Os autovalores representam a variabilidade dos dados em direções específicas no espaço. Os valores singulares são obtidos a partir dos autovalores. Eles são a raiz quadrada dos autovalores positivos, ordenados em ordem decrescente. A matriz S é então construída como uma matriz diagonal, onde os valores singulares são colocados na diagonal principal e os demais elementos são zeros. Quanto maior o valor singular, maior a contribuição desse conceito latente na representação dos dados originais. A redução da dimensionalidade envolve manter apenas os primeiros k valores singulares mais significativos, onde k é a dimensão desejada no espaço de conceitos reduzido.

A matriz D^T (de documentos ou contextos para conceitos), por sua vez, é calculada utilizando os autovetores associados à matriz resultante do produto $A * A^T$. Os autovetores são os mesmos que compõem a matriz T . Eles são normalizados para terem comprimento igual a 1. Isso é importante para garantir que as informações de magnitude não sejam distorcidas. A matriz D^T é construída colocando os autovetores normalizados como colunas, onde cada coluna representa um vetor de documentos ou de contextos no espaço de conceitos.

Embora o LSA seja uma técnica poderosa para análise de texto e redução de dimensionalidade, o modelo também possui algumas limitações que devem ser consideradas. Por exemplo, ao tratar os termos como entidades independentes, ignorando as relações de contexto mais complexas que ocorrem em linguagem natural, nuances de significado que dependem do contexto podem não ser totalmente capturadas. Tratando as palavras de maneira independente, o modelo também não captura a estrutura sintática da linguagem.

Outra limitação importante é que o LSA tende a ter dificuldade em lidar com documentos muito curtos, uma vez que a coocorrência de termos relevantes é menor e a representação no espaço de conceitos pode ser menos robusta. O LSA é uma abordagem estática, o que significa que não é capaz de lidar bem com mudanças no significado das palavras ao longo do tempo ou em diferentes contextos. Modelos mais recentes, como os baseados em redes



neurais, podem lidar melhor com essas nuances.

Embora o LSA capture informações semânticas latentes, os conceitos extraídos nem sempre são facilmente interpretáveis por seres humanos. Isso dificulta a compreensão do que exatamente está sendo capturado em cada dimensão reduzida do espaço de conceitos. Além disso, ele não é capaz de capturar nuances semânticas mais complexas, como a ambiguidade.

Em resumo, o LSA é uma técnica valiosa para muitas tarefas de processamento de linguagem natural, mas é importante estar ciente de suas limitações e considerar outras abordagens, como modelos baseados em redes neurais, para lidar com algumas das desvantagens mencionadas.

10.3 Vetores densos estáticos

Vimos na Seção 10.2 como representar uma palavra por meio de um vetor esparsos e com muitas dimensões, correspondentes às palavras do vocabulário ou aos documentos de uma coleção. Nesta seção, introduziremos uma representação de palavras mais robusta, conhecida por *embeddings*, de **vetores densos** e menores, com dimensões variando entre 50-1000. Essas dimensões não possuem uma interpretação clara do seu significado (Jurafsky; Martin, 2023).

Os vetores são densos, ou seja, seus valores são números reais positivos ou negativos, ao invés de contagens esparsas, na maioria das vezes zeros, como é o caso dos vetores esparsos vistos na Seção 10.2. Vetores densos (daqui em diante, *embeddings*) capturam melhor as relações semânticas e contextuais entre as palavras do que os vetores esparsos. Por exemplo, na representação vetorial esparsa, sinônimos como “alfabeto” e “abecedário” muito provavelmente têm dimensões distintas e não relacionadas, pois esse tipo de modelo pode falhar ao capturar a similaridade entre palavras que estão no contexto de “alfabeto” e “abecedário”. Essa é uma das razões que faz com que os *embeddings* apresentem melhor desempenho em tarefas de PLN do que os vetores esparsos.

Os *embeddings* são aprendidos a partir de *corpora* por meio de algoritmos de aprendizado de máquina supervisionado ou não supervisionado, por exemplo, usando redes neurais artificiais como é o caso do modelo Word2Vec (Seção 10.3.1), ou, ainda, usando representação estatística da matriz de coocorrência de termos, como é o caso do modelo GloVe (Seção 10.3.3).

Os vetores de *embeddings* podem ser **estáticos** ou **dinâmicos**. Os *embeddings* **estáticos** permanecem fixos uma vez aprendidos, ou seja, eles não podem ser ajustados ou modificados para uma tarefa específica. Ao contrário desses, os *embeddings* **dinâmicos** podem ser ajustados em tarefas específicas, se adaptando às nuances específicas da tarefa e ao contexto atual. A escolha entre essas abordagens depende das necessidades da aplicação, do domínio e das características das tarefas em que os *embeddings* serão utilizados.

Nesta seção o foco será apenas na descrição dos *embeddings* **estáticos**, mais especificamente, nos modelos Word2Vec (Seção 10.3.1), FastText (Seção 10.3.2) e GloVe (Seção 10.3.3). Os *embeddings* **dinâmicos** como os modelos ELMo, BERT e GPT são abordados no Capítulo 15.

Os *embeddings* **estáticos** podem ser definidos para diversos tipos de unidades de representação, incluindo palavras, caracteres, subpalavras, sentenças e até mesmo textos com várias sentenças. Por exemplo, considere as sentenças do Exemplo 10.2:

Exemplo 10.2.



1. Vou ao **banco** sacar dinheiro.
2. Adoro sentar no **banco** da praça.

Considerando o contexto da sentença e o senso comum, o termo “banco” na primeira sentença corresponde a uma *instituição financeira* cujo significado é distinto do “banco” da segunda sentença que corresponde a um *assento*. Neste caso, os *embeddings* estáticos definem um mesmo vetor para representar a palavra “banco” nas duas sentenças, independente do contexto.

Formalmente, as unidades de representação e seus vetores são representados por uma matriz

$$M \in \mathbb{R}^{|V| \times d}$$

onde $|V|$ é o tamanho do vocabulário V e d é a dimensão do *embedding*, em geral, um valor entre 50-1000. Cada linha da matriz contém o vetor estático da unidade de representação $u_i \in V$.

Embora a ideia de representar elementos de um texto usando vetores no espaço multidimensional não seja tão recente (vide, por exemplo, (Joos, 1950), (Harris, 1954), (Firth, 1957a), (Osgood; Suci; Tenenbaum, 1957)), somente a partir de 2013 os *embeddings* começaram a ser muito utilizados, com o desenvolvimento e a disponibilização do modelo Word2Vec (Mikolov et al., 2013b), conforme explicado a seguir.

10.3.1 Word2Vec

O Word2Vec é uma técnica de aprendizado de unidades de representações distribuídas proposta por Mikolov et al. (2013b), que tem como objetivo capturar a semântica e a relação entre unidades de representação em um *corpus*, aprendendo *embeddings* estáticos para cada palavra presente no vocabulário de treino (Jurafsky; Martin, 2023). O modelo é baseado na ideia de que palavras que ocorrem em contextos semelhantes têm significados semelhantes. Portanto, ele explora a distribuição de palavras em grandes *corpora* para aprender representações vetoriais que capturam esses padrões.

O Word2Vec possui duas arquiteturas principais: CBOW – *Continuous Bag-of-Words* e Skip-gram. Na arquitetura CBOW, o modelo tenta prever uma palavra-alvo com base em um contexto de várias palavras de entrada. O contexto é definido por um conjunto de palavras vizinhas da palavra-alvo. Ao contrário do CBOW, na arquitetura Skip-gram o modelo tenta prever o contexto (palavras vizinhas) de uma palavra-alvo. Dito de outra forma, o Skip-gram tenta encontrar as palavras que normalmente aparecem no contexto da palavra-alvo. Esse método é, em geral, mais lento de treinar, mas muitas vezes gera representações mais precisas.

As subseções 10.3.1.1 e 10.3.1.2 explicam as arquiteturas CBOW e Skip-gram, respectivamente.

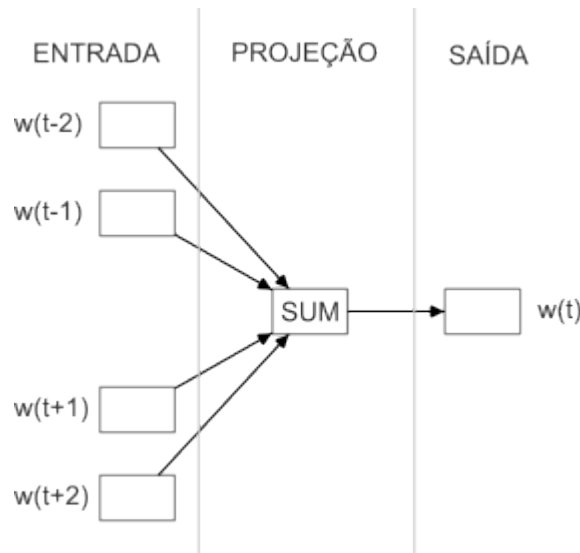
10.3.1.1 CBOW

A ideia principal por trás do CBOW é prever a palavra-alvo com base no contexto. O contexto é definido por um conjunto de palavras vizinhas da palavra-alvo. Por exemplo, na frase “Vou ao **banco** sacar dinheiro.”, o CBOW tentaria prever a palavra “banco” com base no contexto [Vou, ao, sacar, dinheiro], sendo [Vou, ao] o conjunto de palavras anteriores a palavra-alvo e [sacar, dinheiro] o conjunto de palavras posteriores. Dessa forma, o modelo aprende a associação entre as palavras do contexto e a palavra-alvo.



A arquitetura do CBOW é baseada em uma rede neural de uma única camada oculta (Figura 10.4). A camada de entrada possui um neurônio para cada palavra no contexto. A camada oculta (camada de projeção) tem um número fixo de neurônios, que é um hiper-parâmetro definido antes do treinamento. A predição da palavra-alvo $w(t)$ é feita base nas palavras de histórico e palavras futuras e o resultado calculado na saída da projeção é a média de todos os vetores de todas as palavras presentes no contexto da camada de entrada. Este modelo pode prever tanto palavras anteriores como posteriores a uma palavra-alvo em um contexto.

Figura 10.4: Arquitetura CBOW. Prediz a palavra-alvo com base no contexto.



Fonte: Adaptado de (Mikolov et al., 2013b)

O treinamento do modelo utilizando a arquitetura CBOW é dada pela Equação 10.8, onde N é o tamanho da camada de projeção, isto é, o número de neurônios da camada oculta; D é a dimensão dos vetores; e V é o vocabulário. A primeira parte da fórmula, $N \times D$, refere-se à multiplicação de matrizes durante o treinamento. A segunda parte, $D \times \log_2(V)$, está relacionada ao *softmax*, usado para calcular a probabilidade de cada unidade de representação no vocabulário V ser a representação de destino, com base nos vetores das representações de entrada.

$$Q = N \times D + D \times \log_2(V) \quad (10.8)$$

O *softmax* é uma generalização da função *sigmoid* que converte um vetor numérico em um vetor de probabilidades de possíveis saídas (Jurafsky; Martin, 2023), com valores dentro do intervalo $[0,1]$ perfazendo um somatório de 1, como representado na Equação 10.9.

$$\text{softmax}(v_i) = \frac{\exp(v_i)}{\sum_{j=1}^{|V|} \exp(v_j)} \quad (10.9)$$

Assim, a função *softmax* tem por objetivo converter um vetor numérico em um vetor normalizado de probabilidades.



10.3.1.2 Skip-gram

Enquanto o modelo CBOW prevê a representação de uma unidade (e.g. palavra) com base nos contextos em que ela ocorre em um *corpus* de treinamento, o modelo Skip-gram tenta prever o contexto (ou as representações vizinhas) a partir de um alvo. Considerando as representações como palavras. Nesta arquitetura, cada palavra é uma entrada para uma rede neural similar à arquitetura do CBOW, com camada de projeção para prever palavras dentro de um determinado intervalo antes e depois da palavra de entrada, conforme Figura 10.5. Utilizando o mesmo exemplo anterior, o Skip-gram recebe a palavra “banco” como entrada e tenta prever o contexto “Vou, ao, sacar, dinheiro”. Essa abordagem permite que o modelo aprenda a representação de uma palavra, considerando as palavras que normalmente aparecem ao seu redor.

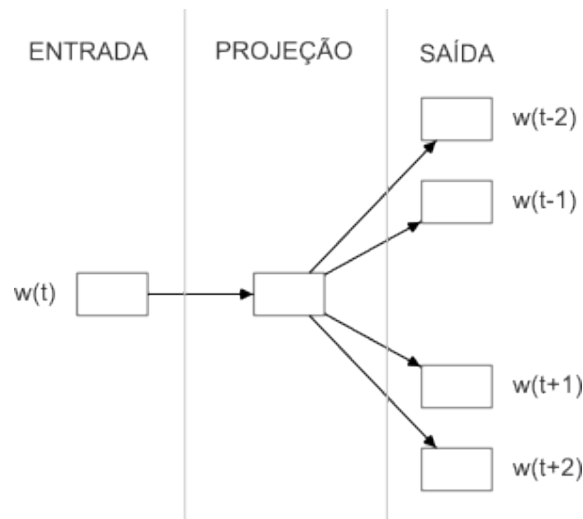
Nesta arquitetura do Skip-gram, aumentar o tamanho do intervalo resulta em vetores melhores; entretanto, isso também aumenta a complexidade computacional do algoritmo. A complexidade do treinamento do modelo utilizando a arquitetura Skip-gram é proporcional à Equação 10.10, onde C é a distância máxima entre as palavras.

$$Q = C \times (D + D \times \log_2(V)) \quad (10.10)$$

O Skip-gram parte da ideia de que é mais provável que uma palavra ocorra próxima a uma palavra-alvo se seu vetor for similar ao vetor desta palavra-alvo. Esse cálculo de similaridade é baseado na similaridade do Cosseno (Seção 10.1.3).

Em suma, o Skip-gram treina um classificador que, dada uma palavra-alvo e seu contexto, calcula uma probabilidade do quão similares são o contexto e a palavra-alvo. Quanto mais distantes forem os vetores do contexto e da palavra-alvo, menos relacionados são a palavra-alvo e o contexto e, portanto, menor será o peso daquele contexto no treinamento (Mikolov et al., 2013b).

Figura 10.5: Arquitetura Skip-gram. Prediz o contexto com base na palavra-alvo.



Fonte: Adaptado de (Mikolov et al., 2013b)

10.3.1.3 Treinamento do Word2Vec

O processo de treinamento do Word2Vec envolve a criação de um vocabulário a partir do *corpus*. Durante o treinamento, o modelo ajusta os valores desses vetores para maximizar a capacidade de prever a palavra-alvo ou o contexto, dependendo da arquitetura escolhida.

Embora considerada mais complexa que a CBOW, a abordagem Skip-gram é mais comumente utilizada com a técnica de amostragem negativa (*Skip-gram with Negative Sampling* – SGNS). Na amostragem negativa apenas um subconjunto de palavras negativas (palavras não contextuais) é selecionado para a atualização dos pesos, ao invés de ajustar os pesos de todas as palavras no *corpus* a cada iteração. Isso torna o treinamento mais eficiente computacionalmente, aprendendo boas representações, especialmente para palavras mais frequentes (Mikolov et al., 2013b).

Uma vez treinado, o modelo Word2Vec é capaz de fornecer representações vetoriais para palavras (*embeddings*), nas quais palavras semanticamente similares são mapeadas para regiões próximas do espaço vetorial.

10.3.1.4 Limitações do Word2vec

Apesar das vantagens de se utilizar o modelo Word2Vec, é importante destacar que o mesmo possui algumas limitações. Embora o modelo Word2Vec capture as relações semânticas entre palavras, as palavras com múltiplos sentidos podem ser ambíguas dependendo do contexto, dificultando a precisão da representação.

Outra limitação do Word2Vec é que ele não lida bem com palavras raras ou fora do vocabulário; conseqüentemente essas palavras podem gerar representações vetoriais pouco confiáveis. Além disso, a arquitetura Skip-gram não captura explicitamente relações sintáticas, tais como relações entre adjetivos, verbos, por exemplo. Sendo assim, para tarefas que exigem um entendimento mais profundo da estrutura gramatical, outros modelos ou técnicas podem ser mais adequados.

Além da questão da representação de palavras do vocabulário, o modelo não leva em consideração questões morfológicas e ignora a estrutura interna das palavras, o que é uma limitação especialmente para línguas morfolologicamente ricas, ou seja, que possuem uma grande variedade de morfemas que podem para expressar diferentes funções gramaticais e que podem ser adicionados, alterados ou combinados para criar diferentes formas de palavras e estruturas gramaticais dentro da língua, como Árabe ou Finlandês.

Por fim, o Word2Vec não lida diretamente com a concatenação de palavras como uma única unidade. Por exemplo, o modelo não conseguiria reconhecer a palavra “pontapé”, se ela não estivesse presente no vocabulário, mesmo se o vocabulário de treino contivesse as palavras “ponta” e “pé”. No entanto, existem variações do modelo que permitem capturar informações contextuais mais ricas, como o modelo Fasttext descrito a seguir.

10.3.2 Fasttext

O Fasttext é uma extensão do modelo Word2Vec criada pelo grupo *Facebook AI Research* (FAIR), que amplia o conceito do Skip-gram, no qual cada palavra é representada como uma combinação de *n-gramas* de caracteres. O modelo leva em consideração não apenas as palavras individuais, mas também as subpalavras (morfemas) que compõem as palavras. Isso permite que o Fasttext consiga melhor representar as palavras fora do vocabulário e conseqüentemente, captura informações contextuais mais ricas, especialmente em idiomas com aglutinação e morfologia complexa.

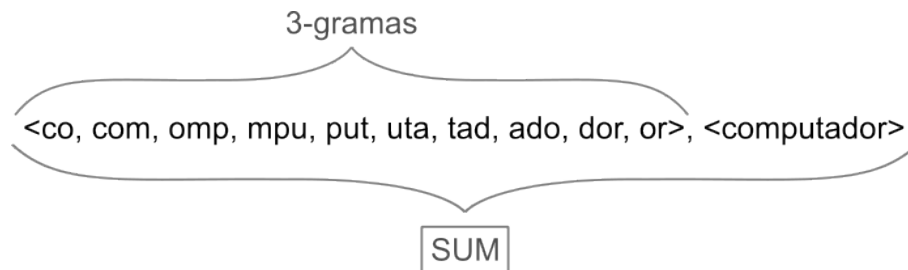


A principal diferença em relação ao Word2Vec está no tratamento das palavras, ou seja, ao invés de tratá-las como uma única unidade, o FastText as divide em n -gramas menores, como n -gramas de caracteres, e as representa como a soma de seus n -gramas.

Mais especificamente, cada palavra w é representada como um saco de caracteres (*bag of characters*). Para cada palavra, são adicionados os símbolos ‘<’ e ‘>’ denotando o início e fim das palavras, respectivamente. Isso permite a distinção entre prefixos, sufixos e outras sequências de caracteres. Além disso, a própria palavra w é incluída no conjunto de n -gramas, para aprender um *embedding* para cada palavra além dos n -gramas dos caracteres.

A Figura 10.6 apresenta um exemplo de saco de caracteres da palavra “computador”, considerando $n = 3$ para o tamanho dos n -gramas. É importante ressaltar que a sequência “<computador>” tem como 3-gramas de caracteres palavras do português como “com” e “dor”; entretanto, as sequências “com” e “dor” são diferentes dos trigramas “com” e “dor” da palavra “<computador>”. Também é possível capturar como informação o prefixo “com” e sufixo “dor” da palavra.

Figura 10.6: Exemplo de representação de trigramas adotados pelo modelo FastText.



Segundo Bojanowski et al. (2017), o modelo extrai os n -gramas para n maior ou igual a 3 e menor ou igual a 6. Diferentes conjuntos de n -gramas podem ser considerados, por exemplo, tomando todos os prefixos e sufixos.

Em termos arquiteturais, o FastText também possui duas arquiteturas assim como o Word2Vec: CBOW – *Continuous Bag-of-Words* e Skip-gram.

A Figura 10.7 exemplifica o funcionamento do FastText com a arquitetura CBOW. Essa arquitetura prediz a palavra-alvo com base no contexto. Cada *embedding* é gerado a partir dos vetores dos n -gramas, neste exemplo, sendo bigramas.

10.3.2.1 Limitações do FastText

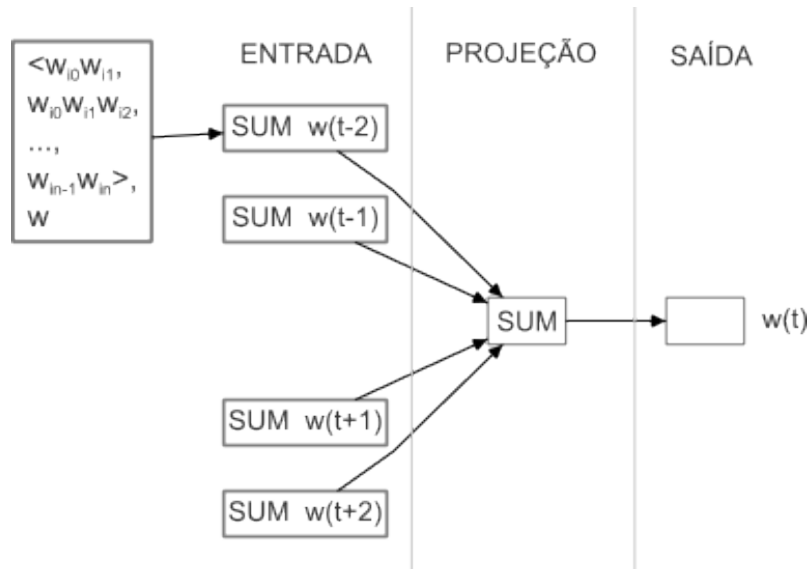
Embora o FastText seja um modelo eficiente e útil para muitas tarefas de processamento de linguagem natural, ele também possui algumas limitações. A primeira é a representação limitada do contexto. Nesse modelo, os textos são representados como a soma das representações de vetor de palavras individuais. Isso significa que o FastText não captura informações de ordem ou estrutura sintática mais complexa nos textos.

Outra limitação importante é que o modelo é sensível ao tamanho do vocabulário. Em outras palavras, o FastText requer uma representação de vetor de palavras predefinida para cada palavra no vocabulário. Isso pode levar a problemas de dimensionamento, quando se lida com vocabulários muito grandes, pois o espaço vetorial se torna maior e o treinamento e a inferência podem se tornar mais lentos e exigentes em recursos computacionais.

Embora o FastText seja eficaz em muitos idiomas, pode não funcionar tão bem em idiomas com morfologia muito complexa, onde as palavras se desdobram em várias formas com



Figura 10.7: Arquitetura FastText-CBOW.



significados diferentes. Em tais casos, modelos que incorporam informações morfológicas mais profundas podem ser mais apropriados.

É importante notar que as limitações do FastText não o tornam inadequado para todas as tarefas de processamento de linguagem natural. Ele continua sendo uma escolha sólida para muitos cenários devido à sua eficiência e simplicidade, mas é importante considerar essas limitações ao decidir qual modelo utilizar em um projeto específico. Em tarefas mais complexas e em idiomas com características particulares, pode ser necessário explorar modelos mais avançados como os modelos contextualizados abordados no Capítulo 15.

10.3.3 GloVe

Ao explorar modelos de *embeddings* estáticos como Word2Vec e FastText, é importante mencionar outra abordagem: o modelo GloVe (*Global Vectors for Word Representation*) (Pennington; Socher; Manning, 2014). Enquanto o Word2Vec e o FastText se concentram principalmente na relação local entre as palavras, o GloVe adota uma perspectiva global, levando em consideração a contagem de coocorrência palavra-palavra em um *corpus*. Essa abordagem permite que o GloVe capture informações de relação semântica e sintática entre as palavras.

O GloVe é um modelo global de regressão log-bilinear, que relaciona as variáveis dependentes e independentes por meio de uma função logarítmica (Pennington; Socher; Manning, 2014). Uma função log-bilinear é uma função não linear que tem como argumentos dois vetores. Neste modelo, a regressão log-bilinear é aplicada para estimar os vetores de palavras e as matrizes de transformação necessárias para mapear as palavras em um espaço vetorial.

Primeiramente, uma matriz de coocorrência é construída a partir de um *corpus* de textos. Essa matriz registra quantas vezes duas palavras aparecem juntas em uma janela de contexto. A partir da matriz de coocorrência, é construída uma matriz de probabilidade que representa a probabilidade condicional de uma palavra ocorrer perto de outra palavra. Essa matriz tenta capturar a relação entre as palavras considerando suas frequências relativas



de coocorrência. O objetivo do GloVe é encontrar representações vetoriais para palavras de forma que a relação entre os vetores corresponda à relação entre suas probabilidades de coocorrência. Isso é formulado como uma função de perda que minimiza o erro entre as relações de coocorrência reais e as estimadas. O modelo é treinado ajustando os vetores de palavras para minimizar a função de perda. Isso é feito usando um algoritmo de otimização, o Gradiente Descendente Estocástico (*Stochastic Gradient Descent – SGD*). O *SGD* é um algoritmo de otimização para ajustar os parâmetros de um modelo de acordo com uma função de custo a ser minimizada.

Ao utilizar uma combinação das vantagens dos métodos existentes (fatoração de matriz global e janela de contexto local), há uma análise das propriedades dos modelos que não eram totalmente exploradas e argumentam que o ponto de partida apropriado para o aprendizado de *embeddings* deve ser com proporções de probabilidades de coocorrência, ao invés das próprias probabilidades (Pennington; Socher; Manning, 2014).

Considerando a matriz de contagens de coocorrência palavra-palavra denotada por X , cuja entrada X_{ij} tabula o número de vezes que a palavra j ocorre no contexto da palavra i e seja $X_i = \sum_k X_{ik}$ o número de vezes que qualquer palavra aparece no contexto da palavra i , tem-se que $P_{ij} = P(j|i) = X_{ij}/X_i$ é a probabilidade de que a palavra j apareça no contexto da palavra i .

De maneira geral, o modelo GloVe busca converter X em matrizes de atributos W , onde as linhas de W são preenchidas por palavras do vocabulário e cada coluna corresponde a uma dimensão no espaço vetorial (Pennington; Socher; Manning, 2014).

A proporção de coocorrência (P_{ik}/P_{jk}) depende de três palavras, i , j , e k , de forma que o modelo mais geral assume a forma:

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (10.11)$$

onde F distingue palavras relevantes de palavras irrelevantes da seguinte maneira:

- Se a palavra k estiver relacionada às palavras i e j (P_{ik} e P_{jk} são grandes), ou não relacionada às palavras i e j (P_{ik} e P_{jk} são pequenos), então o valor de F seria próximo de 1.
- Se a palavra k estiver relacionada exatamente a uma das palavras i ou j , então o valor de F estaria longe de 1.

Como os espaços vetoriais são estruturas inerentemente lineares, a maneira mais natural de codificar a informação presente em P_{ik}/P_{jk} é com diferenças vetoriais. Sendo assim é possível restringir F , modificando a Equação 10.11 para

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}. \quad (10.12)$$

A função de custo J do modelo tem como objetivo aprender vetores de palavras otimizados para prever as probabilidades de co-ocorrência de palavras no *corpus* e é dada pela Equação 10.13:

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (10.13)$$

onde V é o tamanho do vocabulário e b é bias (ou viés) (Jurafsky; Martin, 2023) que leva em conta a frequência de cada palavra. $f(X_{ij})$ é uma função de ponderação introduzida



para compensar X_{ij} , uma vez que $\log(X_{ij})$ é indefinido nesse ponto, e também para equilibrar a contribuição de palavras frequentes e pouco frequentes para o modelo.

Para a função de custo J , a função f deve obedecer às seguintes propriedades (Pennington; Socher; Manning, 2014):

1ª propriedade:

- f deve satisfazer $f(0) = 0$. Se f for contínua, deve desaparecer como $x \rightarrow 0$ rápido o suficiente dado que o $\lim_{x \rightarrow 0} f(x) \log^2 x$ é finito;

2ª propriedade:

- $f(x)$ deve ser não decrescente para que raras coocorrências (pequeno x) não tenham excesso de peso (tem f relativamente grande);

3ª propriedade:

- $f(x)$ deve ser relativamente pequeno para grandes valores de x , para que as coocorrências frequentes não sejam sobrecarregadas.

Partindo dessas propriedades, Pennington; Socher; Manning (2014) propõem a função $f(x)$:

$$f(x) = \begin{cases} (x/x_{max})^\alpha & \text{if } x < x_{max} \\ 1 & \text{c.c.} \end{cases} \quad (10.14)$$

onde x_{max} e α são hiper-parâmetros.

O resultado do modelo é um conjunto de representações vetoriais de palavras que capturam as relações entre as palavras em termos semânticos e sintáticos. Esses vetores podem ser usados em diversas tarefas de PLN.

O GloVe tem capacidade para capturar informações semânticas mais abrangentes em comparação aos outros métodos de representação vetorial de palavras, devido à abordagem de coocorrência global ponderada.

10.3.3.1 Limitações do Glove

Apesar das vantagens do GloVe em relação às tarefas de analogia e similaridade, cabe destacar algumas limitações do método. Em virtude de seu treinamento necessitar de uma matriz de contagem de coocorrência palavra-palavra, o modelo consome muito espaço de memória; além disso, assim como o Word2Vec, o GloVe não consegue lidar com palavras fora do vocabulário e ignora a morfologia das palavras.

Em síntese, GloVe é um modelo de representação vetorial de palavras que utiliza informações de coocorrência global entre palavras no *corpus* e estima os parâmetros do modelo por meio de regressão log-bilinear. Essa abordagem visa capturar relações mais abrangentes entre as palavras e obter vetores de palavras mais informativos e precisos.

10.4 Considerações Finais

As representações vetoriais geradas pelos modelos Word2Vec, Fasttext e GloVe têm uma dimensionalidade menor em comparação com os vetores esparsos baseados em abordagens como TF-IDF e PMI/PMMI, o que ajuda a reduzir o custo computacional



e a dimensionalidade dos dados. Outra vantagem dos *embeddings* é que eles podem ser transferidos e usados como recursos em tarefas de aprendizado de máquina relacionadas, melhorando o desempenho de modelos em tarefas de PLN com conjuntos de dados menores.

Embora os *embeddings* apresentem algumas vantagens em relação aos modelos esparsos, nos *embeddings* estáticos cada palavra tem uma única representação, independentemente do contexto em que a palavra ocorre. Isso limita a capacidade de compreender a ambiguidade nas palavras. Além disso, por se tratarem de modelos estáticos, ou seja, que não são atualizados durante o treinamento da tarefa na qual são empregados, eles podem não se adaptar bem a tarefas específicas, especialmente aquelas que exigem informações contextuais específicas da tarefa como é o caso da análise de sentimentos, resolução de ambiguidades e tradução automática. Há que se considerar, ainda, que a língua natural está em constante evolução, e o significado das palavras pode mudar ao longo do tempo. Os *embeddings* estáticos não conseguem capturar essas mudanças.

Para tentar contornar esses problemas dos *embeddings* estáticos, permitindo o ajuste do modelo durante o treinamento da tarefa específica, várias abordagens baseadas em *embeddings* dinâmicos têm sido estudadas na literatura recentemente. Ao contrário dos vetores estáticos, os vetores dinâmicos se adaptam ao vocabulário e às características específicas da tarefa, capturando informações contextuais, pois consideram o contexto em que cada palavra é usada em um documento ou sentença. Isso permite lidar melhor com palavras polissêmicas, tornando os *embeddings* mais adequados para tarefas que dependem do contexto.

Os modelos de *embeddings* dinâmicos são o assunto do Capítulo 15.



Parte VI

Discurso



Capítulo 11

Modelos discursivos

Paula Christina Figueira Cardoso
Jackson Wilke da Cruz Souza
Roana Rodrigues

11.1 Introdução

No Dicionário Houaiss¹, discurso pode referir-se à “língua em ação, tal como é realizada pelo falante; a um segmento contínuo de fala maior do que uma sentença (Análise de discurso); a um enunciado oral ou escrito que supõe, numa situação de comunicação, um locutor e um interlocutor”; e ainda à “reprodução que alguém faz das palavras atribuídas a outra pessoa”. Diante das possibilidades de definir o que é discurso, nos parece pertinente pontuar quais os limites e o objeto de estudo do nível discursivo para a Linguística e, mais especificamente, para o PLN.

Segundo Barros (2021), na Linguística há diferentes perspectivas teórico-metodológicas para o estudo do *texto* e do *discurso*, porém todas coincidem no fato de considerarem que a análise discursiva “vai além da dimensão da palavra ou da frase, e se preocupa com a organização global do texto; examina as relações entre a enunciação e o discurso enunciado e entre o discurso enunciado e os fatores sócio-históricos que o constroem”. Salientamos que *texto* e *discurso* tendem a ser entendidos como elementos que se complementam. Segundo Lyons (1977), o texto se dá por meio do *discurso*, em que aquele seria qualquer passagem que apresenta a conexão do discurso, falado ou escrito, em um diálogo ou um monólogo.

Por sua vez, no PLN há uma tendência a definir *discurso* como “qualquer segmento conexo de texto ou fala, compreendendo uma ou mais frases ou segmento de frases” (Sidner, 1978). Essa parece ser uma definição bastante genérica, mas que conduz as pesquisas da área a tomarem *texto* e *discurso* como sinônimos. Diversos estudos discursivos em PLN trabalham com textos de diversos gêneros (como redações escolares, textos jornalísticos ou postagens em redes sociais) e tamanhos variados. Assim, a definição proposta por Sidner (1978) nos parece pertinente por não ter concebido *discurso* a partir de uma porção encadeada de duas ou mais sentenças, mas a partir da possibilidade de observação de questões que extrapolam os limites da materialidade e que não têm como fator limitante o tamanho. Ainda sob a perspectiva do PLN, Mitkov (2010) enfatiza que o discurso produzido não é uma mera coleção aleatória de símbolos ou palavras, mas se trata de elementos relacionados e significativos que têm um objetivo comunicativo particular.

Sendo assim, podemos afirmar que, em nível discursivo, uma preocupação comum à Linguística e, em especial, aos estudos de PLN está na relação entre os elementos de um texto, podendo-se, de antemão, depreender que a produção de um texto em si pressupõe um processo de interação e de intenções entre os sujeitos envolvidos em uma determinada

¹Dicionário Houaiss, disponível em: <https://houaiss.uol.com.br/>.



situação comunicativa. De acordo com Oliveira (2008), podemos organizar as relações textuais em duas grandes áreas: *coesão* e *coerência*, que, para a autora, são, na verdade, faces de uma mesma moeda.

Segundo Koch (2003), é possível definir *coesão* como “o fenômeno que diz respeito ao modo como os elementos linguísticos presentes na superfície textual se encontram interligados entre si, por meio de recursos também linguísticos, formando sequências veiculadoras de sentido”. A autora ainda destaca duas modalidades de coesão²: a remissão (reativação de referentes por anáfora, catáfora ou sinalização) e a sequenciação (elementos responsáveis pelo avanço e a continuidade dos sentidos do texto). Por sua parte, *coerência* refere-se “ao modo como os elementos subjacentes à superfície textual vêm a construir, na mente dos interlocutores, uma configuração veiculadora de sentidos” (Koch, 2003, p. 52). A coerência resulta da construção feita pelos interlocutores, por isso, embora parta do texto, envolve uma série de fatores de caráter cognitivo, interacional, situacional e sociocultural. A superfície do texto, conforme ressalta Koch (2003, p. 53), “funciona como pistas ou chaves para orientar o interlocutor na construção do sentido”. Pardo (2005, p. 1) explica o fenômeno da coerência textual nos exemplos de Exemplo 11.1:

Exemplo 11.1.

- a) Embora tenha chovido, as obras continuaram.
- b) João não foi à aula, mas estava doente.

Segundo o autor, apenas o trecho (1a) é coerente, por apresentar um sentido global marcado por uma relação de *oposição* entre as proposições. O trecho (1b), por sua parte, é incoerente, pois “a relação de *oposição* [marcada nesse caso pela conjunção adversativa *mas*] contraria a relação *decausa* que parece mais plausível” (Pardo, 2005). Portanto, é no nível do discurso que um escritor/falante organiza e relaciona as proposições para a produção de um texto com determinados objetivos comunicativos, buscando, assim, satisfazer as suas intenções comunicativas, como persuadir, informar ou pedir algo ao seu leitor/ouvinte.

As relações estabelecidas entre os elementos no interior de um texto para a construção de sentido são bastante complexas, inclusive para a interpretação humana. Por isso, verifica-se a teorização, anotação e o processamento de dados discursivos como grandes desafios para o PLN. Com base nisso, neste Capítulo, não temos a pretensão de findar as discussões sobre o nível discursivo; pelo contrário, nosso objetivo é apresentar um panorama sobre modelos discursivos que vêm sendo utilizados em pesquisas nas (sub)áreas de PLN, além de destacarmos tarefas desenvolvidas e consolidadas a partir desses modelos.

Para tanto, este Capítulo se organiza da seguinte maneira: na Seção 11.2, apresentamos fundamentações teóricas gerais sobre modelos de relações discursivas, exemplificando suas preocupações e potenciais aplicações por meio das teorias GSĐT, SDRT, Teoria de Centering e Teoria das Veias. Na Seção 11.2.1 e na Seção 11.2.2, em contrapartida, descrevemos com algum aprofundamento dois modelos discursivos bastante relevantes nos estudos de PLN no mundo e no Brasil: a *Rhetorical Structure Theory* (RST) e a *Cross-document Structure Theory* (CST). Na Seção 11.3, apresentamos os principais recursos disponíveis e aplicações em PLN que utilizaram modelos discursivos para sua constituição e/ou realização. Em Considerações Finais (Seção 11.4), descrevemos algumas limitações, desafios e conquistas da área.

²Os conceitos de coesão são definidos e discutidos no Capítulo 12.



11.2 Modelos de relações discursivas

Ao longo da exposição desta seção, poderá ficar a impressão de que alguns modelos são mais detalhados que outros. Isso se deve ao fato de que muitos deles não têm sido vastamente utilizados nos últimos anos, especialmente por conta do excelente desempenho que alguns métodos estatísticos e modelagens computacionais recentes vêm apresentando na área de PLN e Inteligência Artificial. Apesar de alguns modelos apresentarem essa questão, eles estão presentes nesta seção devido à aderência a aplicações e desenvolvimento de recursos para o PLN, ou mesmo por terem servido como ponto de partida teórico para outros modelos. Há modelos clássicos que buscam tratar diversos fenômenos discursivos, também nomeados retóricos. A título de exemplo, mencionamos, inicialmente e de maneira concisa, as contribuições da GSDT, SDRT, da Teoria de Centering e da Teoria das Veias.

A teoria de Grosz; Sidner (1986), conhecida como **GSDT** (*Grosz and Sidner Discourse Theory*), visa modelar o aspecto intencional do discurso. Parte-se da ideia de que o autor de um texto possui uma ou mais intenções e estrutura seu conteúdo de forma a satisfazê-las. Identificar as intenções do autor é crucial para compreender a mensagem pretendida. Como as intenções potenciais em um discurso são praticamente ilimitadas, a GSDT organiza-o usando relações de contribuição e satisfação entre as intenções. Essas relações são em número finito e limitadas a dois tipos: a *intenção primária* do discurso e as *intenções subjacentes* aos segmentos do discurso. Define-se, nesta teoria, as seguintes relações: *Dominance*, *Satisfaction-Precedence*, *Supports* e *Generates*.

A relação *Dominance* ocorre quando a intenção subjacente a um segmento A contribui para a intenção subjacente de um segmento B, isto é, A *dominates* B, representado por $(DOM(A,B))$. A relação *Satisfaction-Precedence* ocorre quando a intenção subjacente a um segmento A deve ser satisfeita antes da intenção subjacente a um segmento B, isto é, $SP(A,B)$. As relações *Supports* e *Generates* ocorrem entre o conteúdo dos segmentos. A primeira acontece se a aceitação de um segmento B fornece subsídios para a aceitação do segmento A, então se diz que o conteúdo de B *supports* A ($SUP(A,B)$). A segunda ocorre se a ação descrita em B contribui para a ação descrita em um segmento A ($GEN(B,A)$). No Exemplo 11.2, extraído de Maziero (2016, p. 14), ilustra-se tais relações.

Exemplo 11.2.

- a) A teoria XYZ é bem informativa para muitas tarefas de PLN que requerem conhecimento discursivo, e conta com diversos *parsers* disponíveis.
- b) Seu uso, portanto, é uma ótima alternativa quando se deseja automatizar totalmente uma tarefa de PLN.

Segundo Maziero (2016), no exemplo anterior, a intenção do autor do texto é persuadir o leitor que o uso da XYZ é uma ótima alternativa no campo do PLN (2b), argumentando a favor do modelo da primeira sentença. Podemos dizer, portanto, que há uma relação de DOM (2b, 2a) e SUP (2a, 2b). A teoria não visa explicitar qual a intenção do autor do texto, mas estabelece conexões entre as intenções, além de abordar questões como os focos de atenção e a estrutura linguística.

A teoria **SDRT** (Asher; Lascarides, 2003) – Teoria da Representação do Discurso Segmentado – se interessa em identificar os segmentos discursivos e as relações retóricas entre essas unidades, que podem ser classificadas em dois tipos básicos. Uma análise SDRT abrange todas as etapas do processamento do discurso, incluindo segmentação, identificação de relacionamentos e construção de hierarquias, usando informações semânticas e pragmáticas.



O discurso é representado como um hipergrafo, no qual as arestas são as relações discursivas e os nós representam as Unidades de Discurso Elementar (EDUs) que contém apenas um elemento. O grafo pode ter ainda Unidades de Discurso Complexas (CDUs) que são nós com mais de um elemento simples. As unidades discursivas são conectadas por relações retóricas de *coordenação* ou de *subordinação*. As relações de coordenação conectam segmentos do discurso no mesmo nível hierárquico, enquanto as relações de subordinação ligam um segmento do discurso a outro segmento que está um nível hierárquico abaixo. Asher; Vieu (2005) afirmam que essa distinção (no nível do discurso) possui uma motivação intuitiva, na qual certas partes do texto desempenham um papel subordinativo (menos relevante) em relação às demais. É importante ressaltar que o conjunto de relações, sejam de coordenação ou subordinação, não é fechado, pois estudos recentes já apresentam variações do conjunto original (por exemplo, (Muller et al., 2012)).

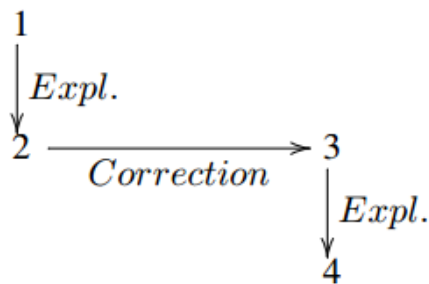
Esta teoria foi bastante explorada para modelar diálogos, pois permite representar contra-argumentação, um fenômeno pouco tratado em outros modelos discursivos (Afantenos; Asher, 2014; Asher et al., 2016; Badene et al., 2019; Li et al., 2020). Portanto, a teoria SDRT possui mecanismos que podem ser aplicados ao tratamento de diálogos, tais como *Question Elaborating*, *Correction* e *Question Answer Pair*. No Exemplo 11.3, Afantenos; Asher (2014) exemplificam um diálogo:

Exemplo 11.3.

- a) [Maria irá falhar em seus exames.]¹ [Ela não estudou muito.]²
- b) [Não, ela estudou muito.]³ [Agora ela tem até olheiras.]⁴

Em (3b), o falante não questiona seu interlocutor sobre sua conclusão (EDU 1), mas expressa discordância em relação à veracidade subjacente àquela conclusão. Isso assume a forma de uma relação de *Correction* entre a EDU 2 do primeiro falante, em (3a), representando o motivo e o contra-argumento do segundo falante. O falante fornece uma razão adicional para suas crenças por meio de uma relação de *Explanation*. Na Figura 11.1, tem-se a representação na forma de gráfico para esse diálogo.

Figura 11.1: Exemplo para relação Contraste – SDRT



Outra preocupação em nível discursivo é a resolução anafórica, elemento fundamental para o estabelecimento das relações de correferência de um texto. A **Teoria de Centering** (Grosz; Joshi; Weinstein, 1995), foca nas relações existentes entre anáforas e visa estabelecer a coerência nos segmentos discursivos adjacentes ao direcionar a atenção para a escolha de uma expressão referencial (discurso local). O principal objetivo da teoria é prever qual entidade discursiva tem maior importância em determinados segmentos, definindo um conjunto de regras e restrições que ditam as escolhas feitas pelos participantes do



discurso, como demonstrado em Exemplo 11.4 e Exemplo 11.5, a seguir, em que a Teoria de Centering fornece meios para tratar essas diferenças.

Exemplo 11.4.

- a) João foi a sua loja de música favorita para comprar um piano.
- b) Ele havia frequentado a loja por vários anos.
- c) Estava excitado porque iria finalmente poder comprar um piano.
- d) Mas quando chegou, a loja estava fechada.

Exemplo 11.5.

- (a) João foi a sua loja de música favorita para comprar um piano.
- (b) Esta era a loja que João frequentou por vários anos.
- (c) Ele estava excitado porque iria finalmente poder comprar um piano.
- (d) Ela estava fechada quando João chegou.

Nos exemplos, adaptados de Grosz; Joshi; Weinstein (1995), tem-se que os dois textos expressam a mesma ideia, mas no Exemplo 11.4 “João” é a unidade central enquanto que no Exemplo 11.5, o foco é alternado entre “João” e a “loja de música”. Percebe-se que as escolhas dos participantes podem variar desde a seleção da estrutura sintática (como em (4d) e (5d) que usam estruturas diferentes para tratar sobre o fato de a loja estar fechada)em até a escolha de expressões referenciais (como o uso de “a loja”, em (4b) e “esta” em (5b) ao tratar do mesmo referente).

Ainda na linha de tratamento de anáforas, há a **Teoria das Veias** (*Veins Theory*), proposta por Cristea; Ide; Romary (1998) , que sugere o estabelecimento de domínios referenciais de acessibilidade para cada unidade discursiva, representado pelas “veias” definidas na RST³. A Teoria das Veias expande as regras de coerência local da Teoria de Centering para abranger a composicionalidade das unidades do discurso (Seno, 2005). A veia de uma unidade é definida como um conjunto de unidades do discurso que podem conter o antecedente de uma anáfora. Para manter a coerência, é fundamental que o antecedente e o termo anafórico estejam presentes no mesmo veio, contribuindo para o discurso global.

No exemplo Exemplo 11.6, extraído de Seno (2005), as unidades 1 e 3 são ditas relevantes. Assim, o antecedente da anáfora “a fábrica” da unidade 4 pode estar presente em uma das unidades 1 e 3. No exemplo, seu antecedente encontra-se em 1.

Exemplo 11.6.

- [1] **A empresa Produtos Pirata Indústria e Comércio Ltda., de Contagem** [2] (na região metropolitana de Belo Horizonte), [3] deverá registrar este ano um crescimento de produtividade nas suas áreas comercial e industrial de 11% e 17%, respectivamente.
- [4] Os ganhos são atribuídos pela diretoria da **fábrica** à nova filosofia.

Os modelos discursivos podem destacar estruturas linguísticas, intencionais, informacionais ou de foco, todos com a principal preocupação de apresentar as relações entre os elementos de um texto para depreender a sua produção e os processos de

³A ser detalhada na próxima seção.



interação e intenções pertencentes a uma situação comunicativa específica. Nesta seção foram apresentadas brevemente bases teóricas dos modelos: GSDT, SDRT, da Teoria de Centering e da Teoria das Veias. Conforme já explicitado, embora existam vários modelos de análise discursiva que partem de reflexões linguísticas e possibilitam aplicações computacionais, nos deteremos, nas próximas seções, à descrição aprofundada de dois modelos discursivos: a RST e a CST, devido à sua relevância no cenário brasileiro.

11.2.1 O modelo RST (Mann e Thompson, 1988)

A **Rhetorical Structure Theory** (RST) é uma teoria linguístico-descritiva que trata da organização do texto utilizando relações retóricas (também nomeadas relações de coerência ou discurso) que existem entre os segmentos discursivos, formando uma estrutura discursiva totalmente conectada, geralmente na forma de árvore (Mann; Thompson, 1988). A RST explica a coerência postulando uma estrutura hierárquica e conectada, na qual cada parte de um texto tem uma função a cumprir, com relação às outras partes do texto (Taboada; Mann, 2006).

Cada proposição é associada a um núcleo (informação principal) ou satélite (informação adicional) de uma relação retórica. Em casos padrões, as relações se estabelecem entre duas proposições, expressas por segmentos adjacentes no texto. Quando a relação conecta um núcleo e um satélite, ela é chamada de mononuclear. Por outro lado, se a relação conectar somente núcleos, ela é chamada de multinuclear.

Mann; Thompson (1988) estabeleceram um conjunto de 23 relações retóricas que podem ser aplicadas a uma grande variedade de textos. Nesse conjunto, cada relação é classificada em semântica (*subject-matter*) ou intencional (*presentational*). As relações semânticas são aquelas que informam o leitor sobre algo, por exemplo, a relação SEQUENCE, cujo efeito pretendido é que o leitor reconheça que há uma sucessão temporal dos eventos apresentados. As relações intencionais alteram a inclinação do leitor para algo, por exemplo, a relação JUSTIFY, cujo efeito pretendido é que o leitor passe a aceitar melhor o direito do escritor de apresentar o núcleo. Vários pesquisadores modificaram e/ou complementaram esse conjunto de relações, como Marcu (1997) e Pardo (2005). No Quadro 11.1 apresenta-se o conjunto de relações de Mann; Thompson (1988) e o tipo de cada relação. Quanto à nuclearidade, as relações multinucleares estão marcadas com um asterisco.



Quadro 11.1. Relações RST

Relações semânticas	Circumstance, Solutionhood, Elaboration, Background, Volitional cause, Non-volitional cause, Volitional result, Non-volitional result, Purpose, Condition, Otherwise, Interpretation, Evaluation, Restatement, Summary, Sequence*, Contrast*
Relações intencionais	Antithesis, Concession, Enablement, Evidence, Justify, Motivation

Conforme se observa no Quadro 11.1, Mann; Thompson (1988) definiram as relações em termos de quatro campos, que devem ser observados pelo analista de um texto durante o processo de construção da estrutura RST. Os campos são restrições sobre o núcleo (N), restrições sobre o satélite (S), restrições sobre a combinação de núcleo e satélite e o efeito que a relação em questão pode causar no leitor. Nos Quadros 11.2 e 11.3, apresentam-se as definições das relações *Antithesis* e *Contrast*, respectivamente.

Quadro 11.2. Definição da relação Antithesis

Anthithesis	<p>Restrições sobre N: O escritor julga N válido</p> <p>Restrições sobre S: Nenhuma</p> <p>Restrições sobre N+S: N e S se contrastam e, por esse motivo, não podem ser válidos simultaneamente. Compreendendo-se S inválido levará o leitor a aceitar melhor N</p> <p>Efeito no receptor: O leitor aceita melhor N</p>
--------------------	--

Quadro 11.3. Definição da relação Contraste

Contrast	<p>Restrições sobre Ns: Não mais do que dois Ns; as situações nos Ns são (a) compreendidas como similares em vários aspectos, (b) compreendidas como diferentes em vários aspectos e (c) comparadas em relação a uma ou mais dessas diferenças</p> <p>Efeito no receptor: O leitor reconhece as similaridades e diferenças resultantes da comparação sendo feita.</p>
-----------------	---

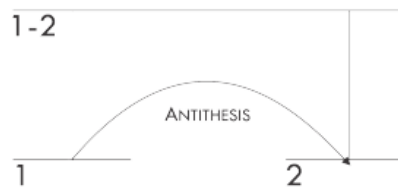
Um grande desafio encontrado na análise RST é a definição da relação retórica entre dois segmentos textuais. Se esse contexto for expandido para um texto inteiro, há diversas possíveis árvores discursivas para um mesmo texto, com segmentos, relações e nuclearidades diferentes. Por exemplo, um analista RST pode identificar que há uma oposição entre duas unidades discursivas e, assim, relações como *Antithesis* e *Concession* poderiam ser úteis na análise, gerando diferentes árvores discursivas. Para amenizar tal situação, o analista deve olhar para o campo efeito da definição das possíveis relações e identificar aquele que está mais saliente para o objetivo do autor do texto (Mann; Thompson, 1988).

Na Figura 11.2, apresenta-se um exemplo da relação mononuclear *Antithesis*, em que os segmentos 1 e 2 não podem ser válidos ao mesmo tempo, pois, ou a “detonação” foi “acidental” ou “proposital”. O segmento 2 é nuclear. Para que a crença do leitor no segmento



2 seja melhor aceita, o segmento 1 deve ser inválido. Na Figura 11.3 exemplifica-se a relação multinuclear *Contrast*.

Figura 11.2: Exemplo para relação Antithesis



1. POUCO DEPOIS DA DETONAÇÃO, OCORRIDA ÀS 10H40 (3H40 EM BRASÍLIA), FONTES DA POLÍCIA MOSCOVITA ADIANTARAM QUE ELA TERIA ACONTECIDO PROVAVELMENTE POR CAUSA DA EXPLOSÃO ACIDENTAL DE UM BUJÃO DE GÁS.
2. - ISTO NÃO FOI UM ACIDENTE, FOI PROPOSITAL - DISSE RESIN, CITADO PELA AGÊNCIA OFICIAL RUSSA "ITAR-TASS".

Figura 11.3: Exemplo para relação Contraste



Para fazer a análise RST de um texto, várias estratégias podem ser utilizadas. Carlson; Marcu (2001) apontam que uma estratégia bem aceita é fazer uma análise incremental, isto é, relacionar primeiro as proposições de uma sentença, o que resultará em uma subestrutura RST, a qual, por sua vez, será relacionada à outra subestrutura. Podem-se montar subestruturas de cada parágrafo do texto isoladamente e depois integrá-las, formando uma única estrutura RST. Se o analista decide por esse tipo de análise, ele pode tirar proveito da estrutura organizacional dada pelo produtor do texto. Por exemplo, se duas proposições estão diretamente relacionadas por *Condition*, é provável que elas sejam expressas em uma única sentença.

11.2.2 CST: Cross-document Structure Theory, 2000

A **Cross-document Structure Theory** (CST) é um modelo teórico derivado da RST, diferenciando-se acerca da quantidade de textos que podem ser analisados e, conseqüentemente, dos fenômenos linguísticos que ocorrem. Esta teoria foi proposta por Radev (2000), com o objetivo de realizar análises semânticas de múltiplos textos



que possuem o mesmo assunto. O autor percebeu que, quando se agrupa textos que possuem o mesmo assunto, fenômenos linguísticos (como redundância, contradição e complementaridade), de estilo e de organização. Em sua proposta, o modelo CST, então, é capaz de traduzir cada fenômeno em diferentes relações, como demonstrado nos exemplos 11.7, 11.8 e 11.9, a seguir.

Exemplo 11.7.

(S1) Todos morreram quando o avião, prejudicado pelo mau tempo, não conseguiu chegar à pista de aterrissagem e caiu numa floresta a 15 quilômetros do aeroporto de Bukavu.

(S2) Todos morreram quando o avião, prejudicado pelo mau tempo, não conseguiu chegar à pista de aterrissagem e caiu numa floresta a 15 Km do aeroporto de Bukavu.

Exemplo 11.8.

(S1) Ao menos 17 pessoas morreram após a queda de um avião de passageiros na República Democrática do Congo.

(S2) O avião explodiu e se incendiou, acrescentou o porta-voz da ONU em Kinshasa, Jean Tobias Okala.

Exemplo 11.9.

(S1) Todos morreram quando o avião, prejudicado pelo mau tempo, não conseguiu chegar à pista de aterrissagem e caiu numa floresta a 15 Km do aeroporto de Bukavu.

(S2) A aeronave se chocou com uma montanha e caiu, em chamas, sobre uma floresta a 15 quilômetros de distância da pista do aeroporto.

As sentenças (S1 e S2) foram retiradas do *corpus* CSTNews (Cardoso et al., 2011), que contém conjuntos de textos escritos em português brasileiro e anotados com o modelo CST. Cada uma das sentenças foram extraídas de fontes jornalísticas distintas, e associadas manualmente em função dos fenômenos identificados. No Exemplo 11.7 existe uma relação de *redundância*, uma vez que o par de sentenças apresenta um conteúdo praticamente idêntico. Já no Exemplo 11.8, há uma relação de *complementaridade*, pois S2 acrescenta que o avião acidentado “explodiu e se incendiou”, em relação à informação em S1. Por fim, no Exemplo 11.9 observa-se a presença de *contradição*, porque S1 informa que a causa da queda do avião foi o mau tempo, enquanto S2 destaca que a causa do acidente foi o choque contra a montanha.

Os modelos apresentados neste Capítulo foram caracterizados como discursivos, pois focalizam aspectos e fenômenos de apenas um texto. Dessa maneira, é necessário destacar que o modelo CST é especialmente caracterizado por sua abordagem semântica, já que é possível identificar, como demonstrado, uma série de fenômenos linguísticos, de estilo e de estrutura. Porém, tais fenômenos se dão de maneira não intencional, diferentemente dos outros modelos em que os fenômenos ocorrem por intencionalidade de quem elabora o texto e, nesse sentido, o estrutura e o organiza de determinada forma.

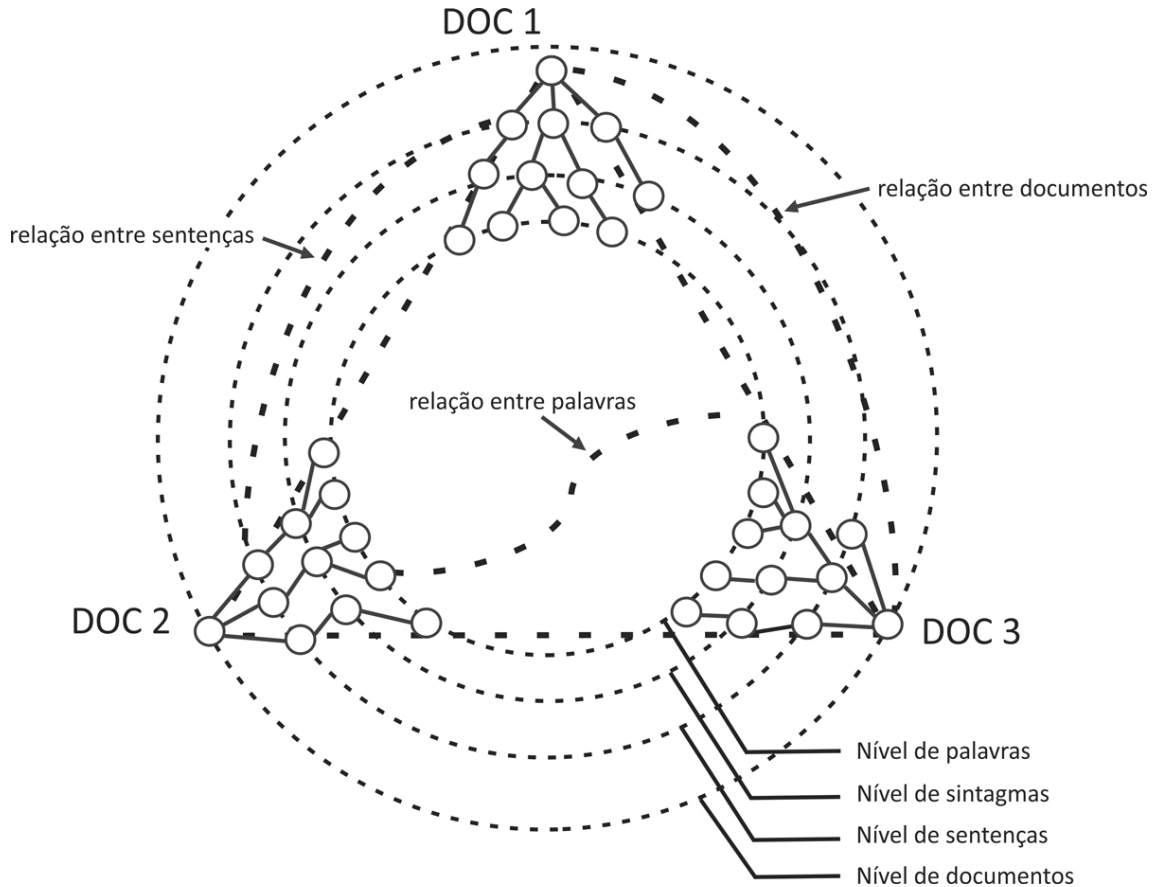
A contradição no Exemplo 11.9, por exemplo, só foi possível de ser identificada porque dois textos foram agrupados e, de maneira manual e/ou automática, foi identificado o



fenômeno (não intencional) em questão. Assim, o que justifica a ocorrência do modelo CST neste capítulo é o fato de ele ocorrer na relação entre textos. Nesse sentido, se dá de maneira discursiva, ainda que caminhe nas margens de uma definição clássica de discurso para o PLN.

De acordo com Radev (2000), as relações CST podem ocorrer entre diferentes unidades informativas, tais como, palavras, sintagmas, sentenças, parágrafos e documentos, formando um grafo, como ilustrado na Figura 11.4.

Figura 11.4: Esquema de relacionamentos CST



Na Figura 11.4, percebe-se que os níveis nos quais as relações CST podem ser identificadas compõem uma hierarquia (palavras → sintagma → sentença → texto), ainda que usualmente isso seja feito em nível sentencial. Cada um dos três documentos (DOC 1, DOC 2 e DOC 3) está representado por um subgrafo, que codifica relações internas aos textos. Os relacionamentos internos a cada texto podem ser caracterizados em nível sintático ou discursivo. As relações CST que podem ser estabelecidas nos diferentes níveis estão representadas por linhas pontilhadas mais grossas.

Ainda sobre a Figura 11.4, destaca-se que:

- Os documentos similares são representados numa hierarquia de palavras, sintagmas, sentenças e os próprios documentos, ou seja, todos esses níveis são considerados na análise;
- Em cada nível da hierarquia podem ocorrer relações CST, apesar de sentenças serem usualmente mais utilizadas nos trabalhos da área;

- O grafo resultante da anotação é provavelmente desconectado, pois nem todos os segmentos dos textos em análise precisam estar relacionados: podem existir segmentos que não se referem diretamente ao mesmo assunto.

Para o inglês, originalmente foram propostas 24 relações CST por Radev (2000). Uma vez que o modelo CST admite haver ambiguidade entre as relações, é natural ter novas propostas de conjuntos de relações. Além disso, determinadas relações podem não ocorrer em certos *corpora* com gêneros textuais específicos. Destacam-se as relações propostas para o modelo CST aplicado ao português brasileiro. Aleixo; Pardo (2008a) chegaram a um conjunto de 14 relações multidocumento. Segundo os autores, a redução justifica-se pela não ocorrência de algumas relações em textos jornalísticos ou ainda por conta da similaridade entre algumas relações, o que resultou no agrupamento de algumas delas, como é o caso de *Equivalence* e *Paraphrase* em apenas *Equivalence*, ou *Elaboration* e *Refinement* em *Elaboration*. No Quadro 11.4, mostra-se o conjunto de relações CST utilizado no *corpus* CSTNews.

Quadro 11.4. Relações CST

Attribution	Identity
Citation	Indirect speech
Contradiction	Modality
Elaboration	Overlap
Equivalence	Subsumption
Follow-up	Summary
Historical background	Translation

Maziero; Jorge; Pardo (2010) propuseram uma tipologia em que as relações CST para o português brasileiro estão categorizadas entre Redundância, Complemento, Contradição, Fonte/Autoria e Estilo. É possível inferir que essa proposta seja, na verdade, uma simplificação do modelo discursivo com foco na implementação computacional, em especial, além ter contribuído com a área de PLN na compreensão de fenômenos linguísticos no contexto de Sumarização Automática Multidocumento.

Mais recentemente, alguns estudos descritivos (Souza, 2015, 2019) apontaram que a organização de relações CST entre Conteúdo e de Apresentação/Forma pode não ser suficiente para caracterizar as relações CST, em especial as relações classificadas, até então, como complementaridade. Tais estudos indicam que algumas relações de redundância (como *Subsumption* e *Overlap*) poderiam ser classificadas como relações de complementaridade, por apresentarem outras informações acerca do mesmo evento.

Vale destacar que o modelo CST contribuiu com a criação de recursos e, conseqüentemente, em aplicações de PLN. Na próxima seção, destacamos alguns deles, entre os outros modelos apresentados aqui.

11.3 Recursos e aplicações para o português brasileiro

A descrição dos fenômenos linguísticos em nível discursivo, a partir dos diferentes modelos de análise, como os descritos neste Capítulo, contribuiu para importantes avanços de diversas aplicações de PLN. Freitas (2022) escreve que para que tais aplicações sejam bem-sucedidas,



uma série de recursos e ferramentas linguístico-computacionais é acionada. Assim, destaca-se a criação de *corpus* como recurso anotado no nível discursivo, de ferramentas que facilitam a anotação automática de dados e de diversas aplicações, como de sumarização (Cardoso, 2014; Uzêda; Pardo; Nunes, 2010), tradução automática (Marcu; Carlson; Watanabe, 2000) e avaliação de redações (Stab et al., 2014).

Na literatura, são encontrados pelo menos dois *corpora* padrão ouro com relações discursivas para o português brasileiro: Summ-it⁴ (Collovini et al., 2007; Fonseca et al., 2016) e CSTNews⁵ (Aleixo; Pardo, 2008b; Cardoso et al., 2011). O *corpus* Summ-it reúne anotações de vários níveis linguísticos, incluindo relações retóricas da RST, correferência e entidades nomeadas. Esse recurso, concebido para promover pesquisas em discurso e sumarização automática, constitui-se de 50 textos jornalísticos do caderno de Ciências da Folha de São Paulo.

O *corpus* CSTNews, por sua vez, contém 50 grupos de textos jornalísticos de assuntos variados, coletados manualmente das fontes de notícias Folha de São Paulo, Estadão, O Globo, Jornal do Brasil e Gazeta do Povo. Assim como o *corpus* Summ-it, CSTNews foi orientado para a sumarização automática, sendo constituído de diversas camadas de anotação, tais como RST e CST, e sumários manuais e automáticos.

A anotação no nível discursivo de textos pode ser feita de forma manual ou automática. Para alguns modelos discursivos existem analisadores automáticos, conhecidos como *parsers* discursivos, que visam a identificação retórica do texto, gerando uma estrutura hierárquica em que as intenções do autor são explicitadas e relacionadas entre si (Maziero, 2016). Para relações RST, se tem conhecimento do *parser* DiZer⁶ (Maziero, 2016; Maziero; Hirst; Pardo, 2015). Treinado com textos acadêmicos e jornalísticos, a ferramenta recebe um texto de entrada, segmenta-o, identifica a nuclearidade e monta a estrutura arbórea com as relações discursivas.

Com a finalidade de facilitar o processo de anotação de *corpus* com CST, foi desenvolvida a ferramenta semiautomática CSTTool⁷ (Aleixo; Pardo, 2008a). A CSTTool possibilita os processos de segmentação dos textos-fonte em nível sentencial e a identificação, em pares, das sentenças lexicalmente relacionadas por meio de medidas de similaridade. Após a indicação dos possíveis pares relacionados, cabe ao anotador escolher uma relação CST adequada. Após a indicação dos possíveis pares relacionados, cabe ao anotador escolher uma relação CST adequada. Para uma análise totalmente automática, está disponível o CSTParser⁸ (Maziero; Pardo, 2012), que recebe como entrada um conjunto de documentos relacionados e segmenta-os em sentenças. Após isso, busca os pares de sentenças mais prováveis de terem algum relacionamento multidocumento por meio de medidas de similaridade.

11.4 Considerações Finais

Lidar com o nível discursivo é um desafio para os estudos em PLN, como já havia sido sinalizado por Dias-da-Silva (1996), um dos pioneiros na área no Brasil. O autor já destacava algumas questões relativas esse nível de análise linguística, como a necessidade de delimitar o objeto de estudo, determinar os limites entre análise textual e discursiva, ou ainda caracterizar o discusso como um processo. Felizmente, algumas dessas perguntas

⁴<https://www.inf.pucreb.br/linatural/wordpress/recursos-e-ferramentas/summ-it/>

⁵<http://nilc.icmc.usp.br/CSTNews/login/about>

⁶<https://github.com/egmaziero/dizer3>

⁷<http://www.nilc.icmc.usp.br/nilc/index.php/tools-and-resources>

⁸<http://www.nilc.icmc.usp.br/CSTParser/>



já foram respondidas, como a definição do objeto de estudo. No entanto, outras questões ainda estão sendo investigadas para encontrar possíveis respostas.

Se, por um lado, ao longo dos últimos anos, percebemos que diferentes tarefas linguístico-computacionais foram sendo demandadas e concebidas discursivamente, como resolução anafórica, por outro, há de se questionar se a análise de sentimentos e emoções, por exemplo, se enquadra no nível discursivo. Como dito anteriormente, o nível discursivo congrega outros níveis de análise linguística e, conseqüentemente, é esperado que determinados fenômenos sejam fronteiros com a Morfologia, Sintaxe, Semântica e Pragmática.

Quanto ao questionamento de Dias-da-Silva (1996) sobre a possibilidade de o discurso ser um processo, é possível que as respostas residam em aprimorar modelos discursivos a partir de descrições linguísticas cada vez mais robustas. Ao longo deste capítulo ilustramos modelos discursivos que por vezes nasceram para suprir expectativas teórico-metodológicas de determinadas aplicações em PLN, mas que não se restringiram a elas. Outros modelos, no entanto, ficaram restritos a determinadas aplicações, podendo esse fato ser explicado por uma maior dependência de humanos para as fases de treinamento dos modelos. Assim, há ainda um vasto campo de pesquisas e descrições linguísticas a serem realizadas em todos os modelos aqui dispostos.

Sabe-se que o desenvolvimento de tecnologias sofisticadas tem substituído a reflexão e supervisão linguísticas por modelos estatísticos, com métodos não compreensíveis para os seres humanos. No entanto, conforme aponta Freitas (2022), o conhecimento linguístico para o PLN não ficará obsoleto por diversos motivos, entre os quais a autora destaca quatro:

- (i) nem todo conhecimento em PLN é voltado para aplicações da indústria, portanto, há pesquisas linguísticas que dependem desse conhecimento para o desenvolvimento de aplicações linguísticas (materiais lexicográficos, didáticos, corretores gramaticais etc.);
- (ii) continua sendo necessária ao menos uma amostra do conhecimento humano para as tarefas em PLN, como na construção de *datasets*, em versões iniciais de sistemas e na avaliação do desempenho da máquina);
- (iii) é elevado o custo (computacional, financeiro e ambiental) das atividades desenvolvidas com base nos métodos estatísticos, por isso, informações linguísticas possibilitam a economia no processamento em comparação com o uso de dados brutos; e
- (iv) desde uma perspectiva filosófica, haver apenas a eficácia – sem compreensão, nem explicação – dos sistemas não é o suficiente, pois a ciência se baseia no paradigma da verdade.

Ao passo que os modelos estatísticos/probabilísticos proporcionam avanços infindáveis a pesquisas em PLN, é necessário pontuar que o fato de não sabermos como a língua(gem) é, de fato, processada em muitos deles isso faz com que muitos desafios ainda perdurem. Muito recentemente, fomos inseridos em abordagens e aplicações que, com apenas um comando, é possível ter conteúdos readequados a exigências estruturais de determinados gêneros textuais (Capítulo 20). Para além dos especialistas, isso pode impressionar muitas pessoas, ainda que estejamos lidando substancialmente com a probabilidade de combinações de *tokens*, sem lançar mão ao sentido das unidades que estão sendo articuladas em sentenças ou em textos.

Nesse sentido, talvez o que tenhamos diante de nós sejam outras e mais complexas perguntas que se alinham às reflexões e provocações de Dias-da-Silva (1996): É possível processar o discurso sem precisar de um componente semântico? É possível compreender



emoções de um texto sem o componente pragmático? É possível realizar análise discursiva sem análises linguísticas e, conseqüentemente, sem os modelos discursivos aqui apresentados?



Capítulo 12

Resolução de Correferência

*Evandro Fonseca
Aline Aver Vanin
Renata Vieira*

12.1 Introdução

No processo de construção de sentidos na língua em uso, interlocutores negociam o universo de discurso de que falam, escolhendo referir-se a algum, ou a alguns, indivíduo(s) cuja identidade estabelecem e da qual garantem a existência (Neves, 2013). Esses referentes, concretizados no texto por **expressões referenciais**, vão atravessá-lo por inteiro, garantindo unidade temática – isto é, a coerência que constitui um texto (Vieira; Faraco, 2019). Fazer referência a algo ou a alguém no mundo é uma ação intrinsecamente ligada à interação, em que se constituem os objetos de discurso, isto é, entidades que constituem termos das predicções, entidades oriundas de uma construção mental, e não de um mundo real (Neves, 2013).

A construção de referentes se dá por cadeias de texto, redes referenciais construídas pelos objetos de discurso que constituem as marcas da textualidade. Uma cadeia de referência, ou cadeia referencial, corresponde à noção de cadeia anafórica, e cadeia coesiva (Roncarati, 2010). Os elos coesivos em um texto são mecanismos semânticos e léxico-gramaticais essenciais para a tessitura textual. Os referentes se ligam por meio de relações de sentido que formam a base para retomadas em um texto (Roncarati, 2010). Nesse sentido, as cadeias coesivas ligam referentes à(s) sua(s) expressão(ões) referencial(is), em que os fios que tecem o texto são articulados por meio de procedimentos e recursos - o que chamamos de **coesão textual** (Vieira; Faraco, 2019). Trata-se de elementos cujos mecanismos gramaticais coesivos estão em consonância, sejam eles por reiteração (ou retomada), por associação (ou ligações de sentidos entre as palavras presentes), ou por conexão entre as orações (por conectores) (Antunes, 2007), os quais garantem que o texto seja coerente em sua extensão. Então, se em um dado texto temos uma dada entidade, como “Maria”, nome próprio, é de se esperar que os seus referentes sejam detectados por relações léxico-semânticas do texto, como, por exemplo: por pronome, “ela”, ou por sintagma nominal, “a professora”, “a ativista”, “a mulher” etc.

12.2 Resolução de Correferência

A Resolução de Correferência a partir de textos é uma tarefa útil e também um dos principais desafios da área de Processamento da Linguagem Natural (PLN). Isso porque essa tarefa depende de diversos níveis de processamento, como análise sintática, morfológica, extração de sintagmas nominais, entre outros. Na literatura, encontramos diversas iniciativas para a



língua portuguesa que abordam esse problema, geralmente separados entre a resolução de anáfora (Basso, 2009; Bick, 2010; Ferradeira, 1993; Rocha, 2000; Vieira et al., 2005) e o estudo da correferência nominal (Fonseca, 2014; Fonseca; Vieira; Vanin, 2016a; Fonseca; Vieira; Vanin, 2014; Freitas et al., 2009). Resolução de Correferência consiste em identificar as diferentes formas que uma mesma menção pode assumir em um discurso. Em outras palavras, esse processo consiste em identificar determinados termos e expressões que remetem a uma mesma referência. Na sentença apresentada no Exemplo 12.1 podemos dizer que [*o único país de a União Europeia a não permitir patenteamento de genes*] é uma correferência de [*A França*], da mesma forma que [*A UE*] é uma correferência de [*a União Europeia*]. Agrupando esses termos formamos grupos de menções referenciais, mais conhecidos como cadeias de correferência.

Exemplo 12.1.

A França resiste como o único país de a União Europeia a não permitir patenteamento de genes. A UE ...

Na presente seção, veremos as definições de conceitos fortemente relacionados à tarefa de Resolução de Correferências, tais como os de referentes, entidades nomeadas, sintagmas nominais, entre outras definições.

12.2.1 Referentes

Referentes, ou menções, podem ser definidos como termos os quais usamos para nos referirmos a determinada entidade em um discurso. Em um texto, essas referências podem aparecer como uma entidade nomeada específica ou como parte constituinte de um sintagma nominal.

12.2.1.1 Entidades Nomeadas

Entidades nomeadas, a grosso modo, são elementos que podem ser referenciados por meio de nomes próprios (Jurafsky; Martin, 2023). Esses nomes próprios podem configurar-se em classes específicas, tais como: Pessoa (nomes de pessoas), Organização (nomes de empresas), Local (nomes de lugares), entre outras. Por meio dos exemplos abaixo, podemos identificar diversas entidades nomeadas (ENs), como Banco Nacional de Desenvolvimento Econômico e Social (a), Apple (b), nomes de bandas musicais (c).

- a) O Banco Nacional de Desenvolvimento Econômico e Social (BNDES), empresa pública federal, é hoje o principal instrumento de financiamento de longo prazo ...
- b) A Apple informou que vendeu 5 milhões de iPhone 5 só em um fim de semana ...
- c) Várias bandas de black metal tiveram influências do punk, tais como Venom, Celtic Frost, Bathory, Sarcófago, Darkthrone, Impaled, Nazarene, Mayhem, Hellhammer, Behemoth, entre outras ...

12.2.1.2 Sintagmas Nominais

São unidades formadas por uma ou mais palavras que, juntas, desempenham uma função sintática específica na frase (Capítulo 6). A natureza de um sintagma depende diretamente do elemento que constitui seu núcleo. Neste capítulo, damos foco a menções expressas por sintagmas nominais. Dito isso, temos então os sintagmas nominais, cujos núcleos



podem configurar-se em nome comum, nome próprio ou um pronome. Os pronomes podem apresentar-se, basicamente, nas formas de pronome pessoal, demonstrativo, indefinido, possessivo ou relativo. Um sintagma nominal geralmente é composto por um determinante (artigo, pronome demonstrativo, pronome indefinido e numeral cardinal) seguido de um substantivo. Por exemplo, na sentença do Exemplo 12.2 “**O especialista**” é um sintagma nominal, e o artigo “**O**” é seu determinante. Por meio do determinante de um sintagma é possível extrair informações valiosas. Isto é, a palavra “especialista”, por si só, pode assumir diferentes papéis. Contudo, “**O especialista**” qualifica uma pessoa do sexo masculino, além de informar quem é o especialista (apenas um, não dois ou mais). Notemos que o determinante carrega informações úteis para o processamento linguístico.

Exemplo 12.2.

O especialista não respondeu todas as perguntas.

Contudo, sintagmas nominais podem configurar-se em apenas substantivos como no Exemplo 12.3 onde temos dois sintagmas nominais sem determinante explícito: “Rio de Janeiro” e “cidade maravilhosa”. Respectivamente, um nome próprio e um substantivo comum seguido de seu adjunto adnominal. Algumas vezes, esse adjunto pode ser predicativo.

Exemplo 12.3.

Rio de Janeiro, cidade maravilhosa

Para entendermos a diferença entre adjunto adnominal e predicativo, basta observarmos que ora um termo pode exercer a função de adjunto, ora de predicativo. Ou seja, enquanto o adjunto adnominal representa o termo acessório da oração, o predicativo se revela como um termo essencial, de modo a deixá-la compreensível, dotada de sentido. Em d), “referência em saúde e segurança” representa parte essencial à constituição do enunciado, pois sem a presença desses termos o entendimento estaria comprometido. Assim, consideramos que se trata de um predicativo, visto que atribui uma característica ao sujeito, cujo núcleo é representado por “cidade”.

d) A cidade que é referência em saúde e segurança.

Em e) constatamos que o termo “limpa” pode perfeitamente ser retirado do contexto oracional sem que isso cause nenhum dano ao perfeito entendimento do discurso. Logo, trata-se de um termo acessório da oração ou adjunto adnominal.

e) A cidade limpa que é referência em saúde e segurança.

12.2.1.3 Tipos de Referentes

Existem três tipos de referentes: referentes específicos, referentes não-específicos e referentes abstratos.

Referentes específicos: Quando a menção de uma entidade, basicamente, identifica-a por meio de um nome comum ou próprio.

f) Microsoft informou que irá resolver o bug que reinicia o Windows Phone em dezembro.



- g) Luiz Inácio Lula da Silva sancionou nesta quarta-feira, 29, a lei que regulamenta as atividades de moto-taxista e motoboy de todo país...
- h) Roger Waters faz seu segundo show em São Paulo.

Em (f) temos um referente específico, isto é, a menção da entidade refere-se diretamente a algo específico, à empresa Microsoft. O referente específico, nesse caso, ainda pode ser classificado como uma entidade do tipo Organização. Existem outros tipos de referentes específicos, como Pessoa (g), Local (h), entre outros. Notemos que em (h) temos dois tipos de referentes específicos, “Roger Waters” e “São Paulo”, respectivamente entidades do tipo Pessoa e Local.

Referentes não-específicos: Quando as menções referem-se a uma entidade de forma não específica (autoridades, funcionários, policiais...), como mostram os exemplos “i”, “j” e “k”.

- i) Policiais invadiram a casa, porém os bandidos já haviam fugido...
- j) Funcionários estão descontentes. Eles afirmam ainda não terem recebido o seu décimo terceiro salário.
- k) Autoridades disseram que estão cansados de fazer as mesmas declarações.

Referentes abstratos: como o próprio nome sugere, são entidades abstratas, “não físicas”. Tratam de estados e qualidades, sentimentos e ações, como: medo, viagem, coragem, felicidade, esforço... Exemplos “l” e “m”.

- l) O medo é algo que deve ser superado. Para isso, concentre-se em seus objetivos.
- m) A viagem foi ótima, porém o tempo podia estar melhor.

12.2.2 Relações Semânticas Referenciais

Nesta subseção temos como foco tornar claro os tipos de relações semânticas que podem indicar uma relação de correferência.

12.2.2.1 Hiperonímia e Hiponímia

Hiperonímia é uma relação semântica que expressa um sentido amplo entre dois termos, partindo de uma classe mais ampla para uma subclasse mais específica, por exemplo: (inseto abelha). Neste caso, o termo “inseto” é um hiperônimo de “abelha”. Já Hiponímia representa uma relação contrária, ou seja, parte de uma classe mais específica para uma classe mais abrangente. Para o exemplo previamente dado temos que “abelha” é um hipônimo de “inseto”. Os hiperônimos e hipônimos são importantes no campo semântico, pois são muito usados na retomada de elementos em um texto, a fim de evitar repetições desnecessárias. No que diz respeito à identificação de menções referenciais em um discurso, na língua portuguesa é comum partirmos de termos específicos para termos mais abrangentes. Dessa forma, a relação de Hiponímia geralmente ocorre com maior frequência.

- n) João e Maria estão muito felizes com o seu cão. O animal é fiel e companheiro.
- o) Nada disso vai fazê-los mudar de carro. O pequeno veículo parece suprir todas as necessidades deles.



12.2.2.2 Sinonímia

Trata-se de uma relação entre dois termos, em que estes, mesmo sendo distintos lexicalmente, possuem significados muito próximos, por exemplo: (menino garoto). É importante referir que muitas vezes os sinônimos podem ter conotações diferentes, dependendo do contexto, como: (gato bichano) e (gato atraente). Em um texto, a utilização de sinônimos de uma palavra é importante para evitar repetições. Assim, um sinônimo é uma palavra que, apesar de ser diferente, tem o mesmo significado (ou semelhante) e, por isso, a sua inclusão não altera o sentido do texto em questão.

- p) Esse carro é maravilhoso. Também, estamos falando de um automóvel de 100 mil reais.
- q) Ana comprou um gato. O bichano adora dormir no sofá.

12.2.3 Correferência, Anáfora e Catáfora

Para o entendimento sobre o que é correferência, é relevante também definirmos anáfora, já que seus conceitos estão relacionados. Anáfora pode ser definida como a retomada de uma expressão apresentada anteriormente em um texto. Quando uma entidade é mencionada pela primeira vez textualmente, temos a evocação da entidade. Durante a leitura da sequência do texto, quando essa entidade é novamente mencionada, temos a realização do acesso a essa entidade. A expressão que faz o acesso é dita como anafórica, e a expressão anterior é dita como seu antecedente (Vieira; Gonçalves; Souza, 2012). Há casos de anáfora em que o termo anafórico e o antecedente são correferentes, isto é, remetem a uma mesma entidade (como os Exemplos “r” e “s” ilustram), mas há também casos de anáfora sem correferência, como podemos ver em “t”.

- r) A Ana comprou um cão. O animal já conhece todos os cantos da casa. Nesse exemplo, o termo anafórico é o grupo nominal “o animal”, que retoma o valor referencial do antecedente, “o cão”. É a relação entre “cão” e “animal” que suporta a correferência.
- s) Maria está com febre. Acho que ela está doente. Notemos que a interpretação referencial do sintagma nominal “ela” depende da sua relação anafórica com o sintagma nominal “Maria”.
- t) João faz 18 anos no dia 2 de Julho de 2001. No dia seguinte, parte para uma grande viagem pela Europa. Já nesse caso, o valor referencial da expressão sublinhada constrói-se a partir da interpretação do antecedente, a expressão adverbial temporal “no dia 2 de Julho de 2001”. Assim, “No dia seguinte” designa o dia 3 de Julho de 2001.

Catáfora: semelhante à anáfora mas em ordem oposta, uma relação catafórica ocorre quando um termo se refere a outro que vem à frente e lhe dá, a partir deste, o seu sentido. Conforme podemos ver no exemplo “u”:

- u) A mãe olhou-o e disse: - Meu filho, estás com um olhar cansado.

Correferência: é um fenômeno que ocorre quando duas ou mais menções em um discurso referem-se a uma mesma entidade. O conjunto de menções a uma mesma entidade no texto é denominado de cadeia de correferência.



- v) O João está doente. Vi-o na semana passada. Neste caso, o pronome “o” é uma anáfora de “João”, pois, para ser compreendido, necessita resgatar a frase anterior para que seu significado seja construído.

Temos também o tipo apostro, que ocorre quando o termo da oração se relaciona a uma entidade para esclarecê-la ou explicá-la, como em “w” e “x”.

- w) Cubatão, a cidade mais poluída do Brasil, localiza-se na Baixada Santista.
- x) Maria comprou várias frutas: mamão, melancia, abacate e uva. Normalmente, o apostro aparece isolado por sinais de pontuação, sendo mais comum aparecer entre vírgulas ou então introduzido por dois pontos. Nos exemplos acima podemos notar que “cidade” é correferente de “Cubatão”, e “mamão, melancia, abacate e uva” são correferentes de “frutas”.
- y) (extraído do *corpus* Summ-it (Collovini et al., 2007)) “A discussão sobre a biotecnologia nacional está enviesada, pois está sendo entendida como sinônimo de transgenia. A opinião é de Miguel Guerra, da UFSC (Universidade Federal de Santa Catarina). Guerra participou do debate”Biotecnologia para uma Agricultura Sustentável”... Para o agrônomo, o Brasil deve buscar o desenvolvimento de transgenias que tentem melhorar as condições da agricultura local...”

No exemplo “y”, as expressões “Guerra” e “o agrônomo” fazem referência à entidade “Miguel Guerra”, já mencionada anteriormente no texto. Para não repetir a mesma expressão, faz-se uso de outra diferente, mas que retoma a mesma entidade mencionada previamente. Esse é um método muito utilizado no processo de escrita para não deixar o texto repetitivo, e está diretamente relacionado a coesão referencial e sequencial. Notemos que a coesão referencial é responsável por criar um sistema de relações entre as menções dentro de um texto, permitindo que o leitor identifique termos e expressões que remetem a uma mesma entidade. Junto a isso temos também a coesão sequencial, responsável por criar condições que auxiliam na progressão textual. De forma geral, as flexões de tempo, as de modo dos verbos e as conjunções são mecanismos responsáveis pela coesão sequencial e auxiliam na coesão referencial.

Esses fatores inferem diretamente nas dificuldades existentes na tarefa de resolução de correferência, dado que estão relacionados diretamente a questões linguísticas e a habilidades cognitivas humanas complexas, de difícil reprodução por sistemas computacionais. Em poucas palavras, o desafio é: como inferir computacionalmente que a palavra “agrônomo”, que está sendo citada dois parágrafos abaixo da expressão “o agrônomo Miguel Guerra”, está se referindo a esta entidade e não a uma outra?

Portanto, o conjunto dessas expressões referenciais relativas a uma mesma entidade de mundo denomina-se **cadeia de correferência**. Esse conjunto é responsável pela construção coesa de um texto e por isso sua importância, já que a coesão é responsável pela compreensão textual. No exemplo “y”, podemos afirmar que “Miguel Guerra” é o antecedente e “Guerra” é a anáfora. Dessa forma, expressões correferentes fazem referência à mesma entidade, enquanto expressões anafóricas e catafóricas podem retomar uma referência ou ativar um novo referente. Anáfora e catáfora pressupõem um par ordenado, enquanto que a correferência remete à ideia de conjunto (Poesio; Stuckardt; Versley, 2016).



12.2.4 Referências Endofóricas e Exofóricas

Quando lidamos com relações referenciais é importante deixarmos claro que estas podem configurar-se em dois tipos: endofóricas e exofóricas. Referências endofóricas são aquelas que antecedem ou sucedem informação dentro de um texto. Essas comumente ocorrem na forma de anáforas ou catáforas, conforme visto nos exemplos anteriores. Já relações exofóricas referem-se a relações que ocorrem fora de um dado texto e necessitam de um prévio conhecimento de mundo, local ou momento para serem identificadas, como em:

z) O Bruxo do Cosme Velho foi homenageado em nossa cidade.

Notemos que no exemplo “z” não existem referências para os termos “Bruxo do Cosme Velho” e “nossa cidade”. É necessário recorrermos ao conhecimento de mundo para inferirmos que “Bruxo do Cosme Velho” refere-se a Machado de Assis. Da mesma forma, a referência do termo “nossa cidade” não está no texto, mas pode estar na memória do leitor ou na memória do escritor.

12.2.5 Correferência, coerência e coesão

Quando lidamos com a resolução de correferência, existem características que geralmente ficam implícitas em textos bem escritos e estruturados, mas que merecem atenção, dado que influenciam diretamente na obtenção de bons resultados. Dentro desse contexto, temos a coerência e a coesão textual. De acordo com Koch; Travaglia (2012) a coerência textual é algo que tem a ver com a boa formação do texto, não em um sentido gramatical, mas sim em nível de interlocução. A coerência é algo que se estabelece na interação, na interlocução ou em uma situação comunicativa entre duas pessoas. Em poucas palavras, a coerência é o que faz com que o texto tenha sentido, devendo ser vista como um princípio de interpretabilidade do texto e também com a capacidade que o leitor possui para calcular seu significado. A coerência é vista também como uma continuidade de sentidos perceptíveis no texto, a qual resulta em uma conexão conceitual cognitiva entre os elementos do texto. Como podemos perceber, a coerência é, ao mesmo tempo, semântica e pragmática aplicadas, pois a forma como construímos nossas ideias pode variar, de acordo com nosso conhecimento de mundo.

Paralelamente ao conceito de coerência temos a coesão. Ao contrário da coerência, a coesão é explicitamente revelada por meio de marcas linguísticas, sendo de caráter linear, dado que manifesta-se na organização sequencial de um texto. Em poucas palavras, a coesão está muito mais ligada à sintaxe e à gramática. Notemos que esses conceitos são muito importantes para a tarefa de Resolução de Correferência, dado que a correferência de um termo e seu antecedente é guiada por essa construção de ideias.

12.3 Abordagens Computacionais para Resolução de Correferência

Na literatura, encontramos uma grande variedade de abordagens que propõem resolver correferência em diversos idiomas, como: inglês, chinês, árabe, espanhol, galego, português, entre outros (Chang et al., 2012; Coreixas, 2010; Fernandes; Santos; Milidiú, 2014; Fonseca, 2014; Fonseca; Vieira; Vanin, 2015; Fonseca; Vieira; Vanin, 2014; Lee et al., 2017; Martschat; Strube, 2015; Ng; Cardie, 2002; Rahman; Ng, 2011a; Soon; Ng; Lim, 2001; Yang et al., 2008). Essas abordagens, em sua maioria, são voltadas para a língua Inglesa e baseadas em aprendizado de máquina. Contudo, é possível encontrarmos alguns modelos baseados em



regras linguísticas (Garcia; Gamallo, 2014; Hou; Markert; Strube, 2014; Lee et al., 2013). Veremos que, diferente dos modelos baseados em regras, o aprendizado de máquina pode se ramificar em diferentes propostas, como *Mention-Pair*, *Entity-Mention*, *Mention-Ranking* e *Antecedent-Trees*.

12.3.1 Modelos Baseados em Regras

Modelos baseados em regras consistem em uma série de passos que definem se duas menções são correferentes entre si. Abordagens baseadas em regras requerem um conhecimento prévio mais aprofundado referente ao idioma e ao domínio a serem tratados. Por exemplo, ao lidarmos com aprendizado de máquina, durante a implementação e seleção de *features*, caso sejam selecionadas *features* irrelevantes, a maioria dos algoritmos de treino consegue detectar e desconsiderar tal característica. Já em abordagens voltadas a regras, não temos essa flexibilidade. Cada regra deve ser elaborada cuidadosamente, pois não temos um modelo estatístico como apoio. Outra característica forte desses modelos é a forma como as menções são agrupadas. Ou seja, em abordagens baseadas em regras não existe etapa de treinamento: definidas as regras, uma menção m_j é comparada com todas as menções que a antecedem e, caso alguma das regras seja satisfeita, essas menções são agrupadas. Esse tipo de método é o mais utilizado pelos modelos de regras atuais (Garcia; Gamallo, 2014; Lee et al., 2013).

Um dos principais modelos de regras contido na literatura foi proposto por Lee et al. (2011). Denominado Stanford Multi-Pass Sieve, é um sistema para a resolução de correferência puramente baseado em regras linguísticas. Seu modelo possui dez *Sieves*/filtros, cujo objetivo é agrupar menções correferentes, caso cada regra ou conjunto de regras sejam satisfeitos. O modelo de Lee et al. foi proposto para o inglês, durante a CoNLL¹ (*Conference on Natural Language Learning*), ficando em primeiro colocado no *ranking* de melhores modelos. Os modelos foram avaliados por meio do *corpus* Ontonotes (Pradhan et al., 2011), em conjunto do CoNLL Scorer (Pradhan et al., 2014). Alguns anos após surgiram outras abordagens semelhantes com o mesmo propósito, como o trabalho de Garcia; Gamallo (2014), voltado ao português, ao espanhol e ao galego. Embora abordagens baseadas em regras possam ser de custoso planejamento, dado que cada idioma possui suas características, estas podem provar-se eficazes e competitivas, principalmente quando há carência por *corpora* anotados.

12.3.2 Modelos Baseados em Aprendizado de Máquina

12.3.2.1 Mention Pair

A primeira delas, e a mais popular, provavelmente por sua simplicidade, é a abordagem baseada em pares de menções. Basicamente, modelos que lidam com essa abordagem optam por efetuar seu treino por meio de pares de menções, de forma a determinar se duas menções são correferentes ou não. Os modelos baseados em pares de menções têm influenciado significativamente os trabalhos que propõem a resolução de correferência utilizando técnicas de aprendizado de máquina nos últimos dezesseis anos (Soon; Ng; Lim, 2001). Modelos baseados em pares de menções visam responder se devem ou não classificar como correferente uma menção m_j com um candidato antecedente m_i . Inicialmente, para treinar um modelo baseado em pares, é necessário extrair características/*features* que

¹<http://conll.cemantix.org/2011/>



possibilitem obter alguma informação proveniente da comparação entre m_i e m_j . No Quadro 12.1, podemos observar as *features* mais utilizadas pela literatura.

Quadro 12.1. *Features* mais comuns na literatura.

<i>Feature</i>	Descrição
Casamento de Padrões	Se m_i e m_j são lexicalmente iguais.
Casamento de Núcleos	Se m_i e m_j possuem o mesmo núcleo.
Alias	Se m_j é sigla de m_i ou vice-versa.
I_Pronome	Se m_i é um pronome.
J_Pronome	Se m_j é um pronome.
Número	Se m_i e m_j concordam em número (singular/plural).
Gênero	Se m_i e m_j concordam em gênero (masculino/feminino).
Nome Próprio	Se m_i e m_j são nomes próprios.
Aposto	Se m_j é aposto de m_i .
Distância entre Sentenças	Distância em sentenças entre m_j e m_i .
Distância entre Sintagmas	Distância em menções à entre m_j e m_i .
Classe Semântica	Se m_i e m_j possuem mesma classe semântica.
Hiponímia	Se m_i e m_j possuem uma relação de hiponímia.
Hiperonímia	Se m_i e m_j possuem uma relação de hiperonímia.
Sinonímia	Se m_i e m_j possuem uma relação de sinonímia.

Um dos grandes desafios ao utilizar uma abordagem baseada em pares de menções se dá no desbalanceamento entre as classes positiva (pares correferentes) e negativa (pares não correferentes). Ou seja, todo modelo requer, além de amostras positivas, amostras de pares negativos. Dada essa premissa, é necessário realizar a construção de pares. Nessa etapa, ao cruzarmos essas menções, conseqüentemente teremos muito mais amostras negativas do que positivas. Objetivando minimizar esse desbalanceamento entre as classes, alguns trabalhos propõem diferentes técnicas para geração de pares.

Soon; Ng; Lim (2001) realizam um pareamento distinto para cada uma das classes: para os pares positivos, dado o conjunto de menções $C = \{m_i, m_j, m_k, m_l\}$ (todas correferentes entre si), apenas as menções imediatamente adjacentes formam pares (Quadro 12.2): $P_p = \{(m_i, m_j), (m_j, m_k), (m_k, m_l)\}$. Para gerar os pares negativos, considere o conjunto de menções $M = \{m_m, m_n, m_o, m_p, m_q\}$ em que apenas as menções m_m e m_q são correferentes. Dentro desse contexto, a última menção deste conjunto, m_q , faz par com todas as anteriores, exceto com m_m : $P_n = \{(m_q, m_p), (m_q, m_o) \text{ e } (m_q, m_n)\}$ (Quadro 12.3).

Notemos que no Quadro 12.2, o conjunto de menções considerado é uma cadeia de correferência. Já no Quadro 12.3, o conjunto de menções não consiste em uma cadeia. Apenas as menções m_m e m_q são correferentes. Logo, não formam par.

Quadro 12.2. Geração de pares positivos proposta por (Soon; Ng; Lim, 2001).

Pareamento de amostras Positivas	
Conjunto de menções	m_i, m_j, m_k, m_l



Pares	m_i, m_j m_j, m_k m_k, m_l
--------------	--

Quadro 12.3. Geração de pares negativos proposta por (Soon; Ng; Lim, 2001).

Pareamento de amostras Negativas	
Conjunto de menções	m_m, m_n, m_o, m_p, m_q
Pares	m_q, m_p m_q, m_o m_q, m_n

Martschat; Strube (2015) propõem uma mesma metodologia para geração de pares positivos e negativos: dado documento D_x , que possua um conjunto de menções $M=\{m_i, m_j, m_k, m_l, m_m, m_n\}$ em que apenas m_l e m_i são correferentes, o conjunto de pares (positivos e negativos) será: $P= \{(m_l, m_k), (m_l, m_j), (m_l, m_i)\}$. Basicamente, para cada par correferente (m_x, m_y) , a geração de amostras negativas será realizada com as menções entre (m_x, m_y) . Notemos que essa construção é efetiva pelo fato de não gerar uma grande quantidade de amostras negativas. No entanto, devido a essa restrição, pode-se perder pares negativos que possuam informações relevantes.

Quadro 12.4. Geração de pares positivos e negativos proposta por (Martschat; Strube, 2015).

Pareamento de amostras Positivas e Negativas	
Conjunto de menções	$m_i, m_j, m_k, m_l, m_m, m_n$
Pares	m_l, m_k m_l, m_j m_l, m_i

Em (Fonseca; Vieira; Vanin, 2015), para um dado conjunto de menções $M=\{m_i, m_j, m_k\}$, temos: $P=\{(m_i, m_j), (m_i, m_k), (m_j, m_k)\}$. Basicamente, cada menção faz par com a próxima, independente de esta ser correferente ou não. Notemos que a quantidade de pares será muito maior que em (Martschat; Strube, 2015). Para minimizar o desbalanceamento entre as classes foi utilizado *random undersampling*, que consiste na seleção aleatória de n pares negativos, em que n é a quantidade de pares positivos. Por meio de experimentos, foi visto que os níveis de balanceamento “1 para 1” (um par positivo para cada par negativo) e “1 para 2” (um par positivo para cada dois pares negativos) foram os que obtiveram melhores resultados.

Quadro 12.5. Geração de pares positivos e negativos proposta por (Fonseca; Vieira; Vanin, 2015).



Pareamento de amostras Positivas e Negativas	
Conjunto de menções	m_i, m_j, m_k
Pares	m_i, m_j m_i, m_k m_j, m_k

12.3.3 Entity-Mention

Diferente do tradicional *Mention Pair*, o *Entity-Mention* (Yang et al., 2008) explora a propriedade de representação do discurso, tendo em vista o conhecimento de quando uma entidade é nova no discurso ou anafórica (semelhante à nossa metodologia de agrupamento proposta). Para conceber os pares, assume-se que uma instância de treino positiva consiste em $\{e_x, m_y\}$, na qual m_y é uma menção ativa e e_x é uma entidade parcial, encontrada antes de m_y . Para cada menção anafórica m_y , uma única instância de treinamento positivo é criada para a entidade parcial à qual m_y pertence. Para os pares negativos, é criado um grupo de instâncias para cada entidade cuja última menção ocorra entre m_y e o antecedente mais próximo de m_y . Por exemplo: considere o conjunto de menções $M = \{e_i, m_j, e_k, m_l, m_m, m_n\}$. Assumindo que neste conjunto tenhamos duas cadeias: $C1 = \{e_i, m_j, m_m\}$ e $C2 = \{e_k, m_n\}$ e m_n é a menção ativa. Teremos então, como conjunto dos pares positivos, $P_p = \{(m_n, e_k)\}$ e como conjunto de pares negativos $P_n = \{(m_n, m_m), (m_n, m_j), (m_n, e_i), (m_n, m_l)\}$. Basicamente, assumindo que m_n representa a menção ativa e e_k representa seu antecedente, notemos que temos duas menções entre elas (m_l e m_m). Nesse caso, toda menção ou cadeia pertencente a m_l e m_m forma par com a menção ativa. Nos Quadros 12.6 e 12.7 temos os pares gerados, considerando m_n como menção ativa. Notemos que a cada iteração a menção ativa será outra e com isso novos pares serão gerados, sempre utilizando o mesmo critério.



Quadro 12.6. Geração de pares positivos proposta por (Yang et al., 2008).

Pareamento de Amostras Positivas	
Conjunto de menções	$e_i, m_j, e_k, m_l, m_m, m_n$
Pares	m_n, e_k

Quadro 12.7. Geração de pares negativos proposta por (Yang et al., 2008).

Pareamento de Amostras Negativas	
Conjunto de menções	$e_i, m_j, e_k, m_l, m_m, m_n$
Pares	m_n, m_m m_n, m_j m_n, e_i m_n, m_l

Outro diferencial deste modelo focado em entidades está na forma de representar suas *features*: os autores definem três tipos de instâncias, que representam como as menções se relacionam: $link(e_x, m_y)$, em que m_y representa uma menção ativa e e_x representa uma entidade parcial; $has_mention(e, m)$, descrevendo todas as menções as quais determinada menção está ligada. Por exemplo, para a cadeia previamente mencionada, $C1 = \{e_i, m_j, m_m\}$, teremos então $has_mention(e_i, m_j)$, $has_mention(e_i, m_m)$; e o último denota as características de cada par de menções, seguindo a seguinte estrutura: $nome_da_feature(m_x, m_y, 0)$, representando respectivamente: o nome da *feature*, o par de menções e um valor binário (0 para falso e 1 para verdadeiro).

12.3.4 Mention-Ranking

No *Mention-Ranking model*, assim como o *Mention pair*, cada instância de treino $i(m_x, m_y)$ representa m_y e sua menção precedente m_x . Basicamente, as *features* que representam uma instância e um método para criar uma instância de treino são idênticas às utilizadas pelo *Mention Pair model*. A diferença reside em rotular as instâncias de treino, assumindo que I_y é um conjunto de instâncias de treino, criadas para a menção anafórica m_y , o *rank* para $i(m_x, m_y)$ em I_y é o rank de m_y entre os candidatos antecedentes, que é 2 se m_x é o antecedente mais próximo de m_y ou 1 caso contrário. Em poucas palavras, o antecedente mais próximo de sua anáfora recebe um *ranking* maior em relação às demais menções. Considere o seguinte conjunto de menções $M = \{m_i, m_j, m_k, m_l, m_m, m_n\}$, contendo as seguintes cadeias $C1 = \{m_i, m_k, m_n\}$ $C2 = \{m_j, m_l\}$. Notemos que, para m_n , teremos as seguintes instâncias:



Quadro 12.8. Instâncias de treino geradas para m_n

I_n	$Ranking$
Pares	
m_m, m_n	1
m_l, m_n	1
m_k, m_n	2

Notemos que m_i não faz par com m_n , mas sim com m_k , dado que m_i é antecedente de m_k . Dado que m_k é o antecedente mais próximo de m_n , o par recebe *ranking* 2. Já os demais pares (considerando m_n) recebem valor 1. Mesmo o *Mention-Ranking* não sendo muito popular, seus resultados são superiores às abordagens baseadas em pares de menções, como podemos ver em (Rahman; Ng, 2011b) e (Martschat; Strube, 2015).

12.3.5 Antecedent-Tree

Na proposta de (Fernandes; Santos; Milidiú, 2014), também baseada em pares de menções, os autores propõem um conjunto de regras, as quais objetivam reduzir a quantidade de pares menos propensos a serem correferentes. Assim, para um dado par de menções, caso pelo menos uma das regras do Quadro 12.9 seja satisfeita, o par é considerado válido para utilizar em seu treinamento (seja ele um par positivo ou negativo).

Quadro 12.9. Conjunto de regras para seleção de pares proposto por (Fernandes; Santos; Milidiú, 2014)

Regra	Descrição – Considera um par como válido se:
Distância	a quantidade de menções entre m_i e m_j não ultrapassar um determinado <i>threshold</i>
Classe Semântica	m_i e m_j possuem mesma classe semântica.
Combinação de Núcleos	o núcleo de m_i combinar com o núcleo de m_j .
Concordância em atributos de discurso	os atributos de discurso combinam para m_i e m_j . Esta regra consiste de um conjunto de regras proposto por Lee et al. (2013), o qual baseia-se em atributos de uma menção e seu falante.
Pronome J	m_j for um pronome e m_i concordar em gênero, número ou fala.
Pronome e Entidade Nomeada	m_j for um pronome e m_i for um pronome compatível ou uma entidade nomeada.

Referente ao motivo dos autores nomearem sua abordagem como *Antecedent-Tree model*, reside na forma de representar o agrupamento de suas menções: para representar o agrupamento de menções correferentes entre si são utilizadas estruturas chamadas de árvores. Uma árvore de correferência é uma árvore cujos nós são dirigidos às menções e os arcos representam alguma relação entre elas. Basicamente, para cada documento é gerado um conjunto de árvores e de sub-árvores, em que cada sub-árvore representa uma menção e seus referentes. Ou seja, cada anáfora pode ser considerada uma raiz ou nodo-pai e seus antecedentes podem ser considerados nodos-filhos.

Notemos que cada abordagem possui uma forma distinta para concepção de suas amostras de treino, assim como que para representar suas estruturas de agrupamento. Martschat;



Strube (2015) propuseram uma forma unificada de representar tais estruturas, os autores a chamam de estrutura latente. Basicamente, uma estrutura latente é representada por um conjunto de *arrays* “V”, “A” e “L”. Analisando-a, podemos verificar que uma estrutura latente pode ser abstraída à forma de um grafo, o qual “V” representa um conjunto de nós/menções; “A” representa o conjunto de arestas, e “L”(label), representa um sinal, positivo ou negativo, informando se dada menção é correferente de outra.

12.3.5.1 Modelos de linguagem

Os modelos de linguagem (Capítulo 15) foram também incorporados às tarefas de resolução de correferência, Joshi et al. (2019) apresentam uma análise do modelo BERT nessa tarefa.

Com os avanços recentes dos modelos de linguagem, abordagens alternativas, como a de Kirstain; Ram; Levy (2021), dispensam a necessidade de representação de *spans*.

12.3.6 Modelos Voltados à Língua Portuguesa

Para a língua portuguesa, Silva (2011) propôs um modelo para a resolução de correferência utilizando o conjunto de etiquetas semânticas providas pelo *corpus* do HAREM (Freitas et al., 2010). Para detectar tais categorias, Silva utilizou o *parser* PALAVRAS (Bick, 2000) e o reconhecedor de entidades nomeadas Rembrandt (Cardoso, 2012). Como base de conhecimento semântico, o autor utilizou o Tep2.0² (Maziero et al., 2008), um *thesaurus* contendo relações de sinonímia e antonímia para a língua portuguesa.

Lidando com aprendizado supervisionado, temos o trabalho de Coreixas (2010), o qual focou nas categorias “Pessoa”, “Local”, “Organização”, “Acontecimento”, “Obra”, “Coisa” e “Outro”. Como recursos, foram utilizados o *corpus* do HAREM (Freitas et al., 2010), o *parser* PALAVRAS (Bick, 2000) e o *corpus* Summ-it (Collovini et al., 2007). De forma a provar que o uso de categorias semânticas pode auxiliar na tarefa de resolução de correferência, o autor compara duas versões de seu sistema: a primeira, sem fazer o uso de categorias semânticas; e a segunda, fazendo uso de categorias. Como resultado, Coreixas mostrou que o uso de categorias podem prover melhorias significativas, dado que o uso de categorias pode auxiliar a determinar se dado par de menções é correferente ou não. O autor também mostrou a importância do conhecimento de mundo para esta linha de pesquisa.

Seguindo uma linha semelhante ao trabalho de Coreixas (2010), Fonseca; Vieira; Vanin (2014) propõem um modelo baseado em aprendizado de máquina, com foco em nomes próprios e nas categorias de entidades “Pessoa”, “Local” e “Organização”. Para detectar as entidades, foram utilizados os recursos Repentino (Sarmiento; Pinto; Cabral, 2006) e NERP-CRF (Amaral, 2013). Adicionalmente, para casos mais genéricos de entidades, os autores utilizaram listas, contendo substantivos comuns, que remetem a determinadas entidades, tais como: [advogado, agrônomo, juiz] para a categoria “Pessoa”, e [avenida, rua, praça, cidade] para “Local”.

Diferente dos trabalhos anteriores (Garcia; Gamallo, 2014) abordam um modelo baseado em regras (semelhante ao de Lee et al. (2013)), mas para múltiplos idiomas (português, espanhol e galego). Em seu trabalho, os autores focam apenas na categoria semântica “Pessoa”. Mais recentemente Fonseca (2018) propôs um modelo baseado em regras linguísticas, similar ao modelo de Lee et al. (2013), mas totalmente voltado ao português.

²<http://www.nilc.icmc.usp.br/tep2>



Adicionalmente, Fonseca (2018) introduziu conhecimento semântico ao seu modelo, provindo do Onto.PT (Gonçalo Oliveira; Gomes, 2014) e uma nova metodologia de agrupamento de menções (Fonseca; Vanin; Vieira, 2018). Basicamente, seu método (Figura 12.1) recebe como entrada uma lista ordenada de menções “M” e devolve uma lista de Cadeias contendo essas menções devidamente agrupadas, de acordo com o critério selecionado.

Figura 12.1: Algoritmo de agrupamento proposto por Fonseca (2018)

```

1: enquanto (tamanho de M > 0) faça
2:   int j ← 0;
3:   int[ ] S;
4:   para cada i ∈ C faça
5:     se M0 tem relação com Ci então
6:       Sj ← Ci
7:       j ← j + +
8:     fim se
9:   fim para
10:  se j > 0 então
11:    int k ← CritérioDeAgrupamento(M0, S, C)
12:    Ck ← M0
13:  senão
14:    C ← criaNovaCadeia(M0)
15:  fim se
16:  M ← Remove(M, 0)
17: fim enquanto

```

O método proposto foi baseado no trabalho de Heim (2008) e consiste em explorar a representação do discurso³. Para isso, assume-se que qualquer menção é nova no discurso se não possuir ligação de correferência com uma ou mais menções antecedentes. Essas ligações são consideradas utilizando o conjunto de regras proposto pelo autor. Assim, sempre que uma menção não possui uma relação referencial (nenhuma regra é satisfeita), uma nova cadeia é gerada. Basicamente utilizou-se uma lista de menções M (esta lista é ordenada na ordem em que as menções ocorrem no texto), contendo todas as menções de um documento de entrada. Cada menção pode ter uma ligação de correferência entre uma ou mais cadeias “C”. Dessa forma, os Ids dessas cadeias são armazenados em um vetor “S” (apenas se M₀ possui alguma relação de correferência com C_i (se alguma regra retorna o valor verdade). O próximo passo é responsável por agrupar uma menção atual M₀ a uma cadeia existente C_k ou criar uma nova cadeia de correferência, usando M₀⁴. Isso depende do critério de agrupamento utilizado. Dos critérios propostos pelo autor, vale mencionar o que obteve melhores resultados: o “Peso por Regra”, o qual para cada menção explora o conjunto C, com o objetivo de encontrar o maior peso (em nível de cadeias). O modelo

³Considera-se como representação do discurso a forma como as ideias são construídas em textos de linguagem natural, considerando sua construção linguística e seu contexto de uso, bem como suas formas de expressões comuns. Em linguística, a Pragmática é o ramo que analisa o uso concreto da linguagem pelos falantes da língua em seus variados contextos.

⁴Notemos que, para cada iteração, M₀ muda.



proposto possui um total de 13 regras e; para cada regra satisfeita, soma-se 1 à pontuação. Assim, se para uma dada cadeia C_x existem duas menções correferentes com uma menção M_0 (M_a e M_b) e M_a e M_b possuem respectivamente três e duas regras com valor verdade, o peso da cadeia será cinco.

12.4 Avaliação da Tarefa de Resolução de Correferência

A tarefa de resolução de correferência é complexa e envolve diferentes níveis de processamento. Logo, avaliar um modelo de correferência não é uma tarefa simples, dado que existem muitos detalhes a serem considerados, como a detecção de menções, agrupamentos realizados, agrupamentos não realizados. Na literatura encontramos cinco métricas propostas para avaliar esses modelos: MUC (Vilain et al., 1995), B-CUBED (Bagga; Baldwin, 1998), $Ceaf_e$, $Ceaf_m$ (Luo, 2005) e BLANC (Recasens; Hovy, 2011). Cada uma dessas métricas visa avaliar uma característica específica de cada modelo. Anualmente, competições como a CoNLL (Pradhan et al., 2012) são realizadas, visando motivar o desenvolvimento de sistemas. Nos anos de 2011 e 2012 essas competições foram voltadas à tarefa de Resolução de Correferência. Com o objetivo de avaliar os modelos participantes por meio de uma pontuação única, a conferência propôs uma nova métrica, chamada CoNLL (Pradhan et al., 2014). A métrica CoNLL consiste na média da medida-F de três outras métricas da literatura, como veremos nessa seção.

12.4.1 Métricas de Avaliação

12.4.1.1 MUC:

Baseada em cadeias, mede quantos agrupamentos de menções são necessários para cobrir as cadeias padrão. Por exemplo, considere que o conjunto K (cadeia de referência) seja composto pelas seguintes ligações (*links*) de correferência $\{AB, BE, CD\}$ e que o conjunto R (cadeia predita pelo modelo) seja composto por $\{AB, CD\}$. Para este caso podemos ver que falta uma ligação no conjunto R . teremos então $Abrangência = \frac{2}{3} = 0,67$ (67%) e $Precisão = \frac{2}{2} = 1$ (100%). De forma mais geral, o cálculo da métrica MUC pode ser obtido por meio das seguintes fórmulas:

$$Abrangência = \frac{\sum_{i=1}^{N_k} (\|K_i\| - \|p(K_i)\|)}{\sum_{i=1}^{N_k} (\|K_i\| - 1)}$$

$$Precisão = \frac{\sum_{i=1}^{N_r} (\|R_i\| - \|p'(R_i)\|)}{\sum_{i=1}^{N_r} (\|R_i\| - 1)}$$

Em que: K_i é i -ésima entidade padrão (*key entity* referência) e $p(K_i)$ é o grupo de partições criado por meio da intersecção de K_i e os *links* preditos pelo modelo; R_i é a i -ésima entidade predita pelo modelo (*Response entity*) e $p'(R_i)$ é o conjunto de partições criadas por meio da intersecção de R_i e K_i . N_k e N_r representam a quantidade de menções padrão e resposta, respectivamente.



12.4.1.2 B-CUBED:

Baseada em menções, gera resultados considerando as menções presentes e ausentes de cada entidade em dada cadeia. Basicamente, a métrica B-Cubed atribui um peso para as menções, baseando-se na quantidade total de menções existentes. Sua abrangência e precisão são obtidas por:

$$Abrangência = \frac{\sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \frac{\|K_i \cap R_j\|^2}{K_i}}{\sum_{i=1}^{N_k} K_i}$$

$$Precisão = \frac{\sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \frac{\|K_i \cap R_j\|^2}{R_j}}{\sum_{i=1}^{N_k} R_j}$$

Em que: K representa o conjunto das *key entities* (menções padrão) e R o conjunto de menções preditas pelo modelo. Por exemplo, dadas as cadeias de referência :

- $C_{K1} = \{A, B, C, D, E\}$;
- $C_{K2} = \{F, G\}$;
- $C_{K3} = \{H, I, J, K, L\}$.

E as cadeias preditas pelo modelo:

- $C_{R1} = \{A, B, C, D, E\}$;
- $C_{R2} = \{F, G, H, I, J, K, L\}$.

Cada menção possuirá o peso de $\frac{1}{12}$, dado que o total de menções existente é 12. Dito isso temos então:

$$Abrangência = \frac{1}{12} * \left[\frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{2}{2} + \frac{2}{2} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} \right] = 1 \text{ (100\%)}$$

$$Precisão = \frac{1}{12} * \left[\frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{5}{5} + \frac{2}{7} + \frac{2}{7} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} + \frac{5}{7} \right] = \frac{16}{21} = 0,76 \text{ (76\%)}$$

12.4.1.3 CEAF:

Baseada no alinhamento de menções e entidades, possui duas variações: CEAF_m (Φ_3) e CEAF_e (Φ_4).

$$\Phi_3(K, R) = \|K \cap R\|$$

$$\Phi_4(K, R) = \frac{2\|K \cap R\|}{\|K\| + \|R\|}$$

$$Abrangência = \frac{\Phi_x}{\sum_{i=1} \|K_i\|}$$

$$Precisão = \frac{\Phi_x}{\sum_{i=1} \|R_i\|}$$

Por exemplo, dadas as cadeias de referência:



- $C_{K1} = \{A, B, C, D, E\}$;
- $C_{K2} = \{F, G\}$.

E as cadeias preditas pelo modelo:

- $C_{R1} = \{A, B, C, D, E\}$.

As métricas CEAF utilizam o alinhamento entre as entidades ou menções para calcular seus resultados, dessa forma C_{K1} será alinhado com C_{R1} e C_{K2} não possuirá um alinhamento, dado que o modelo não obteve tal cadeia. Notemos que o número de menções alinhadas é 5. Portanto $\Phi_3 = 5$. Dito isso, temos:

$$\text{CEAF}_m: \text{Abrangência} = \frac{5}{7} = 0,71 \text{ (71\%)} \text{ e } \text{Precisão} = \frac{5}{5} = 1 \text{ (100\%)}$$

Para CEAF_e , dado que $\Phi_4 = \frac{2 * 5}{5 + 5} = 1$, temos:

$$\text{Abrangência} = \frac{1}{2} = 0,5 \text{ (50\%)} \text{ e } \text{Precisão} = \frac{1}{1} = 1 \text{ (100\%)}$$

Notemos que para a métrica CEAF_m o cálculo de precisão e abrangência é realizado considerando a quantidade de menções, para a métrica CEAF_e esse valor é considerado em Φ_4 . Contudo, para obtenção dos valores de precisão e abrangência, são usados os valores referentes a quantidade de entidades/cadeias.

12.4.1.4 BLANC:

BiLateral Assessment of NounPhrase Coreference avalia tanto *links* de correferência quanto os de não correferência. Basicamente, um *link* de não correferência é formado por duas menções que não são correferentes entre si. A métrica BLANC tem como objetivo recompensar as cadeias de correferência corretas, de forma proporcional ao seu tamanho. Temos, então, C_K e C_R respectivamente como: *links* de correferência padrão e preditos automaticamente e; N_K e N_R como grupo dos *links* de não correferência padrão e preditos automaticamente; Abrangência_C e Precisão_C remetem ao cálculo de abrangência e precisão dos *links* de correferência, e Abrangência_N e Precisão_N , aos *links* de não correferência.

$$\text{Abrangência}_C = \frac{\|C_k \cap C_r\|}{C_k}$$

$$\text{Precisão}_C = \frac{\|C_k \cap C_r\|}{C_r}$$

$$\text{Abrangência}_N = \frac{\|N_k \cap N_r\|}{N_k}$$

$$\text{Precisão}_N = \frac{\|N_k \cap N_r\|}{N_r}$$

Por fim, a precisão e a abrangência da métrica BLANC são calculadas, respectivamente, por meio das médias de Precisão e de abrangência, obtidas entre os *links* de correferência e de não correferência:

$$\text{BLANC}_{\text{Precisao}} = \frac{\text{Precisão}_C + \text{Precisão}_N}{2}$$

$$\text{BLANC}_{\text{Abrangencia}} = \frac{\text{Abrangência}_C + \text{Abrangência}_N}{2}$$

Por exemplo: dados os seguintes links de correferência:



- $C_{K1} = \{A-B, B-C, C-D, D-E\}$;
- $C_{K2} = \{F-G\}$.

E os seguintes links preditos pelo modelo:

- $C_{R1} = \{A-B, B-C, C-D, D-E\}$;
- $C_{R2} = \{F-G, F-I\}$.

Temos, então: $Abrangência_C = \frac{5}{5} = 1$ (100%) , $Precisão_C = \frac{5}{6} = 0,83$ (83%)

Considerando que os links de não correferência representam ligações entre todas as menções que não são referenciais, teremos então:

- $N_K = \{F-A, F-B, F-C, F-D, F-E, G-A, G-B, G-C, G-D, G-E\}$;
- $N_R = \{F-A, F-B, F-C, F-D, F-E, G-A, G-B, G-C, G-D, G-E, I-A, I-B, I-C, I-D, I-E\}$.

$$Abrangência_N = \frac{10}{10} = 1 \text{ (100\%)} , Precisão_C = \frac{10}{15} = 0,67 \text{ (67\%)}$$

$$BLANC_{Precisao} = \frac{0,83 + 0,67}{2} = 0,75 \text{ (75\%)}$$

$$BLANC_{Abrangencia} = \frac{1 + 1}{2} = 1 \text{ (100\%)}$$

12.4.1.5 CoNLL:

Amplamente utilizada para avaliar modelos de resolução de correferência, a métrica CoNLL calcula um score único, baseando-se no cálculo da medida-f das métricas MUC, B^3 e $CEAF_e$:

$$CoNLL = \frac{(F(MUC) + F(B^3) + F(CEAF_e))}{3}$$

12.5 Aplicações

Os ganhos da tarefa de Resolução de Correferência podem ser significativos, principalmente se considerarmos abordagens que utilizam apoio semântico (Fonseca, 2018; Rahman; Ng, 2011a). Em poucas palavras, existem muitas utilidades para a tarefa, e muitas outras tarefas de PLN podem se beneficiar de tal processamento. Na literatura, encontramos alguns trabalhos que fazem uso de tais modelos, como o de Vargas; Pardo (2018). Na presente abordagem, os autores fazem uso da ferramenta de prateleira chamada CORP (Fonseca; Vieira; Vanin, 2016b), até o momento a única ferramenta disponível para a língua portuguesa. Em sua produção os autores mostraram que, por meio da resolução de correferências, foi possível obter ganhos significativos na tarefa de Agrupamento de Aspectos para Análise de Sentimentos.

Muitas outras tarefas de PLN podem se beneficiar de seus resultados; como o Reconhecimento de Entidades Nomeadas (REN) (Amaral, 2013), Extração de Relação entre Entidades Nomeadas (Collovini et al., 2014) (ER), entre outras. Na Figura 12.2, no que diz respeito a tarefa de Reconhecimento de Entidades Nomeadas, considerando a cadeia [o agrônomo Miguel Guerra, de a UFSC, Guerra, Guerra, o agrônomo], podemos dizer que o sintagma nominal “Guerra” pode ser ambíguo e existe a possibilidade de que



modelos de REN (Capítulo 12) não o classifiquem corretamente. Por meio da tarefa de Resolução de Correferências podemos identificar que a menção “Guerra” corresponde ao agrônomo Miguel Guerra e, portanto, inferir uma mesma categoria de entidade nomeada (Pessoa). No contexto de extração de relação entre entidades nomeadas, considerando o sintagma nominal [o agrônomo Miguel Guerra, de a UFSC] é possível identificarmos a seguinte relação (Miguel Guerra, de, UFSC). E, identificando que “Guerra” faz referência a “Miguel Guerra” é possível inferirmos uma relação direta entre “Guerra” e “UFSC”.

Figura 12.2: Resolução de correferência e sua aplicabilidade (imagem extraída utilizando a ferramenta CORP (Fonseca; Vieira; Vanin, 2016b))

A discussão sobre a biotecnologia nacional está enviesada , pois está sendo entendida como sinônimo de transgenia . A opinião é de **[o agrônomo Miguel Guerra, de a UFSC [5]]** (Universidade Federal de Santa Catarina) . **[Guerra [5]]** participou de o debate “Biotecnologia para uma Agricultura Sustentável” , realizado ontem durante a 52ª Reunião_Anual de a SBPC (Sociedade Brasileira para o Progresso da Ciência) , sobre as biotecnologias apropriadas a o desenvolvimento de o país. **[Guerra [5]]** citou a micropropagação de vegetais (produção de mudas em laboratório , feita para evitar doenças e selecionar vegetaissaúdáveis) como exemplo de biotecnologia de baixo custo . Com ela , aumentou -se a produção de moranguinho , em o sul de o país , de 3,2 kg para 60 kg por hectare . Para **[o agrônomo [5]]**, o Brasil deve...



Capítulo 13

Pragmática

*Leidiana Iza Andrade Freitas
Vlândia Pinheiro*

O presente capítulo encontra-se em fase de desenvolvimento e será lançado nas próximas edições, portanto, apresentamos um breve resumo. A emergência e a constituição do domínio pragmático são antes de tudo imputáveis a uma situação de crise da filosofia, ocorrida no final do século XIX, em razão da qual as diferentes correntes de pensamento efetuaram um retorno radical à questão da linguagem. Do ponto de vista linguístico, a pragmática é um conjunto de teorias desenvolvidas com a perspectiva de analisar o uso concreto das linguagens naturais considerando a influência do contexto comunicacional, extrapolando assim a visão da semântica e da sintaxe. Dentre as principais teorias destacam-se a Teoria dos Atos de Fala, de Searle, as Teorias do Inferencialismo Semântico, de Sellars, Dummett e Brandom, e o Princípio Cooperativo, de Grice. Este capítulo terá como objetivo explicar tais teorias e filosofias da linguagem e apresentar os *frameworks* e aplicações de PLN que visam processar textos sob a ótica racional e pragmática das linguagens naturais. Em especial, o capítulo abordará a tarefa de Reconhecimento de Implicação Textual (*Recognizing Textual Entailment* – RTE) e demais sistemas de Inferência em Linguagem Natural (*Natural Language Inference* – NLI).



Parte VII

Dados e Modelos



Capítulo 14

Dataset e corpus

Cláudia Freitas

14.1 Introdução

A preparação de bons *datasets* (ou *corpora* anotados) para o PLN é um empreendimento que costuma envolver conhecimentos variados – de computação e linguística, no mínimo. Neste capítulo, fazemos uma apresentação de conceitos básicos e metodologias relacionados à criação de *datasets* (ou *corpora* anotados)¹. Afinal, se queremos avançar na área, mesmo levando em conta os grandes modelos de linguagem (LLM), precisaremos de *datasets* de alta qualidade, feitos para a nossa língua e cultura.

Um *dataset*, literalmente, é um conjunto (*set*) de dados (*data*). Dados são elementos que, organizados (ou distribuídos) de uma(s) certa(s) maneira(s), isto é, tratados, produzem informação. Praticamente qualquer coisa pode ser um dado. No PLN, os dados que usamos são dados linguísticos; nossa matéria prima é a linguagem humana, e cada língua individualmente.

Os dados podem ser

- palavras, que podem ser classificadas como substantivos, advérbios, verbos etc;
- postagens em rede social, que podem ser classificadas como ofensivas ou não;
- palavras ou unidades maiores, que podem ser classificadas como pessoa, lugar etc;
- pronomes, que podem ser classificados como masculinos, femininos ou neutros;
- documentos ou frases, que podem ser classificados como simples ou complexos;
- pares de frases, que podem ser classificadas como sinônimas (mais precisamente, classificadas quanto ao seu grau de similaridade) ou não;
- segmentos de textos, que podem ser classificados e relacionados como uma pergunta e a resposta a ela associada;
- sequência de caracteres que aparecem entre espaços em branco ou espaços em branco e pontuação, que podem ser classificados como palavras;
- posição que cada palavra ocupa em uma frase ou em um texto inteiro;
- frequência de cada palavra ao longo de um texto;
- sons, que podem ser classificados como fala humana, uma tosse ou uma risada;
- sons de fala, que podem ser classificados como uma palavra (“agente”) ou mais de uma (“a” “gente”).

¹Este capítulo contém adaptações de (Freitas, 2022), onde apresento os bastidores do processo de anotação -- planejamento e esquema de anotação, documentação e concordância entre anotadores - com mais algum detalhe, e de um ponto de vista linguístico. E, apesar de ter sido publicado primeiro, o capítulo (Caseli; Freitas; Viola, 2022) aproveitou muito do que já havia sido escrito para este capítulo.



Partindo dos exemplos acima, o elemento “palavra” pode virar um dado quando atribuímos a ele algum valor, como a sua classe gramatical (substantivo, verbo etc), classe semântica (pessoa, lugar etc), sua posição no texto ou a sua frequência.

No PLN, estes valores podem ser atribuídos aos dados de duas maneiras. A primeira delas é de maneira explícita – por exemplo, com cada palavra associada a uma informação do tipo PoS (classe de palavra, do inglês, *part-of-speech*), sendo essa informação do tipo *substantivo*, *verbo*, *pronome*, *advérbio* etc. Ou cada frase (ou palavra) associada a uma informação do tipo *polaridade de opinião*, sendo essa informação do tipo *positiva*, *negativa*, *neutra*. No primeiro caso podemos dizer que organizamos (ou distribuimos, ou classificamos, ou rotulamos) as palavras do texto conforme sua classe morfosintática, no segundo, podemos dizer que organizamos (ou distribuimos, ou classificamos, ou rotulamos) as frases (ou palavras) do texto conforme sua polaridade. O que há em comum em ambos os casos é a organização (ou classificação) dos dados conforme classes pré-estabelecidas que nos parecem relevantes para explorar o conteúdo linguístico, e a partir delas produzimos informação: por exemplo, se há mais opiniões positivas ou negativas (ou neutras) com relação a um determinado objeto.

Mas nem todo *dataset* com conteúdo linguístico precisa ter seus dados organizados de acordo com atributos “externos” ao texto. Grandes modelos de língua – modelos de previsão de palavras – têm como entrada imensos volumes de texto, sem informação linguística explícita associada (Capítulo 15). A informação capturada é a posição da palavra no texto e a frequência com que é usada, e isso já permite saber muito sobre as palavras, desde que tenhamos muitas delas. Mas não é deste tipo de *dataset* (que contém o que chamamos de *textos crus*) que nos ocuparemos aqui, e nem dos procedimentos que transformam posição e frequência em informação linguística, tema do Capítulo 10. Nosso foco está nos dados linguísticos que possuem alguma organização explícita humana, feita conforme classes pré-definidas por nós – *dados anotados*, que são *elementos* linguísticos que possuem classificações, anotações ou rótulos linguísticos que codificam alguma dimensão do nosso entendimento sobre as palavras, frases ou textos. Porque contêm classificações (ou análises) atribuídas aos elementos linguísticos, estes conjuntos de dados também podem ser considerados *corpora anotados*.

14.1.1 Dataset ou corpus anotado?

Um *corpus* é um conjunto de dados linguísticos. A utilização de *corpus* – palavra latina que significa *corpo* e que tem como plural a palavra *corpora* – vem de longa data nos estudos linguísticos e lexicográficos², e ganha força nos anos 1980 com a popularização dos computadores. A partir daí, e cada vez mais, *corpus* diz respeito a uma coleção de textos que pode ser processada por computadores. O material que compõe um *corpus* (os textos) é coletado com algum propósito (investigar ou explorar algum aspecto da linguagem, teórico ou aplicado) e foi produzido “naturalmente”, isto é, não estamos diante de frases artificialmente inventadas com o objetivo de construir um *corpus*.

Como inicialmente o uso de *corpus* estava vinculado aos estudos linguísticos, o propósito mais comum é (ou era) o de estudar a/uma língua e suas variedades. Mas, à medida que a utilização de *corpus* vai sendo ampliada para áreas além da linguística, como no PLN, os limites (e critérios) do que é um *corpus* vão também se ampliando. A característica da *naturalidade* - enunciados naturalmente construídos em situações reais de trocas linguísticas, aspecto fundamental se o interesse é estudar uma língua – é deixada de

²Lexicografia é a área que se dedica a fazer dicionários.



lado quando pensamos em *corpora* criados artificialmente (automaticamente) a partir de dados “naturais”. Este é o caso de materiais criados por meio de tradução automática, ou por meio do aumento artificial de dados, caso do WinoBias³ (Zhao et al., 2018), criado para estudar viés de gênero e no qual, por meio de regras, foi criado um material balanceado quanto ao gênero com o mesmo número de entidades (pessoas) do gênero masculino e do gênero feminino. Do mesmo modo, nada impede a utilização de textos criados por grandes modelos de linguagem na elaboração de um *corpus*, ou de um *corpus* composto pelas interações linguísticas entre pessoas e máquinas.

Corpus ou *dataset*? Podemos usar três critérios para diferenciar *corpus* e *dataset*: utilidade, tipo de anotação, e tamanho.

Um *corpus*, para ser considerado um *dataset* linguístico no PLN, precisa ter algum **tamanho**. No mínimo, tamanho suficiente para permitir avaliar de maneira confiável o resultado de uma análise automática e, idealmente, tamanho suficiente para treinar um modelo de aprendizado de máquina. Nos estudos linguísticos, conforme o fenômeno pesquisado, quantidade pode não ser uma exigência. Em ambos os casos (nos estudos linguísticos e no PLN), podemos dispor de material já classificado – ou anotado, ou rotulado, ou etiquetado. Esta anotação ou classificação pode ser de diferentes naturezas e pode estar associada a diferentes segmentos de texto (palavras, expressões, frases, parágrafos ou o texto inteiro). Por exemplo:

- Indicar a classe gramatical de uma palavra;
- Indicar se duas frases são sinônimas ou não;
- Indicar se um tweet expressa discurso de ódio;
- Indicar se uma palavra se refere a uma PESSOA;
- Relacionar uma palavra a pronomes e demais outras maneiras pelas quais esta palavra é referida ao longo de um texto;
- Indicar se uma palavra, expressão, ou tweet é favorável ou desfavorável com relação a alguém ou algo;
- Indicar se uma frase é uma boa legenda para uma imagem;
- Indicar, para um trecho de áudio de um enunciado linguístico, a sua transcrição textual;
- Indicar qual sentimento uma palavra ou segmento de texto evoca.

Todos esses rótulos ou anotações têm em comum o fato de codificarem a interpretação humana, assim outro critério relevante para a distinção entre *corpus* e *dataset* é o **tipo de anotação**. A anotação é um procedimento de análise de textos – um procedimento interpretativo – ainda que esta dimensão nem sempre seja lembrada. Mesmo quando a anotação é incluída em textos de maneira totalmente automática, ela está (ou deveria estar) tentando reproduzir a análise humana. E aqui, além do tamanho, temos outra diferença entre os *corpora* anotados usados no PLN e nos estudos linguísticos: nos estudos linguísticos é possível (e comum) a utilização de *corpora* que foram anotados de maneira completamente automática, desde que a anotação automática seja de qualidade. Já quando pensamos em *datasets* para o PLN, como sua utilidade está na avaliação e treinamento de ferramentas ou modelos, as anotações são feitas por pessoas, ou feitas por máquinas e revistas por pessoas. Neste caso, *datasets* são equivalentes a *corpora padrão ouro* (*gold standard*).

³<https://github.com/uclanlp/corefBias#readme>



É possível que a diferença quanto à nomeação – *corpus* anotado ou *dataset* – também se deva à **utilidade** em foco: quando aquilo que a Linguística chama de *corpus* anotado passa a ser usado como material para treinar modelos de linguagem, ele também pode ser visto como um *dataset*. Vejamos a apresentação do já referido WinoBias, criado para a verificação de viés de gênero na resolução de correferência (Capítulo 12). Retomaremos o WinoBias mais à frente, mas interessam aqui as duas primeiras frases do resumo do artigo que apresenta o material (grifo meu)⁴:

Apresentamos um novo benchmark, WinoBias, para resolução de correferência com foco no viés de gênero. Nosso ***corpus*** contém sentenças no estilo do esquema de Winograd com entidades correspondentes às pessoas às quais se faz referência por meio de sua profissão .

Já no repositório do projeto onde está o WinoBias, encontramos a seguinte apresentação (grifo meu)⁵:

Analisamos diferentes sistemas de resolução de anáfora para entender as questões de viés de gênero presentes em tais sistemas. Dando uma mesma frase como entrada para o sistema, mas apenas mudando o gênero do pronome na frase, há variação no desempenho dos sistemas. Para demonstrar a questão do viés de gênero, criamos o ***dataset*** WinoBias.

Nem todo *dataset* tem dados linguísticos, e nem todo conjunto de dados linguístico é um *corpus* anotado. Nem todo *corpus* anotado é considerado, no PLN, um *dataset* linguístico. Neste capítulo, trataremos de *datatsets* – *corpora* padrão ouro – que são conjuntos de dados com classificações linguísticas atribuídas por pessoas (Figura 14.1).

Figura 14.1: Nosso ponto de partida



14.1.2 Anotação linguística

A anotação é uma **atividade de classificação**: temos um conjunto de classes (as etiquetas) – também chamado de *tagset* – previamente definidas e critérios que guiam a classificação. O

⁴“Our corpus contains Winograd-schema style sentences with entities corresponding to people referred by their occupation (e.g. the nurse, the doctor, the carpenter).” <https://aclanthology.org/N18-2003.pdf>

⁵“We analyze different resolution systems to understand the gender bias issues lying in such systems. Providing the same sentence to the system but only changing the gender of the pronoun in the sentence, the performance of the systems varies. To demonstrate the gender bias issue, we created a WinoBias dataset.” <https://github.com/uclanlp/corefBias/tree/master>



que torna a anotação interessante – ou desafiadora – é que a classificação é feita levando em conta o contexto do enunciado linguístico. Nos exemplos do Quadro 14.1, temos anotações de classes de palavras (anotação de PoS) e anotação de polaridade, utilizada na tarefa de análise de sentimentos. A anotação de polaridade, nos exemplos, está presente em dois níveis: no nível da palavra propriamente – cada palavra recebe uma etiqueta indicando se a polaridade é positiva [+], negativa [-] ou neutra [0] – e no nível da frase – cada frase recebe uma etiqueta indicando se a polaridade é positiva, negativa ou neutra com relação ao objeto sendo analisado. A anotação pode ser codificada (ou formalizada) de diferentes maneiras, e a codificação do Quadro 14.1 é apenas ilustrativa. Na Seção 14.4.4 retomaremos aspectos de codificação e formalização.

Quadro 14.1. Exemplos de frases anotadas com PoS e polaridade de sentimento

1. TRISTE _{_SUBST_[0]} é _{_V_[0]} uma _{_ART_[0]} palavra _{_SUBST_[0]} de _{_PREP_[0]} 6 _{_NUM_[0]} letras _{_SUBST_[0]} [frase neutra]
2. Um _{_NUM_[0]} dos _{_PREP+ART[0]} livros _{_SUBST_[0]} mais _{_ADV_[0]} tristes _{_ADJ_[0]} que _{_PRON-Rel_[0]} já _{_ADV_[0]} li _{_V_[0]} [frase neutra?]
3. Sofri _{_V_[0]} com _{_PREP_[0]} a _{_ART_[0]} protagonista _{_SUBST_[0]} a _{_PREP_[0]} cada _{_PRON_[0]} nova _{_ADJ_[0]} página _{_SUBST_[0]} ; Sofri _{_V_[0]} quando _{_ADV_[0]} o _{_ART_[0]} livro _{_SUBST_[0]} acabou _{_V_[0]} [frase positiva]
4. Nunca _{_ADV_[0]} Sofri _{_V_[0]} tanto _{_ADV_[0]} para _{_PREP_[0]} ler _{_V_[0]} um _{_ART_[0]} livro _{_SUBST_[0]} [frase negativa]

Como já indicado, é fundamental ter em mente que a anotação é uma **atividade interpretativa**. Como em boa parte das vezes o que está sendo anotado é o resultado de um amplo consenso, ou do “senso comum”, ficamos com a impressão (errada) de que estamos diante de uma classificação objetiva. Decidir, por exemplo, se a palavra “TRISTE”, na frase 1, deve ser classificada como *substantivo* ou *adjetivo* é uma decisão que irá depender da teoria adotada na anotação. Na frase 2, analisar a palavra “um” como *artigo* ou *numeral* também é fruto de uma escolha (teórica), e não a codificação de um dado objetivo. E, embora aqui o aspecto interpretativo da anotação pareça um detalhe teórico que só interessa a linguistas, veremos na Seção 14.4.5 consequências nem um pouco irrelevantes desta crença na objetividade da classificação, que aparece disfarçada de senso comum.

Quanto à **importância do contexto**, vemos pelos exemplos que ainda que as frases 1 a 3 contenham palavras de polaridade de sentimento considerada negativa (“triste” e “sofri”), no exemplo (1) a frase não tem polaridade (ou tem polaridade neutra), no exemplo (2) é difícil decidir se estamos diante de um julgamento de valor (negativo) sobre o livro ou se diante de uma constatação, por isso a polaridade sugerida é seguida com um “?”, e no exemplo (3) temos uma frase que indica uma opinião positiva sobre um livro, apesar da menção ao sofrimento.

14.2 Datasets pra quê?

A existência de *datasets* linguísticos, ou *corpora* padrão ouro, é fundamental para uma série de tarefas e aplicações de PLN. E por que fundamental? São três os motivos, todos igualmente importantes.



O primeiro deles é consequência da popularização, no PLN e na IA, de métodos baseados em aprendizado de máquina (AM): precisamos de exemplos do que se precisa aprender. E mesmo com os avanços da área, **bons exemplos**⁶ **continuam necessários**, com a vantagem de agora os algoritmos necessitarem de uma quantidade menor deles, graças ao ajuste fino (Capítulo 15). Já se sabe que quanto mais cuidado na preparação dos dados, melhor o desempenho dos modelos – melhor a qualidade das predições, além da possibilidade de se usar menos dados⁷. Um bom *dataset*, fruto de uma anotação cuidadosa, pode ser visto como um atalho para o aprendizado, como um empurrãozinho que damos nos modelos para que atinjam logo o melhor resultado possível.

Então um motivo para investirmos na criação de *datasets* linguísticos é fornecer exemplos para que certos tipos de aprendizado possam acontecer de maneira eficiente.

O segundo motivo que torna *datasets* fundamentais em diversas tarefas de PLN é que eles **facilitam o processo de avaliação** de um sistema, ferramenta ou modelo, e de **comparação** entre eles. Isto porque se a anotação codifica, no *corpus*, a compreensão humana sobre algo, e o que queremos das máquinas em certas tarefas é que elas reproduzam esta compreensão humana, a melhor maneira de saber em que medida um resultado é bom é comparando-o com o entendimento humano. Nesse contexto, poder dispor de um *corpus* padrão ouro facilita muito as coisas. Sem ele, a alternativa para a avaliação é selecionar uma amostra do material analisado automaticamente e avaliar. Embora esta seja a única opção disponível em certos casos, não é ideal porque dificulta comparações com o desempenho de outros modelos/sistemas/ferramentas. Isto é, se cada ferramenta for avaliada de maneira “independente” a partir de uma amostra do seu resultado, será difícil uma comparação com os resultados de outras ferramentas. Outra desvantagem da avaliação por meio da análise de uma amostra, ainda que mais facilmente contornável, é que quando separamos uma amostra para fazer uma análise de erros, é mais fácil perceber aquilo que foi analisado de maneira errada (falsos positivos) do que aquilo que não foi analisado, mas deveria ter sido (falsos negativos)⁸.

O último motivo é um desdobramento dos anteriores: a partir do momento em que temos condições de treinar, avaliar e comparar resultados, temos condições de **avancar no PLN**, uma vez que o avanço na área pode ser medido pelo desempenho em tarefas. Ou seja, no PLN, um *dataset* é criado para ajudar a resolver algum problema ou tarefa⁹. Assim, para que o *dataset* exista, houve uma pergunta/problema/tarefa anterior que motivou a sua existência – alguma dimensão do PLN foi percebida como sensível e foi considerada digna de uma medição: o quanto o PLN é bom em responder perguntas, encontrar palavras de um certo tipo semântico, relacionar palavras que se referem à mesma entidade ao longo de um texto, traduzir, resumir ou simplificar um texto, dentre tantas outras.

E aqui aproximamos *datasets* e **avaliações conjuntas** (*shared tasks*). Uma avaliação

⁶Bons exemplos são exemplos representativos, variados e analisados de maneira consistente, como veremos na Seção 14.3.

⁷Os trabalhos (Souza; Nogueira; Lotufo, 2020) e (Pires et al., 2023) trazem dados da língua portuguesa a respeito disso, e a palestra de Rodrigo Nogueira sobre a adaptação e modelos de linguagem para o Português também: Adaptando Modelos de Linguagem para o Português: Passado, Presente e Futuro - com Rodrigo Nogueira

⁸Quando estamos tratando de *datasets* para avaliação no contexto de aprendizado de máquina, é importante que o material usado na avaliação não tenha sido utilizado nem nas etapas de treinamento e nem de validação do modelo. Esta separação do *dataset* em partes diferentes é importante para que a avaliação seja de fato um teste, e não uma avaliação em que o modelo está “roubando”, uma vez que seu desempenho está sendo testado nos mesmos dados em que foi treinado.

⁹Já um *corpus* anotado padrão ouro pode ter sido criado com a intenção de estudar um determinado fenômeno linguístico.



conjunta (ou uma *shared task*) tem como principal objetivo incentivar a pesquisa e desenvolvimento de uma área, uma vez que fornece uma estrutura experimental comum (os mesmos conjuntos de dados e as mesmas medidas de avaliação) (Santos, 2007). Além disso, avaliações conjuntas também são maneiras de divulgar uma tarefa. Se achamos que um determinado aspecto do PLN precisa ser avaliado, ou precisa de atenção, a criação de uma avaliação conjunta que a tematize é um caminho. E, como já sinalizado, avaliações conjuntas só existem se existem *datasets* associados (e quanto melhor o *dataset*, mais bem-sucedida a avaliação).

Voltando um pouco no tempo para exemplificar, foi a criação de um *corpus* padrão ouro (chamado “Coleção Dourada”) no âmbito da avaliação conjunta HAREM¹⁰, em 2007-2008, que permitiu o avanço na tarefa de identificação e classificação de entidades mencionadas em português. Foi a criação do *corpus* ASSIN que permitiu a realização da avaliação conjunta ASSIN (Avaliação de Similaridade Semântica e Inferência Textual)¹¹, levando ao avanço em tarefas de similaridade semântica e inferência em português.

No entanto, em ambos os casos estamos diante de tarefas (identificação de entidades mencionadas, de similaridade semântica) que já existiam para outras línguas e que careciam de recursos – *datasets* padrão ouro são *recursos* (Capítulo 1) – para que pudessem ser abordadas também em língua portuguesa. Mas que outras tarefas poderíamos ter? Que desafios ou tarefas o PLN tem e ainda não foram abordados por falta de recursos? Por isso, além de avaliar e treinar, um *dataset* permite, ainda que indiretamente, **pautar os rumos do PLN**, o que não é pouca coisa.

É definindo tarefas que vamos abrindo caminhos no PLN – tanto caminhos previamente explorados para outras línguas, mas ainda não pavimentados para o português, quanto caminhos realmente novos, ainda não explorados em nenhuma língua. Um problema, ou tarefa, é enfrentado quando temos os meios para fazê-lo – e a construção de conjuntos de dados é um dos meios de que precisamos, considerando as abordagens atuais. Para quais tarefas – quais práticas de linguagem – queremos ajuda das máquinas?

Com a utilização cada vez maior de grandes modelos de linguagem que se alimentam de imensos volumes de dados, estratégias e abordagens para mitigar a presença de viés indesejado – como as manifestações, na linguagem, de comportamentos racistas, sexistas, xenofóbicos, dentre outros – têm preocupado pesquisadoras e pesquisadores de PLN (Capítulo 24) e têm sido um caminho explorado no que se refere à criação de *datasets*. O que seriam tarefas voltadas para a “monitoria de diversidade”? O já mencionado WinoBias, por exemplo, foi criado para avaliar a presença de viés de gênero. Em 2023 foi lançada uma avaliação conjunta para a detecção de homofobia/transfobia com *datasets* em inglês, espanhol, hindi, tâmil e malaiala – mas não para português, porque naquele momento não havia *datasets*¹². Levando em conta a proliferação de conteúdo produzido automaticamente por grandes modelos de linguagem, Ignat et al. (2023) sugerem, por exemplo, o desenvolvimento de modelos capazes de identificar as partes interessadas no conteúdo gerado e seus tipos de interesse, como lucros comerciais ou interesses políticos. Novamente, para que tais modelos existam, precisaremos de *datasets*. E por este exemplo, vemos a imensa responsabilidade atribuída aos *datasets* – especificamente, aos dados, como veremos na Seção 14.2.1, e às pessoas responsáveis pela classificação dos dados, como veremos na Seção 14.4.5.

¹⁰<https://www.linguateca.pt/HAREM/>.

¹¹http://propor2016.di.fc.ul.pt/?page_id=381 e <https://sites.google.com/view/assin2/>.

¹²Informações e *datasets* disponíveis em <https://codalab.lisn.upsaclay.fr/competitions/11077>.



14.2.1 Sobre a importância dos dados

Quando indicamos que, para o AM, *datasets* são relevantes porque fornecem exemplos e permitem criar modelos, nem sempre nos damos conta da importância dos dados de um ponto de vista qualitativo. Isto é, pensamos nos dados como recursos de que precisamos dispor em abundância para produzir bons modelos de linguagem (Capítulo 15), mas não como fonte de “visões de mundo”. Afinal, se uma língua constrói e representa as visões de mundo e crenças de seus falantes, um modelo de língua (ou de linguagem) irá, mesmo que indiretamente, codificar visões de mundo e crenças subjacentes aos dados linguísticos com que foi treinado. De forma mais direta: se, no paradigma do AM com representações distribuídas, todo o conhecimento vem dos dados, precisamos entender bem que dados são esses.

Um exemplo famoso da importância dos dados vem da pesquisadora Robyn Speer, que criou um algoritmo de Análise de Sentimento baseado em representações distribuídas (Capítulo 10), usando como dados textos publicados da internet. Ela percebeu que o algoritmo estava classificando restaurantes mexicanos de maneira negativa, o que não encontrava respaldo na pontuação dada pelos usuários para avaliar os restaurantes e nem nos próprios textos das resenhas dos restaurantes. O motivo da avaliação ruim dos restaurantes mexicanos era que a palavra *mexicano* sempre aparecia associada a *ilegal: imigrantes mexicanos* estavam associados a *imigrantes ilegais*, levando à classificação de restaurantes mexicanos como algo negativo. A experiência e suas lições estão relatadas no post “*How to make a racist AI without really trying*” (“Como fazer uma IA racista sem fazer muito esforço”).

E, apesar das tentativas de mitigar viés indesejado utilizando pessoas comuns para classificar dados (Seção 14.4.5), ainda há muito por fazer. Um estudo de 2021, por exemplo, mostrou que textos ficcionais criados pelo modelo de linguagem GPT-3 reproduzem estereótipos de gênero: se há um personagem feminino, ele tem muitas chances de estar associado a palavras que remetem a ambiente doméstico/familiar e à aparência/corpo; se o personagem é masculino, há muitas chances de estar associado a palavras que remetem à política, guerra e tecnologia (Lucy; Bamman, 2021). E por que isso? Justamente porque é o padrão (e viés) encontrado nos dados. Em uma exploração da caracterização de personagens literários em obras de língua portuguesa que já estão em domínio público – obras brasileiras e portuguesas –, encontramos um padrão bastante parecido quando analisamos as caracterizações atribuídas preferencialmente a personagens masculinos e personagens femininos: personagens femininos são caracterizados sobretudo quanto à aparência, com destaque para a beleza (ou ausência dela); personagens masculinos caracterizados sobretudo quanto a traços de caráter como excelência, coragem, liberdade e sabedoria (Freitas; Santos, 2023).

Ou seja, os estereótipos produzidos pelo modelo de linguagem nada mais são do que a reprodução dos padrões que já estão nos dados. E os padrões que estão nos dados nada mais são do que comportamentos linguísticos que estão na nossa sociedade. Os dados são sempre um registro do que foi dito em algum tempo/espaço, são sempre dados do passado, e por isso, invariavelmente, modelos de previsão estão fadados a refletir o passado. Isso nos leva a uma reflexão: podemos, por meio de um passado alterado, prever um futuro que queremos? Que futuro queremos? Em outras palavras, será que *datasets* construídos artificialmente com o objetivo de eliminar vieses indesejados, ao estilo do WinoBias, que artificialmente (e indiretamente) constroem relações igualitárias de gênero, raça ou etnia, seriam capazes de produzir linguagem e, conseqüentemente, de criar mundos, mais igualitários? Quais as



implicações éticas disso? “Quem controla o passado, controla o futuro.”, dizia o slogan do estado totalitário do romance *1984*, de George Orwell.

14.3 Características de um bom *dataset* linguístico

Quando falamos de um bom *dataset* linguístico, ou de um bom *corpus* anotado, são cinco as características desejáveis: consistência, variedade, representatividade, documentação detalhada e tamanho.

- **Consistência** – por consistência, entenda-se que fenômenos semelhantes devem ser anotados (isto é, analisados) da mesma maneira. A consistência permite que algoritmos de aprendizado de máquina generalizem a partir dos dados e que as avaliações sejam confiáveis. Em outras palavras: se para aprender são necessários exemplos, mas os exemplos são inconsistentes – às vezes *Brasil* em construções do tipo *morar no Brasil* está anotado como LOCAL, às vezes está anotado como ORGANIZAÇÃO –, não será possível uma boa generalização, pois os dados ruidosos trarão dificuldade para esse processo.
- **Variedade** – na medida do possível, um bom *dataset* deve ser variado (ou balanceado) com relação aos fenômenos para os quais foi construído. Por exemplo, se desejamos um *dataset* para a tarefa de análise de sentimento/opinião com resenhas de produtos, um compilado de avaliações de páginas do tipo “Reclame Aqui” não é recomendado, pois contrará, invariavelmente, muito mais avaliações negativas do que positivas. Se desejamos um *dataset* para reconhecimento de fala (Capítulo 3), é importante que ele contemple a diversidade dos sotaques brasileiros.
- **Representatividade** – esperamos que os textos que compõem o *corpus/dataset* sejam representativos do tipo de texto que será alvo da aplicação, isto é, ao qual o modelo baseado nesse *dataset* será aplicado. Se a ideia é criar um modelo/ferramenta que irá procurar informações em relatórios técnicos, não é indicado que o modelo seja gerado a partir de um *dataset* que contém apenas textos jornalísticos, por exemplo. Por outro lado, considerando o que foi dito sobre a presença de viés indesejado nos dados, a representatividade pode ser uma armadilha. Dados representativos poderão conter também estereótipos, então talvez seja mais prudente pensar na representatividade como uma característica que não é absoluta. Podemos desejar um material que seja representativo da “vida real” em vários aspectos, mas talvez não em todos¹³.
- **Documentação** – um bom *dataset* é um *dataset* bem documentado, que informa a origem do conteúdo textual, as classes de anotação e as diretrizes usadas para anotar, as características e/ou formação dos anotadores. É por meio da documentação, por exemplo, que poderemos saber que o conteúdo de um *dataset* de análise de sentimento foi extraído do site “Reclame Aqui”, e que talvez seja pouco indicado para aprendizado equilibrado de avaliações positivas e negativas. Como já mencionado, outro ponto cada vez mais importante diz respeito aos cuidados relativos à presença de viés

¹³De um ponto de vista linguístico, a representatividade é uma característica bastante discutível. Se pensamos em *corpora* “gerais”, de que representatividade estamos falando? Ser representativo de algo significa ser uma parte (uma amostra) que contém as principais propriedades e características do todo que ela representa. Quando nosso objeto é uma língua, sabemos qual é o todo? Até onde vai uma língua? No mundo do PLN, a representatividade está associada sobretudo às tarefas – que por sua vez estarão associadas a um ou mais tipos de texto –, o que torna a busca pela representatividade mais factível.



indesejado nos dados. Uma vez que não é possível obter dados sem viés nenhum—línguas não são neutras, e uma língua constrói e dissemina visões de mundo e valores de uma comunidade linguística—, o material com que o PLN trabalha (enunciados reais proferidos por pessoas reais, e anotados ou revisados por pessoas reais) pode acabar incluindo e reproduzindo preconceitos. Com o objetivo de minimizar limitações científicas e éticas decorrentes de conjuntos de dados enviesados, tem sido proposto que se inclua na documentação de *datasets* informação detalhada a respeito dos falantes que produziram tais dados ou dos anotadores que analisaram e classificaram os dados em termos de *gênero*, *classe social*, *idade*, *etnia* e o que mais for possível obter (sempre respeitando questões de privacidade). Esta preocupação não é nova (Couillault et al., 2014), mas tem importância crescente no PLN (veja-se, mais recentemente, (Bender; Friedman, 2018)).

- **Tamanho** – por fim, o tamanho é um aspecto que precisa ser levado em conta. E por tamanho não me refiro apenas ao tamanho do *corpus* em *tokens* ou *palavras*, mas também à quantidade de fenômenos rotulados. Na anotação de PoS e de sintaxe, por exemplo, todas as palavras recebem alguma etiqueta; na anotação de entidades, polaridade ou correferência, por outro lado, apenas algumas palavras são classificadas, isto é, recebem uma etiqueta. Consequentemente, um *dataset* de anotação morfossintática poderá ser menor, em número de palavras, do que um de correferência ou de entidades. Ou seja, uma coisa é o número de palavras (ou de *tokens*), e outra o número de palavras (ou *tokens*) que recebem alguma etiqueta. Uma das características dos LLMs (grandes modelos de linguagem) é terem sido treinados com *datasets* gigantescos. Para conseguirem produzir material com o volume desejado e em quantidade de tempo razoável, grandes empresas fazem uso da anotação colaborativa (ou anotação *crowdsourcing*), que cada vez mais é alvo de críticas e preocupações, como veremos na Seção 14.4.5.

Por fim, como nem sempre teremos à disposição *datasets* com todas as características desejáveis, sobretudo no que se refere à variedade e tamanho, podem ser utilizadas diferentes técnicas para aumentar artificialmente o conjunto de dados linguísticos (método chamado de *data augmentation*).

14.4 Por onde começar?

Para criar um *dataset*, precisaremos de

- um problema ou tarefa: por exemplo, classificar enunciados como ofensivos ou não; classificar palavras ou grupos de palavras como remetendo a PESSOAS etc.
- etiquetas de anotação (*tagset*), com as quais iremos classificar os dados;
- instruções sobre como classificar os dados, conhecidas como esquema ou diretrizes (*guidelines*) de anotação;
- compilação ou criação de um *corpus* adequado para esta tarefa;
- uma maneira de codificar a anotação;
- pessoas responsáveis pela anotação;
- uma ferramenta de anotação (se for o caso);
- estratégias para otimizar o processo de anotação (se for o caso);
- maneiras de avaliar o resultado da anotação, isto é de avaliar a qualidade do *dataset* produzido (embora não tenha a ver com “por onde começar?”, avaliar a qualidade do material produzido é parte importante da preparação de *datasets*).



14.4.1 Definição do problema ou tarefa

O primeiro passo é ter clara a motivação para a criação do *dataset* – que problema ele se propõe a ajudar a resolver? Assim, é fundamental ter clareza sobre a motivação para a classificação dos dados: ao organizar os livros em uma estante, diferentes motivações (encontrar rapidamente os livros de que preciso ou tirar fotos de uma estante de livros para decoração) selecionam diferentes critérios para classificação (assunto, por um lado, e tamanho do dos livros, por outro), e levam a diferentes resultados – diferentes maneiras de dispor os livros na estante. Do mesmo modo, uma palavra como “Sofri” em “Sofri para terminar o livro” pode ser classificada como “negativa” se a motivação para a classificação é encontrar opiniões, e também pode ser classificada como um “verbo” se a motivação para a classificação é o comportamento morfológico. A sequência “Abri mão” em “Abri mão do prêmio” pode ser classificada como uma unidade se a motivação para a classificação é encontrar unidades de sentido (equivalente a “abdicar”, por exemplo), mas considerada duas unidades se a motivação para a classificação é construir um corretor gramatical, já que será necessário verificar, na correção, se as várias formas do verbo “abrir” estão corretas (“eles abriram mão do prêmio”, “abrirei mão do prêmio”).

14.4.2 Conjunto de etiquetas e instruções: o esquema de anotação

A tarefa/problema bem definidos também são cruciais para o desenvolvimento das classes de anotação (as etiquetas, e também chamamos de *tagset* o conjunto de etiquetas), que podem ser poucas como “positivo” “negativo” ou “neutro”, no caso de anotação de polaridades, ou mais de dez, como na anotação morfossintática.

Definir um conjunto de etiquetas e sua utilização refletem uma maneira de ver a tarefa. Por isso, quanto mais bem definido o problema (a tarefa), mais chances de sucesso. Caso seja necessário criar um esquema de anotação, devemos logo responder às seguintes perguntas: Qual o objetivo da anotação? A que ela serve?

Além disso, um esquema de anotação deve favorecer a generalização, mas sem perder informatividade. Esta generalização é o que fazemos quando, na frase (a), anotamos *nunca* e *tanto* como advérbios; *sofri* e *ler* como verbos; quando anotamos as frases (a) e (b) como frases positivas com relação a um alvo; ou quando anotamos, na frase (c) *sarampo*, *caxumba* e *rubéola* como DOENÇAS e *tríplice viral* como VACINA. Em todos os casos, embora cada uma das palavras ou frases comentadas (e anotadas) seja uma palavra/frase diferente, dizemos que elas são de um mesmo tipo – algumas são do tipo *advérbio*, outras do tipo *verbo*; algumas do tipo *positivo*; outras do tipo *negativo*; algumas do tipo DOENÇA, outras do tipo VACINA. E, apesar de as igualarmos, mantemos informação relevante (para as tarefas). Poderíamos classificar, por exemplo, *tanto* e *livro* como palavras de um mesmo tipo, porque ambas terminam em *o*, mas esta classificação não carrega informação relevante; ou podíamos classificar *sarampo*, *caxumba*, *rubéola* e *tríplice viral* com palavras do campo da SAÚDE ou MEDICINA. Em ambos os casos, embora até estejamos favorecendo alguma generalização, propondo classes mais amplas, estamos perdendo informação.

- a. Nunca_{ADV}[0] sofrir_V[-] tanto_{ADV}[0] para_{PREP}[0] ler_V[0] um_{ART}[0] livro_{SUBST}[0] [frase negativa]
- b. Sério, sofrir pra terminar o livro, o livro ainda consegue ser pior que os filmes. [frase negativa]
- c. Sarampo[DOENÇA], caxumba[DOENÇA] e rubéola[DOENÇA] são combatidas pela tríplice=viral[VACINA].



Na definição de um esquema de anotação é fundamental uma rodada inicial de anotação para as primeiras observações e ajustes. Porque uma coisa é a teoria e, outra, o contato com os dados. Nessa primeira rodada de anotação, pode ser que classes que julgávamos claras não estejam tão claras assim quando as palavras estão em contexto. Ou pode ser que as classes sugeridas sejam insuficientes para dar conta do que o *corpus* apresenta, e então é necessário criar novas classes. Assim, ao longo de um processo de anotação, as etiquetas iniciais (provisórias) podem ser confirmadas, ou os dados podem levar à reformulação das categorias iniciais, e o processo de anotação recomeça.

O processo de anotação com refinamento do esquema de anotação segue as seguintes etapas:

- Levantamento bibliográfico sobre o que já existe relacionado à questão, em termos teóricos e aplicados: *Existem anotações do mesmo tipo, ou diretamente relacionadas? Existem datasets para tarefas similares?*
- Elaboração de um esquema de anotação que contenha as primeiras generalizações acerca do fenômeno observado, isto é, a primeira proposta de etiquetas (classes);
- Aplicação dessas etiquetas a uma amostra mais ampla;
- Refinamento progressivo do esquema de anotação.

As instruções podem ser complexas como um manual ou uma gramática (como a documentação do *treebank* Bosque¹⁴, e da coleção dourada do HAREM¹⁵, que codificam morfossintaxe, e entidades genéricas, respectivamente), ou podem conter apenas alguns parágrafos de explicação, como ilustrado no Quadro 14.2, que traz (uma versão traduzida) das instruções para a anotação do *corpus* SNLI (*Stanford Natural Language Inference*), que contém 560 mil pares de frases. Nesse caso, o problema (ou tarefa) associado ao *corpus* é a identificação de certos tipos de relação semântica entre duas frases. As relações de interesse são *acarretamento*, *contradição*, *neutra*. Essas, portanto, são as etiquetas atribuídas aos pares de frases. As instruções são interessantes porque não pedem que os anotadores classifiquem pares de frases com as referidas etiquetas, pede “apenas” que produzam frases a partir de certas instruções, e voltaremos a esse *corpus* (e essas instruções) na próxima seção.

Quadro 14.2. Instruções de anotação do *corpus* SNLI (Bowman et al., 2015)

Vamos te mostrar a legenda de uma foto. Não vamos te mostrar a foto. Usando apenas a legenda e o seu conhecimento de mundo:

- **Escreva uma legenda alternativa que seja definitivamente uma descrição verdadeira da foto. Exemplo: para a legenda “Dois cães correndo em um campo.” você pode escrever “Existem animais ao ar livre”.**
- **Escreva uma legenda alternativa que possa ser uma descrição verdadeira da foto. Exemplo: para a legenda “Dois cães correndo em um campo.” você pode escrever “Alguns cachorros estão correndo para pegar um graveto.”**
- **Escreva uma legenda alternativa seja definitivamente uma descrição falsa da foto. Exemplo: para a legenda “Dois cães correndo em um campo.” você poderia escrever “Os animais de estimação estão sentados em um sofá”. Isso é diferente da categoria talvez correta porque é impossível para os cães correr e sentar.**

¹⁴<https://www.linguateca.pt/Floresta/BibliaFlorestal/>.

¹⁵https://www.linguateca.pt/aval_conjunta/HAREM/directivas.html.



14.4.3 Escolha do *corpus*

A escolha do *corpus* também está diretamente associada à tarefa – se o interesse está na detecção de opinião, é pouco indicado um *corpus* da Wikipédia, por exemplo, a não ser que o problema seja justamente a busca por opinião em textos que supostamente não deveriam conter opinião. Além disso, outras preocupações devem estar associadas à seleção do material:

- Tem direitos autorais ou pode ser usado (e disponibilizado) livremente?
- Leva em conta ou viola a privacidade de quem escreve (Capítulo 23)?
- Como lidar com conteúdo duplicado ou repostado, comum na internet?
- Como é a qualidade do texto?
 - Produzido originalmente em formato eletrônico ou resultado de processamento por OCR?
 - Possui muitas imagens, gráficos, tabelas? Neste caso, como lidar com este material?
 - O texto será normalizado (tudo em minúsculas, por exemplo), ou será mantida a grafia original?

Também é possível que o material textual que compõe um *corpus* tenha sido especificamente produzido para o próprio *corpus*, como o exemplo do SNLI mostrou¹⁶. Outro exemplo de *dataset* em que o material textual é produzido pelas pessoas com a finalidade de criar o *dataset* é o SQuAD (*Stanford Question Answering Dataset*), criado para a tarefa de perguntas e respostas (para a língua inglesa). Na tarefa de criação do *corpus*, as pessoas receberam parágrafos de documentos da Wikipédia, e sobre este material deveriam formular 5 perguntas sobre o conteúdo, e respondê-las. Além disso, não era possível usar o recurso de copiar & colar, o que forçou as pessoas a usarem suas próprias palavras na formulação das perguntas e das respostas. Nestes casos, a etapa de seleção do *corpus* deixa de existir, e é substituída pela etapa “Crie uma maneira engenhosa de produzir certos fenômenos linguísticos em grande escala”. Do mesmo modo, não há anotação ou classificação propriamente, uma vez que os enunciados criados já “nascem” organizados conforme certas classes (do tipo *pergunta* e do tipo *resposta*, ou do tipo *contradição*).

14.4.4 Codificação

Na preparação de um *dataset* é preciso decidir o formato dos dados (txt, csv, tsv, JSON, XML ou outros formatos) e o formato da anotação propriamente. Muitas vezes, o formato é determinado pela ferramenta que se usa para anotar; outras vezes, a ferramenta é escolhida em função (também) dos arquivos que suporta.

Por exemplo, um formato de anotação bastante comum (para a anotação de entidades mencionadas, mas não só) é o formato IOB (Inside-Outside-Beginning), ou BIO, em que o B significa o início (*begin*) de uma entidade, o I (*in*) a continuação dela, e o O (*out*), indica que a palavra em questão não pertence à entidade. Por exemplo, em “Ana conheceu a Serra da Mantiqueira ontem”, ou em “Sarampo e rubéola são combatidas pela tríplice viral” teríamos o seguinte, usando um *token* por linha:

¹⁶Embora, no PLN, não haja qualquer problema em considerar o SNLI um *corpus*, é possível que, de um ponto de vista dos estudos linguísticos, este material seja discutível no que se refere à dimensão *naturalidade*, isto é, “enunciados naturalmente produzidos”.



Figura 14.2: Anotação no formato IOB para entidades mencionadas.

```
Ana          B-PESSOA
conheceu    O
a           O
Serra       B-LOCAL
da          I-LOCAL
Mantiqueira I-LOCAL
ontem       B-TEMPO
```

```
Sarampo     B-DOENÇA
e           O
rubéola     B-DOENÇA
são         O
combatidas  O
pela        O
tríplice    B-VACINA
viral       I-VACINA
```

Já a anotação de dependências sintáticas, quando feita pela abordagem Universal Dependencies (Capítulo 6), segue o formato CoNLL-U, que por sua vez é um formato TSV (*tab-separated values*, isto é, valores separados por tabulação). São características deste formato:

- Cada frase tem um identificador;
- Na linha seguinte temos o texto da frase;
- Na linha seguinte começa a representação das palavras (*tokens*) da frase, com um *token* por linha;
- A separação entre uma frase e outra é feita por uma linha em branco;
- Cada *token* contém anotação em 10 campos separados por uma tabulação. Cada campo codifica diferentes informações morfosintáticas, e informações não preenchidas são marcadas com o caractere especial “_”.

A Figura 14.3 mostra uma frase no formato CoNLL-U¹⁷.

Figura 14.3: Anotação no formato CoNLL-U para dependências sintáticas.

```
# sent_id = 1
# text = Ana conheceu a Serra da Mantiqueira ontem.
1 Ana Ana PROPN - Gender=Fem|Number=Sing 2 nsubj - -
2 conheceu conhecer VERB - Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin 0 root -
3 a o DET - Definite=Def|Gender=Fem|Number=Sing|PronType=Art 4 det -
4 Serra Serra PROPN - Gender=Fem|Number=Sing 2 obj - -
5-6 da - - ADP - - 7 case - -
5 de de ADP - - 7 case - -
6 a o DET - Definite=Def|Gender=Fem|Number=Sing|PronType=Art 7 det -
7 Mantiqueira Mantiqueira PROPN - Number=Sing 4 nmod - -
8 ontem ontem ADV - - 2 advmod - -
9 . . PUNCT - - 2 punct - -
```

¹⁷Mais informações sobre o formato em <https://universaldependencies.org/format.html>.



Também é preciso decidir qual segmento de texto será anotado. Na preparação de *datasets* para detecção de desinformação e discurso de ódio, ou para análise de sentimento, por exemplo, é possível tanto uma anotação localizada, que classifica palavras e expressões ou, por outro lado, uma anotação que classifica segmentos maiores como frases ou documentos inteiros. Na Figura 14.4, temos um exemplo adaptado do *dataset* HateXplain¹⁸, utilizado na tarefa de detecção de discurso de ódio (e que foi utilizado para avaliar a capacidade do modelo de linguagem GPT de identificar este tipo de discurso). A apresentação do material explica que cada postagem no *dataset* é anotada de três perspectivas diferentes: (i) a classificação em 3 classes “frequentemente usadas para este tipo de anotação” (“ódio”, “ofensivo” ou “normal”), (ii) a comunidade-alvo (ou seja, a comunidade que foi vítima de discurso de ódio/discurso ofensivo na postagem) e (iii) as justificativas para a classificação, ou seja, as partes do texto que justificam a decisão de anotar algo como *odioso*, *ofensivo* ou *normal*. A Figura 14.4 traz um exemplo (traduzido e adaptado) do material, que está no formato JSON. Cada campo codifica o seguinte:

- Id: identificador único de cada postagem
- Anotadores: traz as anotações produzidas por cada anotador
- Label: indica a etiqueta (ou classe) atribuída a cada postagem. Os valores possíveis ódio (0), normal (1) ou ofensivo (2)
- [anotador_id]: o identificador exclusivo atribuído a cada anotador
- [target]: O alvo da postagem (no caso, adaptamos para os “extraterrestres”, dentre os quais estão os marcianos)
- justificativas: os elementos do texto (os *tokens*) selecionados como justificativa para a etiqueta atribuída pelos anotadores. Cada justificativa representa uma lista com valores 0 ou 1. Um valor de 1 significa que o *token* faz parte da justificativa selecionada pelo anotador. Para obter o *token* específico, podemos usar a mesma posição de índice em “post_tokens”
- post_tokens : A lista de *tokens* representando a postagem que foi anotada.

Como podemos ver pela Figura 14.4, diferentes porções do texto (diferentes *tokens*) foram selecionadas como justificativas para a classificação geral (ódio, ofensivo ou normal). Além disso, a mesma postagem foi considerada “discurso de ódio” para duas pessoas (anotadores 4 e 3), mas apenas “discurso ofensivo” para uma delas. E com isso passamos a uma dimensão fundamental da anotação: as pessoas que anotam.

14.4.5 Anotação: sabedoria de especialistas ou sabedoria da multidão?

A anotação é valiosa porque codifica o entendimento humano sobre alguma coisa, mas esta maneira de apresentar a anotação dá a entender que o “entendimento humano” sobre mundo é homogêneo, o que sabemos não ser verdade (e a divergência das interpretações sobre a classificação da frase dos marcianos é um pequeno exemplo). Além disso, nem todas as pessoas têm o conhecimento necessário para fazer todos os tipos de anotação.

Quando mencionamos anotadores **especialistas**, a referida especialidade pode ser de diferentes naturezas: conhecimento dos conceitos linguísticos (por exemplo, de classes de palavras, anáfora, acarretamento) e/ou do domínio (por exemplo, medicina em um *corpus* de Medicina) e/ou da tarefa em questão (anotar certos elementos usando certas ferramentas e seguindo certas instruções de anotação).

¹⁸<https://huggingface.co/datasets/hatexplain#dataset-structure> e <https://github.com/hate-alert/HateXplain>.



Figura 14.4: Anotação de discurso de ódio no formato JSON.

```

{
  "id": "24198545_gab",
  "anotadores": [
    {
      "label": 0, # ódio
      "anotador_id": 4,
      "target": ["extraterrestres"]
    },
    {
      "label": 0, # ódio
      "anotador_id": 3,
      "target": ["extraterrestres"]
    },
    {
      "label": 2, # ofensivo
      "anotador_id": 5,
      "target": ["extraterrestres"]
    }
  ],
  "justificativas":[
    [0,0,0,1,1,1],
    [0,0,0,1,0,0],
    [0,0,0,1,0,0]
  ],
  "post_tokens":
  ["marcianos", "são", "o", "câncer", "dessa", "nação"]
}

```

Inicialmente, boa parte das anotações utilizadas no PLN tinha como fonte ou inspiração os níveis de análise linguística ou alguma teoria linguística: morfologia, sintaxe, semântica, pragmática, papéis semânticos. E, por demandarem este tipo de conhecimento especializado, linguistas sempre foram responsáveis pela anotação. Mas isto não quer dizer que linguistas estão aptos a fazer todo e qualquer tipo de anotação, justamente porque anotações (ou boa parte delas) precisam ser feitas por especialistas, e linguistas são especialistas em linguagem. Há anotações que, idealmente, precisarão do apoio ou supervisão de outros profissionais, como em projetos de anotação e/ou de criação de *datasets* de áreas como medicina, direito, genética, geologia, dentre tantas outras.

Ao lado de tarefas que demandam conhecimento especializado, há tarefas que demandam “apenas” capacidade de leitura e fluência de escrita, como atribuição de orientação semântica (positiva ou negativa) a enunciados, identificação de entidades genéricas como PESSOA, ORGANIZAÇÃO, ou a criação de frases de determinados tipos, como no *corpus* SNLI, apresentado na Seção 14.4.2. E tarefas de PLN que no fim dos anos 2000 seriam vistas com descrédito por serem complexas e precisarem de uma quantidade brutal de dados são realidade 20 anos depois. A criação dos já referidos *datasets* SNLI e SQuAD são exemplos dessa tendência, apoiada por grandes empresas que pagam (mal) para que pessoas do mundo todo produzam dados de treino para as máquinas. Trata-se da *anotação colaborativa*,



ou anotação feita por “trabalhadores de multidão” ou “microtrabalhadores” (*crowdworkers*). Anotações colaborativas envolvem um grande número de anotadores e possibilitam produzir *datasets* enormes em uma quantidade de tempo relativamente baixa.

Apesar de possibilitar a geração de dados em larga escala, esta forma de anotar é alvo de muitas críticas, que vão desde a falta de comprometimento dos anotadores com a tarefa, remuneração baixa e exposição de quem trabalha a material com conteúdo repulsivo (por exemplo, classificar postagens em tarefa de detecção de discurso de ódio) (veja também Capítulo 24) até a crítica de que o que se chama de inteligência artificial é o fruto de milhares de trabalhadores invisíveis e precarizados¹⁹.

A falta de comprometimento com a tarefa pode levar a características indesejadas nos *datasets*, que por sua vez irão impactar tanto o que é aprendido quanto a avaliação da tarefa. O estudo de Gururangan et al. (2018), por exemplo, mostrou que os anotadores do *corpus* SNLI usaram estratégias bastante previsíveis, como o emprego de (i) advérbios de negação para criar frases com *contradição*, ou de (ii) hiperônimos (relação entre *animal* e *gato*, por exemplo) para frases com *acarretamento* (as instruções para a tarefa estão no Quadro 14.2, na Seção 14.4.2). Em consequência disso, a previsibilidade tornou a tarefa artificialmente fácil para as máquinas, levando a números de desempenho enganosamente altos.

A Amazon Mechanical Turk (AMT)²⁰ é a mais antiga plataforma para este tipo de anotação, e ficou mais conhecida pelo público pelas reportagens sobre quem são e as condições desumanas a que estão submetidos os “anotadores”, chamados *turkeys*. Admitindo que muito do que tem sido gerado pelas IAs tem como fonte os dados produzidos por estas pessoas, não chega a ser um grande exagero pensar que elas são as representantes do nosso senso comum.

Assim, outra crítica a este tipo de trabalho é que, se na imensa maioria das vezes o trabalho envolve anotações que codificam um certo senso comum, não sabemos exatamente de quem é este senso comum. A decisão sobre classificar um determinado enunciado como discurso de “ódio”, “ofensivo” ou “normal” (tarefa codificada no *dataset* HateXplain e em muitos outros do mesmo tipo), por exemplo, é feita com base em alguns exemplos fornecidos pela empresa responsável pela anotação. Mas será que, em temas como esses, o que desejamos é a codificação do “senso comum”? Como garantir que a anotação não vai justamente reforçar e amplificar, uma vez que provavelmente irá alimentar modelos de linguagem, o comportamento que deveria detectar e suprimir? Vejamos a apresentação do HateXplain, traduzida abaixo:

Antes de iniciar a anotação, os anotadores são explicitamente avisados de que a tarefa exige algum conteúdo de ódio ou ofensivo. Preparamos instruções que explicam claramente o objetivo da tarefa de anotação, como anotar os segmentos de texto e também incluímos uma definição para cada categoria. Fornecemos vários exemplos com classificação, comunidade-alvo e segmentação das anotações para ajudar os anotadores a entender a tarefa.

¹⁹A intensa divulgação e popularização de grandes modelos de linguagem tem chamado a atenção para a maneira pela qual estes *datasets* são construídos, veja-se <http://www.uol.com.br/tilt/reportagens-especiais/a-vida-dura-de-quem-treina-inteligencias-artificiais/>, <https://www.bbc.com/portuguese/geral-49234093> e <http://time.com/6247678/openai-chatgpt-kenya-workers/>.

²⁰O nome inspirado no “Turco Mecânico”, um robô jogador de xadrez do século 18 que os adversários enfrentavam pensando estar competindo contra uma máquina, quando na verdade havia um mestre de xadrez escondido lá dentro <https://www.bbc.com/portuguese/geral-49234093>.



A apresentação faz parecer que a identificação de algo como “discurso de ódio” (em oposição a “discurso ofensivo”, ou “normal”) é trivial. A manifestação de discurso de ódio, no Brasil, é crime previsto por lei²¹, mas os limites entre discurso de ódio e liberdade de expressão são alvo de discussão inclusive no meio jurídico. É razoável confiar no senso comum, ou na sabedoria da multidão, para indicar às máquinas aquilo que especialistas estão debatendo?

E voltamos à relevância da participação de especialistas nos processos de anotação. Se queremos codificar conhecimento, precisamos **qualificar** este conhecimento. Se estamos tentando fazer as máquinas nos ajudarem a classificar um determinado conteúdo, e esta classificação tem implicações jurídicas e limites pouco definidos, é importante que especialistas (do direito e dos direitos humanos, por exemplo) tomem parte no processo. E, neste caso específico, é uma maneira de restringir apenas a especialistas o contato com conteúdo sensível. Sem contar o óbvio: quanto melhor a curadoria, isto é, quanto melhor a classificação dos dados, melhor a qualidade das previsões que serão feitas e menos dados são necessários para um bom desempenho.

Por fim, quando menciono especialistas não-linguistas não quero dizer que o conhecimento linguístico formalizado só é útil em projetos de anotação linguística explícita, isto é, em projetos que envolvem conhecimento de teorias linguísticas. Diferentes tipos de anotação podem se beneficiar se o material já contém alguma anotação linguística anterior, e o ideal é que profissionais de diferentes especialidades colaborem. A anotação linguística, mesmo a mais simples como PoS, já é uma primeira organização dos dados textuais brutos, já codifica informação. O que difere a anotação linguística – de PoS, sintaxe, semântica etc. – das demais é que, por serem genéricas e fornecerem um primeiro nível de organização nos dados, facilitam e impulsionam outros tipos de anotação, como veremos na Seção 14.5.

14.4.6 Ferramentas de anotação

É importante contar com ferramentas que auxiliem o processo de anotação e revisão. Devido à expansão da atividade de anotação para além de tarefas estritamente linguísticas, há um grande número de ferramentas, algumas gratuitas, criadas para este fim. Em geral, a escolha da ferramenta é influenciada pela natureza da anotação ou da tarefa, e pontos para se levar em conta são:

- o formato dos arquivos de entrada e de saída;
- se a ferramenta suporta receber textos com outros tipos de anotação, ou apenas o texto cru que será anotado;
- se é possível corrigir (ou editar) anotações usando regras/padrões linguísticos, ou a revisão/edição só pode ser feita caso a caso;
- se há diferentes maneiras de visualizar os textos que serão anotados;
- se há opção de atalhos usando teclado (precisar clicar para anotar, apesar de aparentemente mais fácil, é pouco produtivo para a maioria das pessoas que anotam);
- se é possível trabalhar online ou apenas localmente;
- se a ferramenta suporta diferentes anotadores trabalhando simultaneamente;
- se a ferramenta é customizável;
- o tempo de familiarização necessário para começar a usar.

²¹Veja-se esta página do Ministério Público Federal dedicada ao tema: <https://respeitediferenca.mpf.mp.br/www/discurso-odio.html>.



Em geral, teremos sempre uma tensão entre uma ferramenta com várias funcionalidades, mas cuja utilização é mais difícil, e uma ferramenta mais fácil de usar, mas que oferece menos funcionalidades. Alguns exemplos de ferramentas são

- BRAT²²: para anotação de entidades mencionadas, correferência, dependências sintáticas, entre outras.
- Label Studio²³: para anotação de entidades mencionadas, perguntas e respostas, análise de sentimento, entre outras.
- Inception²⁴: para diversos tipos de anotação semântica e discursiva.
- WebAnno²⁵: para anotação morfológica, sintática e semântica.
- Arborator²⁶: para anotar e buscar dependências sintáticas em formato UD.
- ET: dois ambientes integrados para buscar, revisar e avaliar dependências sintáticas em formato UD: Interrogatório²⁷, para buscar e revisar anotações e Julgamento²⁸, para avaliar .

14.4.7 Estratégias de anotação

Um *dataset/corpus* padrão ouro pode ser construído (i) de maneira totalmente manual por uma única pessoa, por grupo pequeno ou por centenas ou milhares de pessoas, ou (ii) de maneira híbrida, quando é feita uma primeira rodada de anotação automática que depois é revista por pessoas²⁹. Este será o tema da Seção 14.5.

14.4.8 Formas de avaliação

Quem garante que um determinado *dataset* é bom, isto é, que os dados que contém são confiáveis, diversificados, representativos, codificados de maneira consistente e adequados à tarefa? Cada tarefa tem suas especificidades, e é avaliada de uma maneira. Este será o tema da Seção 14.6.

14.5 Procedimentos e estratégias de anotação e revisão

Depois de definido o problema ou a tarefa, escolhido o *corpus*, o esquema de anotação, a codificação, a ferramenta e as pessoas responsáveis pela anotação – e nada impede que a anotação seja feita por uma única pessoa, ainda que esta não seja a situação ideal – é hora de anotar propriamente.

Como vimos, diferentes práticas podem ser consideradas anotação:

- Ler (ou ouvir) um enunciado e atribuir uma etiqueta ao enunciado todo ou a partes dele;
- Produzir uma frase (enunciados em geral) a partir de determinadas instruções;
- Produzir perguntas e respostas a partir de um texto.

²²<https://brat.nlplab.org/>

²³<https://labelstud.io/>

²⁴<https://inception-project.github.io/>

²⁵<https://webanno.github.io/webanno/>

²⁶<https://arborator.icmc.usp.br>

²⁷<https://github.com/alvelvis/Interrogat-rio>

²⁸<https://github.com/alvelvis/Julgamento>

²⁹Já quando falamos de um *corpus* anotado, além dessas duas maneiras, existe a possibilidade de uma anotação 100% automática.



O foco desta seção está nas anotações do primeiro tipo, e em enunciados escritos. Quando a anotação consiste em atribuir uma etiqueta a um elemento do texto, é possível começar usando uma lista de palavras vindas de um léxico ou recursos lexicais (Capítulo 4 e (Freitas, 2022)). A língua portuguesa dispõe de alguns, para diferentes tarefas³⁰.

14.5.1 Palavras, regras e padrões linguísticos na construção de um *corpus* padrão ouro

Quando a anotação parte de uma lista de palavras (um léxico), a cada vez que uma palavra de interesse é encontrada, ela recebe uma etiqueta (ou mais de uma, caso o *esquema de anotação* preveja isso).

Por exemplo, o Emocionário³¹ contém uma lista de palavras que, em algum contexto, descrevem algum tipo de emoção e/ou sentimento da língua portuguesa³². O léxico tem o formato de uma lista de lemas associados a uma classe de emoção/sentimento.

Mas, como as palavras podem assumir sentidos diferentes conforme o contexto em que estão sendo usadas, listas de palavras, sozinhas, são insuficientes. Será necessária uma **revisão** da classificação inicial das palavras (da anotação) para corrigir os erros. Léxicos são sempre ótimos pontos de partida, mas não são eficientes para uma boa anotação. Por exemplo, quando usamos o léxico do Emocionário para anotar um texto, o verbo *chocar* é anotado com a emoção SURPRESA nas duas frases abaixo, mas apenas a ocorrência 2 remete à SURPRESA, e deveria ter sido anotada.

1. Após a queda, sua moto chocou-se com o muro e pegou fogo.
2. O mundo chocou-se com a notícia da morte do navegador.

Por isso, após a aplicação do léxico, ficamos diante da necessidade de revisão. Tanto a revisão quanto a anotação podem se beneficiar se o *corpus* já contiver uma análise linguística prévia: quanto mais informação (anotação) disponível no *corpus*, mais otimizada pode ser a revisão. Para a língua portuguesa, podemos contar com modelos de anotação morfossintática com desempenho bastante satisfatório, e esta anotação prévia pode ajudar o trabalho de revisão semântica se estamos diante de anotadores humanos especializados em conhecimentos linguísticos/morfossintáticos. Voltando aos exemplos 1 e 2, contar com uma **anotação prévia** e com uma *ferramenta adequada* permite listar os sujeitos do verbo “chocar”, facilitando a análise dos casos corretos e dos que precisam ser corrigidos. Ou seja, ao ler uma lista com palavras como “mundo”, “moto”, “caminhão”, “país”, “padre”, “avião”, “asteróide” e “Maria” e já sabendo que estas são palavras que estão sendo usadas como *sujeito* do verbo “chocar” (“mundo se chocou”, “moto se chocou”, “país se chocou” etc), podemos dividir as palavras em três grupos, como no Quadro 14.3.

³⁰Ver GitHub do Brasileiras em PLN: <https://github.com/brasileiras-pln>

³¹<https://www.linguateca.pt/acesso/corpos/dicemocoos.iso.txt>

³²Para mais informações, veja <https://www.linguateca.pt/Gramateca/Emocionario.html>.



Quadro 14.3. Distribuição de palavras que exercem a função de *sujeito* do verbo “chocar”

Grupo 1 = surpresa	Grupo 2 = colisão	Grupo 3 = dúvida
<p>mundo país</p>	<p>moto caminhão avião asteróide</p>	<p>Maria</p>

Das 7 ocorrências listadas, precisaremos ler o contexto apenas daquelas listadas no grupo 3, já que pessoas podem ficar *surpresas* (grupo1) mas também podem *colidir* com outras pessoas ou objetos (grupo 2). Após a análise do contexto e decisão sobre os casos de dúvida, precisaremos eliminar a etiqueta SURPRESA quando os sujeitos forem aqueles listados nas palavras do Grupo 2. Com uma ferramenta adequada, podemos fazer todas essas alterações de uma única vez. Será preciso escrever uma regra, verificar se ela faz o que era esperado, e aplicá-la ao *corpus*. A regra poderia ser algo do tipo:

```
token.lemma = "palavras_do_grupo2" and token.deprel == "nsubj"
and token.head_token.lemma == "chocar" >> token.emo == "0"
```

que utiliza informação morfossintática (codificada nos atributos lemma e deprel, que indica a função sintática). Ou seja, a regra diz³³:

Se

O lema do *token* é alguma das palavras do grupo 2 (informação do atributo *lemma*) E

A função sintática do *token* é sujeito (informação do atributo *deprel*, e *nsubj* é o código para a função de *sujeito*) E

O “pai” do sujeito é o lema “chocar” (informação do atributo *token.head_token.lemma*)

Então

A anotação de emoção do *token* será zerada (informação do atributo *emo*)

E por que não usar esta forma de trabalhar, baseada em regras, para todo o PLN? Isto é, se o interesse está na anotação de emoção, por que precisamos de um *dataset* para treinar e criar um modelo, por que não podemos fazer tudo por meio de regras? Acontece que esta forma de trabalhar é bem pouco prática/eficaz para textos diferentes daqueles que originaram as regras de revisão, e também para *corpora* realmente grandes. Para o “mundo controlado”³⁴ do *dataset*, abordagens baseadas em léxicos e regras são uma boa estratégia, mas nem tudo poderá ser (bem) resolvido por regras, e isto se deve a uma propriedade das línguas: independentemente do tipo de texto e do assunto tratado, sempre haverá um número imenso de fenômenos que não se repetem, e para os quais não teremos regras³⁵.

³³O exemplo é baseado na utilização da ferramenta Interrogatório <https://github.com/alvelvis/Interrogatorio>.

³⁴“Controlado” porque sabemos onde começa e onde termina, isto é, é um conjunto finito de enunciados linguísticos.

³⁵E não custa lembrar que os anos iniciais do PLN foram dominados pela crença de que o processamento linguístico poderia ser inteiramente resolvido por meio de regras linguísticas.

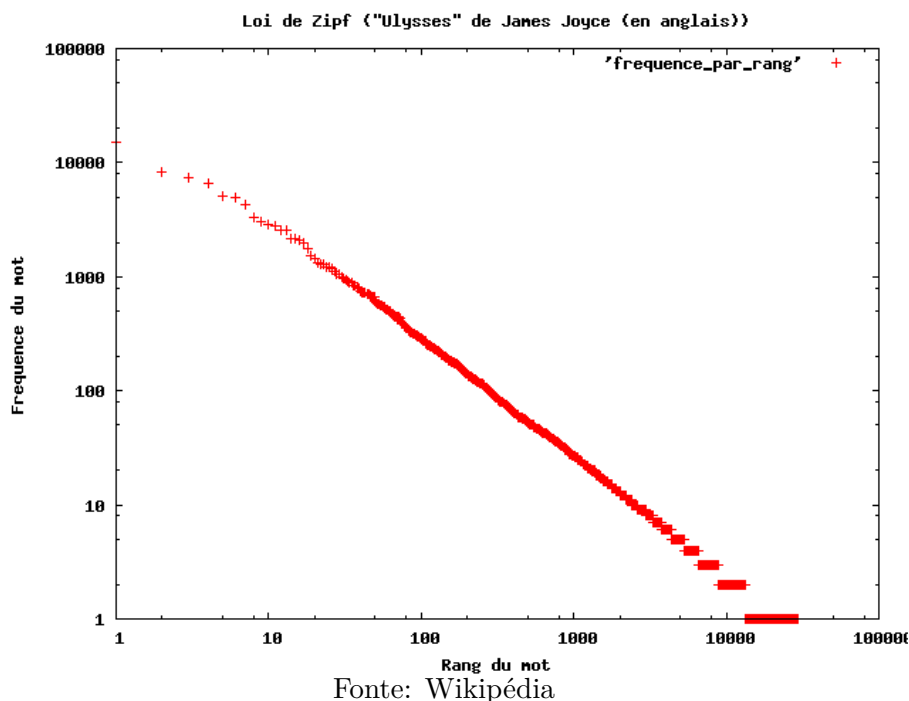


Quando observamos a distribuição das palavras em um *corpus*, sempre veremos muitas palavras com baixa ocorrência – muitas palavras que ocorrem 5 vezes, ainda mais palavras que ocorrem 4 vezes, ainda mais palavras que ocorrem 3 vezes, ainda mais palavras com ocorrência 2, e ainda mais palavras com apenas uma única ocorrência. Temos uma imensa proporção de palavras do *corpus* que são ocorrências singulares, de casos que ocorrem apenas uma vez. Este fenômeno tem um nome: *hapax legomenon* (*hapax legomena*, no plural), termo que vem do grego e que significa “sendo dito uma vez”.

Esta distribuição segue um padrão: poucos casos com muitas ocorrências, um número intermediário de casos com frequência média, e um número enorme de casos de frequência baixa. Além disso, quanto menor a frequência, mais palavras compartilham essa mesma frequência. Por isso há mais palavras de frequência 1 do que palavras de frequência 2, mais palavras de frequência 2 do que palavras de frequência 3 etc. Em *corpora* grandes, 40% a 60% das palavras são *hapax legomena* e outros 10% a 15% são *dis legomena* (ocorrem 2 vezes). E esta distribuição não se aplica apenas a palavras isoladas, mas a qualquer fenômeno, como a distribuição das entidades em um *corpus*, ou a estrutura dos sintagmas nominais. Este tipo de fenômeno é capturado pela lei de Zipf, uma lei empírica formulada no contexto da Linguística, e assim nomeada devido ao trabalho do linguista George Kingsley Zipf (1902–1950).

A Figura 14.5 mostra a distribuição das palavras do livro *Ulisses*, de James Joyce, e vemos que a frequência de uma palavra é inversamente proporcional à sua posição no *ranking* de frequências (isto é, a palavra na posição 1 tem frequência muito maior que a palavra na posição 10 mil). E vemos que o padrão se repete na Figura 14.6, que mostra a distribuição das 10 milhões de palavras mais frequentes em 30 wikipédias, isto é, em 30 línguas diferentes.

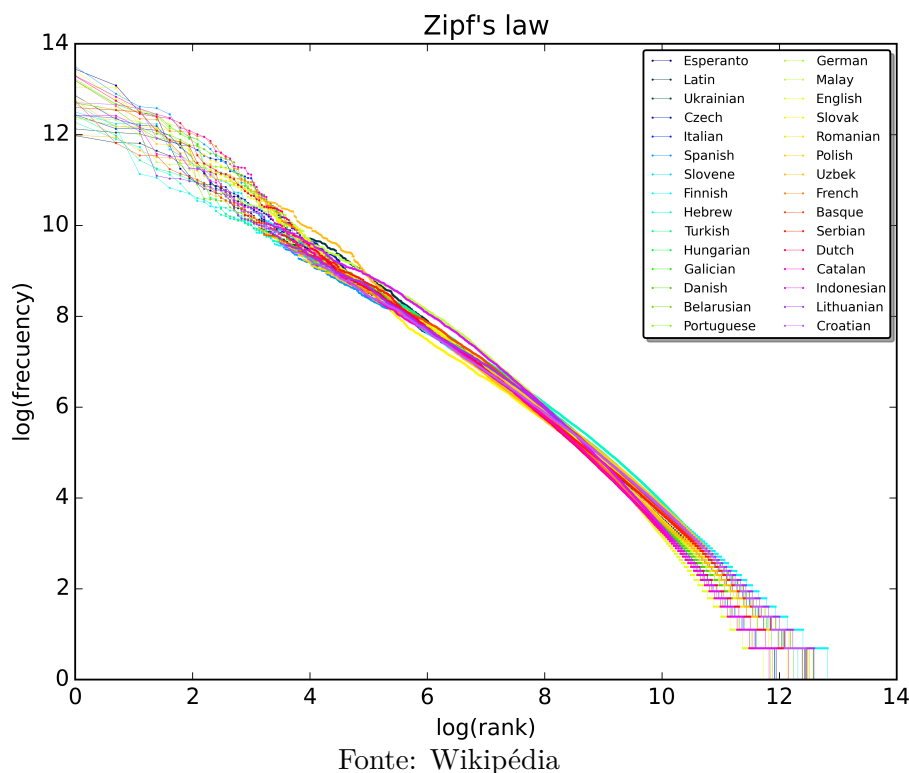
Figura 14.5: Distribuição das palavras do livro *Ulisses*, de James Joyce.



Quando ficamos cientes desta propriedade distribucional, conseguimos entender melhor



Figura 14.6: Distribuição das 10 milhões de palavras mais frequentes em 30 wikipédias.



por que as regras linguísticas são importantes, mas até um certo ponto: as regras capturam regularidades, e regularidades só existem se existe repetição. Por outro lado, se descartamos os casos com frequência 2 ou 1 (que não se repetem, portanto), estaremos olhando para uma língua mutilada, da qual uma imensa parte foi dispensada. Como uma boa parcela da língua não se repete, é difícil, apenas com regras, conseguir generalizar fenômenos. Assim, não importa a quantidade colossal de dados que tenhamos à disposição, sempre teremos uma proporção enorme de casos raros e não previstos. Por isso, abordagens baseadas em regras serão limitadas, e por isso também o aprendizado de máquina apresenta bons resultados: desde que haja dados, algoritmos estão cada vez melhores em prever eventos raros. Apesar da ressalva, o combinado léxico & regras continua sendo uma ótima maneira de construir *datasets* e *corpora* padrão ouro, considerando a alternativa de ler cada frase isoladamente, e anotar cada fenômeno de uma vez. A elaboração do PetroNER, que contém anotação de entidades da área de petróleo, seguiu esta maneira de trabalhar. O *corpus* tem 615 mil *tokens*, quase 20 mil entidades anotadas, e partiu de um léxico inicial de quase 390 mil instâncias de entidades (palavras) fornecidas por especialistas, mas nem todas foram encontradas no *corpus*. Foram criadas quase 2 mil regras para rever e melhorar a aplicação do léxico inicial ao *corpus*, e, mais da metade delas (56,2%) foi aplicada apenas uma vez. Quando reaplicamos essas regras a um novo *corpus*, com as mesmas características do PetroNER, precisamos criar mais de mil novas regras para que ele ficasse padrão ouro, e quase 55% das regras usadas no PetroNER não foram aplicadas.



14.5.2 Revisão de anotação

A revisão de uma anotação pode ser motivada pelo reconhecimento de que aquilo que se considerava padrão ouro, no fim das contas, não era tão ouro assim, ou porque estamos diante de um *corpus* novo, de um gênero ou domínio novos, o que acaba trazendo novos desafios para a anotação ou, como vimos na seção anterior, ou como parte de uma estratégia de anotação.

O que estas situações têm em comum é já contarem com uma anotação, ainda que de má qualidade. Por isso, o que precisamos neste momento é de **estratégias** para encontrar erros ou inconsistências na anotação. Por que estratégias? Um caminho óbvio é ler cada frase no *corpus*, e esse pode ser o primeiro passo se este é o primeiro contato com o *corpus* e conforme o tipo de anotação. Mas, como única abordagem para a detecção de erros, é uma estratégia limitada, por dois motivos: (i) uma mesma frase pode conter erros de diferentes naturezas, o que tem como consequência dificuldade em manter o foco e a consistência da revisão, deixando o processo de revisão mais suscetível a erros e mais demorado; (ii) conforme o tipo de anotação que está em jogo, há fenômenos que não costumam trazer erros – o reconhecimento de certas formas como artigos, verbos, advérbios, sujeitos, por exemplo – e rever cada unidade da frase envolve rever também estes casos mais fáceis. Ou seja, analisar *token a token* (ou palavra por palavra) também pode ser um desperdício de energia, em *certos casos*. Wallis (2003), por exemplo, recomenda trocar uma revisão linear, *token a token*, por uma **revisão transversal**, que permitiria ver os fenômenos em questão de forma ampliada e garantiria uma revisão consistente, como ilustramos na seção anterior com o exemplo de *chocar* e SURPRESA. Por isso – porque analisar linearmente, frase a frase, é pouco eficaz – uma etapa importante do processo de revisão consiste em encontrar uma maneira de detectar erros e corrigi-los.

O trabalho de revisão de anotação pode se tornar um imenso labirinto, e para chegar até o (padrão) ouro precisaremos de um mapa e de um guia que nos ajudem a encontrar os erros ou inconsistências: um caminho que seja, de preferência, rápido e seguro, isto é, que nos permita encontrar erros no menor tempo possível sem abrir mão da qualidade. A preocupação é dupla: do ponto de vista do resultado, buscamos análises consistentes e adequadas – uma anotação padrão ouro –; do ponto de vista do processo de revisão, buscamos eficiência – chegar aos melhores resultados consumindo pouca energia, isto é, em pouco tempo.

14.5.3 Até onde precisamos rever?

Quem já trabalhou com revisão de *corpus* sabe que o trabalho parece não ter fim – e por isso também é importante termos estratégias para conduzir e finalizar a anotação.

Do ponto de vista do aprendizado de máquina, a presença de alguns erros aleatórios não chega a prejudicar justamente porque não há o que ser aprendido, já que não será possível generalizar a partir de erros aleatórios. Mas o fato de um *dataset* ruidoso não influenciar a generalização não significa, necessariamente, que não seja necessário torná-lo o melhor possível. Em primeiro lugar, porque nem todas as ferramentas ou tarefas precisam estar associadas ao paradigma de aprendizado automático. Em segundo lugar, e mais importante, porque se quisermos que o *dataset* produzido também sirva para avaliação, a presença de erros será sempre um problema, uma vez que pode penalizar sistemas injustamente caso a análise automática esteja correta, mas a anotação padrão ouro esteja errada.

De todo modo, no ambiente de aprendizado automático, o que temos visto é que é sim possível aprender com dados um pouco ruidosos, e que uma anotação de alta qualidade não



se reflete, necessariamente, em facilidade de generalização. O trabalho de Souza; Freitas (2023) traz alguns dados para informar a discussão no que se refere à língua portuguesa.

Na preparação do *treebank* PetroGold v3³⁶, cada rodada de revisões foi acompanhada de uma rodada de “avaliação de aprendizagem” (uma *avaliação intrínseca*, como veremos na Seção 14.6) com o objetivo de verificar o impacto das melhorias linguísticas na aprendizagem automática. Os resultados estão na Tabela 14.2³⁷:

Tabela 14.2: Evolução de um *corpus* comparando quantidade de revisões e melhoria no aprendizado de máquina.

	<i>Tokens</i> revistos	F1
V1	–	88,53
V2	8.802	88,82
V3	9.314	90,22

Pela Tabela 14.2, vemos que apesar do esforço de revisão, o impacto na aprendizagem é limitado, sobretudo entre as versões 2 e 3. No fim das contas, um *corpus* maravilhosamente anotado leva a um desempenho muito semelhante àquele treinado em um *corpus* “apenas” bem anotado. No entanto, levar um melhor desempenho não é garantia absoluta de um *dataset* melhor, como veremos na próxima seção.

14.6 Como avaliar a qualidade do *dataset*?

Na Seção 14.2, vimos o que são e qual a relevância das avaliações conjuntas, que avaliam modelos e ferramentas a partir de um mesmo conjunto de dados/*dataset*. Nesta seção, o foco está em avaliar a qualidade dos *datasets*, já que um *dataset* de baixa qualidade não será capaz de dar suporte a uma avaliação confiável, seja ou não uma avaliação conjunta, e irá gerar um modelo de linguagem (ou previsões) de baixa qualidade.

Cada tarefa de PLN tem seus próprios métodos de avaliação. No entanto, quando tratamos de *datasets* anotados, alguns elementos da avaliação são comuns às diferentes tarefas. Uma vez que a anotação pode ser considerada uma tarefa de classificação, a avaliação também é feita nesses moldes.

Para avaliar um modelo ou ferramenta de classificação é comum utilizar as medidas de *precisão* e abrangência. A **precisão** mede (avalia) se a classificação que foi feita está correta (se a palavra analisada como *verbo* é realmente um *verbo*, ou se um comentário classificado como *ofensivo* é realmente *ofensivo*). A **abrangência** mede (avalia) se tudo o que deveria ter sido encontrado (e classificado) foi encontrado e classificado corretamente (se todos os verbos ou comentários ofensivos foram encontrados). A *precisão* mede a *qualidade* das classificações realizadas; a *abrangência* mede a qualidade da quantidade de elementos classificados, isto é, indica se tudo aquilo que deveria ter sido encontrado foi, de fato, encontrado.

Para calcular *precisão*, *abrangência* e a medida F (que é uma média harmônica entre ambas), classificamos os resultados da seguinte maneira:

- verdadeiro positivo (VP): o elemento foi detectado pela análise automática e foi classificado de forma correta.

³⁶<https://petroles.puc-rio.ai/files/Corpora/petrogold-v3.zip>

³⁷O trabalho está detalhadamente descrito em (Souza, 2023) e (Souza; Freitas, 2023).



- verdadeiro negativo (VN): o elemento foi detectado pela análise automática, mas foi classificado de forma errada.
- falso positivo (FP): o elemento foi detectado pela análise automática, mas não deveria.
- falso negativo (FN): o elemento não foi detectado pela análise automática, mas deveria.

Para calcular a precisão fazemos:

$$Precisão = \frac{VP}{VP + FP}$$

Para calcular a abrangência fazemos:

$$Abrangência = \frac{VP}{VP + FN}$$

Para calcular a medida F fazemos

$$F = 2 * \frac{Precisão * Abrangência}{Precisão + Abrangência}$$

Uma ferramenta pode ser muito precisa – todas as classificações que ela faz são corretas – e pode, igualmente, ter uma baixa abrangência – apesar de acertar bastante, há muitos casos que ficam de fora. Em geral, há uma tensão entre essas duas medidas: se afrouxamos a abrangência para encontrar mais casos, podemos diminuir a precisão, trazendo muitos casos errados. E, tentando melhorar a precisão, corremos o risco de perder em abrangência. Por isso, um bom desempenho se reflete em um equilíbrio entre essas medidas, e esta é a proposta da **medida F**: indicar em um único número uma combinação entre precisão e abrangência que reflita o desempenho geral.

Usamos medidas de F1, precisão e abrangência para avaliar modelos e ferramentas quanto à capacidade de generalizar a partir dos dados a que foram expostos no treinamento. Mas o quanto os dados permitem essa generalização? A capacidade de generalizar a partir dos dados está associada aos algoritmos utilizados, mas *datasets* também têm um papel nessa história – para o bem e para o mal –, pois algoritmos não fazem mágica.

14.6.1 Concordância entre-anotadores

No que se refere a *datasets* com dados anotados, além de bem documentado, de tamanho suficiente, variado e adequado à tarefa, um bom *dataset* é aquele no qual as anotações foram feitas de maneira consistente.

Nos *datasets* criados por meio de anotação *crowdsourcing*, algumas estratégias usadas para garantir a consistência são a revisão das análises por outros anotadores (e apenas aquelas anotações em que não foi necessária correção são consideradas) e o descarte de respostas ou análises desviantes.

Na anotação feita por equipes menores, em geral compostas por especialistas, a consistência de uma anotação é avaliada por meio da concordância entre anotadores (ou **concordância inter-anotadores**), que nada mais é do que a comparação entre duas ou mais anotações (análises) humanas. Quanto maior o índice de concordância, isto é, quanto mais convergência entre as análises, mais confiáveis elas são. Na concordância entre anotadores, a ideia de uma anotação *correta* é substituída pela ideia de uma anotação *consistente* (todos os anotadores analisaram os fenômenos da mesma maneira). O raciocínio subjacente é este: se diferentes pessoas, seguindo as mesmas instruções



(esquema e documentação de anotação), analisaram um fenômeno da mesma maneira, esta análise é confiável.

Levando em conta o trabalho envolvido na anotação, é comum que a verificação da concordância seja feita utilizando apenas uma amostra do *corpus*, isto é, temos as mesmas pessoas anotando os mesmos textos apenas em um subconjunto do *corpus*. O resultado da comparação destas anotações é um número que nos diz o grau de confiança que podemos ter nas análises/anotações; que nos diz o quanto as análises/anotações convergiram (ou divergiram), e a partir dele temos uma estimativa de o quão consistente está a anotação no *corpus* todo. Uma alta concordância entre anotadores é indicativa do potencial de reprodutibilidade, isto é, da possibilidade de reprodução das análises por outras pessoas (e espera-se que as máquinas sejam capazes de reproduzir estas mesmas análises).

Quando os resultados da concordância entre anotadores indicam uma baixa consistência entre as análises, é possível explorar algumas alternativas:

- Melhoria das instruções de anotação, como a inclusão de exemplos, tanto positivos (“faça assim nesses casos”) quanto negativos (“não faça assim nesses casos”);
- Aumento do tempo de familiarização com a tarefa e com o fenômeno sendo analisado, para que as pessoas responsáveis pela anotação estejam seguras do que estão fazendo;
- Reformulação das classes de anotação (do *tagset*). Os resultados da concordância entre anotadores podem ser baixos porque o conjunto de classes não está bem desenhado/modelado, mesmo que ele corresponda às classes de uma teoria.

A prática de medir a concordância entre anotadores é comum não apenas para avaliar a qualidade e consistência da anotação de um *dataset*, mas também em outras tarefas que visam avaliar a qualidade (e confiança) da análise humana. Este é o caso, por exemplo, da **anotação de erros de tradução automática**, uma das maneiras de avaliar a qualidade de uma tradução automática (Seção 18.3.4).

Algumas medidas utilizadas para o cálculo da concordância entre anotadores foram propostas por Cohen (1960), Fleiss (1971) e Krippendorff (1970). O Cohen’s Kappa foi originalmente projetado para medir a concordância entre apenas dois anotadores. Se a intenção é calcular a concordância entre múltiplos anotadores, é possível então levar em conta a média de cada par possível de anotadores. Já o Fleiss’ Kappa pode ser utilizado com mais de dois anotadores, e o Alfa de Krippendorff é flexível para ser aplicado em cenários com múltiplos anotadores, permitindo considerar diferentes níveis de desacordo.

A interpretação dos valores de Kappa irá depender da tarefa avaliada. Em termos gerais, Cohen sugere que valores 0 indicam nenhuma concordância, 0.01–0.20 indicam concordância fraca a nula, 0.21–0.40 indicam concordância razoável, 0.41–0.60 indicam concordância moderada, 0.61–0.80 indicam concordância substancial e 0.81–1.00 indicam uma concordância quase perfeita.

Por exemplo, na anotação semântica que visa a desambiguação de sentidos (Capítulo 8), as medidas de concordância costumam ser baixas, em torno de 0.7. Já na anotação sintática, que aparentemente seria mais complexa, é comum números superiores a 0.9.

Por fim, existem diferentes maneiras de lidar com as discordâncias, partindo da ideia de que alguns tipos de divergência entre análises humanas são mais graves (ou mais inofensivos) do que outros. O coeficiente Kappa de Cohen, por exemplo, pode ser usado na versão ponderada ou não ponderada. A diferença entre **Kappa ponderado** (*weighted Kappa*) e **Kappa não ponderado** (*non-weighted Kappa*) está na maneira como eles lidam com as discordâncias: o Kappa não ponderado trata todos os pares de categorias/etiquetas da



mesma forma, ou seja, todas as discrepâncias são tratadas igualmente, independentemente de sua natureza ou importância. Por outro lado, o Kappa ponderado atribui diferentes pesos às discordâncias entre as categorias/etiquetas, baseando-se em uma escala que reflete a importância atribuída a essas categorias. Por exemplo, em uma escala Likert de 1 a 5, o Kappa não ponderado considera que, no cálculo uma concordância (mais precisamente, de uma discordância), os valores 1, 2, 3, 4 e 5 são equivalentes, isto é, são, todos eles, indicativos de discordância. Já o Kappa ponderado reconhece que apesar dos valores 1, 2, 3, 4 e 5 indicarem a presença de uma discordância, o valor 1 está mais próximo de 2 do que 5, indicando que a discordância entre 1 e 5 é mais grave ou relevante que uma discordância entre 1 e 2.

14.6.2 Avaliação intrínseca

Por meio de uma avaliação da concordância entre anotadores ficamos sabendo que o material está consistente do ponto de vista da análise humana. Porém, não temos muitas pistas sobre o quanto esta análise humana – ou, o quanto o *dataset* (a combinação de análise humana + dados) – permite generalizar. Além disso, consistência na amostra utilizada no cálculo da concordância entre anotadores não significa, necessariamente, que não haja lapsos na anotação ao longo do material.

No PLN, **avaliação intrínseca** é, tradicionalmente, uma avaliação dos modelos e ferramentas (não do *dataset*), e é calculada utilizando medidas de precisão e abrangência apresentadas no início desta seção. Como o nome indica, a avaliação intrínseca é intrínseca à tarefa que está sendo avaliada. Ou seja, se temos uma ferramenta/modelo que faz anotação de sintaxe, ou de entidades, a ferramenta/modelo será avaliada nesta tarefa (parece óbvio, mas pode não ser assim, como veremos na **avaliação extrínseca**). Considerando a geração de um modelo de aprendizado de máquina, para que seja possível realizar uma avaliação intrínseca é importante que o material tenha um *tamanho* que permita todo o ciclo do aprendizado: treino, validação e teste. A partição de *teste* é uma parte do *dataset*, e apenas esta parte será alvo da avaliação (Capítulo 15).

Quando pensamos na avaliação intrínseca de *datasets* (e não dos modelos/ferramentas), estamos fazendo uma *inversão*, ou uma mudança de perspectiva. Se na avaliação intrínseca “original” verificamos a capacidade do modelo ou ferramenta de generalizar a partir dos dados a que foram expostos, na avaliação intrínseca de *datasets* verificamos (indiretamente) o quanto o *dataset* (mais especificamente, a natureza dos textos + a classificação dos dados) permitiu esta generalização, levando em conta as características do modelo/ferramenta. A partir dessa mudança de perspectiva, quando olhamos para o desempenho de um modelo e vemos números de precisão, abrangência e medida F, não vemos apenas uma avaliação do modelo ou ferramenta, ou o quão bom o modelo/ferramenta é, mas também até onde os dados permitiram ir, considerando os limites do modelo/ferramenta. Podemos entender da seguinte maneira: avaliação intrínseca é uma **análise dos erros** de um modelo/ferramenta que pressupõe que o material que serviu de treino está bem anotado e que o modelo/ algoritmo tem um desempenho que não é aleatório. Na Tabela 14.3, por exemplo, vemos o resultado de uma avaliação intrínseca de um modelo/ferramenta que tem um desempenho geral de 89,09%, um desempenho muito bom, portanto. O que vemos na tabela, no entanto, é o desempenho de cada classe individualmente (considerando um *tagset* com 30 classes/etiquetas).



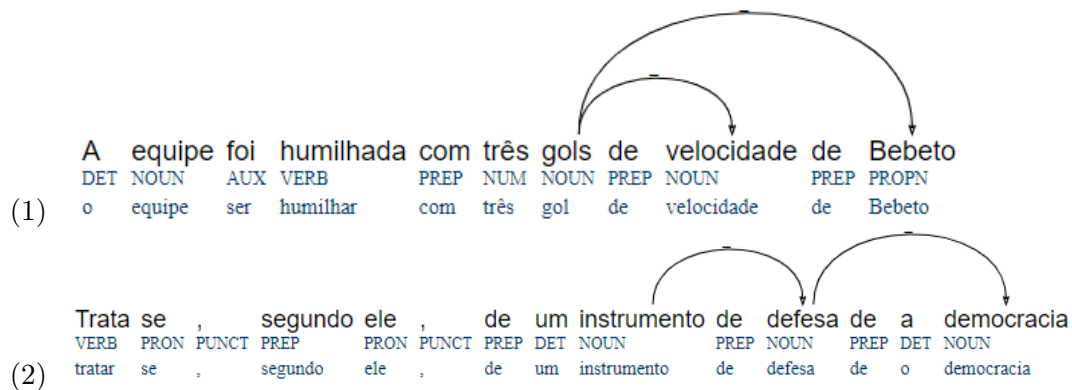
Tabela 14.3: Desempenho de um modelo (desempenho global de 89,09%) para cada classe aprendida

Etiqueta	Qtde	F1	Etiqueta	Qtde	F1	Etiqueta	Qtde	F1
classe 1	1786	99,27%	classe 11	314	89,81%	classe 21	564	74,47%
classe 2	101	99,01%	classe 12	346	89,02%	classe 22	185	71,89%
classe 3	1920	98,80%	classe 13	1269	88,02%	classe 23	78	62,82%
classe 4	178	96,07%	classe 14	140	86,43%	classe 24	29	62,07%
classe 5	447	94,85%	classe 15	166	86,14%	classe 25	92	61,96%
classe 6	319	93,73%	classe 16	1375	84,73%	classe 26	435	61,84%
classe 7	129	93,02%	classe 17	154	83,77%	classe 27	115	60,00%
classe 8	332	92,77%	classe 18	308	80,19%	classe 28	81	55,56%
classe 9	248	92,34%	classe 19	213	77,93%	classe 29	36	41,67%
classe 10	666	90,09%	classe 20	110	77,27%	classe 30	14	78,57%

Na tabela, vemos que as classes 14 a 30 foram as mais difíceis, e que as classes/etiquetas 23 a 29 foram especialmente difíceis. Vemos também que, exceto pela classe 26, as classes 23 a 30 são menos frequentes (na partição teste) que as demais. Assim, quando trocamos de perspectiva e olhamos para o desempenho nos dados individualmente, e não para o desempenho global do modelo, pensamos: como fazer para melhorar a classificação prevista nesses casos críticos, sem piorar os demais? O que esses casos têm de especial? Acrescentar mais exemplos para cada classe resolve? Ou o problema está nas instruções de anotação, que não deixam claro como exatamente classificar em certos casos? Ou não há o que fazer, e o problema de generalização é apenas da ferramenta/modelo?

A limitação desta maneira de avaliação é, justamente, a impossibilidade de diferenciar uma dificuldade de generalização oriunda do algoritmo de aprendizado de uma dificuldade de generalização oriunda de análises inconsistentes ou da falta de exemplos suficientes no material padrão ouro. Por isso, é importante garantir que há **consistência** na análise humana, o que vemos na concordância entre anotadores. Outra limitação desta forma de avaliar é que ela se restringe à análise da partição ‘teste’ de um *dataset*, uma partição pequena. Com tantas limitações, qual a vantagem?

Por meio do resultado de uma avaliação intrínseca das classes, podemos verificar o desempenho relativo à generalização de cada classe utilizada, individualmente (e sempre em contraste com o padrão ouro), e com esse olhar de análise iremos tentar melhorar os resultados (ou seja, melhorar o modelo) “melhorando” o conjunto de dados, e mexendo apenas nele. Há classificações mais difíceis de serem aprendidas? Por que? Essas perguntas têm como pano de fundo a constatação de que nem sempre o que é fácil para as pessoas é fácil para as máquinas. Para uma pessoa escolarizada, por exemplo, é simples diferenciar a estrutura (1) da estrutura (2). Mas de um ponto de vista formal temos exatamente a mesma sequência: substantivo + preposição + substantivo + preposição + substantivo.



Na frase (1), no entanto, o último substantivo da sequência está relacionado ao primeiro elemento (*Bebeto* está associado aos *gols*, e não à *velocidade*); na frase (2), o último substantivo da sequência está relacionado ao segundo elemento (*democracia* está associado à *defesa*, e não a *instrumento*).

O que podemos fazer para que esta distinção fique mais nítida para as máquinas? Mais exemplos ajudam? Ou o material já tem exemplos suficientes de ambas as estruturas (e neste caso não há o que fazer)?

Uma das vantagens deste tipo de avaliação – que joga luz sobre os *datasets*, e não sobre as ferramentas – é a possibilidade de perceber, de forma localizada, os obstáculos para generalização, e então atuar de forma direcionada para construir *datasets* “otimizados” para tarefas. Trata-se de uma forma de avaliar a qualidade da anotação que sinaliza, a partir do “ponto de vista das máquinas”, onde há espaço para melhoria no conjunto de dados.

Esta é uma abordagem que nos permite uma visão simultaneamente quantitativa e qualitativa; panorâmica e detalhada dos resultados de uma análise automática, capaz de guiar intervenções e melhorias onde de fato elas são necessárias. Além disso, repetindo o ciclo de aprendizagem após as intervenções é possível verificar se a solução proposta (inclusão de mais exemplos, redefinição das classes, melhoria das instruções de anotação) realmente facilitou a generalização. Ou seja, esta é uma abordagem não só para avaliar, mas que permite também medir o quanto as revisões melhoram um *corpus*, e nos ajuda a decidir um ponto de corte na revisão, respondendo à pergunta “Até onde precisamos rever?”, da Seção 14.5.3.

A segunda vantagem desta abordagem já foi esboçada nos parágrafos anteriores: uma avaliação intrínseca de *datasets* permite verificar o resultado de experimentações na classificação dos dados; ela ajuda e informa a tomada de decisões no processo de anotação. Se há duas ou mais maneiras legítimas e adequadas de analisar um mesmo fenômeno, qual escolher? Facilitar a generalização deve ser um critério relevante na escolha, no contexto do PLN. Por exemplo, na frase exemplo (1) do início deste capítulo (“TRISTE é uma palavra de 6 letras”), a palavra TRISTE foi analisada como um *substantivo* (e não como *adjetivo*), mas essa é uma escolha de anotação: decidir se as classes de palavras são propriedades das palavras (e, portanto, estáticas, e nesse caso TRISTE seria um *adjetivo* em qualquer contexto) ou se as classes são atribuídas em função do papel que exercem na frase (e, portanto, dinâmicas, e nesse caso TRISTE seria um *substantivo* no contexto da frase) deveria ser fruto de uma escolha que leva em conta as tarefas e demandas do PLN (Capítulo 6). Qual maneira de classificar leva à maior generalização sobre a anotação de PoS? E qual maneira de classificar leva ao melhor desempenho em uma outra tarefa – na análise sintática, por exemplo? Esta segunda pergunta se relaciona à *avaliação extrínseca*, tema da próxima seção.

14.6.3 Avaliação extrínseca

A avaliação extrínseca verifica se a informação codificada no *dataset* melhorou o desempenho em tarefas consideradas mais complexas. Por isso, está associada à **adequação**, e não à consistência. Assim como na avaliação intrínseca, esta é uma forma de avaliar aplicada tradicionalmente a modelos ou ferramentas, mas *datasets* também podem ser avaliados dessa maneira (novamente fazendo uma mudança de perspectiva e tomando o ponto de vista dos dados/*dataset*, e do que permitem generalizar). A avaliação é extrínseca porque ela não avalia aquilo que, diretamente, o modelo/ferramenta faz, ou aquilo que o *dataset*



codifica. Ela avalia indiretamente, verificando o quanto outras tarefas são beneficiadas quando aquilo que o *dataset* codifica é levado em conta. Ou: uma avaliação extrínseca verifica o quanto a informação codificada em um *dataset* é adequada para as tarefas mais complexas que o *dataset* pretende auxiliar. Por trás dessa ideia, o reconhecimento de que, no PLN, a codificação de certos tipos de informação (principalmente informação linguística especializada, como sintaxe, semântica etc.) é um meio, e não um fim, para resolver as tarefas de PLN. Ou seja, um *dataset* sintático, para tarefas de PLN, interessa não porque codifica informação sintática, mas porque, dispondo de informação sintática, é possível melhorar o desempenho em outras tarefas, como extração de informação³⁸ (Capítulo 17).

O estudo de Nooralahzadeh; Øvrelid (2018), por exemplo, estava interessado em verificar a contribuição de três tipos de gramáticas diferentes (ou, de três esquemas de anotação) na tarefa de extração de relações semânticas entre entidades, com o objetivo de saber qual o tipo de anotação gramatical (ou, de representação gramatical) mais adequado à tarefa. Ou seja, o estudo queria saber qual gramática ajudaria mais o modelo/ferramenta a chegar aos melhores resultados na extração de relações entre entidades. Para tanto, dispunham de um modelo que utilizava, como entrada, informação de dependências sintáticas³⁹ (Capítulo 6), e prepararam *datasets* anotados conforme as três abordagens gramaticais: as chamadas dependências sintáticas de Stanford, as dependências sintáticas do projeto Universal Dependencies e dependências sintáticas usadas nas tarefas CONLL⁴⁰ (ou seja, exatamente os mesmos textos anotados, a única diferença era o esquema de anotação sintática utilizado em cada um).

Voltando ao exemplo da palavra TRISTE, podemos imaginar dois *datasets* de PoS, que contém exatamente os mesmos documentos, mas em um deles as classes são atribuídas independentemente do contexto (e “triste” será sempre um adjetivo); e no outro as classes são atribuídas conforme o papel que exercem em cada frase. Não é difícil prever que a opção pelas classes estáticas (“triste” é sempre um adjetivo) será de mais fácil generalização, e levará a um desempenho melhor em uma avaliação intrínseca. Mas será que essa é a melhor opção se desejamos “aprender” informação sintática?

Em resumo, a avaliação intrínseca verifica a consistência interna da anotação (o que não necessariamente é sinônimo de qualidade, embora, em geral, seja), mas nada nos diz sobre a adequação do *dataset* para uma tarefa específica, e aqui a avaliação extrínseca pode ser útil. E nem sempre um *dataset* que obtém melhor desempenho na avaliação intrínseca leva aos melhores resultados quando em uma avaliação extrínseca.

No entanto, a avaliação extrínseca só pode ser realizada se a) dispomos do *dataset* para a “tarefa subsequente” – no estudo de Nooralahzadeh; Øvrelid (2018) foi utilizado o *dataset* de uma *avaliação conjunta* de extração de relações entre entidades; b) dispomos de diferentes versões do mesmo *corpus*, que variam apenas quanto ao esquema de anotação (as etiquetas e maneira de utilizá-las): diferentes tagsets de PoS; diferentes representações sintáticas, diferentes representações de papéis semânticos etc.

Para a língua portuguesa, dispomos – ainda que de maneira tímida – de alguns *corpora* com diferentes anotações de um mesmo tipo de atributo, como vemos na Tabela 14.4, que lista *corpora* que são fruto de projetos de pesquisa acadêmica, e estão disponíveis

³⁸Para os estudos linguísticos a anotação sintática pode ser um fim, quando o interesse está em estudar aspectos sintáticos da língua.

³⁹Importante notar que a tarefa não pressupõe a existência de uma anotação sintática anterior.

⁴⁰Considerando o foco deste capítulo, basta sabermos que são três maneiras diferentes de codificar a informação sintática, e para detalhes é possível consultar o artigo original.



para uso. MacMorpho⁴¹ e Bosque, os mais antigos, são os que contêm mais variações, e para o Bosque-UD⁴² também há uma versão em que os casos de omissão do sujeito (sujeito oculto, nos termos da Gramática Tradicional) foram explicitados (o material foi criado com o objetivo de verificar o quanto a explicitação de sujeitos nas frases é capaz de facilitar o processamento linguístico ou o quanto a omissão do sujeito – característica da língua portuguesa, mas não da língua inglesa – dificulta o processamento automático do português)⁴³. O mais recente deles é o PetroGold⁴⁴.

Tabela 14.4: Variações de atributos linguísticos em *corpora* do português

ATRIBUTO	VARIAÇÃO	CORPUS	REFERÊNCIA
PoS	Tagset LácioWeb clássico	MacMorpho	(Aluísio et al., 2003)
	Tagset LácioWeb modificado	MacMorpho	(Fonseca; Rosa, 2013)
	UD	MacMorpho	(Freitas et al., 2018)
		Porttinari	(Pardo et al., 2021)
		PetroGold	(Souza; Freitas, 2023)
Sintaxe	PALAVRAS/Floresta Sintá(c)tica	Bosque-UD	(Rademaker et al., 2017)
		Bosque	(Freitas; Rocha; Bick, 2008)
	UD	Porttinari	(Pardo et al., 2021)
		PetroGold	(Souza; Freitas, 2023)
		Bosque-UD	(Rademaker et al., 2017)
	UD	Bosque-UD com sujeitos ocultos explicitados	(Freitas; Souza, 2021)
	UD - Petrolês	PetroGold	(Souza; Freitas, 2023)
	PALAVRAS/Floresta Sintá(c)tica	Bosque	(Freitas; Rocha; Bick, 2008)

E qual desses é o melhor? Ou, qual dessas maneiras de representar o conhecimento linguístico é melhor? Aliás, melhor para que?

Em termos de avaliação intrínseca, precisamos garantir condições semelhantes de avaliação: mesmo algoritmo/ferramenta e mesmo conjunto de treino e teste. Todos os *corpora/datasets* citados estão públicos.

Quanto à adequação à tarefa, a avaliação extrínseca é uma forma interessante de avaliar a qualidade de um *dataset*, mas para que seja possível utilizá-la é fundamental a existência de um segundo *dataset*, que codifica a tarefa mais complexa. E voltamos ao que foi mencionado no início deste capítulo: a relevância de *datasets* para o avanço do PLN.

⁴¹O *corpus* MacMorpho e a documentação estão disponíveis em <http://nilc.icmc.usp.br/macmorpho/> e o MacMorpho-UD em <https://github.com/own-pt/macmorpho-UD>.

⁴²O Bosque anotado com a abordagem UD está disponível em https://universaldependencies.org/treebank-s/pt_bosque/index.html. Demais versões estão disponíveis em <https://www.linguateca.pt/Floresta/levantamento.html>

⁴³A preparação do *corpus* está descrita em (Freitas; Souza, 2021) e o *corpus* está disponível em <https://github.com/alvelvis/desocultando-sujeitos>

⁴⁴O *corpus* está disponível em https://github.com/UniversalDependencies/UD_Portuguese-PetroGold/tree/master



14.7 Em resumo...

- *Datasets* são importantes para o PLN porque permitem avaliar, treinar e pautar os rumos do PLN. (Seção 14.2)
- *Datasets* bons são consistentes (boa avaliação intrínseca), bem documentados, de tamanho adequado e com dados adequados. (Seção 14.3)
- A anotação de um *dataset* não precisa ser 100% perfeita para funcionar no aprendizado de máquina. (Seção 14.5.3)
- A preparação de um *dataset* se beneficia se a revisão/anotação é feita de maneira sistemática. (Seção 14.5)
- Avaliação intrínseca não significa, necessariamente, melhor adequação à tarefa, que é medida pela avaliação extrínseca. (Seção 14.6)

Um bom projeto cuidadoso de anotação, por sua vez, deverá levar em conta (Seção 14.4):

- Clareza quanto ao fenômeno que será anotado (que se reflete em um bom esquema de anotação);
- Escolha do *corpus* adequado (um *corpus* composto por relatórios de pesquisa é pouco adequado para a anotação de ironia, por exemplo);
- Conhecimento linguístico – para identificação e descrição do fenômeno anotado;
- Conhecimento do problema – para um melhor recorte das classes;
- Uma boa dose de inspiração ou criatividade – para chegar ao equilíbrio em termos de granularidade e generalização;
- Um outro tanto de experimentação – para validação e reformulação das classes, se for o caso;
- Verificação quanto à eficiência da anotação (por exemplo, o tempo levado e o nível de treinamento/conhecimento necessário por parte de quem vai anotar);
- Infraestrutura adequada – diretivas, documentação e ferramenta;
- Avaliação (verificação da concordância entre os anotadores).

A avaliação de um *dataset* pode levar em conta diferentes perspectivas (Seção 14.6):

- A concordância inter-anotadores avalia a consistência das anotações humanas;
- A avaliação intrínseca também avalia a consistência, mas privilegia a capacidade de generalizar a partir dos dados e a possibilidades de experimentações na classificação dos dados, tendo em vista o aprendizado de máquina;
- A avaliação extrínseca leva em conta a adequação de uma anotação para uma determinada tarefa.

Agradecimentos

Agradeço muitíssimo à Helena Caseli e à Graça Nunes pelas sugestões e provocações que deixaram o capítulo bem melhor (espero!).



Capítulo 15

Modelos de Linguagem

Aline Paes
Daniela Vianna
Jessica Rodrigues

15.1 Relembrando a Hipótese Semântica e Definindo Modelos

Da segunda década em diante do século XXI, testemunhamos um avanço significativo no desenvolvimento e popularização do aprendizado de representações numéricas para linguagem. Na época de escrita deste capítulo, **modelos de linguagem computacionais**, em particular os gerados por redes neurais são utilizados para representar textos escritos, fala, e até mesmo especificações que não são consideradas como parte da “linguagem natural”, como, por exemplo, formalizações matemáticas (Geva; Gupta; Berant, 2020; Gong et al., 2022; Li et al., 2023c; Piękos; Malinowski; Michalewski, 2021), código [Wang et al. (2021b); Li et al. (2023b)]¹², e até codificação de informações genéticas e moleculares (Brandes et al., 2022; Nijkamp et al., 2022). Os modelos de linguagem produzidos por redes neurais tanto geram como consomem textos mapeados para representações numéricas.

Mas por que seria importante representar informações essencialmente simbólicas em um formato numérico? A resposta mais simples e direta é que os computadores gostam de números. Seguindo ao porquê, a pergunta que segue é **“como representar tais informações simbólicas em um formato numérico, de forma a capturar sua semântica ?”** A segunda parte da pergunta – a tentativa de captura da semântica – é o ponto-chave, uma vez que simplesmente representar os componentes da língua em um formato numérico poderia guiar para uma simples representação por indexação. Ou seja, cada caractere ou palavra – ou cada componente léxico – poderia ser mapeado para um número distinto. Entretanto, tais números não teriam nenhuma conotação semântica. Assim, o arcabouço adotado de forma mais ampla para resolver este problema é mapear os componentes da língua para vetores em um espaço semântico, seguindo a hipótese distribucional. Como melhor detalhado no Capítulo 9, a hipótese distribucional tem como mote inferir significado a partir do contexto em que as palavras ocorrem. Apenas para ter uma ideia, considere, por exemplo o texto a seguir, em que a palavra “*bruble*” não pertence à língua portuguesa (até onde sabemos):

¹<https://github.com/features/copilot/>

²<https://openai.com/blog/openai-codex>



Exemplo 15.1.

Sempre deixo as notificações do meu *bruble* desligadas. Mas no outro dia, estava escrevendo uma mensagem no meu *bruble* e as notificações de vários aplicativos apareceram na tela.

Pelo contexto, podemos inferir que a palavra “*bruble*” seria “celular” e é justamente nesta motivação que a semântica distribucional se coloca. Indo além, segundo Firth, o significado de uma palavra pode ser depreendido pelas palavras que coocorrem com ela, ideia difundida pelo slogan “*you shall know a word by the company it keeps*” (Firth, 1957b), que, no contexto do significado das palavras, podemos adaptar para algo como “Diga-me com quem andas, e te direi quem és”.

Embora contexto possa contemplar diversas definições, para a geração de modelos semânticos distribucionais, contexto é definido pela coocorrência de itens. A coocorrência pode ser traduzida para: itens que aparecem próximos uns dos outros ou ainda itens que aparecem em contextos similares.

Os modelos de linguagem mais recentes apresentam uma significativa sinergia com a hipótese distribucional. Por um lado, eles se fundamentam na hipótese distribucional, uma vez que assumem que o contexto pode ser usado para a predição de uma ou mais palavras; por outro lado, modelos de linguagem podem gerar as representações numéricas que sumarizem os contextos em que as palavras ocorrem, permitindo a investigação da hipótese distribucional em termos de similaridade. Nesta sinergia, os modelos de linguagem mais recentes que geram representações vetoriais de forma dinâmica se sobrepõem às limitações dos métodos distribucionais estáticos mais clássicos, uma vez que os vetores de um mesmo item podem ser diferentes dependendo do contexto em que ele aparece.

Mas antes de entrarmos em detalhes sobre os modelos de linguagem atuais, temos uma pergunta ainda mais básica a ser respondida: **O que é um modelo?** Um modelo é uma simplificação de um fenômeno complexo, no nosso caso, uma simplificação da língua que possa ser representada por ferramentas computacionais. Embora um modelo tente capturar as nuances do fenômeno real, justamente por ser uma simplificação, ele não tem a intenção de substituir o fenômeno real, mas representá-lo para auxiliar o nosso entendimento ou resolver algumas tarefas. Porém, idealmente, o modelo deve manter alguma consistência com o fenômeno real. Por isso, um modelo de linguagem deveria respeitar os princípios léxicos, sintáticos e semânticos, componentes essenciais de qualquer linguagem, natural ou não.

Também, um modelo deveria considerar o mesmo funcionamento do fenômeno real. Mas como a questão de como nosso cérebro processa e produz linguagem continua em aberto (Berwick; Chomsky, 2017), nos modelos de linguagem computacionais, assume-se que um texto escrito ou falado é oriundo de um processo de completação. Em suas primeiras abordagens, definia-se que um modelo de linguagem computacional deveria ser capaz de completar a próxima palavra em uma sequência, considerando todas as palavras que vieram antes. Por exemplo, considerando a sentença “Vamos completar o texto com a próxima ...”, um modelo poderia completá-la com “palavra”. Atualmente, alguns modelos também podem considerar completar partes de uma sequência considerando palavras (ou *tokens*) que vieram antes ou depois do elemento que se deseja completar, seguindo uma abordagem inspirada no teste Cloze (Santos et al., 2002; Taylor, 1953). Por exemplo, seguindo o caso anterior, poderíamos ter “Vamos ...o ...com a próxima palavra”, onde ... poderiam ser preenchidos com palavras. Um modelo de linguagem computacional não precisa estar



restrito a completar uma única palavra, mas sim uma sequência delas, independente de serem as próximas palavras, ou palavras em outras posições da sequência.

Nas próximas seções, vamos entender melhor como essas tarefas são abordadas em termos computacionais.

15.2 Modelos de Linguagem Probabilísticos

Em termos computacionais, a modelagem probabilística de linguagem é a tarefa que atribui uma probabilidade a uma sequência de palavras. Ou seja, o modelo assume que existe uma probabilidade associada à existência de uma sequência de palavras $p_{1:i}$, representada por $P(p_{1:i})$, onde i representa a posição da última palavra na sequência considerada. Usando a regra da cadeia da probabilidade, a fórmula pode ser definida como:

$$P(p_{1:i}) = P(p_1)P(p_2|p_1)P(p_3|p_{1:2})P(p_4|p_{1:3}) \dots P(p_i|p_{1:i-1}) \quad (15.1)$$

Apenas uma observação: multiplicações de valores menores que um podem fazer com que o resultado seja zero, considerando a limitação dos computadores em manipularem números em ponto flutuante. Chamamos esse problema de *underflow*. Para aliviá-lo, podemos usar log e somar os termos, ao invés de multiplicar:

$$\begin{aligned} \log P(p_{1:i}) = & \\ & \log P(p_1) + \log P(p_2|p_1) + \log P(p_3|p_{1:2}) + \\ & + \log P(p_4|p_{1:3}) + \dots + \log P(p_i|p_{1:i-1}) \end{aligned} \quad (15.2)$$

Observe que na fórmula, temos uma sequência de tarefas de predição de palavra, onde o objetivo é prever uma palavra condicionando-a às palavras precedentes. Assim, pensando na completção discutida anteriormente, assumimos que a tarefa de completar uma sequência de palavras com uma próxima palavra é definida por uma distribuição de probabilidade condicional³ das palavras que poderiam completar a sequência, dadas as palavras que vieram antes na sequência, ou seja:

$$P(p_i|p_1, \dots, p_{i-1}) \quad (15.3)$$

onde p_i é uma palavra do vocabulário, i é a sua posição na sequência, p_1 é a primeira palavra da sequência e p_{i-1} é a última palavra da sequência. Modelos de língua que seguem esta formulação são chamados de *modelos autorregressivos ou causais* e são frequentemente empregados para tarefas que envolvem geração de texto. A ideia é simples: (1) use o modelo probabilístico para escolher o próximo *token*; (2) adicione o *token* gerado na sequência de entrada; (3) repita. Mas no passo (1), quando falamos que um *token* é gerado pelo modelo, o que acontece, na verdade, é que um *token* é escolhido de acordo com uma distribuição de probabilidade aprendida pelo modelo. Tal distribuição de probabilidade é definida para um vocabulário, o conjunto de *tokens* que o modelo conhece.

Voltando ao nosso exemplo anterior, ele seria modelado pela seguinte distribuição de probabilidade condicional

$$P(p_i|\text{Vamos, completar, o, texto, com, a, próxima})$$

onde

³Probabilidade condicional é a probabilidade de um evento ocorrer, dado que outro evento já ocorreu. A probabilidade condicional é representada por $P(A|B)$, o que significa a probabilidade do evento A ocorrer, dado que o evento B já ocorreu.



$$P(\text{palavra}|\text{Vamos, completar, o, texto, com, a, próxima})$$

poderia ter um valor de, digamos, 0,88. No caso de uma palavra pouco provável, digamos, CHUTEIRA, esse valor poderia ser bem pequenino, digamos, 0,00001 (por enquanto, assuma que esses valores vieram do além).

Entretanto, não é computacionalmente eficiente considerar toda a sequência anterior para prever a próxima palavra na sequência. Embora modelos probabilísticos sejam apelativos, principalmente pela sua simplicidade, eles sofrem da “maldição da dimensionalidade”: modelar a distribuição conjunta de, digamos, sequências de 10 palavras, com um vocabulário de 100.000 palavras, traz a enorme quantidade de $100.000^{10} - 1$ parâmetros.

Então, podemos simplificar ainda mais o modelo, assumindo a suposição de Markov (Markov, 1954), que dita, informalmente, que apenas o passado mais recente é importante para o futuro. Assim, considerando a suposição de Markov, assume-se que prever a próxima palavra é independente das outras palavras na sequência, dada a última palavra vista. Ou seja,

$$P(p_i|p_{1..i-1}) \approx P(p_i|p_{i-1})$$

No nosso exemplo, consideraríamos apenas a palavra PRÓXIMA para prever a palavra PALAVRA (desculpem a redundância), ou seja, $P(\text{palavra}|\text{próxima})$. Este modelo é conhecido como *bigrama*, por considerar apenas um par de palavras na probabilidade condicional. Generalizando, um unigrama consiste em considerar a probabilidade a priori de apenas uma palavra, $P(p_i)$, um bigrama consiste em considerar duas palavras $P(p_i|p_{i-1})$, um trigrama consiste em considerar as duas palavras anteriores $P(p_i|p_{i-1}, p_{i-2})$, e assim por diante. Generalizando ainda mais, um modelo n-grama é representado por $P(p_i|p_1, \dots, p_{i-n})$.

Perceba que existe uma troca na decisão de que valor de n considerar. Enquanto valores menores de n tornam o modelo probabilístico mais eficiente de ser computado, por outro lado, eles perdem precisão. Considerando nosso exemplo, é mais fácil de prever que p_i seria PALAVRA se pensarmos na sequência anterior completa. Olhando apenas para PRÓXIMA, a gama de palavras que fariam sentido vir depois é muito maior. Entretanto, conforme veremos a seguir, essas probabilidades precisam vir de algum lugar (não do além), e esse lugar são textos existentes (ou melhor dizendo, o *corpora*). Quanto maior for a sequência considerada, mais rara será a sua aparição no *corpora*, o que pode prejudicar o cálculo do valor de probabilidade para uma determinada sequência.

15.2.1 Estimando as probabilidades a partir de *corpora*

Para estimar as probabilidades das sequências de palavras, usaremos um conjunto de textos. Quanto maior e mais diverso o conjunto, maior é a chance dele conter muitas variações de sequências. Mas também, mais demorado será o seu processamento. Considerando a probabilidade frequentista, para calcular a probabilidade condicional $P(p_i|p_{1:i})$, podemos simplesmente usar contagem. Vamos começar de um modelo bigrama para depois generalizarmos. Neste caso,

$$P(p_i|p_{i-1}) = \frac{c(p_{i-1}, p_i)}{c(p_{i-1})}$$

onde $c(p_{i-1}, p_i)$ representa quantas vezes a sequência formada pelas duas palavras p_i e p_{i-1} apareceram nos textos, mais precisamente, quantas vezes p_{i-1} aparece antes de p_i , e $c(p_{i-1})$ representa quantas vezes a palavra p_{i-1} aparece no texto.

De forma similar, para estimar as probabilidades de um modelo trigrama, temos que



$$P(p_i|p_{i-1}, p_{i-2}) = \frac{c(p_{i-2}, p_{i-1}, p_i)}{c(p_{i-2}, p_{i-1})}$$

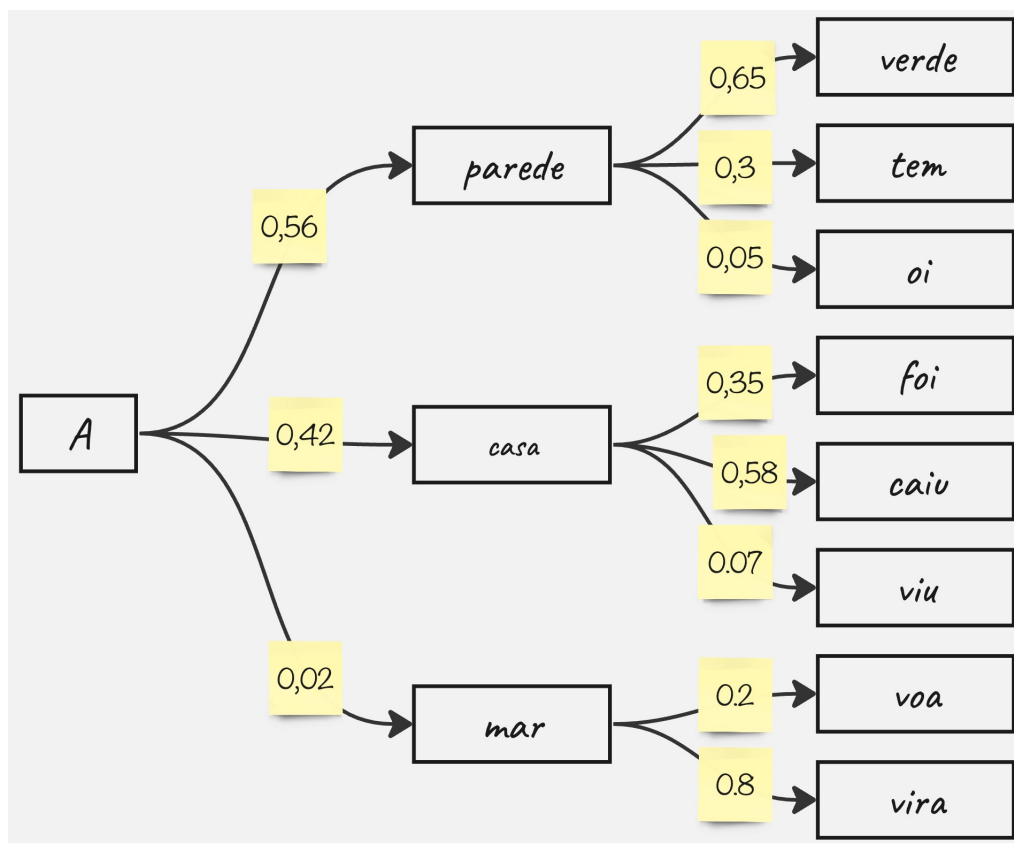
Ou seja, $c(\dots)$ representa quantas vezes um dado n-grama ocorreu no texto.

O modelo n-grama também serve para calcular as probabilidades mesmo de sequências de palavras que não apareceram no conjunto de treinamento. Ou seja, a probabilidade de uma sequência não vista de palavras será obtida a partir da concatenação de gramas menores que formam a sequência. Entretanto, perde-se informação ao não considerar contextos maiores. Outro problema é desconsiderar a similaridade entre palavras, que poderia servir para devolver probabilidades de palavras ou sequências não vistas durante o treinamento. Modelos de linguagem neurais tentam abordar esses problemas com métodos mais sofisticados de aprendizado de máquina do que simplesmente contagem.

15.2.2 Usando o modelo probabilístico

A escolha do próximo *token* conforme a probabilidade pode seguir diferentes algoritmos. O primeiro que pode nos vir à mente é um processo guloso, ou seja, a cada iteração, escolhemos o *token* com a maior probabilidade. Assim, para escolher a palavra p_t na iteração t , usamos $p_t = \arg \max_p P(p|p_{1:t-1})$. No exemplo da Figura 15.1, supondo que o *token* “A” já foi emitido, a busca gulosa escolherá a palavra “casa” e depois “caiu”.

Figura 15.1: Exemplo de geração de sentença com a busca gulosa



Entretanto, um processo guloso pode trazer um sério problema, que é a falta de diversidade. Também, pode ser que uma escolha conjunta seja melhor do que uma



escolha individual. Mas a escolha individual deixa a busca um pouco míope em relação ao que ainda está por vir. Assim, é bem comum utilizar outros mecanismos. Um desses outros mecanismos é a busca em feixe, que armazena possíveis escolhas, desde que esse armazenamento não ultrapasse o limite máximo de feixes. No exemplo da Figura 15.1 e assumindo um feixe de tamanho dois, a busca, além de guardar “(A, casa)” também guardaria “(A, parede)”. Na próxima iteração, teríamos as seguintes possibilidades: (A, PAREDE, VERDE) e (A, CASA, CAIU), com a primeira opção tendo probabilidade $0.4 \times 0.9 = 0.36$ e a segunda possibilidade com probabilidade $0.5 \times 0.4 = 0.2$.

Ainda assim, pode ser difícil de garantir muita diversidade, tanto nas palavras geradas na mesma execução como na geração em diferentes execuções. Uma escolha aleatória do tipo $p_t \sim P(p_t | p_{1:t-1})$ poderia trazer a desejada diversidade. Porém, os resultados também podem ficar bem incoerentes (Holtzman et al., 2020). Um procedimento que ajuda um pouco é usar um valor de *temperatura* parametrizável. Valores mais altos de temperatura produzem saídas mais aleatórias, enquanto valores menores fazem com que as saídas sejam mais similares, ou seja, mais determinísticas. Outras abordagens também existem, incluindo amostragem das top-k palavras, com uma redistribuição da massa de probabilidade apenas entre essas k (Fan; Lewis; Dauphin, 2018), ou amostragem baseada em um limiar p , que escolhe o menor conjunto de palavras cuja probabilidade acumulada exceda p (Holtzman et al., 2020).

15.3 Modelos de Linguagem Neurais

O uso de n-grams discutido na seção anterior é uma forma de generalizar e tornar eficiente o cálculo da probabilidade de uma sequência de palavras. Outra forma de atender às necessidades de generalização – ou seja, calcular uma probabilidade para uma sequência de palavras ao usar um modelo, mesmo que a sequência não tenha aparecido durante o treinamento do modelo⁴ – é considerar que a probabilidade associada a um modelo de linguagem é uma função e “aprender” tal função. Redes Neurais (Goodfellow; Bengio; Courville, 2016) são métodos de aprendizado de máquina conhecidos por sua propriedade de aproximação universal de funções. Ou seja, dada uma rede neural com ao menos uma camada escondida e um número suficiente de neurônios, ela é um aproximador universal de funções contínuas no espaço de interesse (Hornik; Stinchcombe; White, 1989). Caso você queira entender melhor como funciona uma rede neural, os capítulos 5 e 6 de (Goodfellow; Bengio; Courville, 2016) são uma boa introdução (dentre muitas outras referências).

15.3.1 Um Contexto Histórico

A ideia de usar redes neurais para aprender funções que representem modelos de linguagem pode parecer recente, mas não é. Na verdade, as primeiras tentativas datam do início da década de 90, com o trabalho de Miikkulainen; Dyer (1991). Ainda na década de 90, também foram propostas técnicas baseadas em redes neurais para prever a probabilidade do próximo caractere (Schmidhuber; Heil, 1996). Mas os modelos que mais se assemelham aos modelos de linguagem neurais da era de *Deep Learning* (redes neurais profundas) foram propostos no início dos anos 2000, de forma independente, com os trabalhos *Can artificial neural network learn language models?* (Xu; Rudnicky, 2000) e *A neural probabilistic language model* (Bengio et al., 2003).

⁴Lembre que, em um modelo de linguagem probabilístico, treinar o modelo é encontrar as probabilidades por contagem.



Enquanto o primeiro caso usava uma forma limitada de rede neural, sem camadas escondidas e limitando a predição a apenas uma palavra, ou seja, modelando apenas unigramas e bigramas, o segundo caso já apresentava várias características e fundamentos encontrados nos modelos de linguagem neurais modernos. A proposta do primeiro trabalho era aprender (i) funções de representações distribuídas para cada palavra $P(w)$, que consideraria a vizinhança das palavras nos textos de treinamento, bem na linha do que vimos no Capítulo 10. Mas além da probabilidade das palavras, o modelo também aprenderia de forma simultânea (ii) a função de probabilidade associada a uma sequência de palavras, a partir das probabilidades das palavras. Assim, mesmo que no momento de usar o modelo aparecesse uma sequência de palavras não vista durante o treinamento, ainda seria possível obter a probabilidade da sequência, a partir das palavras e sequências similares vistas durante o treinamento.

No modelo proposto, a rede neural é utilizada para prever a próxima palavra, dadas as palavras anteriores. Para tanto, seus pesos são treinados para aprender a função de probabilidade do modelo de linguagem a partir da maximização da log-verossimilhança dos dados de treinamento. De forma mais específica, um exemplo de treinamento é uma sequência de palavras de tamanho T , $p_1 \dots p_T$, com cada palavra $p_i \in V$, onde V é um vocabulário finito de palavras de uma língua. O objetivo da rede neural é representar uma função $f(p_i, \dots, p_{i-n+1}) = \hat{P}(p_i | p_1^{i-1})$, onde n é o tamanho de uma janela de contexto. Assim como no modelo puramente probabilístico, pode-se obter um modelo da probabilidade conjunta de sequências de palavras a partir do produto destas probabilidades condicionais. A função $f(p_i, \dots, p_{i-n+1})$ é decomposta em duas partes: (i) um mapeamento C de qualquer palavra $p_i \in V$ para um vetor $C(p_i) \in \mathbb{R}$ – um **embedding**⁵ da palavra; e (ii) uma função g que mapeia uma sequência de vetores, capturados a partir de C , ou seja, $C(p_{i-n+1}), \dots, C(p_{i-1})$ para uma distribuição de probabilidade condicional da próxima palavra p_i . A saída da função g é um vetor cujo k -ésimo elemento estima a probabilidade $\hat{P}(p_i = k | p_1^{i-1})$. Ou seja, a função f é uma composição das funções g e C : $f(k, p_{i-1}, \dots, p_{i-n+1}) = g(k, C(p_{i-1}), \dots, C(p_{i-n+1}))$. A função g será parametrizada pelos pesos ω aprendidos pela rede neural. Os parâmetros do modelo são $\Theta = (C; \omega)$, descobertos a partir da minimização da função de custo $L = \frac{1}{T} \sum_i \log f(p_i, p_{i-1}, \dots, p_{i-n+1}; \Theta) + R(\Theta)$.

O que aconteceu com esse modelo para ele não ficar tão famoso como os modelos de linguagem neurais atuais? O treinamento da tal rede neural era extremamente ineficiente e impraticável na época, um problema que começou a ser resolvido alguns anos depois com o advento das Unidades de Processamento Gráfico (GPUs). As GPUs ajudaram a impulsionar a era do *Deep Learning* (Goodfellow; Bengio; Courville, 2016) ao focarem na realização de cálculos matriciais (tudo que uma rede neural quer) em tempos muito menores do que se o mesmo cálculo fosse feito em uma Unidade Central de Processamento (CPU).

15.3.2 Modelos de Linguagem Neurais Modernos

Considerando as limitações que discutimos no Capítulo 10 ao se definir embeddings de forma estática, vários métodos desenvolvidos a partir de 2017 passaram a construir embeddings de forma **dinâmica**, considerando o contexto da sentença no momento do uso, e por isso comumente denominados de **embeddings contextualizados**. Isso quer dizer que as unidades de representação (*tokens*) podem ter embeddings distintos, definidos no momento

⁵Uma representação vetorial densa de baixa dimensionalidade, aprendida a partir de textos. Veja mais no Capítulo 10.



em que eles são aplicados. Considere, por exemplo, as sentenças do Exemplo 15.2:

Exemplo 15.2.

1. Sentei no banco da praça.
2. O banco estava sem notas de R\$ 200,00.
3. O banco estava super cheio hoje!

A palavra “banco” na sentença 1) evoca mais o sentido de assento, embora também seja possível pensar em outros significados. A sentença 2) evoca mais o sentido de estabelecimento comercial financeiro. A sentença 3), apesar de evocar mais o segundo sentido, também poderia estar falando de um assento cheio de pessoas. Sendo assim, uma lista estática de palavras e seus embeddings falharia em retornar representações distintas para estas diferentes interpretações.

Considere ainda o exemplo Exemplo 15.3:

Exemplo 15.3.

Em frente à agência do banco de Pineapólis, existe um banco amarelo que data da década de 50, onde várias pessoas famosas já pararam para descansar e algumas vezes entoar uma melodia.

Observe que a palavra “banco” aparece duas vezes na mesma sentença, com dois significados distintos. Ainda assim, um método de geração de embeddings contextualizados deve ter a habilidade de devolver representações vetoriais distintas para os dois *tokens*.

Para tanto, a unidade de representação é associada a um *embedding* a partir do **contexto** corrente em que ela aparece, onde contexto, em geral, é definido nos modelos de linguagem por uma sequência de *tokens* que aparecem antes e depois do *token* em questão. No exemplo anterior, teríamos *embedding* distintos para os diversos “bancos” mencionados. Na verdade, o *embedding* poderia diferir até mesmo para *tokens* do tipo “banco” com a mesma semântica, devido aos diferentes outros *tokens* que aparecem em seus contextos. Entretanto, ainda se espera que quando mais próxima for a semântica do *token*, mais próximos fiquem os vetores no espaço vetorial.

Uma outra vantagem associada aos embeddings contextualizados é a possibilidade de representar informação que vai além do idioma. Esses embeddings são chamados de *cross-lingual* (Agirre, 2020). Ou seja, é possível que os embeddings associados às palavras “mãe” e “mother” estejam próximos no espaço vetorial, mesmo que ambas as palavras estejam em idiomas distintos.

Para que os embeddings de um *token* sejam gerados conforme o contexto dinâmico em que aparecem, a forma de recuperação e de armazenamento precisam ser diferentes daquelas que discutimos com os embeddings estáticos. Lá, poderíamos armazená-los em uma tabela e recuperá-los pela indexação da palavra. Já os embeddings contextualizados são recuperados a partir de uma função que tem como entrada a sequência em que a unidade de representação de interesse está inserida. Por exemplo, para devolver como saída o *embedding* da palavra “banco” a partir da sentença “Sentei no banco da praça.”, teremos $emb_{\text{banco}} = f(e_{\text{sentei}}, e_{\text{no}}, e_{\text{banco}}, e_{\text{da}}, e_{\text{praça}})$, onde f é a função de geração do *embedding* e emb_{palavra} é a sua saída. Cada palavra que será entrada da função precisa primeiro ser transformada para uma representação vetorial (e_{palavra}). Outra observação importante é que, no nosso exemplo, a própria palavra é entrada da função. Nem sempre isso acontece, para evitar a influência da própria palavra na representação gerada.



A função f pode assumir diferentes formas. Uma possibilidade seria simplesmente recuperar os embeddings estáticos de cada palavra no contexto e executar alguma forma de agregação, conforme discutido no Capítulo 10.

Porém, temos alguns problemas em simplesmente usar uma função de agregação. Um deles é que não teríamos diferença entre os embeddings de um *token* e os embeddings da sentença em que ele está inserido. Com essa forma simplificada de simplesmente agregar os embeddings estáticos, tanto a palavra “banco” como a palavra “praça” na sentença acima, teriam a mesma representação final. Mesmo que os *tokens* de interesse fossem removidos da entrada, outros problemas surgiriam, incluindo a falta de consideração com a ordem das palavras e palavras semanticamente distintas, porém lexicalmente idênticas, tendo a mesma representação em sentenças distintas.

Assim, torna-se necessário considerar outras funções mais elaboradas. Mas na dificuldade de se definir que função seria essa, por que não descobri-la **automaticamente**? Essa é a ideia da geração de embeddings contextualizados a partir de redes neurais.

A aplicação de embeddings contextualizados para abordar tarefas de PLN inclui dois aspectos: a geração dos embeddings e a sua utilização em tarefas finais. Dois principais métodos para a geração de embeddings contextualizados se destacaram entre 2017 e 2023: as redes neurais recorrentes – incluindo CoVe (McCann et al., 2017) (*Context Vectors*) e ELMo⁶ (*Embeddings from Language Models*) (Peters et al., 2018)) – e os **Transformers** (Vaswani et al., 2017) – incluindo BERT⁷ (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2019) e GPT⁸ (*Generative Pre-trained Transformer*) (Brown et al., 2020). Vamos aqui seguir a ordem cronológica, primeiro falando dos modelos baseados em redes recorrentes, que surgiram primeiro, para depois falarmos dos modelos baseados em Transformers.

15.3.2.1 Modelos de Linguagem com Redes Recorrentes

Representar sequências de tamanhos variáveis é uma habilidade essencial para modelar a linguagem: sentenças não são obrigadas a conterem uma quantidade fixa de palavras; e, a ordem em que as palavras aparecem pode fazer toda a diferença para a sua sintaxe e sua semântica. Redes Neurais recorrentes abordam ambos os aspectos: aceitam entradas de tamanho variável e consideram a ordem dos componentes da entrada para induzir um vetor que represente uma sequência. Veja um esquema de uma rede neural recorrente na Figura 15.2.

De forma abstrata, um modelo de linguagem baseado em redes recorrentes opera gerando uma palavra a partir de uma sequência de palavras anteriores, seguindo os passos abaixo:

1. Calcula-se o vetor de *embedding* $h_t^0 = X_t \mathbf{E}$, onde \mathbf{E} é uma matriz de dimensão $|V \times N|$, X_t é um vetor *one-hot*⁹ do tamanho do vocabulário, ou seja, $|1 \times V|$ representando uma palavra, e t representa a t -ésima palavra da sequência sendo gerada
2. Calcula-se a saída da camada escondida $h_t^1 = fn \left(\mathbf{W}_h \begin{bmatrix} h_t^0 \\ h_{t-1}^1 \end{bmatrix} \right)$, onde $\begin{bmatrix} h_t^0 \\ h_{t-1}^1 \end{bmatrix}$ representa a concatenação dos vetores associados à saída da camada escondida fisicamente anterior (h_t^0) e da camada escondida do instante anterior (logicamente anterior) (h_{t-1}^1), \mathbf{W}_h é

⁶Inspirado pelo nome de um dos personagens de Vila Sésamo.

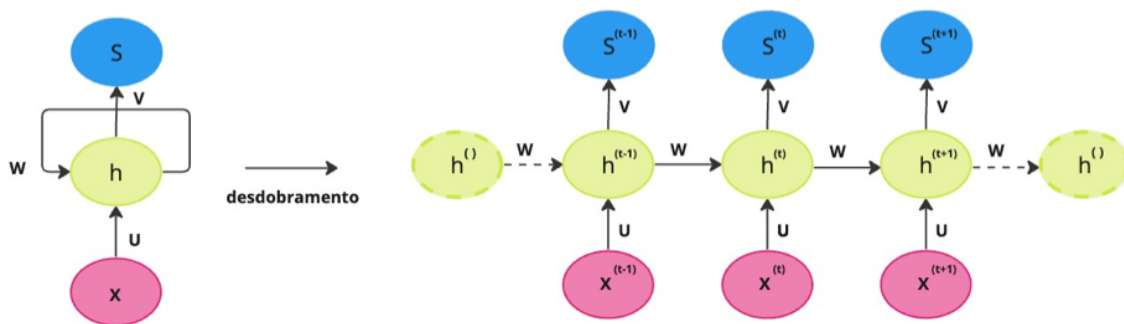
⁷Também inspirado pelo nome de um dos personagens de Vila Sésamo.

⁸Até onde sabemos, nenhum personagem de Vila Sésamo tem esse nome.

⁹Um vetor *one-hot* tem uma posição apenas igual a 1 e as demais são zero. A posição 1 se refere a exatamente um elemento.



Figura 15.2: Esquema clássico de uma rede recorrente, com X representando a entrada, h representando o estado oculto e S representando a saída da rede. À esquerda da figura, temos o esquema físico da rede, demonstrando como ela é implementada. Observe que a entrada se conecta ao estado oculto por meio de uma matriz de pesos U , o estado recorrente se conecta a si mesmo por meio de uma matriz de pesos W e o estado oculto também se conecta à saída por meio da matriz de pesos V . À direita da figura, temos a versão da rede conforme a sua operação em tempo de execução: X^t representa uma unidade da entrada (por exemplo, um *token*) na posição t da sequência de entrada. As matrizes de peso são fixas para todas as posições.



a matriz de pesos da camada escondida, e fn é uma função de ativação, por exemplo, a tangente hiperbólica. Este passo pode se repetir diversas vezes, dependendo de quantas camadas escondidas a rede tiver. O sobrescrito indica a camada da rede.

3. Calcula-se a saída $y_t = \mathbf{W}_o h_t^1$, onde \mathbf{W}_o representa a matriz de pesos da camada de saída.
4. Calcula-se a distribuição de probabilidade $p_t = \text{softmax} y_t$.
5. Resgata-se a palavra com o maior valor de probabilidade na tabela *one-hot*.
6. O processo continua até encontrar um *token* de fim de sequência, ou até alcançar uma saída máxima.

Pensando em uma geração *token a token*, é necessário ter algum *token* de início, que represente a camada anterior, para o primeiro *token*. Ele servirá para indicar a camada logicamente anterior usada, (h_{t-1}^1). As matrizes de pesos são os componentes aprendidos na rede. Para o aprendizado, pode-se considerar um conjunto de textos e fazer a tarefa de predição ser devolver a palavra correta na t -ésima posição, para t de 1 até um valor qualquer.

15.3.2.1.1 Embeddings from Language Models – ELMo

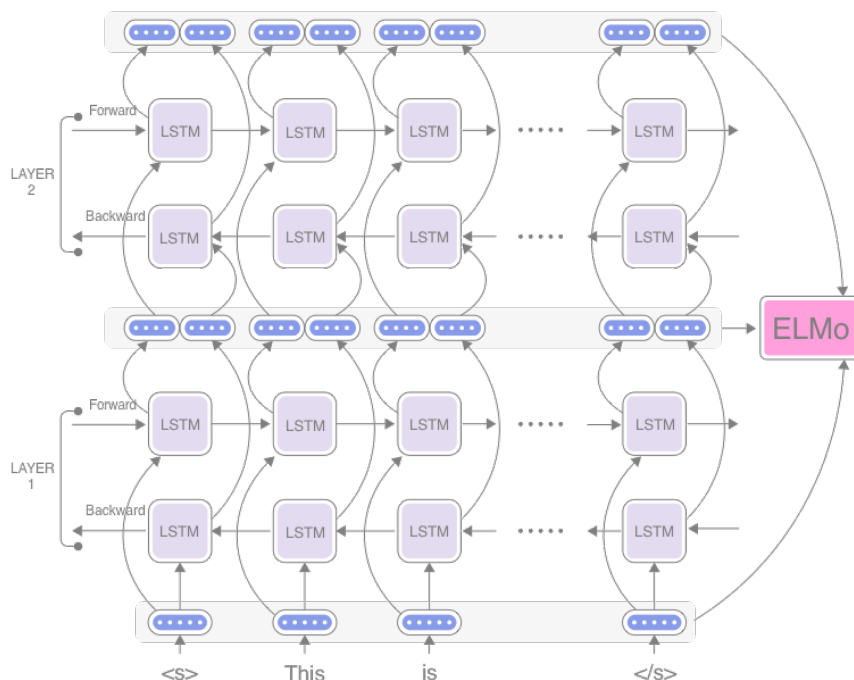
O ELMo é um modelo de linguagem que opera em uma rede neural recorrente ¹⁰ com várias camadas. Assim, cada camada pode ser usada para gerar uma representação contextualizada de um *token*. As camadas de redes recorrentes do ELMo olham para a

¹⁰O ELMo, assim como vários outros métodos de PLN, utiliza um tipo especial de rede recorrente chamada de *Long Short-Term Memory* (Hochreiter; Schmidhuber, 1997). LSTMs conseguem manipular entradas de tamanhos maiores do que redes recorrentes tradicionais, pois o que será aprendido, esquecido ou devolvido como saída é controlado durante o aprendizado por um tipo especial de rede neural chamada de *gate*. As redes recorrentes tradicionais diferentes das LSTMs frequentemente falham em manipular sequências longas, pois uma informação de longa distância pode se perder ao longo das várias recorrências, um problema conhecido como “gradiente que some”.



frente e para trás na sentença (são chamadas de redes recorrentes bidirecionais), dando origem a duas representações, uma para cada direção. Então, cada *token* pode ter um conjunto de representações, mais precisamente $2L + 1$ representações, onde L é a quantidade de camadas da rede. A multiplicação por 2 é devido às duas direções. E de onde vem o 1? É que o ELMo também inclui nesse conjunto de representações a entrada não contextualizada do *token*. Lembra que falamos antes que de todo modo os geradores de embeddings contextualizados devem iniciar por alguma representação vetorial? Mesmo os métodos que geram representações contextualizadas, precisam ter de onde começar. Então, o ELMo também inclui a representação “descontextualizada” e_{t_i} , que é a representação de entrada do *token*, como uma possível representação. Ou seja, cada camada $j \in \{1, 2, \dots, L\}$ da rede produz as representações $\text{emb}_{t_i, j} = (\vec{h}_{t_i, j}, \vec{h}_{t_i, j})$ para o *token* t_i . Assim, o *token* terá um conjunto $\text{emb}_{t_i} = (e_{t_i}, \{\vec{h}_{t_i, 1}, \vec{h}_{t_i, 1}, \dots, \vec{h}_{t_i, L}, \vec{h}_{t_i, L}\})$ de possíveis representações. A Figura 15.3 exibe um esquema da arquitetura do ELMo. O aprendizado de um modelo de linguagem baseado em redes recorrentes segue o algoritmo *backpropagation through time* (Werbos, 1990) a partir de um conjunto enorme de textos.

Figura 15.3: Esquema da arquitetura do ELMo.



Fonte:

<http://www.realworldnlpbook.com/blog/improving-sentiment-analyzer-using-elm.html>

15.3.2.1.2 Utilização dos embeddings do ELMo

Mas como podemos usar esses embeddings para resolver uma tarefa? Por exemplo, suponha que a tarefa seja classificar uma publicação em uma rede social como sendo um comentário tóxico ou não¹¹. Essa é uma tarefa de **classificação**. Uma forma de resolvê-la

¹¹Um texto é considerado um comentário tóxico quando ele contém expressões ou intenções indesejáveis de desagrado, violência psíquica, física e emocional, crueldade, negativismo, manipulação, egoísmo etc. Veja mais sobre esse assunto no Capítulo 23.



é treinar um classificador, que receberá um texto e informará se esse texto possui conteúdo tóxico ou não. Tanto para treinar o classificador como para usá-lo, a entrada textual precisa ser transformada para uma informação numérica, que é a língua que o computador entende. No caso que estamos falando aqui, a informação numérica será obtida justamente a partir dos embeddings. Com o ELMo, podemos obter esses embeddings de duas formas: (i) juntando todos os elementos acima em um único vetor, por exemplo, os somando, ou seguindo uma operação mais simples, (ii), por exemplo, selecionando somente aqueles que estão na última camada, ou seja, considerando somente $(\vec{h}_{t_i,L}, \vec{h}_{t_i,L})$. Mais precisamente, o *embedding* que o ELMo gera para um *token* é definido por

$$emb'_{t_i} = \gamma^{\text{tarefa}} \sum_{j=0}^L s_j^{\text{tarefa}} [\vec{h}_{t_i,j}; \vec{h}_{t_i,j}]$$

onde variamos j de 0 até L , para incluir o *embedding* da primeira camada (o descontextualizado), $[_;_]$ indica uma operação de concatenação, γ^{tarefa} é um hiperparâmetro relacionado à tarefa, e s_j^{tarefa} são os pesos da camada, normalizados por uma função Softmax¹².

Observe que você pode experimentar outras variações. Por exemplo, podemos usar as camadas mais próximas da saída apenas, fazendo o somatório começar em $j = k$, onde k é uma posição intermediária na rede. Também é possível concatenar o *embedding* descontextualizado com a saída da última camada.

15.3.2.1.3 Embeddings de sentenças

Até agora falamos de embeddings de *tokens*. Mas a maioria das tarefas considera entradas que são frases, ou um texto, ou seja, uma sequência de *tokens*. Na verdade, embora seja possível recuperar os embeddings de qualquer tipo de unidade de representação a partir do ELMo, incluindo caracteres, palavras, frases, textos, a saída default das implementações mais comuns¹³ são os embeddings de uma sentença. Eles são obtidos a partir de uma operação de amostragem por média (*mean pooling*) dos embeddings de *tokens* da última camada da rede, conforme discutimos antes. Perceba que isso é bem diferente do que a função simples que mencionamos antes, uma vez que as representações vetoriais passam por várias transformações matemáticas dentro da rede neural.

15.3.2.1.4 ELMo para português

Como usual, o modelo ELMo foi originalmente treinado e avaliado na língua inglesa. Mas existem versões deste modelo treinadas para as variantes brasileira e europeia do português (Rodrigues et al., 2020), disponibilizadas na biblioteca oficial do ELMo, a Allen NLP¹⁴. O modelo foi treinado em tarefas de similaridade sintática e comparado com sucesso a representações estáticas também treinadas para o português.

15.3.2.2 Modelos de Linguagem baseados em Transformers

Embora as redes recorrentes possam resolver tarefas sequenciais e não demandem entradas de tamanho fixo, o que parece perfeito para tarefas de PLN, elas têm um grande problema:

¹²A função de Softmax, também conhecida como regressão logística multinomial, é definida como $\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{|\mathbf{V}|} e^{z_j}}$ e tem como característica principal receber $|\mathbf{V}|$ valores reais e convertê-los para uma distribuição de probabilidade de $|\mathbf{V}|$ saídas possíveis, onde $|\mathbf{V}|$ é o tamanho do vocabulário.

¹³Por exemplo, usando o TensorFlow: <https://tfhub.dev/google/elmo/2>.

¹⁴<https://allenai.org/allennlp/software/elmo>



sua característica sequencial faz com que elas não sejam paralelizáveis, ou seja, uma rede recorrente não pode ser separada em vários componentes para serem treinados em paralelo. Tal característica torna o treinamento das redes recorrentes bem ineficiente, o que acarreta em um outro problema: as entradas não podem ser muito grandes e nem exigirem uma dependência de longa distância. Mesmo que as redes do tipo *Long Short-Term Memory* (LSTMs) tenham aliviado o problema da dependência de longa distância com o uso dos mecanismos de *gate*, eles acarretam em redes com mais parâmetros para serem treinados, o que de novo nos leva à questão da ineficiência.

Bem, esse é um problema para modelar línguas com redes neurais recorrentes, uma vez que um texto pode ser enorme e ainda assim trazer componentes importantes lá no início para serem usados lá no fim. Considere, por exemplo, a frase do Exemplo 15.4:

Exemplo 15.4.

A garota de blusa amarela com uma frase em que os verbos estavam em letras pretas, que andava tão rápido e nunca em linha reta, a ponto de passar pelas nossas vistas como se fosse quase um furacão, tinha na parte de trás da sua blusa uma frase atribuída a Gandhi: “Acreditar em algo e não vivê-lo, é desonesto”.

Caso quiséssemos saber qual é a cor das letras em que a palavra “Acreditar” foi escrita, teríamos que conectar “Acreditar” com “verbo” e ver no início da frase que eles são escritos em preto. Claro que esse é um exemplo exagerado, mas pare para contar quantas palavras estão entre a cor da letra e o primeiro verbo da frase de Gandhi. Uma rede recorrente teria que aprender tais conexões, apesar da distância.

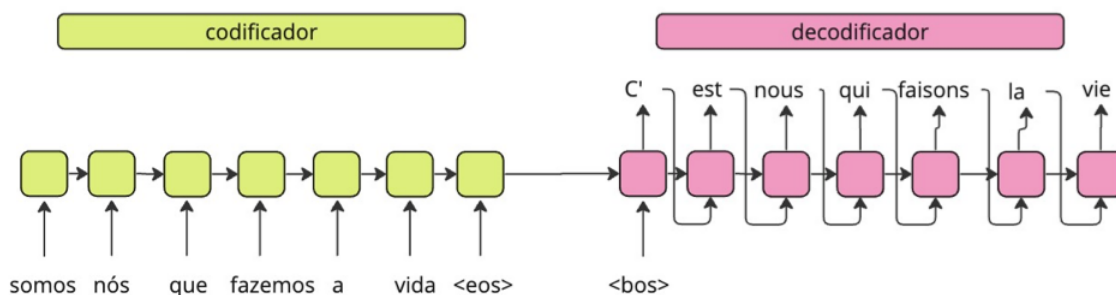
Outro ponto que precisamos mencionar antes de chegar aos Transformers (Vaswani et al., 2017) do título, que não são os mesmos dos filmes e brinquedos, mas que guardam muitas semelhanças, são as tarefas de PLN em que modelos de linguagem são costumeiramente usados: as tarefas de geração de sequências, no nosso caso, sequências de letras, palavras, textos. Tais sequências não são meramente concatenações de palavras, pois elas devem obedecer a princípios sintáticos e semânticos. Ainda, a geração de sequências não envolve apenas gerar textos do zero, ou completar frases, mas também gerar sequências a partir de outras sequências. Neste caso, a tarefa é chamada de forma genérica na literatura de *sequence-to-sequence* ou “seq2seq” (Cho et al., 2014; Sutskever; Vinyals; Le, 2014). Por exemplo, as tarefas de tradução automática, sumarização, respostas a consultas complexas, entre outras, requerem que a entrada seja um texto (uma sequência) e que a saída também seja um texto (outra sequência).

Do ponto de vista da modelagem da arquitetura de uma rede neural para resolver tarefas “seq2seq”, o mais comum é considerar dois grandes componentes: o primeiro, chamado de *encoder* ou codificador, é responsável por processar a sequência de entrada – para nós a sequência de letras, *tokens*, palavras, frases, e codificá-la como um vetor de números (as redes neurais gostam de números), chamado de *vetor de contexto*; o segundo componente, chamado de *decoder* ou decodificador, é responsável por receber e processar o vetor de contexto e transformá-lo na sequência de saída – uma sequência de letras, *tokens*, palavras, frases. Veja um diagrama de alto nível deste processo na Figura 15.4, que exemplifica uma tarefa de tradução automática.

O codificador é uma rede neural – ou várias delas – e o mesmo vale para o decodificador. Logo, as redes neurais têm seus parâmetros aprendidos com o foco de receber uma sequência-fonte e devolver a sequência-alvo desejada. Outro ponto importante é que a rede precisa da representação numérica dos itens na sequência de entrada. Assim, ou podemos ter uma

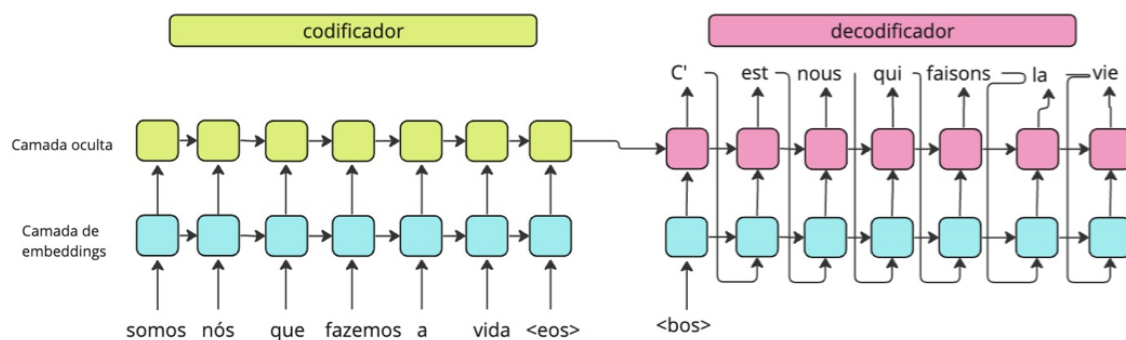


Figura 15.4: Exemplo do esquema básico de uma arquitetura seq2seq, exemplificada com uma tarefa de tradução de uma sentença em português para o francês. Os *tokens* “” e “<bos>” indicam o fim e início de sentença. Perceba que, assim como em uma rede recorrente tradicional, a saída da iteração anterior serve como entrada para a próxima iteração. No codificador, cada iteração apenas armazena informação, enquanto no decodificador, existe uma entrada e uma emissão de *token* a cada iteração.



camada inicial que faz a transformação de um vetor *one-hot* para um vetor de embeddings, ou podemos recuperar os embeddings das palavras a partir de um modelo pré-treinado. A Figura 15.5 traz um exemplo da arquitetura anterior, detalhando a camada de embeddings.

Figura 15.5: Esquema básico de uma arquitetura seq2seq, exemplificada com uma tarefa de tradução de uma sentença em português para o francês. Os *tokens* “” e “<bos>” indicam o fim e início de sentença. Aqui, a camada de embeddings está explícita.

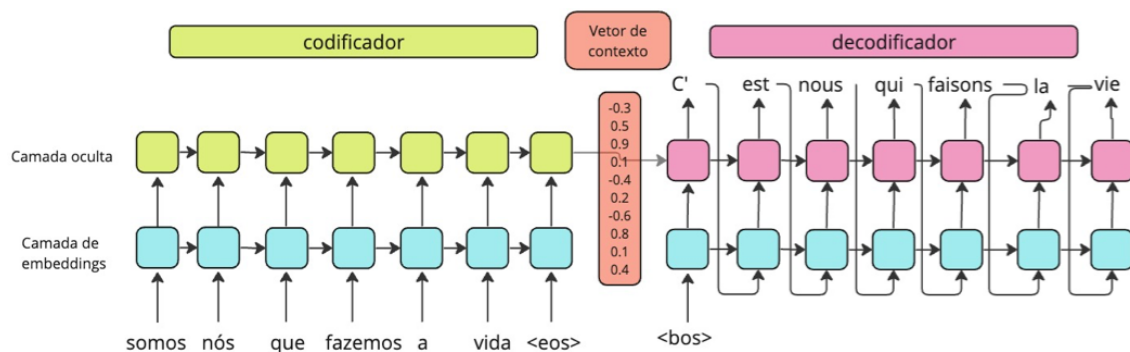


Bem, o codificador e o decodificador podem muito bem ser redes recorrentes, com uma ou mais camadas, do tipo LSTM, ou alguma outra variação. Em tais casos, o último estado escondido da rede, no sentido lógico, ou seja, obtido após processar o último item da sequência, será o vetor de contexto. A Figura 15.6 explicita o vetor de contexto, que antes estava representado de forma implícita como a seta de ligação entre o codificador e o decodificador.

Aqui temos um problema: é complicado assumir que esse último estado escondido, codificado como o vetor de contexto, conseguirá capturar todos os aspectos necessários para resolver a tarefa, ainda mais se a sequência de entrada for grande. Para lidar com este problema, pesquisadores elaboraram uma nova estratégia, chamada de *mecanismo de*



Figura 15.6: Esquema básico de uma arquitetura seq2seq, exemplificada com uma tarefa de tradução de uma sentença em português para o francês, com o vetor de contexto conectando o estado final do codificador e a entrada do decodificador.



atenção¹⁵ (Bahdanau; Cho; Bengio, 2015; Luong; Pham; Manning, 2015).

¹⁵Alguns consideram que o mecanismo de atenção é uma forma de explicação, enquanto outros discordam veementemente disso. Veja as discussões: (Bibal et al., 2022; Jain; Wallace, 2019; Wiegrefe; Pinter, 2019).



15.3.2.2.1 Atenção!

O objetivo do mecanismo de atenção – na verdade, um conjunto adicional de parâmetros para a rede – é que os itens mais relevantes da entrada recebam uma valoração maior no vetor de contexto. Mas outros itens também podem receber algum valor. A ideia é mais ou menos assim, e deixe de lado a língua natural só por um minuto, para um exemplo mais abstrato: suponha que você quer aprender a assar um bolo de chocolate. Vamos chamar “Assar o bolo de chocolate” de consulta. Você pode pegar o livro de receitas da sua avó para te ajudar. O livro é composto de diversas receitas, que vamos chamar de chaves. O que você quer é encontrar a receita mais adequada, e para isso, todas as receitas vão receber alguma valoração. A receita do bolo de chocolate perfeito deve ter um valor maior em relação aos demais, mas uma receita de bolo de chocolate com morango, também pode receber alguma relevância. Mas uma receita de Tiramissu deveria ter uma relevância bem pequenininha. O mecanismo de atenção segue essa ideia: a saída é a consulta, a informação que precisa ser gerada a partir da entrada, as chaves. Para definir a chave mais relevante, são calculados pesos de atenção, que definirão o vetor de contexto.

Para considerar a relevância de diferentes itens na entrada, o codificador não considera que apenas o último estado escondido da rede será o vetor de contexto, mas que todos os estados escondidos, ou seja, todos os estados obtidos após o processamento de cada item da sequência, também podem participar do vetor de contexto. Mas agora o decodificador terá mais trabalho, pois ele precisará decidir o que fazer com esses vários vetores antes de gerar os itens da saída, e ainda considerando que é necessário focar nas partes mais relevantes para a resolução da tarefa. Assim, antes de gerar a saída pelo decodificador, são executados os seguintes passos:

1. Computar uma pontuação para cada estado escondido, também chamada de pontuação de alinhamento, seguindo a Equação 15.4.
2. Passar as pontuações combinadas – por concatenação, em geral, ou o vetor resultante da equação anterior, pensando em termos de representações matriciais – por uma função de “softmax”, para capturar alguma noção de probabilidade da relevância, produzindo os pesos de atenção.
3. Multiplicar cada estado escondido (lembrando que ele é representando por um vetor) pelos pesos de atenção, de forma a tornar os estados escondidos mais relevantes com valores ainda maiores, e obter o efeito oposto para os estados menos relevantes. O resultado deste passo será o vetor de contexto.

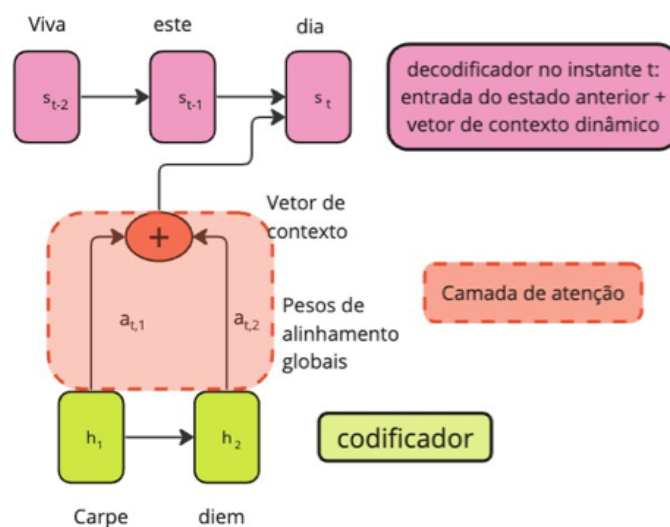
$$att = \mathbf{W}_{\text{combinado}} \times \tanh(\mathbf{W}_{\text{decod}} \times \mathbf{H}_{\text{decod}} + \mathbf{W}_{\text{codif}} \times \mathbf{H}_{\text{codif}}) \quad (15.4)$$

onde $\mathbf{W}_{\text{codif}}$ e $\mathbf{W}_{\text{decod}}$ representam matrizes de pesos (parâmetros) aprendidos e $\mathbf{H}_{\text{decod}}$ representam estados escondidos. Observe a semelhança com uma rede neural de uma camada escondida com a função de ativação de tangente hiperbólica. Veja um exemplo ilustrativo na Figura 15.7.

O mecanismo de atenção calculado desta forma também é chamado de mecanismo aditivo ou mecanismo de Bahdanau, proposto em Bahdanau; Cho; Bengio (2015). Existe um segundo tipo de atenção, proposto em (Luong; Pham; Manning, 2015), chamado de mecanismo de atenção multiplicativo. As diferenças principais são que o decodificador produz um estado intermediário a partir do estado escondido anterior antes de calcular os pesos de atenção, e as pontuações de alinhamento podem ser de três tipos: (i) multiplicando os estados escondidos do codificador e decodificador apenas, ou seja, $att = \mathbf{H}_{\text{decod}} \times \mathbf{H}_{\text{codif}}$



Figura 15.7: Ilustração do mecanismo de atenção aditivo para a tradução da expressão em latim *carpe diem* para a expressão em português “Viva este dia”.



Fonte: Adaptado de (Bahdanau; Cho; Bengio, 2015)

(ii) multiplicando uma matriz de pesos aprendidos ao resultado da multiplicação em (i), ou seja, $att = \mathbf{W} \times \mathbf{H}_{decod} \times \mathbf{H}_{codif}$, e (iii) somando os estados escondidos do codificador e decodificador, que são multiplicados por uma matriz de pesos, passam pela função de ativação da tangente hiperbólica e são finalmente multiplicadas a uma matriz de pesos, $att = \mathbf{W} \times \tanh(\mathbf{W}_{combinado}(\mathbf{H}_{decod} + \mathbf{H}_{codif}))$. Este último caso é o mais similar ao mecanismo aditivo, mas os estados escondidos compartilham uma matriz de pesos, diferente da Equação 15.4. Ao final, o vetor de contexto é concatenado com o estado do decodificador no instante anterior, para produzir uma nova saída.

O mecanismo de atenção apresentado até agora é chamado de mecanismo de atenção geral, uma vez que ele tenta encontrar os componentes da entrada que são mais relevantes para gerar a saída. *Transformers* fazem uso de um mecanismo de atenção adicional, chamado de *auto-atenção*, em que a captura da relevância é feita entre os elementos de uma mesma sequência, usualmente da entrada.

15.3.2.2 Finalmente os Transformers

Mas vamos finalmente entender o que são esses tais Transformers, uma arquitetura de rede neural proposta em 2017 e que faz uso do mecanismo de atenção, entre outros componentes conhecidos de redes neurais, e cujo esquema está representado na Figura 15.9. Um Transformer tem dois componentes principais, adivinhem só, um componente de codificação e um componente de decodificação. Entretanto, diferente do que falamos antes nos modelos seq2seq, Transformers não são constituídos por redes neurais recorrentes. Com isso, é possível paraleliza-los e alcançar tempos de treinamento mais eficientes para modelos de linguagem do que usando redes recorrentes. Mas não é só isso: o uso extensivo de mecanismos de atenção, combinados a outros componentes, fez dos Transformers e suas diversas variações o estado da arte em diversas tarefas de PLN, ao menos até o momento de escrita deste livro (Wolf et al., 2020). Eles são o componente principal dos modelos de linguagem em larga escala (em inglês, *large language models* ou LLMs) que deram o que falar no início do ano de 2023, principalmente com a vasta disponibilidade de agentes de



conversação e suas interfaces de programação de aplicações¹⁶.

Vamos entender do que esses codificadores e decodificadores são compostos, já que não são redes recorrentes. O codificador é, na verdade, uma pilha de sub-codificadores, enquanto o decodificador é uma pilha de sub-decodificadores. No artigo original, essas pilhas tinham seis componentes, mas poderia ser qualquer outra quantidade. Os sub-codificadores possuem estruturas idênticas e são constituídos de dois outros componentes: um mecanismo de auto-atenção e uma rede neural completamente conectada de uma camada.

Antes de explorar os demais componentes, vamos observar como funciona o mecanismo de auto-atenção. Considere que cada item da sequência (um *token*, uma palavra) é representado por um *embedding*. Como antes, o vetor de *embedding* pode ter sido pre-treinado. Considerando o ponto de entrada de um transformer como sendo o tal vetor de embeddings, são criados três vetores a partir de cada palavra ou *token*. A implementação é toda matricial, para fazer bom uso das GPUs, mas podemos abstrair para vetores, para facilitar o entendimento. Vamos então considerar que temos a frase *viva este dia*. *Transformers* fazem uso de *tokens* de subpalavras, para aliviar o problema das palavras que estariam fora de um vocabulário pre-treinado, conforme apresentado no Capítulo 4. Mas, para simplificar, vamos assumir que cada palavra é um *token*. Temos então três *tokens* na frase, que serão representados pelos vetores x_i – viva, x_2 – este, e x_3 – dia, que são os embeddings de cada palavra. A partir de cada um deles, criamos três outros vetores, q , k e v , de *query* (consulta), *keys* (chaves) e *values* (valores)¹⁷, respectivamente (lembra do exemplo do bolo?). O vetor q se refere a um item de interesse que está sendo codificado. O vetor k se refere aos demais itens da sentença. O vetor v representa a codificação do valor dado a cada item, considerando o item de interesse. Para o nosso exemplo, temos então, os vetores q_1 , k_1 e v_1 para a palavra “viva”, q_2 , k_2 e v_2 para a palavra “este” e q_3 , k_3 e v_3 para a palavra “dia”.

Como os vetores são obtidos? Como é de praxe com redes neurais, usando matrizes de pesos aprendidas com os dados. Assim, multiplicando o vetor x_1 pela matriz de pesos associadas às *queries*, \mathbf{W}_Q , temos o vetor q_1 . O mesmo vale para os demais itens, ou seja, para obter o vetor k_1 , multiplicamos x_1 por uma outra matriz de pesos \mathbf{W}_k , e para obter v_1 , multiplicamos x_1 por uma outra matriz de pesos \mathbf{W}_v . Vamos agora calcular o peso de atenção, para, dada uma palavra que está fazendo às vezes de *query*, identificarmos quais são as *keys* mais relevantes para produzir o vetor de valoração. Observe que essa intuição está inserida nas matrizes de peso aprendidas. Então, assumindo inicialmente que a query é a palavra “viva”, multiplicamos seu vetor q_1 por cada uma das keys, k_1 , k_2 e k_3 . Observe que depois o mesmo será feito para as demais palavras. Os valores multiplicados são divididos por 8¹⁸. A seguir, como antes, os valores passam por uma função de softmax, para que eles sejam transformados em probabilidades. A soma de todas as probabilidades vai ser sempre igual a um. Ou seja, ficamos com a ideia de que cada *key* é mais ou menos importante para cada palavra query, de acordo com o valor computado pela softmax. Agora aparecem os vetores de valoração. Os valores calculados pelo softmax são multiplicados por cada um dos vetores de *values*, para codificar a importância das demais palavras na valoração. Finalmente, esses valores são somados, produzindo um valor final, chamado de z_1 para a primeira palavra, que será passado adiante para a rede neural completamente

¹⁶<https://openai.com/blog/introducing-chatgpt-and-whisper-apis>

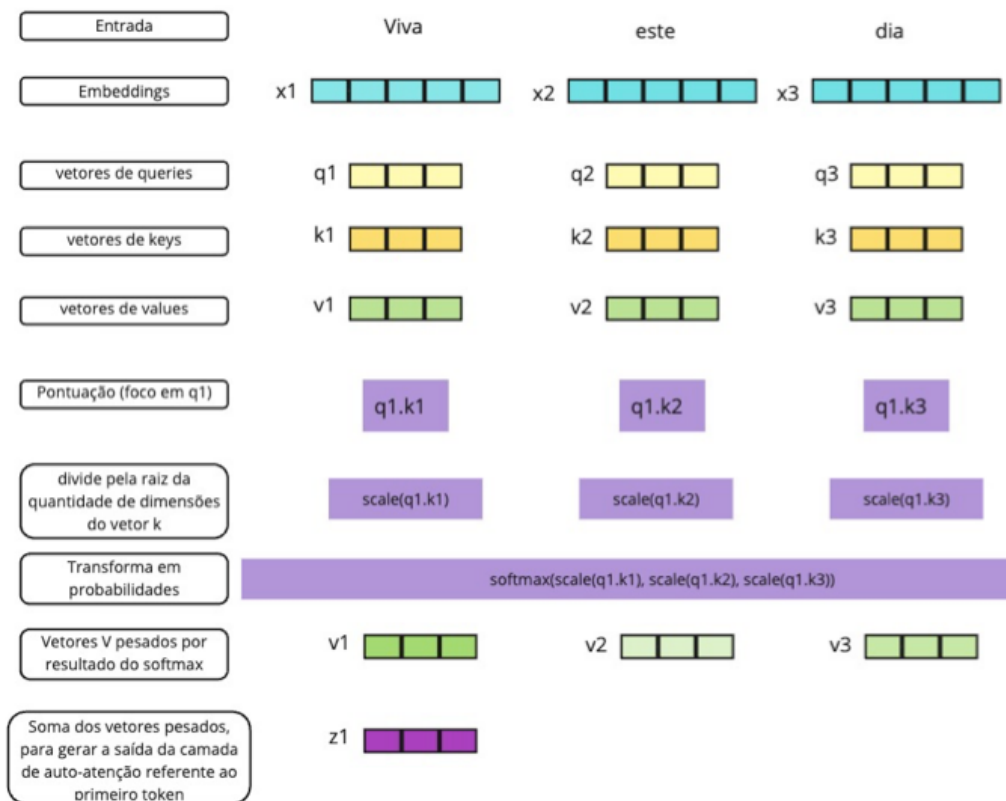
¹⁷Para manter as letras clássicas que referenciam os vetores, vamos manter os nomes em inglês.

¹⁸A explicação no artigo é que 8 é a raiz quadrada da dimensão dos vetores de *keys*, que no artigo é 64. Essa divisão é uma espécie de normalização, para deixar os cálculos dos gradientes mais estáveis. Outros valores poderiam ser usados, especialmente com vetores de outras dimensões.



conectada. Veja um esquema do processo para a primeira palavra na Figura 15.8.

Figura 15.8: Exemplo do mecanismo de auto-atenção para codificar a frase “Viva este dia”.



Fonte: Adaptado de <http://jalammr.github.io/illustrated-transformer/>

15.3.2.2.3 Atenção em múltiplas versões

A linguagem é sujeita a várias complexidades que podem fazer o mecanismo de atenção não ser suficiente para representá-las. Dependendo da sentença de entrada, podemos ter variações na atenção. Isso é bem comum em problemas de correferência. Por exemplo, na frase “Eles não levaram os livros nos compartimentos porque eles eram muito pequenos”, o segundo “eles” pode se referir a vários outros pronomes e substantivos na frase. Ainda, dependendo do contexto, as palavras podem ter vários significados, um conceito chamado de polissemia. Então, para capturar diferentes representações de uma palavra, bem como diferentes interações da palavra com os demais componentes, Transformers incluem um nível de paralelismo no processamento da entrada, a partir de um componente adicional chamado de *multi-heads attention*. No artigo original, os Transformers incluem oito versões paralelas do mecanismo de atenção, fazendo com que tenhamos oito vezes três matrizes de *keys*, *values*, *queries* inicializadas aleatoriamente, o que permite que tais matrizes capturem diferentes aspectos da entrada.

Lembre que o mecanismo de atenção produz uma matriz final Z após processar a entrada a partir das diferentes matrizes de peso Q , K , V . Anteriormente, falamos que o resultado do passo de atenção é submetido a uma rede neural completamente conectada. Para que a rede completamente conectada consiga lidar com as várias matrizes Z geradas em paralelo, elas são concatenadas e multiplicadas por uma outra matriz de pesos adicional,



gerando, finalmente, uma única matriz Z que representa o resultado do mecanismo de atenção em suas várias versões. Como essa matriz de pesos adicional também é aprendida, Transformers dão a chance de alguma das versões do mecanismo de atenção paralelo ter mais ou menos relevância que algum outro, dependendo dos dados de treinamento.

Mas quantas matrizes são treinadas, não é mesmo? Pare um momento para fazer uma conta de quantos pesos um Transformer precisa treinar, considerando os componentes que apresentamos até aqui. O que isso pode fazer com o meio ambiente, se um décimo das pessoas do planeta resolvessem treinar seu próprio Transformer?

15.3.2.2.4 E as posições das palavras??

Duas motivações foram apresentadas para construir modelos de linguagem a partir de redes recorrentes: (i) permitir entradas de tamanho variável e (ii) permitir que o aprendizado tenha acesso à ordem das palavras e absorva a diferença que vem de ordens distintas, bem como a importância da ordem para tarefas sintáticas e semânticas. A recorrência é o mecanismo utilizado para atender a estas duas motivações no ELMo, por exemplo. Mas Transformers não incluem nada de recorrência. E agora?

A bem da verdade, para permitir o treinamento de forma eficiente, as implementações das redes recorrentes já não deixavam a entrada ser tão variável assim. Para que os tensores sejam formados e manipulados de forma eficiente, é comum que algumas implementações preencham frases com símbolos nulos e organizem frases que tenham o tamanho mais aproximado o possível, para que eles fiquem nos mesmos lotes e ajudem na manipulação dos tensores. De certa forma, Transformers possuem uma entrada de tamanho pré-definido. O tamanho pré-definido, em geral, é até bem menor do que gostaríamos para manipular textos um pouco mais longos, por questões de desempenho. Na Seção 15.4 falaremos de como este problema tem sido abordado. Mas é possível lidar com sentenças de tamanhos distintos nos Transformers, adotando alguma das abordagens abaixo:

- Quando a sentença de entrada tem menos *tokens* que a quantidade de *tokens* de entrada esperada pelo modelo: esse é o caso mais fácil, quando a sentença é preenchida com valores nulos, um processo chamado de *padding*.
- Quando a sentença de entrada tem mais *tokens* que a quantidade de *tokens* de entrada esperada pelo modelo: duas soluções podem ser adotadas. A mais simples é truncar a entrada, removendo elementos do início ou do fim da sentença. Outra forma mais elaborada é quebrar a sentença em janelas com elementos sobressalentes entre elas e passar esses pedaços ou janelas várias vezes no modelo.

O outro problema, a ordem das palavras, exige a inclusão de um componente adicional no modelo, uma vez que a ordem é de extrema relevância para a sintaxe e a semântica, e portanto também para modelos que tentam aprender a resolver tarefas sintáticas ou semânticas. Assim, Transformers incluem um tipo especial de *embedding*, chamado de *codificador de posição* (*positional encoding*), para contemplar alguma informação sobre as posições dos *tokens* durante o aprendizado. O codificador posicional é um vetor a mais somado ao vetor de embeddings de entrada de cada *token*. Embora, em um primeiro pensamento, possa parecer mais direto considerar um valor simples de posição, como por exemplo, um índice, essa abordagem traria alguns problemas. O primeiro é que o valor pode ficar muito grande, dependendo de quantas palavras temos, e o modelo poderia se confundir achando que esses valores altos têm alguma importância. Mesmo se o valor fosse normalizado entre 0 e 1, diferentes tamanhos de sentenças trariam diferentes valores, o que também atrapalharia a generalização do aprendizado.



Assim, o codificador de posição define um vetor de valores contínuos do mesmo tamanho do *embedding* de entrada do *token*, para que seja possível somá-los. Para incorporar mais uma ideia de distância entre as palavras, ou de posição relativa, do que uma ideia rígida de posição, o vetor é obtido a partir de uma função que intercala entre a aplicação de seno ou cosseno. Mais precisamente, para codificar a informação de posição de um *token* que está em uma posição k na sequência de entrada, considerando cada posição i do vetor posicional, fazemos:

$$p(k, 2i) = \sin\left(\frac{k}{n^{\frac{2i}{d}}}\right)p(k, 2i + 1) = \cos\left(\frac{k}{n^{\frac{2i}{d}}}\right)$$

onde d é a dimensão do vetor posicional e n é um valor pré-definido¹⁹. Para posições pares do vetor de saída, aplica-se o seno e para posições ímpares, aplica-se o cosseno. A Tabela 15.1 apresenta um exemplo simplificado da aplicação do codificador posicional.²⁰

Tabela 15.1: Exemplo da computação do codificador posicional, considerando um vetor de saída de quatro dimensões apenas e $n=10$

<i>Token</i>	índice na sentença	i=0	i=1	i=2	i=3
viva	0	0	1	0	1
este	1	0,8415	0,5403	0,3109	0,9504
dia	2	0,9093	-0,4161	0,5911	0,8607

15.3.2.2.5 Mecanismo residual e normalização

O último subcomponente dos Transformers que precisamos falar é a inclusão de duas conexões residuais (He et al., 2016) dentro da subcamada de codificação. A conexão residual surgiu na área de visão computacional, com a motivação que redes neurais com muitas camadas podem esquecer uma informação importante de entrada após ela passar por muitos processamentos. Em geral, esse esquecimento se dá pelo problema do gradiente que vira zero, depois de muitas multiplicações de valores menores que um (Hochreiter, 1991) durante o *backpropagation*²¹. A conexão residual justamente evita parte dessas transformações multiplicativas, pulando algumas delas.

No caso dos Transformers, além de evitar que o treinamento se perca com multiplicações de valores muito pequenos, a motivação é que os embeddings de representação de palavras também continuem a ser aproveitados de alguma forma, trazendo uma ideia de representação local dos *tokens* para a subcamada de codificação. Ou seja, ao permitir que a informação sem ser processada pela camada de auto-atenção e que a informação sem ser processada pela camada completamente conectada sejam consideradas, é como se a rede estivesse lembrando da representação original do *token*, quando necessário. Para tanto, a saída da camada de auto-atenção é somada com a entrada original, que por sua vez é somada com

¹⁹No artigo original, foi definido o valor de 10.000.

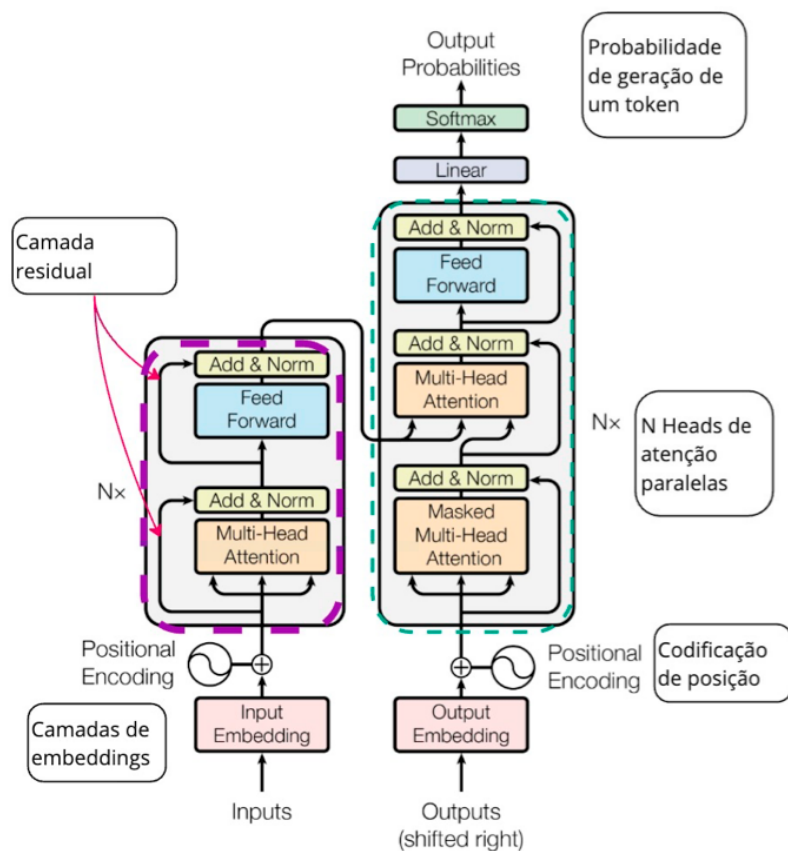
²⁰Para quem quiser entender a derivação do porquê essas funções fazem sentido para representar a informação de posição, podem debruçar nesta detalhada explicação: <https://timodenk.com/blog/linear-relationships-in-the-transformers-positional-encoding/> ou para uma ideia mais geral, aqui: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/.

²¹O algoritmo *backpropagation* é o mais utilizado para aprender os pesos de uma rede neural, usando o método de otimização de gradiente descendentes visando minimizar o erro do modelo. Veja mais detalhes em <https://www.deeplearningbook.com.br/algoritmo-backpropagation-parte-2-treinamento-de-redes-neurais/>.



a saída da camada completamente conectada, preservando, e repassando para a frente de alguma forma, a entrada original.

Figura 15.9: Arquitetura Transformer. O pontilhado violeta representa o codificador e o pontilhado verde representa o decodificador.



Fonte: Modificado a partir de (Vaswani et al., 2017).

A arquitetura Transformer original possui em seu componente de codificação seis subcamadas de codificadores, com as camadas internas completamente conectadas tendo 512 neurônios artificiais intermediários (na camada oculta, ou escondida) e oito *heads* de atenção.

Só mais um detalhe para fecharmos o nível interno de uma camada de codificação: para ajudar no aprendizado dos gradientes, antes da camada de atenção e antes da camada completamente conectada, temos uma camada de normalização. Na verdade, tem uma pequena confusão com essa camada. A figura original dos Transformers (Figura 15.9) aponta que a normalização acontece após o processamento da camada de atenção e após o processamento da camada completamente conectada (pós-normalização). Entretanto, no código original está diferente²²: na verdade, no código temos uma pré-normalização, que ocorre antes do cálculo dos valores de atenção. Argumenta-se que isto ajuda a lidar melhor com os gradientes (Xiong et al., 2020). Mas esta discussão, de onde inserir a camada de normalização, e onde ela apresenta mais vantagens, ainda é um ponto de investigação em aberto.

²²Veja mais em <https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html>.



15.3.2.2.6 Decodificador e comunicação entre codificador e decodificador

A arquitetura do componente Decodificador é bastante parecida com a arquitetura do Codificador. Ambas são uma pilha de sub-camadas, com cada sub-camada incluindo uma camada de auto-atenção, de normalização, um componente residual e uma camada de rede completamente conectada. Porém, aqui temos um componente a mais, que é uma camada de atenção convencional, que se comunica com o Codificador. Assim, após o processo de codificação, as matrizes de atenção K e V serão a entrada para a camada de atenção convencional do decodificador. Já a matriz Q vem mesmo da camada anterior, justamente para que essa camada ajude o modelo a decidir o que ele precisa considerar para gerar a saída.

Uma outra diferença é que a camada de auto-atenção do decodificar é chamada de mascarada, uma vez que ela não tem acesso aos *tokens* que estão em uma posição posterior a um certo *token*. Assim, no decodificador, considerando o processamento de um *token* t_i , a camada de auto-atenção do decodificador só terá acesso aos *tokens* $\{t_0, t_1, \dots, t_{i-1}\}$ para calcular o valor da auto-atenção. Este comportamento tem a ver com o que se espera de um decodificador: que ele gere um próximo *token*, dados os *tokens* anteriores a ele, mas sem saber o futuro de antemão. Assim, o decodificador lembra muito o processo do modelo probabilístico que discutimos antes: (1) o processamento parte dos *tokens* “anteriores”, que inicialmente é apenas um *token* especial de início de sentença; a entrada do decodificador considera esses *tokens* anteriores e a camada de atenção convencional considera as matrizes geradas pelo codificador; (2) um *token* é gerado pelo modelo; (3) o processo se repete.

Entretanto, nos modelos probabilísticos, fica clara a existência das probabilidades, enquanto até agora só falamos em vetores. *Transformers* incluem uma última camada de rede neural, justamente para resolver tal discrepância. Assim, a última camada da arquitetura recebe a saída do decodificar (um vetor) e a processa com uma camada linear completamente conectada. A saída da camada linear completamente conectada é um vetor de *logits*²³ do tamanho do vocabulário, que representam uma pontuação associada a cada palavra do vocabulário. Finalmente, tais valores de pontuação passam por uma operação de SOFTMAX, para converter esses valores reais em valores que fiquem entre 0 e 1, representando a probabilidade de que o decodificador emita cada uma das palavras do vocabulário.

15.3.2.2.7 Arquiteturas que instanciam Transformers

Tarefas que lidam com linguagem têm sido abordados por diferentes instanciações de Transformers: podemos considerar a arquitetura completa, podemos considerar apenas o componente codificador, ou podemos considerar apenas o componente decodificador. Ainda, é possível não usar todas as camadas existentes no modelo original, mas subconjuntos (ou até mesmo superconjuntos) delas.

15.3.2.2.8 Codificador: BERT e seus amigos

A arquitetura mais utilizada que considera apenas o componente codificador chama-se BERT²⁴, de *Bidirectional Encoder Representations for Transformers* (Devlin et al., 2019). O BERT foi treinado em duas versões, uma chamada *base* e outra chamada *large*. A versão *base* possui 12 subcamadas de codificação, que por sua vez incluem camadas completamente conectadas com 768 unidades de neurônios artificiais intermediários e 12

²³Neste contexto, *logits* são basicamente logaritmos de probabilidades não normalizados.

²⁴Também é uma brincadeira com um personagem do famoso desenho infantil *Sesame Street*, no ar desde 1969, e que no Brasil ganhou uma versão chamada Vila Sésamo.



heads de atenção. A versão *large* é composta de 24 subcamadas de codificadores, com suas camadas completamente intermediárias tendo 1024 neurônios artificiais intermediários e as camadas de atenção com 16 *heads*.

Em ambas as arquiteturas, a entrada para o BERT tem uma limitação de 512 *tokens*, devido, principalmente, ao processamento quadrático do mecanismo de atenção, que considera todos os *tokens* para cada *token* em seus cálculos. O primeiro *token* é um especial chamado de [CLS], cujo uso ficará mais claro quando falarmos do processo de treinamento e inferência com modelos de linguagem. O BERT também pode receber duas sentenças (também ficará mais claro daqui a pouco), e nesse caso elas são separadas com um outro *token* especial chamado [SEP]. Para cada *token* da entrada, o BERT produz um vetor de saída de 768 ou 1024 posições, dependendo da configuração *base* ou *large*. A saída completa de um modelo BERT é um tensor de quatro dimensões, com a primeira representando a quantidade de subcamadas de codificação (12) mais a camada dos embeddings de entrada, totalizando 13, a segunda representando a quantidade de lotes (voltaremos nele ao falar do treinamento), a terceira a quantidade de *tokens* na entrada, e a quarta o tamanho da camada escondida. Veja um exemplo abaixo, onde a quantidade de *tokens* é 26 devido ao processo de tokenização em *subtokens* (na verdade, são 24 *tokens*, pois o primeiro e o último são os *tokens* especiais, [CLS] e [SEP]). O código do exemplo segue o *framework* HuggingFace (Wolf et al., 2020).

```
import torch
# carregando os módulos do framework HuggingFace
from transformers import AutoTokenizer, AutoModel

# carregando o tokenizador
tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")

# carregando o modelo BERT BASE
model = AutoModel.from_pretrained("bert-base-uncased")

# frase de Epicteto, um filósofo estoico
s = "a riqueza não consiste em ter grandes posses, mas em ter poucas
necessidades"

# frase tokenizada
input_sentence = torch.tensor(tokenizer.encode(s)).unsqueeze(0)

# saída do modelo
# output_hidden_states=True faz com que tenhamos acesso a todas as
camadas na posição 2 da variável de saída
out = model(input_sentence, output_hidden_states=True)

print("Numero de camadas: ", len(out[2]))
print("Numero de lotes: ", len(out[2][0]))
print("Numero de tokens: ", len(out[2][0][0]))
print("Numero de neurônios artificiais: ", len(out[2][0][0][0]))

Numero de camadas: 13
Numero de lotes: 1
Numero de tokens: 26
Numero de neurônios artificiais: 768
```

Por curiosidade, os *tokens* gerados são como seguem abaixo. Observe que os espaços não são *tokens*, mas estão aqui apenas para conseguirmos separar um *token* de outro.



```
a ri ##que ##za [UNK] consist ##e em ter grande ##s posse ##s , mas
em ter po ##uca ##s nec ##ess ##idad ##es
```

onde as duas tralhas indicam que o *token* é, originalmente, parte de uma palavra e [UNK] é um *token* para indicar que não foi possível encontrar aquele componente no vocabulário. No caso acima, provavelmente é devido à presença de uma palavra com til, o que não existe na língua inglesa.

Um modelo de linguagem segue um vocabulário e o vocabulário do BERT original segue a língua inglesa. Embora o processo de tokenização em subpalavras consiga identificar algumas palavras, toda a definição dos pesos durante o processo de aprendizado será feito com base em um vocabulário de um idioma distinto. E quando falamos de uma língua, não é apenas o vocabulário que é relevante, mas as regras sintáticas, seus significados, e até mesmo características culturais e da sociedade.

Então, temos duas opções: ou considerar um modelo que tenha se deparado com um vocabulário de vários idiomas, ou treinar um modelo separado para uma língua. Para o primeiro caso, temos, por exemplo, o modelo **BERT multilingual** (chamado de mBERT), treinado com textos da Wikipedia de 104 idiomas. Com o mesmo teste feito antes, teríamos agora 18 *tokens*, ao invés de 26, mostrando que, ao menos, mais palavras são reconhecidas. Em particular, usando o mBERT, o processo de tokenização devolve

```
a riqueza não consiste em ter grandes posse ##s , mas em ter poucas
necessidade ##s
```

Para o segundo caso, temos alguns modelos treinados especificamente para o português (outras línguas também, mas nosso interesse aqui é na nossa linda língua materna). O modelo **BERTPT** (Feijó; Moreira, 2020) foi treinado com um vocabulário de tamanho 30.000, assim como o modelo BERT original, porém mantendo a configuração original de maiúsculas e minúsculas e sinais diacríticos (os acentos). Foram usados 4,8GB de textos, considerando textos em português do Brasil e europeu, tanto mais formais, como Wikipedia-PT²⁵ e EuroParl²⁶, como textos mais informais, como Open Subtitles²⁷. No total, foram considerados 992 milhões de *tokens*. A arquitetura utilizada foi a base. O modelo BERTPT apresentou resultados melhores em bases de dados compostas por textos mais informais.

O modelo **BERTimbau**²⁸ (Souza; Nogueira; Lotufo, 2020) também partiu da arquitetura do BERT, mas treinou duas versões, uma a partir da arquitetura *base* e outra a partir da arquitetura *large*. Assim como no BERTPT, são mantidas as letras maiúsculas e minúsculas e acentos e o tamanho do vocabulário também é de 30.000 *tokens*. O conjunto de textos usados para treinar os modelos foi o brWaC (Wagner Filho et al., 2018), que é composto de textos em português do Brasil, contendo 2,68 bilhões de *tokens* e 3,53 milhões de documentos, e após uma fase de pré-processamento ficou com 17,5GB de textos. Outra diferença em relação ao BERTPT é que a tokenização utilizou o algoritmo BPE, enquanto o BERTPT e o BERT original seguem o algoritmo WordPiece, ambos mencionados no Capítulo 4. Apenas para fins de comparação com o exemplo anterior, abaixo temos o resultado da tokenização usando o tokenizador do BERTimbau, que fica com um *token* a menos que o mBERT. O resultado foi gerado com a versão disponibilizada no hub de

²⁵<https://dumps.wikimedia.org/backup-index.html>

²⁶<https://www.europarl.europa.eu/>

²⁷<http://opus.nlpl.eu/OpenSubtitles-v2016.php>

²⁸Uma brincadeira com o berimbau.



modelos HuggingFace²⁹ (Wolf et al., 2020). Na maioria dos resultados apontados no artigo, o BERTimbau supera o mBERT.

```
a riqueza não consiste em ter grandes posse ##s , mas em ter poucas
necessidades
```

O último modelo para português baseado no BERT que citaremos aqui é o **Albertina** (Rodrigues et al., 2023), treinado em duas variantes, português europeu (Albertina PT-PT) e português do Brasil (Albertina PT-BR). A versão PT-BR também foi treinada com o brWaC. Já a versão PT-PT foi treinada com um subconjunto de textos em português extraídos da versão de Janeiro de 2023 do *corpus* Oscar (Abadji et al., 2022) e de outros três *corpora* constituídos de documentos do parlamento europeu e português. No total, foram utilizados oito milhões de documentos contendo 2,2 bilhões de *tokens*.

Uma diferença crucial do Albertina para os modelos anteriores é que a arquitetura base não é a do BERT, mas sim uma versão estendida com duas novas técnicas, chamada DeBERTa (do inglês, *Decoding enhanced BERT with disentangled attention*) (He et al., 2021). A primeira modificação diz respeito ao mecanismo de atenção. Lembre que nos Transformers, um *token* é representado pela soma do seu vetor inicial de embeddings e do seu vetor de codificação de posição. No DeBERTa, e consequentemente no Albertina, temos dois vetores que são processados separadamente (daí o *disentangled*, ou desemaranhado em português), onde o vetor de codificação de posição representa a posição relativa de um *token* i em relação a um *token* j . O valor de atenção cruzada de dois *tokens* é calculado como $A_{i,j} = \{\mathbf{H}_i, \mathbf{P}_{ij}\} \times \{\mathbf{H}_j, \mathbf{P}_{ji}\}^\top$, onde \mathbf{H}_i representa o vetor de embeddings do *token* i e \mathbf{P}_{ij} representa a posição relativa do *token* i em relação ao *token* j . A outra modificação tem a ver com a tarefa de treinamento genérica das arquiteturas Transformers e voltaremos nela na seção seguinte.

Para não perder o costume, veja abaixo o resultado do processo de tokenização usando a versão PT-BR disponibilizada no HuggingFace³⁰. Observe que a representação é diferente das anteriores: cada *token* que é o início de uma palavra recebe um ‘_’ como prefixo. O tokenizador também tem a diferença de tratar espaços como se eles fossem parte do *token*.

```
_a _rique za _não _consist e _em _ter _grande s _posses , _mas _em
_ter _pou cas _necess idades
```

Também temos alguns modelos BERT treinados para tweets em português³¹. Certamente, existem vários outros modelos treinados para o português que não apareceram aqui. Observem que esta não é para ser mesmo uma lista exaustiva.

Existem diversas outras arquiteturas que estendem, melhoram, modificam, ou treinam com mais dados ou com outros parâmetros o componente codificador dos Transformers. Exemplos incluem ROBERTa (Liu et al., 2019), que incluiu modificações no treinamento e usa o algoritmo de tokenização BPE ao invés do WordPiece; DistillBERT (Sanh et al., 2019), que se vale de um processo de destilação de conhecimento para aproximar os pesos do modelo original e obter um modelo menor que o BERT; ALBERT (Lan et al., 2020), que introduz três mecanismos – fatorização das matrizes de embeddings, compartilhamento de pesos e uma nova forma de treinamento – para obter um modelo mais eficiente que o BERT; ELECTRA (Clark et al., 2020), que também muda a forma de treinamento do

²⁹<https://huggingface.co/neuralmind/bert-base-portuguese-cased>

³⁰<https://huggingface.co/PORTULAN/albertina-ptbr>

³¹<https://huggingface.co/melll-uff/bertweetbr> e <https://huggingface.co/pablocosta/bertabaporu-base-uncased>, por exemplo



BERT para obter embeddings melhores com um processo mais eficiente; dentre muitos outros.

15.3.2.2.9 Decodificador: GPT e seus vizinhos

A saída do codificador em um modelo Transformer é uma representação vetorial, que para resolver uma tarefa final ainda precisa passar por algum outro processo. Mas essa representação pode ser bem robusta, uma vez que ela olha ambos os lados direito e esquerdo de um *token* ao construir sua representação vetorial. Já o componente decodificador tem uma característica autorregressiva, e a sua saída pode ser mesmo um texto, mas cada *token* só pode olhar para aqueles que vieram antes dele na sequência. Como falamos antes, esse componente é o que remonta de fato ao propósito original de um modelo de linguagem computacional: gerar o próximo *token*, dados os *tokens* anteriores a ele, ou seja, gerar um texto. A família de arquiteturas GPT (Radford; Narasimhan, 2018) (do inglês, *Generative Pre-trained Transformer*, ou Transformer Gerativo Pré-treinado) usa blocos decodificadores da arquitetura Transformer para funcionar como um modelo autorregressivo de geração de texto. Claramente, sem o componente codificador, não temos o mecanismo de atenção tradicional como parte das entradas do decodificador, como acontece na arquitetura Transformer.

Em sua primeira versão, o GPT era bem similar ao componente decodificador do Transformer, sendo composto de 12 subcamadas de decodificadores com 12 *heads* de auto-atenção mascaradas de dimensão 768, e camadas escondidas completamente conectadas de 3072 dimensões. A tokenização também é baseada em subpalavras, mas segue o algoritmo BPE ao invés do WordPiece, como o BERT. A segunda versão, chamada criativamente de GPT-2 (Radford et al., 2019) veio em quatro versões de tamanhos variados: GPT-2 small, com 12 subcamadas de decodificadores e dimensão dos embeddings de 768, a versão GPT-2 medium, com 24 subcamadas de decodificadores e dimensão dos embeddings de 1024, a versão GPT-2 large, com 36 subcamadas de decodificadores e dimensão dos embeddings de 1280, e a versão GPT-2 *extra large*, com 48 subcamadas de decodificadores e dimensão dos embeddings de 1600. A camada de normalização passou a estar na entrada de cada subcamada e adicionou-se uma outra camada de normalização após o último bloco de auto-atenção. O código a seguir, que também usa o HuggingFace, apresenta exemplos de geração de texto usando o GPT2.

```
# modelo multilingual
model_name = "sberbank-ai/mGPT"
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)
model.cuda()
model.eval()

# geração de texto default
def cond_gen(tokenizer, model, prefix):
    # encode context the generation is conditioned on
    input_ids = tokenizer.encode(prefix, return_tensors='pt').cuda()

    # generate text until the output length (which includes the context
    # length reaches 50)
    greedy_output = model.generate(input_ids, max_length=50)
    return list(map(tokenizer.decode, greedy_output))[0]

# imprimir a saída do modelo
```



```

def print_output(output):
    print("Output:\n" + 100 * '-')
    print(output)

# usando beam search
def cond_gen_beam(tokenizer, model, prefix, ngram=1):
    input_ids = tokenizer.encode(prefix, return_tensors='tf')
    beam_output = model.generate(
        input_ids,
        max_length=50,
        num_beams=5,
        no_repeat_ngram_size=ngram,
        early_stopping=True
    )
    return beam_output[0]

# usando top-k sampling
def cond_gen_sample(tokenizer, model, prefix):
    input_ids = tokenizer.encode(prefix, return_tensors='tf')
    sample_output = model.generate(
        input_ids,
        do_sample=True,
        max_length=50,
        top_k=0
    )
    return sample_output[0]

# manipulando o parâmetro de temperatura
def cond_gen_sample_temp(tokenizer, model, prefix, temp=0.5):
    input_ids = tokenizer.encode(prefix, return_tensors='tf')
    sample_output = model.generate(
        input_ids,
        do_sample=True,
        max_length=50,
        top_k=0,
        temperature=temp
    )
    return sample_output[0]

prefix = 'Eu gosto de'
output = cond_gen(tokenizer_pt, model_pt, prefix)
print_output(output)

```

Output:

Eu gosto de fazer o que gosto, mas não sou muito de fazer o que não gosto.

```

output = cond_gen_beam(tokenizer, model, prefix)
print_output(output)

```

Output:

Eu gosto de pensar que a vida é muito curta, mas eu não posso viver sem ela.

Não importa o quanto você se preocupe com as coisas boas e ruins do mundo; ninguém pode ser fel



```
output = cond_gen_sample_temp(tokenizer, model, prefix, 0.7)
print_output(output)
```

Output:

```
-----
Eu gosto de ler, mas não sou leitora compulsiva.
Eu gosto de livros que me dão vontade de ter saudades, e quando eu
vejo uma resenha que me encanta, eu leio uma história
```

As mudanças arquiteturais da versão original para a versão 2 não foram profundas, exceto pelo tamanho e conseqüente quantidade de parâmetros treinados. Porém, no artigo do GPT-2 começou-se a vislumbrar um modelo mais geral, que pudesse executar várias tarefas (aprendizado de múltiplas tarefas, ou agnóstico de tarefas (Collobert; Weston, 2008)), mesmo sem ser treinado novamente para cada uma delas (configuração (Romera-Paredes; Torr, 2015)), e usando apenas a geração de texto como uma abstração de qualquer outra tarefa mais específica. O argumento era que o pré-treinamento em um conjunto grande e diverso de textos seria suficiente para que o modelo pudesse lidar com problemas com os quais não havia sido explicitamente treinado. Por exemplo, o artigo exemplifica que algumas sentenças nos textos usados para o pré-treinamento já eram exemplos de tradução de uma língua para a outra, o que faria o modelo aprender a traduzir naturalmente.

Este foi o principal motivador para o desenvolvimento da versão 3 da arquitetura GPT, chamada de GPT-3 (Brown et al., 2020). A ideia seria que durante o pre-treinamento (que vamos entender na próxima seção) o modelo consegue desenvolver indiretamente habilidades de geração de texto que poderiam ser usadas para resolver diversas tarefas, como tradução e resposta a perguntas, entre outras. Tais habilidades poderiam ser resgatadas em tempo de execução de acordo com a tarefa pedida, um processo chamado de *aprendizado em um contexto* (Dong et al., 2023). Três configurações foram discutidas no artigo, que já eram objeto de estudo de outros trabalhos voltados para o aprendizado a partir de poucos exemplos. A primeira configuração se chama *zero-shot* e explora cenários em que o modelo recebe como contexto uma descrição da tarefa (que até poderia ser opcional, dependendo da tarefa) e um *prompt*³², e espera-se que o modelo responda a partir destes dois componentes apenas, sem nenhum tipo de ajuste nos seus pesos. Ou seja, nenhum exemplo é dado para o modelo³³. Por exemplo, abaixo temos uma descrição e *prompt* para tradução automática³⁴.

```
Traduza de português para francês: # descrição da tarefa
penso, logo, existo: # prompt
---
je pense, donc je suis # saída do modelo
```

A segunda configuração chama-se *one-shot* e, neste caso, um exemplo completo é fornecido

³²Um *prompt* é um texto em linguagem natural (em oposição à linguagem de programação) que especifica uma instrução do que deve ser feito. Um *prompt* pode ser formulado como uma pergunta, uma observação, um questionamento, ou ainda representar uma tarefa específica, por exemplo, classificar um texto. *Prompts* podem ter um formato livre, mas modelos de linguagem são bastante sensíveis ao conteúdo textual dele. Afinal, modelos como GPT estão completando um texto (neste caso, o *prompt*), condicionados ao que tem neste texto.

³³Note que chamamos de exemplo, mas para quem está acostumado com aprendizado de máquina, este não seria um exemplo para ajustar os parâmetros do modelo. Seria apenas um exemplo mesmo.

³⁴Estamos usando exemplos em português, mas na época, as habilidades do modelo em lidar com várias línguas não foi demonstrada no artigo. Mas mencionou-se que no treinamento do GPT-2, os textos foram filtrados para que ficassem apenas aqueles em inglês. Para o GPT-3, esse requerimento foi relaxado e 7% dos textos estavam em outras línguas.



para o modelo como parte do contexto.

```
Traduza de português para francês: # descrição da tarefa
penso, logo, existo: je pense, donc je suis # exemplo fornecido
conhece-te a ti mesmo: # prompt
---
Connais-toi toi-même. # saída do modelo
```

Em um outro exemplo (bem hipotético, o GPT-3 não foi testado para tais habilidades em português), perceba que a descrição da tarefa e o exemplo podem estar juntos:

```
concordância incorreta em português: há bastante alunos reprovados
conjugação correta em português: há bastantes alunos reprovados # exm
conjugação incorreta em português: vejo muito alunos no corredor
conjugação correta em português: # prompt
---
vejo muitos alunos no corredor # saída do modelo
```

A terceira configuração chama-se *few-shot* e a diferença para as anteriores é apenas que mais de um exemplo são fornecidos para o modelo ter como base. Intuitivamente, tais exemplos, a descrição da tarefa e o *prompt* direcionarão o modelo para certos pesos que ativarão as distribuições de probabilidade de geração de textos para o ponto correto. É mesmo como se ele estivesse sempre completando textos que tenham alguma coerência com o que foi visto antes. Vamos lembrar que modelos gerativos são desenhados para prever o próximo *token* a partir dos *tokens* anteriores e é isso que está sendo feito aqui. No nosso último exemplo, *tokens* anteriores são desde “concordância incorreta ...” até “... correta em português:”, removendo os comentários (o que vem depois da tralha).

Em termos de arquitetura, foi usada a mesma do GPT-2, mas com uma alteração no mecanismo de atenção para fatorizar de forma esparsa as matrizes de atenção (Child et al., 2019) e reduzir a complexidade do mecanismo de atenção de $\mathcal{O}n^2$ para $\mathcal{O}n\sqrt{n}$ ³⁵. Foram treinados oito modelos de diferentes tamanhos, variando de 12 a 96 camadas e dimensão de embeddings de 768 a 12.288. O maior deles, com 175 bilhões de parâmetros foi o que obteve os melhores resultados, em geral, e que ganhou o privilégio de ser o GPT-3. O artigo também apontou as limitações correntes do GPT-3 e potenciais aplicações perigosas, que valem a pena a leitura.

A partir de 2022, a empresa que desenvolveu o GPT, chamada OpenAI³⁶ começou gradativamente a disponibilizar versões melhoradas do GPT-3, introduzindo novas formas de treinamento que se valem de cada vez mais textos e modelos cada vez maiores. Por exemplo, a empresa disponibilizou novas versões e interfaces de programação de aplicações (APIs) para os modelos que estenderam o GPT-3, chamados de “text-davinci-003” e “code-davinci-002”, que passaram a ser genericamente chamados de GPT-3.5. O GPT-3.5 é a base para o famoso ChatGPT³⁷, veja mais no Capítulo 20, um agente de conversação genérico e que foi disponibilizado para o mundo testar no finzinho de 2022. Na época da escrita deste capítulo, a versão mais nova do GPT era o GPT-4.

Um outro modelo interessante e que também se vale da geração autorregressiva dos decodificadores é o XLNet (Yang et al., 2019). Ele melhora a modelagem autorregressiva, que tem como desvantagem não ter um olhar bidirecional, ao considerar todas as possíveis

³⁵A notação de Big O descreve complexidade de um código em termos algébricos (Cormen et al., 2001; Knuth, 1997).

³⁶<https://openai.com/blog/chatgpt>

³⁷<https://chat.openai.com/>



permutações de uma sequência durante o aprendizado. Ele também usa um esquema de codificação posicional relativo, proposto na arquitetura Transformer-XL (Dai et al., 2019), mas com algumas reparametrizações com foco em remoção de ambiguidade. Assim como ocorre com as arquiteturas codificadoras, muitas outras abordagens têm sido propostas a todo momento. Algumas delas voltarão ao nosso radar quando falarmos das tendências correntes da área na Seção 15.4.

15.3.2.2.10 Instâncias de um Transformer inteiro: T5 e seus aliados

Os Transformers também podem ser utilizados ou instanciados com seus dois componentes, o codificador e o decodificador. A desvantagem é que estes modelos costumam ser maiores e precisam de mais dados para serem treinados. A vantagem é que o modelo tanto terá os benefícios da codificação de atenção bidirecional, como de ter pronto um modelo de linguagem para geração de texto. Mas um dos grandes benefícios é permitir que os problemas sejam tratados sempre do ponto de vista da geração de textos, permitindo que um mesmo modelo possa resolver várias tarefas. Ou seja, se queremos fazer tradução, vamos receber um texto inicial e completá-lo com a tradução em outra língua. Se queremos responder a consultas, a pergunta é a entrada e a resposta é a saída. E por aí vai. E o uso de instruções ajudam o modelo a completar o texto da maneira apropriada para resolver a tarefa específica.

Uma arquitetura que desempenha esse papel de ser um consumidor e produtor de texto (*text-to-text*) é o T5 (Raffel et al., 2020). A arquitetura do T5 segue os Transformers, mas o *embedding* de posição é relativo, seguindo o deslocamento de posição entre as matrizes de chave e de consulta que estão sendo comparadas. Outros pontos de mudança são a remoção do valor de viés da camada de normalização e a inclusão da camada de normalização após o caminho residual. O modelo foi avaliado com diversos conjuntos de dados para tarefas variadas, incluindo análise de sentimentos, similaridade de sentenças, desambiguação de palavras, resolução de correferências, entre outras. Assim como já falamos no GPT, para permitir que diversas tarefas pudessem ser abordadas pelo mesmo modelo, os autores usaram uma instrução limitada, que funcionava quase como um hiperparâmetro (no artigo, chama-se **prefixo específico de tarefa**).

Um outro modelo também bastante utilizado e discutido na literatura é o BART (*Bidirectional and Auto-Regressive Transformers*) (Lewis et al., 2020a)³⁸ Seu objetivo é mapear um texto com ruídos (corrompido) para o texto original, uma tarefa que lembra reconstrução de imagens. Duas arquiteturas são apresentadas no artigo: uma com seis camadas de codificadores e decodificadores cada e outra com 12.

A exemplo do que já vimos antes, o T5 também foi treinado para lidar com português, dando origem ao modelo denominado de PTT5 (Carmo et al., 2020). O modelo foi avaliado em uma base de similaridade de sentenças (ASSIN 2) e de reconhecimento de entidades nomeadas (HAREM). Observe que o modelo é limitado às instruções dessas duas tarefas. Além do modelo monolingual, temos a versão multilingual do T5, o mT5 (Xue et al., 2021), treinada por pesquisadores da Google com o *corpus* mC4, uma versão do *corpus* Common Crawl para 101 idiomas, incluindo o português. E temos ainda a versão multilingual do BART (Liu et al., 2020), treinado inicialmente com um subconjunto do Common Crawl para 25 idiomas (mBART) que não incluiu o português, mas sua extensão para 50 idiomas (mBART-50 (Tang et al., 2020)), sim.

³⁸Afinal, além de Sesame Street, também gostamos dos Simpsons.



15.3.2.2.11 *Embeddings* de Sentenças com Transformers

Podemos recuperar embeddings contextualizados tanto para *tokens*, como para combinações de *tokens*, o que inclui sentenças e textos. Quando uma combinação de *tokens* resulta em uma palavra, estamos falando de um *embedding* de palavra. Quando recuperamos os embeddings de uma frase, estamos falando de um *embedding* de sentença.

Modelos baseados em Transformers permitem recuperar embeddings de sentenças basicamente de duas formas: construindo os embeddings a partir das médias dos embeddings de cada *token* na sentença, usando uma ou mais camadas da arquitetura (em geral, usamos as quatro últimas camadas), ou usando os embeddings de saída do primeiro *token*, o [CLS]. Perceba que a agregação por média não é o mesmo processo discutido no Capítulo 10 para obter embeddings de sentenças a partir de representações estáticas. Com Transformers, além dos *tokens* influenciarem uns aos outros com o mecanismo de atenção, também temos o codificador de posição, que influenciará na saída final. Uma outra possibilidade é treinar modelos que consigam devolver embeddings de sentenças de entrada, o que é feito costumeiramente tendo como base a tarefa de similaridade semântica e arquiteturas siamesas, como no modelo *sentence-transformers*³⁹ (Reimers; Gurevych, 2019, 2020), ou treinamentos contrastivos, como a abordagem SIM-CSE (Gao; Yao; Chen, 2021).

Mas como todos esses modelos podem ser usados, treinados, aprendidos, refinados? É o que vamos discutir a seguir.

15.3.3 Treinamento e Ajustes em Modelos de Linguagem Neurais

Na seção anterior, vimos como funcionam os modelos de linguagem neurais, além de abordarmos diversos modelos em português, ressaltando detalhes de suas arquiteturas e *corpora* usado para treinar os pesos iniciais de cada um dos modelos. Esse passo inicial de treinamento, que é parte do modelo disponibilizado em arcabouços como TensorFlow e HuggingFace, é chamado de **pré-treinamento**. O pré-treinamento (ou *training from scratch*, ou treinamento do zero) refere-se a uma técnica de treinamento de redes neurais profundas (*deep neural network*) que no caso de modelos de linguagem usa uma quantidade expressiva de textos sem nenhum rótulo ou anotação, com o intuito de gerar um modelo de propósito geral capaz de “entender” linguagem.

O fato dos textos não terem rótulos ou anotações é o que permite usar uma quantidade enorme de textos, pois sabemos que anotar exemplos é uma tarefa custosa e que requer um tempo precioso de especialistas. Ainda assim, o pré-treinamento de um modelo de linguagem é uma tarefa desafiadora em muitos aspectos, incluindo a necessidade do uso intensivo de uma quantidade significativa de recursos computacionais por longos períodos de tempo. Além do alto custo envolvido no pré-treinamento desses modelos, ainda é preciso levar em consideração os impactos ambientais resultantes do alto consumo de energia.

De modo geral, o pré-treinamento de um modelo de linguagem engloba os seguintes passos:

1. **Escolher *corpora*** (Capítulo 14): a escolha dos *corpora* ideais é um passo importante no treinamento de modelos de linguagem. Mas quais seriam as características desses *corpora* ideais? Essa é uma pergunta difícil de responder, uma vez que a escolha dos *corpora* vai depender muito dos objetivos finais desse modelo, além é claro, da disponibilidade de tais *corpora*. A escolha dos *corpora* pode ser baseada no domínio a

³⁹<https://www.sbert.net/>



ser explorado. Por exemplo, um modelo pré-treinado com textos do Twitter, pode ser mais equipado para resolver tarefas que envolvam textos com uma linguagem mais informal. A escolha também pode basear-se na língua. O objetivo pode ser treinar um modelo monolingual ou até mesmo multilingual. Um outra possibilidade, muito adotada com os modelos de linguagem grandes, é usar *corpora* bem variados, formados por *corpus* de diferentes domínios e línguas. Veja mais sobre as particularidades e requerimentos da criação e escolha de *corpora* no Capítulo 14.

2. **Limpar e pré-processar os textos:** embora modelos de linguagem neurais não precisem de muitos passos de limpeza e pré-processamento, como costumava ser feito para o treinamento de modelos de aprendizado de máquina anteriores, ainda é necessário executar uma normalização dos textos, ainda que simples. Tal passo inclui remover caracteres especiais, remover URLs e remover textos que tenham apenas poucos caracteres. Como muitos dos textos são coletados da Web, também costuma-se remover etiquetas HTML, para que estas não sejam confundidas com palavras ou caracteres importantes.
3. **Treinar o tokenizador** (Capítulo 4): como vimos no Capítulo 4, tokenização é o processo de dividir o texto em unidades menores, chamadas de *tokens*. Esse é um passo importante no treinamento de modelos de linguagem, podendo impactar o desempenho final de tais modelos. Neste passo, podemos optar em usar um tokenizador pré-treinado, como por exemplo o tokenizador do GPT-3, ou podemos treinar um tokenizador do zero, assim como fazemos com os modelos de linguagem. O treinamento do tokenizador também requer a escolha de *corpora*. Neste caso, podemos usar os mesmos *corpora* escolhidos para treinar os pesos do modelo de linguagem. Além da escolha dos *corpora*, é necessário definir o tipo de tokenizador a ser treinado. Escolhas populares são o *Byte-pair encoding* (BPE) (Provilkov; Emelianenko; Voita, 2020), usados por modelos como o GPT-3, e o WordPiece (Schuster; Nakajima, 2012), usado por modelos como o BERT. Ambos dividem as palavras em sub-palavras, para acomodar melhor palavras que não tenham aparecido durante o treinamento do tokenizador. Aliás, treinamento aqui é justamente definir como as palavras serão “quebradas” (ou não) para definir os *tokens* do modelo. Veja mais sobre o processo de tokenização no Capítulo 4.
4. **Definir a arquitetura do modelo:** a escolha da arquitetura adotada para treinar um modelo de linguagem depende de muitos fatores, entre eles a disponibilidade de recursos de alto-desempenho para treinamento do modelo. A arquitetura pode ser uma rede neural recorrente ou uma rede neural baseada em Transformers. A arquitetura envolve quais serão os componentes em termos de camadas, *heads* e funções de ativação. Como podemos ter uma quantidade combinatória de tipos de componentes, costuma-se usar uma arquitetura pré-definida, como BERT ou GPT. Mas nada impede de uma pessoa definir a sua arquitetura do zero.
5. **Definir a função objetivo ou tarefa intermediária:** a tarefa intermediária é responsável por guiar o aprendizado do modelo. Algumas das tarefas mais utilizadas em modelos atuais serão exploradas na Seção 15.3.3.1.
6. **Definir os hiperparâmetros:** nas seções anteriores, falamos muito em “matriz de pesos” e “pesos da camada de atenção”; esses pesos são considerados parâmetros do modelo e são aprendidos com auxílio dos dados de treinamento. Já os hiperparâmetros são parâmetros que ajudam a controlar o processo de treinamento. Eles podem influenciar na qualidade final do modelo, como também na velocidade do treinamento. Esses são alguns dos hiperparâmetros mais comuns usados para o treinamento e



ajuste de modelos de linguagem:

- Taxa de aprendizagem (*learning rate*): a taxa de aprendizagem está relacionada ao algoritmo de otimização usado para atualizar os pesos do modelo a partir dos dados de treinamento. A grosso modo, a medida que os dados de treinamento circulam pela rede, os pesos do modelo são atualizados até que se alcancem pesos ideais. A taxa de aprendizagem controla o tamanho dessas atualizações e, conseqüentemente, afeta diretamente a convergência do modelo e o tempo de convergência.
- *batch size*: número de amostras dos dados de treinamento, ou seja, número de seqüências de texto, que são processadas ao mesmo tempo antes de cada atualização dos pesos do modelo. O valor ideal do *batch size* vai depender da arquitetura e tarefa alvo. É importante ressaltar que quanto maior o *batch size*, maior o consumo de memória, o que pode tornar o treinamento proibitivo em muitos casos.
- Número de épocas (*number of epochs*): é o número total de vezes que todos os exemplos de treinamento passarão pelo modelo.
- Taxa de regularização (*dropout rate*): é usada para controlar o problema de sobreajuste, ou seja, evitar que o modelo se ajuste perfeitamente ao conjunto de treinamento perdendo sua capacidade de generalização na presença de novos dados.

7. **Avaliação:** após o treinamento, é importante avaliar a qualidade e coerência dos textos gerados pelo modelo. Um modelo também pode ser avaliado em relação ao seu desempenho na realização de uma tarefa final de PLN, como as tarefas de sumarização e análise de sentimentos. Veremos mais detalhes na Seção 15.3.4.

O pré-treinamento considera um objetivo genérico, de geração ou preenchimento de texto, que não requer nenhuma anotação por parte de especialistas. Entretanto, este modelo genérico pode ser ainda melhorado tendo em vista uma tarefa final. Assim, modelos pré-treinados podem ser ajustados de acordo com um domínio ou uma tarefa específica, o que chamamos de **continuado ou ajuste fino** (*fine-tuning*), como veremos na Seção 15.3.3.2.

Resumindo, o treinamento de um modelo envolve a seleção dos *corpora*, o pré-processamento e limpeza desses dados, a seleção de uma arquitetura e tarefa intermediária, o treinamento em si e a avaliação do modelo pré-treinado. Em seguida, visitaremos algumas das tarefas intermediárias mais populares.

15.3.3.1 Tarefa Intermediária para o Pré-treinamento

Durante o pré-treinamento de um modelo de linguagem, uma ou mais funções objetivo ou tarefas intermediárias são utilizadas para guiar o aprendizado do modelo a gerar texto, ou, de forma mais genérica, a predizer partes do texto que estejam faltando. O intuito é que o modelo passe a ter uma compreensão estatística da(s) língua(s) em que foi treinado. Vários objetivos foram propostos na literatura, alguns a nível de *token* e outros a nível da sentença. Todos eles têm em comum o intuito de se basearem em uma tarefa de treinamento auto-supervisionada, ou seja, em que as saídas dos exemplos podem ser geradas de forma automática. Discutiremos alguns deles a seguir, começando pelas duas tarefas mais amplamente adotadas na literatura:

1. **Modelagem de linguagem mascarada (em inglês, *Masked Language Modeling* (MLM))** (Devlin et al., 2019): esta tarefa é inspirada no teste



Cloze (Taylor, 1953)⁴⁰ e foi proposta para treinar modelos bidirecionais, como o BERT. Neste caso, os textos de entrada são alterados para que em cada uma das sequências, uma porcentagem dos *tokens* seja substituída pelo *token* especial [MASK]. O objetivo é estimar os *tokens* mascarados levando em consideração o contexto dos demais *tokens* da sequência. Por exemplo, suponha a sentença mascarada do Exemplo 15.5, em que a original é atribuída a Sêneca:

Exemplo 15.5.

Apressa-te a viver [MASK] e pensa que cada [MASK] é, por si [MASK], uma vida.

O objetivo do modelo seria encontrar as palavras mais adequadas para entrar no lugar de [MASK]⁴¹. Perceba que o modelo poderia encontrar palavras adequadas diferentes das originais, mas que ainda seriam plausíveis. Por exemplo, a primeira máscara poderia ser substituída por “muito”, embora no texto original (ao menos na versão traduzida para o português, a palavra seja “bem”. Por isso, avaliar o resultado de modelos de linguagem com tarefas de predição de texto é tão complexo.

2. **Modelagem de linguagem causal ou autorregressiva (em inglês, *Casual Language Modeling (CLM)*):** Esta é a tarefa que mais se assemelha à tarefa de modelagem de linguagem como definimos no início deste capítulo, ou seja, o objetivo é completar o próximo *token* em uma sequência considerando apenas os *tokens* anteriores. Diferente da tarefa anterior, em que a sequência é vista como um todo, com apenas as posições com máscara faltando, aqui o modelo só pode atender aos *tokens* da esquerda, diferentemente dos modelos bidirecionais como o BERT.

As duas tarefas acima são as mais comumente empregadas como tarefas intermediárias na literatura. Entretanto, outras também já foram exploradas:

1. *Replaced Token Detection (RTD)* (Clark et al., 2020): quando falamos de MLM, vimos que a entrada do modelo é corrompida pela substituição de *tokens* originais da sentença, pelo *token* especial [MASK]. No caso do RTD, um gerador, que pode ser um modelo de linguagem menor, é utilizado para gerar *tokens* ambíguos que serão usados no lugar do *token* [MASK]. Esses *tokens* ambíguos, embora incorretos, são próximos do significado semântico do *token* original. Agora, ao invés de ter que prever o *token* mascarado, como ocorre quando usamos a MLM, o objetivo é identificar se um *token* é o *token* original da sentença de entrada ou se ele é um *token* gerado pelo gerador. Um exemplo seria: dada a sentença original “A professora ensinou o novo conteúdo”, uma sentença após o RTD poderia ser “A professora aprendeu o novo conteúdo”;
2. *Shuffled Word Detection (Shuffle)* (Yamaguchi et al., 2021): nesta tarefa, uma porcentagem dos *tokens* de entrada são aleatoriamente embaralhados antes de serem processados pelo modelo. O objetivo do modelo é identificar dentre os *tokens* da sequência de entrada, aqueles que foram inicialmente embaralhados. Considerando a sentença “O gato sentou no tapete da sala”, uma sentença embaralhada seria “O gato no tapete da sala sentou”;

⁴⁰O teste Cloze, segundo o dicionário Cambridge, consiste em um texto com algumas palavras removidas, e o leitor deve preencher tais palavras faltantes com algo que faça sentido. Ele foi proposto para medir o nível de leitura de um indivíduo.

⁴¹Em tradução livre, a sentença original é “Apressa-te a viver bem e pensa que cada dia é, por si só, uma vida”.



3. *Token Order Permutations*: é a tarefa utilizada para treinar o modelo XLNet (Yang et al., 2019). Como nos modelos de linguagem autorregressivos, o objetivo é prever um *token* com base no contexto dos *tokens* anteriores, só que agora, a probabilidade de um *token* é condicionada a todas as permutações de *tokens* em uma sequência. Assim, o modelo consegue aprender o contexto de forma bidirecional, mas sem se restringir à ordem original da sequência, como nos modelos baseados no BERT. Na teoria, são geradas todas as sentenças possíveis a partir da permutação dos *tokens* da sentença original. Na prática, apenas uma amostra dessas sentenças permutadas são usadas durante o treinamento. Exemplos de sentenças seriam “Eu amo chocolate / amo eu chocolate / amo chocolate eu / chocolate eu amo / chocolate amo eu etc.”;
4. *Next Sentence Prediction* (NSP): é uma função objetivo que foi usada para treinar o modelo BERT em conjunto com a função MLM. O objetivo da NSP é aprender a relação entre duas sentenças, ou seja, se elas são sentenças contíguas ou não. Exemplos positivos são criados através da extração de sentenças consecutivas presentes nos *corpora* usados para treinar o modelo. Já os exemplos negativos são criados através do pareamento de duas sentenças oriundas de diferentes documentos dos *corpora*. Alguns estudos (Joshi et al., 2020; Liu et al., 2021) mostraram que NSP não funciona bem, ou é desnecessária para algumas tarefas. Por essa razão, modelos como o RoBERTa (Liu et al., 2021), removeram NSP do seu pré-treinamento;
5. *Sentence-Order Prediction* (SOP): essa tarefa tenta estimar se duas sentenças consecutivas estão na ordem correta ou não, ou seja, se elas tiveram ou não sua ordem invertida (Lan et al., 2020). Ao contrário da tarefa NSP, que cria exemplos negativos através da concatenação de sentenças extraídas de documentos diferentes, na SOP os exemplos negativos são criados usando duas sentenças consecutivas extraídas do mesmo documento, só que agora elas terão suas ordens invertidas. Os exemplos positivos são criados usando a mesma técnica adotada por NSP. Essa pequena alteração na construção dos exemplos negativos força o modelo a fazer uma distinção mais refinada com relação a ordem e coerência das sentenças.
6. *Translation language modeling* (TLM): foi proposto em (Conneau; Lample, 2019) e utilizado em conjunto com as funções objetivo MLM e CLM para treinar o modelo XLM. A TLM é uma extensão da MLM, uma vez que também usa o *token* especial [MASK] para mascarar *tokens* da sequência original. Só que agora, ao invés de usar sequências na mesma língua, o modelo XLM concatena duas sentenças de línguas diferentes, como por exemplo uma sentença em inglês e outra em português. Depois, *tokens* das duas sequências concatenadas são aleatoriamente substituídos pelo *token* [MASK]. Para prever um *token* mascarado na sentença em português, o modelo pode atender (mecanismos de atenção, Seção 15.3.2.2 tanto a outros *tokens* da sentença em português quanto a *tokens* da sequência em inglês).

A escolha de funções objetivo não é o único desafio para o pré-treinamento de modelos de linguagem. Outro fator relevante e com grande impacto na capacidade e qualidade final do modelo é a escolha dos *corpora*. Modelos de linguagem devem ser treinados com uma grande quantidade de dados de alta qualidade. Mesmo que não seja necessário anotar os dados, montar essas grandes coleções de dados deve ser um tarefa cuidadosa, ainda que exaustiva e demorada. O ideal é garantir que esses *corpora* sejam o mais diversos possível e sem enviesamentos, polarização e textos maliciosos, o que requer um grande esforço de filtragem e pré-processamento dos dados. Hoje em dia, vários esforços são feitos no sentido de minimizar os efeitos do uso de *corpora* contendo textos maliciosos. Por exemplo, técnicas



como treinamento adversarial (Kianpour; Wen, 2020) são utilizadas para expor os modelos a textos maliciosos com o intuito de ensinar esses modelos a reconhecer tais tipos de textos. Outra técnica que tem se tornado frequente, é o uso de humanos para revisar e moderar o texto gerado por modelos de língua. Assim, esse tipo de informação pode ser utilizada para melhorar o modelo de forma iterativa.

Apesar de os modelos de linguagem serem treinados usando coleções vastas e diversas de textos, eles podem se tornar obsoletos, uma vez que essas coleções são estáticas. Com o tempo, o modelo pode não ser capaz de gerar e reconhecer textos sobre eventos atuais. Por exemplo, um modelo treinado com textos anteriores a Setembro de 2022 pode não ser capaz de reconhecer que o atual monarca da Inglaterra é o Rei Charles III, uma vez que sua mãe, a Rainha Elizabeth II, faleceu em Setembro de 2022. Além disso, dada a variedade de domínios existentes e a forma dinâmica como novas tendências e culturas emergem ao longo dos anos, é muito difícil garantir que um modelo de linguagem será capaz de entender e resolver de forma precisa as mais diversas tarefas do PLN. Na seção seguinte discutiremos formas de usar novas coleções de dados para ajustar um modelo a tarefas e domínios específicos.

15.3.3.2 Ajustes em Modelos de Linguagem Neurais

Uma das formas encontradas para atualizar modelos de linguagem é o que chamamos de treinamento continuado (em inglês, *continued pre-training*) (Gururangan et al., 2020; Jin et al., 2022; Ke et al., 2023).

No treinamento continuado, o modelo é treinado por mais algumas iterações ou épocas usando uma coleção de textos diferente dos *corpora* utilizados no pré-treinamento, mas mantendo a mesma tarefa intermediária. Ou seja, o treinamento continuado, assim como o pré-treinamento, é um processo de treinamento auto-supervisionado. Tradicionalmente, o treinamento continuado pode ser dividido em dois tipos: o treinamento continuado com foco na adaptação da tarefa (*Task Adaptive Pre-Training*, TAPT) e o treinamento continuado com foco na adaptação do domínio (*Domain Adaptive Pre-Training*, DAPT). No caso do TAPT, o treinamento continuado ocorre com a utilização de uma coleção de textos não-rotulados relacionados a uma tarefa específica, por exemplo a tarefa de análise de sentimentos. A coleção não precisa ser grande, mas precisa representar bem diferentes aspectos da tarefa alvo. Já no DAPT, o foco não é a tarefa, mas sim o domínio. Neste caso, o modelo é treinado por mais algum tempo utilizando uma coleção de textos que tratam de algum domínio específico. Por exemplo, um domínio pode ser a biomedicina ou até mesmo artigos científicos sobre inteligência artificial. Mais recentemente, foi proposto o treinamento continuado baseado em instruções (*Prompt-based Continued Pre-training*, PCP), que seria uma combinação do treinamento continuado tradicional (TAPT) com o ajuste de instruções (*instruction tuning*) (Shi; Lipani, 2023) (veja mais na Seção 15.4. Assim como no TAPT e DAPT, no PCP a função objetivo original, ou tarefa intermediária, é utilizada durante o treinamento continuado. Mas neste caso, teremos dois tipos de entrada: os textos não-rotulados relacionados a tarefa alvo, como no TAPT; e, os *prompts* ou instruções também relacionadas a tarefa alvo.

O treinamento continuado é um dos métodos utilizados para adaptar um modelo pré-treinado a alguma tarefa (TAPT) ou domínio específico (DAPT). Outro método que também permite a adaptação de modelos é o método do *ajuste fino* (*fine-tuning*) (Howard; Ruder, 2018). Enquanto o treinamento continuado não requer textos rotulados e usa a mesma tarefa intermediária adotada durante o pré-treinamento do modelo, no ajuste fino



usamos textos rotulados e uma função objetivo específica da tarefa alvo, por exemplo, a tarefa de classificação. Os dois métodos resultam no ajuste dos pesos do modelos. Entretanto, por ser mais específico e focar totalmente na tarefa alvo, através de dados rotulados e o uso de uma função objetivo específica, o ajuste fino costuma requerer menos dados para promover o ajuste dos pesos do modelo pré-treinado, além de resultar em um modelo altamente ajustado ao contexto da tarefa final. Com isso, podemos dizer que o ajuste fino resulta em um tempo de treinamento menor do que o treinamento continuado, o que também vai impactar no custo final de geração do modelo.

Se pensarmos no treinamento dos modelos de linguagem como um processo que pode ocorrer em duas etapas, o pré-treinamento do modelo seria a primeira etapa e o treinamento continuado e/ou ajuste fino, seria a segunda etapa. Nessa primeira etapa, o modelo é treinado depois de serem definidas a arquitetura e a tarefa intermediária, além da seleção e pré-processamento de grandes coleções de textos a serem utilizados no aprendizado. Já na segunda etapa, os pesos do modelo são ajustados para um domínio e/ou tarefa específica. Apesar da possibilidade de um ajuste dos modelos pré-treinados, essa etapa de ajuste não é obrigatória. Tanto o modelo pré-treinado, como o modelo ajustado podem ser utilizados em diversas tarefas de PLN.

Um exemplo de tarefa é a análise de sentimentos. Aqui, vamos considerar a tarefa de análise de sentimentos como um problema de classificação binária com dois sentimentos possíveis: positivo e negativo. Dada uma coleção de treinamento composta por sentenças rotuladas, o objetivo final é treinar um classificador capaz de classificar novas sentenças em um desses dois sentimentos, positivo ou negativo. Um exemplo de sentença rotulada seria: “Maria gostou muito do computador”, sentimento positivo. Neste caso, podemos usar um modelo de linguagem pré-treinado para gerar representações vetoriais dessas sentenças, os *embeddings*. Dados os embeddings e os rótulos, podemos usar qualquer algoritmo de classificação, como máquina de vetores de suporte (SVM, *Support-Vector Machine*) (Cortes; Vapnik, 1995) ou regressão logística (Tolles; Meurer, 2016), para treinar um classificador capaz de categorizar novas sentenças não rotuladas em um dos dois sentimentos. A extração de *features* (*feature extraction*) ou embeddings, pode ser feita usando tanto um modelo pré-treinado, como também um modelo ajustado.

Ainda considerando a tarefa de análise de sentimentos, poderíamos ajustar um modelo de linguagem de diversas formas. No caso do TAPT, poderíamos usar uma coleção de dados de análise de sentimentos sem a necessidade dos rótulos. No caso do DAPT, precisaríamos de uma coleção de dados associada ao domínio em questão, mas também sem a necessidade dos rótulos. Por exemplo, se a tarefa é analisar o sentimento dos consumidores em relação a marcas de carros, poderíamos então usar no ajuste uma coleção de dados contendo opiniões de consumidores sobre marcas de carros. Note que aqui, o foco não é a tarefa de análise de sentimento, mas sim o domínio. Outro tipo de ajuste possível seria o ajuste fino. Neste caso, usaríamos um coleção de dados de análise de sentimentos para opiniões de consumidores sobre marcas de carros. No ajuste fino, a coleção precisa ser rotulada, uma vez que o modelo de linguagem será ajustado usando a tarefa final. Como estamos tratando a análise de sentimentos como uma tarefa de classificação binária, a tarefa final usada nos ajustes é a tarefa de classificação. Para realizar o ajuste, podemos adicionar uma camada de classificação à arquitetura do modelo e então ajustar os pesos do modelo usando os textos da coleção rotulada.



15.3.4 Avaliação de Modelos de Linguagem Neurais

Com o crescente número de modelos de linguagem disponíveis, é bem desafiante decidir qual a melhor maneira de avaliar a qualidade ou capacidade desses modelos. Tradicionalmente, modelos de linguagem são avaliados por métricas como perplexidade (*perplexity*), entropia cruzada (*cross-entropy*) e bits-por-caracter (*bits-per-character*, BPC). Esse tipo de avaliação é comumente chamada de avaliação intrínseca, com o modelo sendo avaliado através do seu desempenho na tarefa intermediária. No caso dos modelos de linguagem, a tarefa intermediária é prever o próximo *token* de uma sequência. Uma outra forma de avaliar modelos de linguagem é aplicá-los diretamente na resolução de uma tarefa final e então avaliar o quanto a qualidade da solução melhorou. Por exemplo, se estamos na dúvida entre adotar o modelo A ou o modelo B para resolver uma tarefa de classificação ou uma tarefa de reconhecimento de voz, podemos aplicar os dois modelos, A e B, e então medir qual das duas soluções produziu os melhores resultados. Esse tipo de avaliação é chamada de avaliação extrínseca. Apesar da avaliação extrínseca ser considerada a melhor maneira de avaliar a capacidade de um modelo de linguagem em resolver uma tarefa específica, ele é um processo de alto custo e que envolve longos tempos de execução.

Como foi dito no parágrafo anterior, a avaliação intrínseca não depende de nenhuma tarefa final específica, ela considera apenas a qualidade do modelo na geração do próximo *token* da sequência. Para entender os conceitos de perplexidade, entropia cruzada e bits-por-caracter, precisamos primeiro falar de entropia. A ideia de entropia foi proposta em 1951 por C. E. Shannon (Shannon, 1951) para medir a quantidade média de informação que é transmitida por cada letra de um texto. Shannon também definiu entropia da seguinte forma: “Se a linguagem for traduzida em dígitos binários (0 e 1) da forma mais eficiente, a entropia é o número médio de dígitos binários necessários por letra da linguagem original”. No contexto de linguagem, entropia é a quantidade de informação contida em um caractere em uma sequência de texto infinita. A entropia (H) é definida como:

$$H = - \sum p(i) \log(p(i))$$

onde i é o próximo *token* a ser gerado pelo modelo e $p(i)$ é a probabilidade do *token* i ser escolhido como o próximo *token* da sequência, dados os *tokens* anteriores. Podemos dizer que, se um modelo captura bem a estrutura de uma língua, consequentemente a entropia do modelo deve ser baixa.

Nós vimos na Seção 15.2, Equação 15.3, que a tarefa de completar uma sequência de palavras com uma próxima palavra é definida por uma distribuição de probabilidade condicional das palavras que poderiam completar a sequência, dadas as palavras que vieram antes na sequência. Assim sendo, o modelo de língua tem como objetivo aprender uma distribuição Q , a partir de uma amostra de texto, que seja próxima da distribuição P , que é a distribuição empírica da língua. Para medir o quão próximas são essas duas distribuições, muitas vezes usamos a entropia cruzada, definida como:

$$H(P, Q) = - \sum_i P(i) \log Q(i) = H(P) + D_{KL}(P \parallel Q)$$

onde $H(P)$ é a entropia da distribuição empírica P e $D_{KL}(P \parallel Q)$ é a divergência de Kullback-Leibler de Q para P , ou seja, a entropia relativa de P com relação a Q . A divergência de Kullback-Leibler (Joyce, 2011) é uma medida estatística que, neste caso, mede o quão diferente é a distribuição de probabilidade Q da distribuição de probabilidade de referência P .



O conceito de perplexidade está totalmente relacionado ao conceito de entropia e entropia cruzada. A perplexidade é entendida como uma medida de incerteza e é definida como a exponencial da entropia cruzada:

$$PPL(P, Q) = 2^H(P, Q)$$

Teoricamente, quanto menor a perplexidade, melhor o desempenho do modelo em prever o próximo *token* da sequência.

Também seguindo a linha da entropia, temos a métrica bits-por-caracter que mede o número médio de bits necessários para representar um caracter. Ou seja, seguindo a definição de entropia dada por Shannon, podemos dizer que a entropia é o número médio de BPC.

Até aqui, temos falado muito em “número médio de bits” e entropia a nível de caractere. Mas quando revisitamos as seções anteriores, notamos que dependendo do tokenizador adotado pelo modelo de língua, o texto de entrada pode ser quebrado em palavras, sub-palavras e até caracteres. Sendo assim, sempre que vamos comparar modelos de linguagem diferentes, é importante atentar para o tipo de tokenização usada pelo modelo e então, ajustar as métricas de acordo. Outro detalhe importante que precisa ser observado, é o tamanho máximo de contexto permitido por um modelo de linguagem, uma vez que, em geral, modelos de linguagem com comprimento de contexto mais longo costumam ter um valor de entropia cruzada menor quando comparado com modelos com comprimento de contexto menores.

Outra forma de avaliar e comparar diferentes modelos de linguagem, é através do uso de *benchmarks*, conforme discutido de forma extensiva no Capítulo 14. *Benchmarks* para modelos de linguagem são conjuntos de dados referentes a várias tarefas linguísticas que ajudam a avaliar a capacidade dos modelos no entendimento e geração de texto. O uso de *benchmarks* permite uma padronização com relação aos dados e métricas, o que é fundamental para que experimentos possam ser replicados e comparados em diferentes estudos. Além disso, o uso de *benchmarks* permite o monitoramento da evolução dos modelos com o passar do tempo. Entre os *benchmarks* mais populares está o GLUE (*General Language Understanding Evaluation*)⁴² (Wang et al., 2018) e o SuperGLUE⁴³ (Wang et al., 2019), ambos focados na língua inglesa. Para a língua portuguesa, temos o Poeta (*Portuguese Evaluation Tasks*), que inclui 14 bases de dados de tarefas finais, incluindo similaridade textual, análise de sentimentos, perguntas e respostas, entre outros. Apesar dos benefícios trazidos pelo uso de *benchmarks*, é preciso ficar atento a possíveis limitações, como a existência de vieses nas coleções de dados e a falta de representatividade e diversidade nos textos, o que pode impactar a generalização dos resultados dos modelos, além da escassez de coleções multilíngues. Dentre as métricas disponíveis no GLUE e SuperGLUE estão a acurácia, o F1-score e o Coeficiente de Correlação de Matthews (em inglês, *Matthews Correlation Coefficient* ou MCC).

No contexto de tarefas de classificação, a acurácia é a fração de previsões que o modelo acertou e pode ser definida como:

$$\text{Acurácia} = \frac{vp + vn}{vp + vn + fp + fn}$$

onde *vp* (verdadeiro positivo) é o número de amostras positivas que foram classificadas corretamente, *vn* (verdadeiro negativo) é o número de amostras negativas que foram

⁴²<https://gluebenchmark.com/>

⁴³<https://super.gluebenchmark.com/>



classificadas corretamente, fp (falso positivo) é o número de amostras negativas que foram classificadas como positivas e fn (falso negativo) é o número de amostras positivas que foram classificadas como negativas.

O F1-score é a média harmônica da precisão e revocação, podendo ser definida como:

$$\text{F1-score} = 2 \frac{\text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}}$$

A precisão e revocação podem ser definidas como:

$$\begin{aligned} \text{precisão} &= \frac{vp}{vp + fp} \\ \text{revocação} &= \frac{vp}{vp + fn} \end{aligned}$$

Embora as fórmulas estejam focadas em tarefa binárias, considerando exemplos positivos e negativos, é possível generalizá-las para qualquer quantidade de classes. De forma similar, é sempre possível reduzir uma tarefa com qualquer quantidade de classes para uma avaliação binária.

O Coeficiente de Correlação de Matthews (*Matthews Correlation Coefficient*, MCC) é outra métrica que baseia-se nos números de verdadeiro positivo (vp), verdadeiro negativo (vn), falso positivo (fp) e falso negativo (fn). MCC foi proposta com a classificação binária em mente (Matthews, 1975) e pode ser definida como:

$$\text{MCC} = \frac{vp \times vn - fp \times fn}{\sqrt{(vp + fp)(vp + fn)(vn + fp)(vn + fn)}}$$

Métricas de avaliação automática para geração de texto não é algo novo. Uma das métricas mais utilizadas é o ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*), proposta em 2004 por Chin-Yew Lin (Lin, 2004) com o intuito de avaliar resumos gerados por técnicas de sumarização de texto. O ROUGE calcula o número de sobreposições de unidades, como n-gramas, entre uma referência e o texto candidato a ser avaliado. Também no contexto da avaliação automática para geração de texto, foram propostas as métricas BERTScore (Zhang et al., 2020) e BARTScore (Yuan; Neubig; Liu, 2021). Ambas usam modelos de linguagem pré-treinados para tentar avaliar a qualidade do texto gerado. BERTScore usa os embeddings gerados por modelos como o BERT para calcular a similaridade (similaridade de cosseno) entre os *tokens* da sequência gerada e os *tokens* da sequência de referência. Então, métricas como precisão, revocação e *F-measure* são calculadas. Já o BARTScore usa um modelo pré-treinado baseado na arquitetura *encoder-decoder* (BART) para avaliar o texto gerado em diferentes perspectivas, incluindo coerência e factualidade. A ideia é tratar a avaliação da geração de texto como um problema de geração de texto, ou seja, usar o próprio modelo *encoder-decoder* para converter o texto de entrada no texto de saída e vice-versa.

15.4 Tendências

15.4.1 A Era dos *Large Language Models* (LLMs)

O termo *Large Language Models* (LLMs), que podemos traduzir como Modelos de Linguagem Grandes, Modelos de Linguagem Enormes, ou Modelos de Linguagem de Larga-Escala, tem se popularizado para referenciar qualquer modelo de linguagem neural. Entretanto, neste livro consideramos que LLMs se diferenciam dos demais modelos pré-treinados devido a:



- a sua **quantidade enorme de parâmetros**. Embora não exista um limite inferior universalmente aceito, tipicamente, modelos que são chamados de LLMs na literatura possuem mais de um bilhão de parâmetros, mas podendo alcançar centenas de bilhões (Zhao et al., 2023).
- o seu enquadramento na categoria de métodos de **IA Gerativa** (ou generativa). Tais modelos têm como função primária a geração de conteúdo, que no caso dos LLMs traduz-se em geração de texto.
- as suas **habilidades emergentes**, que não costumam ser observadas em modelos menores (Wei et al., 2022a). Argumenta-se que tais habilidades não poderiam ser observadas ao examinar sistemas menores, um fenômeno similar à transição de fase observada em sistemas físicos. A habilidade emergente mais comumente observada é a possibilidade de **utilizar LLMs sem nenhum treinamento adicional** que vá atualizar seus parâmetros por meio de otimização de gradientes. Ao invés deste ajuste específico, eles podem aproveitar seu pre-treinamento e serem utilizados a partir de instruções em linguagem natural – os *prompts* – e/ou demonstrações da tarefa a partir de um ou mais exemplos. Esta habilidade é conhecida como **aprendizado em contexto** (em inglês, *in-context learning* ou *few-shot prompt* (Brown et al., 2020), manifestando-se de forma curiosa com os modelos abordando tarefas para as quais não foram explicitamente treinados. Neste caso, o modelo recebe ou não uma instrução e pares de exemplos de entrada e saída, com o teste no final, conforme discutimos com o GPT. A tarefa do modelo será prever os próximos *tokens* após a última entrada de teste. A Tabela 15.2 traz um exemplo, considerando a tarefa de análise de sentimentos, mas assumindo um modelo pre-treinado que não foi ajustado para ela.

Uma outra habilidade emergente é a estratégia de **cadeia de pensamento** (CoT, do inglês, *chain-of-thought*) (Wei et al., 2022b). Discutivelmente, tal estratégia exibiria habilidades de “raciocínio” dos LLMs, embora esta seja uma terminologia polêmica. A estratégia CoT permite que os modelos retornem passos intermediários da resposta final, em tarefas que requerem múltiplos passos de raciocínio para serem resolvidas, usando instruções do tipo “EXPLIQUE PASSO A PASSO ...” ou similares.

Tabela 15.2: Exemplo de teste da habilidade emergente de aprendizado de contexto em LLMs. Os exemplos de 1 a 3 constituem em pares de entrada e saída para o modelo, enquanto a última linha, chamada de teste, apresenta apenas a entrada para o modelo. O modelo completa tal entrada com a saída em verde. Exemplos extraídos da base de dados TweetSentBR (Brum; Nunes, 2018)

Exemplo	Entrada	Saída
1	“Vitor é gracinha demais #MasterChefBR”	Positivo
2	“O #MasterChefBR tá na mesma vibe do #BBB: odeio todos.”	Negativo
Teste	“Que tensoooooooooo cozinhar com plateia!” #MasterChefBR”	Negativo

Alguns desses modelos são: BLOOM (Scao et al., 2022), Chinchilla (Hoffmann et al., 2022), Galactica (Taylor et al., 2022), Gopher (Rae et al., 2021), GPT-3 (Brown et al., 2020), LaMDA (Thoppilan et al., 2022), LLaMA (Touvron et al., 2023), Sabiá (Pires et al., 2023)



(sim, a língua portuguesa também tem um LLM para chamar de seu), PaLM (Chowdhery et al., 2022), entre vários outros⁴⁴.

Considerando a língua portuguesa, o modelo Sabiá foi treinado com um subconjunto em português do *corpora* ClueWeb 2022 (Overwijk; Xiong; Callan, 2022). O ClueWeb foi tokenizado com o tokenizador do GPT-2 e o processo resultou em 7,8 bilhões de *tokens*. O Sabiá usou uma estratégia de treinamento continuado a partir dos modelos LLaMA – nas versões com 7 bilhões e 65 bilhões de parâmetros – e do modelo GPT-J (Wang; Komatsuzaki, 2021), que contém 6 bilhões de parâmetros.

Observe que a maioria desses modelos são a base de algum agente de conversação que surgiu no fim de 2022 e início de 2023: o GPT é o modelo utilizado pelo famoso agente de conversação ChatGPT⁴⁵ (GPT-3.5 e GPT-4), o LaMDA é o modelo utilizado pelo BARD⁴⁶, LLaMA é o modelo do Vicuna⁴⁷ e o Sabiá é o modelo utilizado pela MariTalk⁴⁸. Vamos falar mais um pouco destes agentes no Capítulo 20.

Algumas habilidades emergentes têm sido observadas popularmente nos agentes de conversação mencionados. Por exemplo, a Figura 15.10 exibe a saída de um programa em Python “implementado pelo” ChatGPT. Observe, entretanto que, embora o modelo de linguagem não tenha sido treinado explicitamente para escrever programas em Python, ele pode ter se deparado com situações como essa em seu pre-treinamento, uma vez que foi treinado com textos da Internet. Ainda, os demais passos do processo de desenvolvimento do ChatGPT envolvem o ajuste de instruções e o alinhamento com feedback humano por meio do aprendizado por reforço (Ouyang et al., 2022; Stiennon et al., 2020; Ziegler et al., 2019). Assim, pode ser que durante estes passos o modelo tenha sido ajustado para lidar com este tipo de instrução. Mas ele não tem como ter sido ajustado para todas as tarefas possíveis com as quais ele tem se deparado.

A Figura 15.11 exibe um exemplo da outra habilidade emergente que mencionamos, a estratégia CoT para resolver um problema matemático simples.

Sempre é bom ressaltar que os modelos podem apresentar comportamentos diferentes de acordo com a entrada apresentada a eles. E ainda que usar o mesmo *prompt* mais de uma vez não é garantia de retorno da mesma resposta, dada a natureza probabilística dos modelos gerativos. Por exemplo, considere a interação com o BARD representada na Figura 15.12, onde o objetivo era traduzir uma frase famosa do latim para o português. Ambas foram tentativas frustradas, em que ele nem se deu ao trabalho de responder em português. Mas observe o que acontece com uma instrução diferente, exibida na Figura 15.13. Embora a segunda instrução seja mais informativa, não necessariamente é este o motivo da tentativa ter sido bem sucedida. Este tipo de estratégia é chamado de *hard prompt tuning* ou **engenharia/desenho de prompts** (Liang et al., 2022; Schick; Schütze, 2021) e consiste em modificar as entradas para tentar obter saídas distintas.

Prompts também são usados para mapear exemplos de tarefas distintas para uma entrada em linguagem natural, na tentativa de se obter uma resposta também em linguagem natural por parte do modelo. Por exemplo, suponha o exemplo a seguir, extraído do *dataset* de reconhecimento de emoções apresentado em (Cortiz et al., 2021):

- “o que eu acho incrível nesse filme é que o Harry Potter é a própria referência à

⁴⁴Veja um catálogo em <https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>.

⁴⁵<https://openai.com/blog/chatgpt>

⁴⁶<https://bard.google.com/>

⁴⁷<https://lmsys.org/blog/2023-03-30-vicuna/>

⁴⁸<https://chat.maritaca.ai/>



Figura 15.10: Saída de um programa em Python escrito pelo ChatGPT. Ele ainda explica ao final da saída o que são as funções e alerta que o programa foi configurado para funcionar apenas até o número 100, e que o usuário poderia fazer eventuais ajustes.

```

python
Copy code

def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def find_prime_pairs_with_difference_3():
    prime_pairs = []
    for num in range(2, 100): # Procura por números primos até 100 (você po
        if is_prime(num) and is_prime(num + 3):
            prime_pairs.append((num, num + 3))
    return prime_pairs

def main():
    prime_pairs = find_prime_pairs_with_difference_3()

    if prime_pairs:
        print("Pares de números primos com diferença de 3:")
        for pair in prime_pairs:
            print(pair[0], "e", pair[1])
    else:
        print("Nenhum par de números primos com diferença de 3 encontrado at

if __name__ == "__main__":
    main()
    
```

mágica”⁴⁹

Para que exemplos como esse possam ser classificados por um modelo de linguagem autorregressivo, podemos embuti-lo na seguinte instrução:

- “o que eu acho incrível nesse filme é que o Harry Potter é a própria referência a mágica.” A emoção expressa nesta mensagem é | tristeza | raiva | admiração | confusão | curiosidade

onde o texto em azul veio do *dataset* (o exemplo) e o texto em vermelho é a instrução.

Mas esta é apenas uma entre muitas formas possíveis de se escrever o *prompt* para a tarefa de reconhecimento de emoções. Ademais, cada tarefa distinta pode ter quantidades e formatos distintos de entrada. Por exemplo, uma tarefa de inferência textual precisa incluir

⁴⁹<https://github.com/diogocortiz/PortugueseEmotionRecognitionWeakSupervision/blob/main/README.md>



Figura 15.11: Exemplo da estratégia CoT com o agente de conversação ChatGPT (realizado em 01 de agosto de 2023)

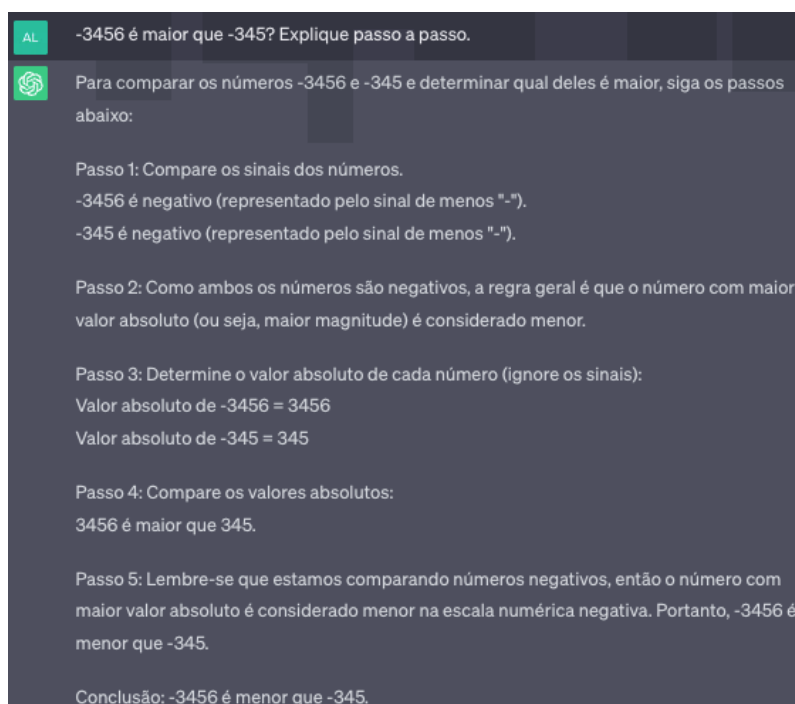
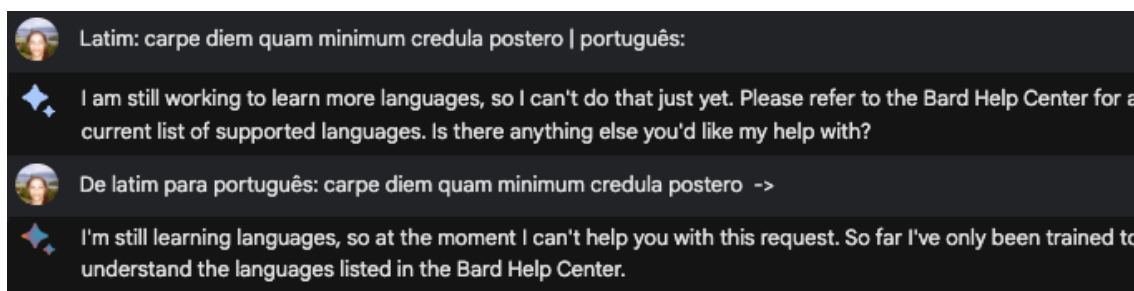


Figura 15.12: Tentativa de tradução do latim para o português com o BARD.



dois componentes, a premissa e a hipótese, ambas em azul no texto abaixo, extraídas da base de dados ASSIN-2 (Real; Fonseca; Gonçalo Oliveira, 2020):

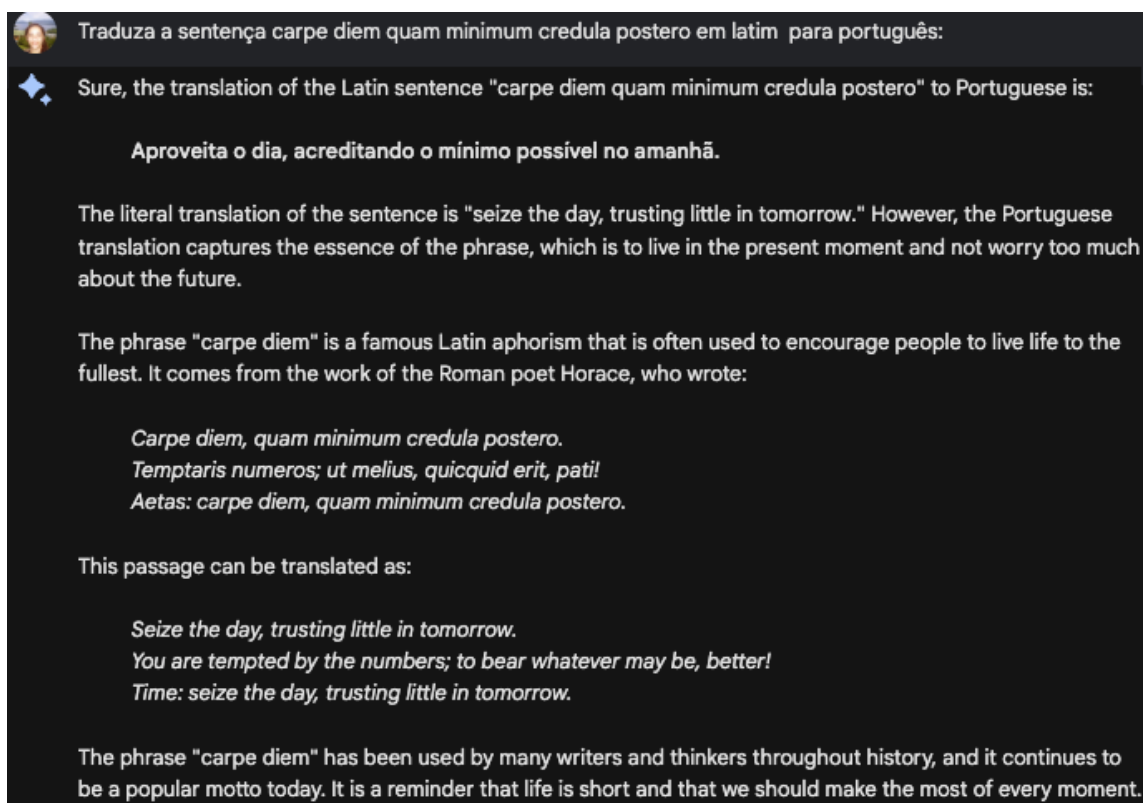
- “Suponha a frase:” “Uma mulher está dirigindo um carro e está conversando animadamente com o carinho que está sentado ao lado dela.” Podemos inferir que A mulher e o carinho estão viajando de carro. Sim ou Não?

Existem arcabouços que podem nos ajudar na criação de *prompts*. Um dos mais completos é o PromptSource⁵⁰, que inclui a coleção P3 (*Public Pool of Prompts*). O P3 é composto de mais de 2000 opções de *prompts* para diversas tarefas de PLN, porém tudo em inglês. Entender como os LLMs produzem as saídas de acordo com as entradas que são dadas para eles por meio de *prompts* é um campo de estudo recente, porém bastante ativo, desde os primeiros resultados dos LLMs (Xie et al., 2022; Xu et al., 2023).

⁵⁰<https://github.com/bigscience-workshop/promptsources>



Figura 15.13: Tentativa de tradução do latim para o português com o BARD, com um *prompt* diferente, mais informativo.



15.4.2 Treinamento Eficiente de Modelos de Linguagem Neurais

Embora LLMs possam ser usados sem nenhum ajuste em seus pesos, eles acabam por ficar muito dependentes dos *prompts* e da exposição implícita que o modelo teve para uma certa tarefa durante o seu pre-treinamento. Assim, o desempenho de modelos que se baseiam apenas na habilidade emergente gerativa pode ficar bem abaixo do desempenho de um outro modelo, ainda que menor, que é ajustado para uma tarefa específica (Raffel et al., 2020). Mas como ajustar um modelo de bilhões de parâmetros de forma razoavelmente eficiente? Para responder a esta pergunta, novas abordagens sugerem que o treinamento seja feito em apenas partes dos modelos, ou com estratégias baseadas em reparametrização das matrizes de pesos.

Para o primeiro caso, podemos mencionar três estratégias:

1. *Soft prompt tuning*, ou apenas *prompt tuning* (Lester; Al-Rfou; Constant, 2021). Neste caso, o modelo fica congelado, exceto por uma quantidade adicional de k parâmetros numéricos ajustáveis – por isso o *soft* – que são concatenados no início dos embeddings do texto de entrada. Esses k parâmetros serão treinados de acordo com a tarefa-alvo, usando o algoritmo clássico de retro-propagação. Observe a diferença entre as versões *hard* e *soft*: a primeira não tem ajuste de parâmetros, se baseando apenas na troca de palavras na instrução, enquanto a segunda é diferenciável, ou seja, o *prompt* é composto por um conjunto de pesos ajustáveis.
2. *Prefix tuning* (Li; Liang, 2021). Nesta estratégia, pesos ajustáveis são acrescentados no início de cada bloco dos Transformers. Observe que o modelo “original” permanece



congelado, sem ajustes, assim como na abordagem de *soft prompt tuning*. Porém, enquanto lá pesos ajustáveis aparecem apenas no início dos embeddings de entrada, que seriam mesmo o local de inserção das instruções, aqui eles são concatenados no início de cada bloco do Transformer. Ainda, antes da concatenação, eles passam por duas camadas de redes neurais completamente conectadas, para garantir que o prefixo esteja em um mesmo espaço de representação vetorial que a entrada do bloco. Ou seja, o processo de adaptação de prefixos, teoricamente, é mais custoso que o processo de adaptação de *prompts*. Assim, a ordem de processamento do bloco do Transformer se torna: CAMADA COMPLETAMENTE CONECTADA PARA PROCESSAMENTO DOS PROMPTS -> CONCATENAÇÃO DA SAÍDA ANTERIOR COM A ENTRADA DO MODELO -> AUTO-ATENÇÃO -> NORMALIZAÇÃO -> CAMADA COMPLETAMENTE CONECTADA DO TRANSFORMER -> NORMALIZAÇÃO (desconsiderando as conexões residuais).

3. Adaptadores (Houlsby et al., 2019). Adaptadores também acrescentam pesos ajustáveis adicionais a cada bloco do Transformer, mas não no início do bloco e sim no meio do bloco. Assim, os adaptadores são camadas de rede neural completamente conectadas, com uma função de ativação não-linear entre elas, introduzidas imediatamente antes da camada de normalização. Ou seja, a ordem de processamento se torna: AUTO-ATENÇÃO -> ADAPTADOR -> NORMALIZAÇÃO -> CAMADA COMPLETAMENTE CONECTADA DO TRANSFORMER -> ADAPTADOR -> NORMALIZAÇÃO (desconsiderando as conexões residuais, para facilitar a comparação com os *prompts*).

Já as abordagens baseadas em reparametrização tem o método de adaptação baseado no posto das matrizes de peso, LoRA (do inglês, *Low-Rank Adaptation*) (Hu et al., 2022) como seu principal representante. A motivação principal vem de um estudo anterior, que apontou que modelos ajustados para uma nova tarefa possuem uma dimensão menor que os modelos pre-treinados (Aghajanyan; Gupta; Zettlemoyer, 2021), ou seja, que eles poderiam ser decompostos para matrizes menores sem perder informação. Dessa forma, o método aprende como decompor as matrizes de atualização dos gradientes para postos menores.

LoRA também se motiva no espaço de memória necessário para armazenar as mudanças nas matrizes de peso durante o seu treinamento. Nesta mesma direção, abordagens baseadas em quantização, que guardam os pesos de treinamento em variáveis tipadas com menos precisão, também têm sido foco de investigação recentemente⁵¹ (Dettmers et al., 2023).

Um outro ponto a ser considerado com o uso de LLMs (e até mesmo LMs) é o tamanho da entrada. Considerando que a o método de atenção tem uma complexidade de ordem quadrática, a maioria dos modelos baseados em Transformers usualmente limitam a sua entrada em cerca de 500 a 1024 *tokens*. Entradas maiores, em geral, precisam ser truncadas. Mesmo abordagens que consideram matrizes de atenção esparsas, como Longformer (Beltagy; Peters; Cohan, 2020), ainda têm limitações. Abordagens recentes armazenam e recuperam camadas de decodificação ou separam entradas em pedaços de tamanho menores, possibilitando lidar com textos de tamanhos até 500 mil *tokens* (Bertsch et al., 2023; Ivgi; Shaham; Berant, 2023; Wu et al., 2022).

⁵¹<https://huggingface.co/blog/hf-bitsandbytes-integration>



15.4.3 Estratégias de Treinamento para Agentes de Conversação: alinhamento e feedback humano

Um modelo de linguagem não é treinado explicitamente para interagir com usuários, apenas para completar sentenças. Para criar um agente de conversação tendo como base um modelo de linguagem, é necessário incluir no modelo a habilidade de tentar responder ao usuário de acordo com a sua intenção expressa nas instruções, ou seja se alinhar ao diálogo, acompanhando a conversa. Idealmente, o agente de conversação também deve evitar respostas indevidas que poderiam levar a comportamentos nocivos.

Assim, em (Ouyang et al., 2022) os autores tinham como motivação “tornar modelos de linguagem **úteis** – ajudando os usuários a resolver tarefas – **honestos** – não inventando informação ou levando o usuário para uma falsidade – e **inofensivos** – não causando algum mal físico, psicológico ou social”. É menos desafiador indicar se os modelos atuais conseguem ter a primeira característica. Entretanto, as duas últimas não podemos afirmar com convicção que foram alcançadas, nem mesmo pelos modelos mais atuais.

Para tornar possível tal alinhamento entre a saída de um modelo de linguagem e a intenção do usuário, recorre-se a uma outra camada de aprendizado, o **Aprendizado por Reforço com Feedback Humano**, ou a partir de preferências humanas, (do inglês, *Reinforcement learning from Human Feedback* ou RLHF) (Christiano et al., 2017) uma abordagem que já havia se mostrado frutífera em visão computacional. O aprendizado por reforço se vale de uma função de recompensa: caso a saída seja adequada, a recompensa é positiva; caso contrário, devolve-se uma penalidade. Ou seja, se o modelo estiver devolvendo uma resposta adequada – e aqui o adequado seria a resposta que obedecesse aos três princípios acima, de utilidade, honestidade e inofensibilidade – então a recompensa seria positiva.

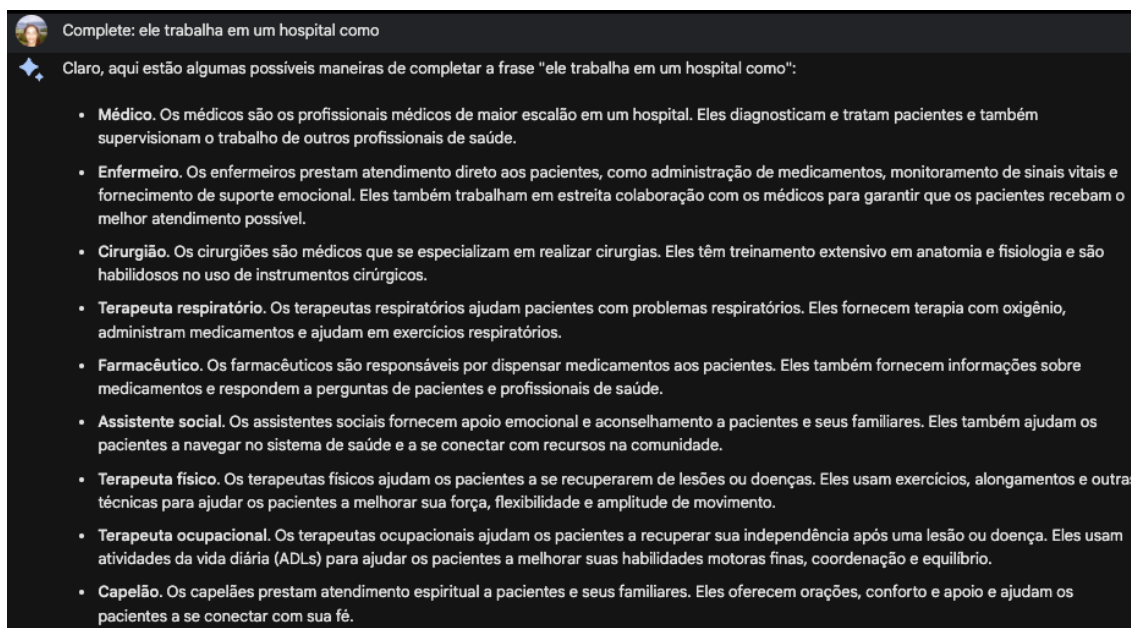
Acontece que definir tal valor de recompensa não é trivial e este tem sido um dos grandes desafios da área de aprendizado por reforço. Nos agentes de conversação, o que é feito, é aprender um modelo de recompensa a partir de exemplos. Pares de *prompts* e respostas são gerados, usualmente de forma automática pelos modelos, por questões de escala. Mas nada impede que esses pares também sejam curados em outros conjuntos de dados ou definidos por pessoas (Zhou et al., 2023). A partir daí, anotadores vão dizer quais são as suas saídas preferidas, geralmente usando mais de um modelo para ter alguma base de comparação. Então, considerando essa saída, o modelo de recompensa pode ser treinado.

Embora os agentes de conversação baseados em modelos de linguagem já estejam sendo usados para resolver problemas reais, muito ainda precisa ser alcançado. Em particular, duas limitações podem fazer com que tais modelos ainda não estejam prontos para serem adotados em larga escala e em aplicações sensíveis: (i) ainda são poucas as línguas que tais modelos conseguem lidar, se compararmos com a quantidade de línguas que temos no mundo; e (ii) ainda existe um viés social negativo embutido em tais modelos. Para ver um exemplo, considere os exemplos das Figuras 15.14 e 15.15 e perceba a diferença ao completar uma frase para o gênero masculino e feminino.

Isso no faz pensar, à medida que fechamos este capítulo, que é nossa responsabilidade como pesquisadores e desenvolvedores de tecnologia, que devemos considerar o potencial positivo e negativo dos modelos de linguagem de revolucionarem a nossa compreensão e interação com a tecnologia, e de estabelecerem novas formas de comunicação. Devemos, sim, celebrar os progressos que foram feitos, os insights que foram obtidos e as barreiras que foram rompidas. Mas à medida que a pesquisa em modelos de linguagem avança, e novas ferramentas a partir deles surgem e se tornam tão populares, precisamos ter em mente os limites que continuam a ser desafiados e a responsabilidade ao se usar e aplicar



Figura 15.14: Exemplo de viés de gênero para o português no agente de conversação BARD. Aqui, solicitou-se ao agente completar a frase “Ele trabalha no hospital como”.

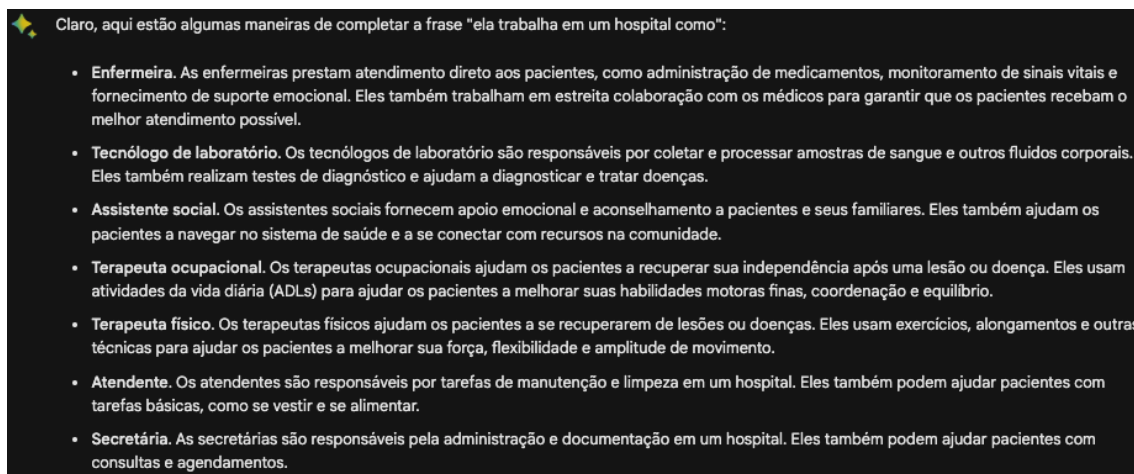


modelos cujas saídas ainda fogem da nossa compreensão. Assim, lembrando Alan Turing, mas do que nunca podemos afirmar que “*We can only see a short distance ahead, but we can see plenty there that needs to be done*”⁵², em particular se considerarmos a língua portuguesa e tantas outras milhares espalhadas no planeta que ainda carecem da nossa atenção (sem trocadilhos) e compreensão.

⁵²Tradução nossa: Somos capazes de ver apenas uma curta distância à frente, mas já podemos ver que muito ainda precisa ser feito.



Figura 15.15: Exemplo de viés de gênero para o português no agente de conversação BARD. Aqui, solicitou-se ao agente completar a frase “Ela trabalha no hospital como”.



Parte VIII

Aplicações



Capítulo 16

Recuperação de Informação

Viviane P. Moreira

16.1 Introdução

A necessidade de organizar a informação é inerente à espécie humana – bibliotecas existem desde pelo menos o ano 2600 AC. Dado o grande volume de informação que as bibliotecas armazenam, a partir dos anos 1960, iniciaram-se esforços a fim de automatizar o armazenamento e a busca de materiais bibliográficos através da computação. Esses esforços marcaram o início da área de recuperação de informação (RI). A RI trata de encontrar, a partir de grandes coleções, material (geralmente documentos) de natureza não estruturada (geralmente texto) que satisfaça uma necessidade de informação (Manning; Schütze; Raghavan, 2008). Em outras palavras, o objetivo central da RI é a busca, ou seja, a tarefa de encontrar material **relevante** a partir de uma consulta de um usuário. Esta tarefa é comumente conhecida por **recuperação ad hoc**. Apesar de a RI poder ser aplicada a diferentes tipos de dados não estruturados como imagem, áudio, vídeo etc. o foco deste capítulo é **informações textuais**.

16.1.1 Relação com o PLN

Há muita interseção entre RI e PLN, pois ambas lidam com a linguagem humana. Contudo, há algumas diferenças. A primeira diferença é quanto à origem – enquanto que o PLN teve origem na inteligência artificial e na linguística computacional, a RI teve origem na biblioteconomia e na ciência da informação¹. Outra diferença é quanto ao escopo – podemos dizer que o escopo do PLN é mais abrangente (i.e., compreensão e geração de linguagem) enquanto que o da RI é mais restrito às tarefas relacionadas à busca por informação.

16.1.2 O Foco da Recuperação de Informação

A **tarefa central** da RI é casar a consulta de um usuário com os documentos que são potencialmente relevantes a ela. A principal dificuldade é que os termos utilizados pelo usuário podem não ter sido usados nos documentos relevantes. Este problema é conhecido como **incompatibilidade de vocabulário** (*vocabulary mismatch*) e é decorrente de dois fenômenos comuns na linguagem: a **sinonímia** e a **polissemia**. A sinonímia refere-se ao fato de usarmos palavras diferentes para nos referirmos ao mesmo conceito, e.g., “bergamota”, “tangerina” e “mexerica”. O problema para RI é que uma consulta pelo termo “bergamota” não consegue recuperar documentos relevantes que contêm “tangerina”. A polissemia refere-se ao fato de que uma mesma palavra pode apresentar sentidos distintos, e.g., “manga”

¹Para um relato mais detalhado sobre as origens da RI, há o artigo de Lesk (1995).



que tanto pode ser a fruta como a parte da vestimenta que cobre o braço da pessoa. O efeito negativo da polissemia para a RI é a recuperação de documentos que contêm a palavra pesquisada, mas não no sentido pretendido. Ao longo dos anos, houve uma vasta gama de propostas de solução para este problema. Uma visão geral dessas propostas é fornecida na Seção 16.5.

Até meados dos anos 1990, o interesse em RI estava restrito a bibliotecários, jornalistas e profissionais do direito (i.e., profissões que tinham bastante necessidade de buscar informações). Com a popularização da Internet e dos motores de busca para a web, a RI ganhou muita importância. Sistemas de RI fazem parte da vida diária de uma grande parte da população mundial. Há estimativas que o Google, o motor de busca mais utilizado, receba cerca de 100 mil consultas por segundo e tenha 4,3 bilhões de usuários². Os desafios de se lidar com coleções contendo bilhões de documentos (i.e., páginas web) motivaram o desenvolvimento de novos algoritmos e técnicas, objetivando tanto eficiência (baixo custo computacional) quanto eficácia (qualidade do resultado).

Um sistema de RI também pode ser utilizado como um componente em tarefas de PLN como sistemas de perguntas e respostas e de detecção de plágio. A vantagem é que a RI consegue recuperar documentos candidatos com um custo computacional baixo. Assim, as fases subsequentes que requerem comparações exaustivas usando modelos mais custosos podem trabalhar com um conjunto menor de documentos.

16.1.3 O Conceito de Relevância

Vimos na Seção 16.1.2 que a tarefa central de RI é recuperar itens que sejam **relevantes** a uma necessidade de informação expressa por meio de palavras-chave. A relevância é um julgamento feito **pelo usuário** que indica o quão bem um documento satisfaz a consulta. Em sua forma mais simples, ela pode ser tratada como binária – cada documento é considerado como **relevante** ou como **não relevante**. Também é possível utilizar diversos níveis de relevância como por exemplo: “muito relevante”, “moderadamente relevante”, “marginalmente relevante” e “não relevante”.

É comum ouvirmos que um motor de busca ou sistema de RI retorna os resultados em **ordem de relevância**. Esta afirmação não está tecnicamente correta pois a **relevância é subjetiva** e só pode ser atribuída pelo usuário. Dois usuários podem fazer a mesma consulta q e julgar um mesmo documento d de forma diferente – enquanto um usuário considera que d é relevante, o outro usuário pode considerar que d não é relevante para q . Os sistemas de RI ordenam os documentos utilizando uma função de ranqueamento baseada em evidências que o sistema consegue calcular (e.g., estatísticas de ocorrências das palavras nos documentos, tamanho dos documentos etc.). Idealmente, espera-se que esta função de ranqueamento esteja positivamente correlacionada com a ideia subjetiva de relevância dos usuários.

16.1.4 Principais Livros

A RI é uma área de pesquisa por si só e este capítulo não pretende ser uma revisão exaustiva. vamos focar nos aspectos centrais da área e na sua interseção com PLN. Há diversos livros em **língua inglesa** que são referência em RI. Para o leitor que deseja saber mais sobre a área, indicamos os seguintes livros:

²<https://wpdevshed.com/how-many-people-use-google/>



- Introduction to Information Retrieval de Manning, Schütze e Raghavan (Manning; Schütze; Raghavan, 2008), publicado em 2008 pela Cambridge University Press e disponível gratuitamente online³.
- Search engines: Information retrieval in practice de Croft, Metzler, e Strohman (Croft; Metzler; Strohman, 2010), publicado em 2010 pela Addison-Wesley e disponível gratuitamente online⁴.
- Modern Information Retrieval - the concepts and technology behind search de Baeza-Yates e Ribeiro-Neto (Baeza-Yates; Ribeiro-Neto, 2011), publicado em 2011 pela Addison-Wesley.

Em português, há uma versão resumida do livro Modern Information Retrieval publicada em 2013 (Baeza-Yates; Ribeiro-Neto, 2013). Esta versão contém os principais capítulos do livro original.

16.1.5 Organização deste Capítulo

Nesta primeira versão do livro **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**, o capítulo sobre Recuperação da Informação tem como foco as operações de pré-processamento e indexação (Seção 16.2), nos modelos clássicos (Seção 16.3), no paradigma de avaliação (Seção 16.4) e em técnicas de melhoria de qualidade (Seção 16.5). Também indicamos algumas bibliotecas e ferramentas amplamente utilizadas (Seção 16.6). As próximas versões do livro irão incluir tópicos avançados como o ranqueamento neural.

16.2 Visão Geral de um Sistema Típico de Recuperação de Informação

A Figura 16.1 mostra uma visão geral do processo de RI. Em uma ponta, há a usuária que tem uma **necessidade de informação** e, na outra ponta, temos uma coleção de documentos textuais. A consulta da usuária é submetida a um sistema de RI (representado pelo retângulo laranja). As Etapas 1 e 2 ocorrem offline, pois o índice precisa estar pronto antes que o sistema possa receber consultas. Na Etapa 1, os documentos da coleção são pré-processados para então, na Etapa 2, serem indexados. Na Etapa 3, o sistema executa sobre a consulta as mesmas operações de pré-processamento que foram aplicadas na Etapa 1. O texto pré-processado da consulta é utilizado, na Etapa 4, para buscar no índice os documentos que mais bem atendam a consulta. Os resultados da consulta são então retornados à usuária sob a forma de uma lista ordenada.

16.2.1 Pré-Processamento

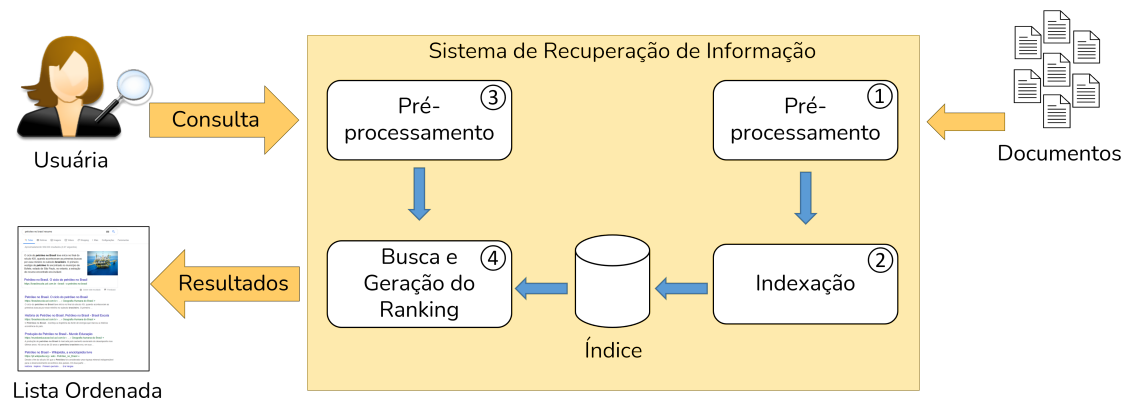
As operações de pré-processamento são muito semelhantes às estudadas para as tarefas de PLN relatadas no Capítulo 4. É preciso definir a nossa **unidade de indexação**, ou seja, o que é um **documento** no contexto tratado. Por exemplo, uma notícia, um comentário, um e-mail, um artigo científico, um livro, um capítulo do livro etc.

³<https://nlp.stanford.edu/IR-book/>

⁴<https://ciir.cs.umass.edu/irbook/>



Figura 16.1: Visão Geral do Processo de RI.



16.2.1.1 Tokenização

Dada uma sequência de caracteres e uma unidade de indexação, a **tokenização** irá separar a sequência em *tokens* (i.e., unigramas de palavras). Nesse processo, é comum descartarmos caracteres de pontuação e mantermos apenas as palavras (i.e., também chamadas de termos). Embora à primeira vista este processo pareça bastante trivial (i.e., podemos apenas considerar que os caracteres não alfabéticos como símbolos e sinais de pontuação sejam separadores), há vários casos especiais que podemos precisar tratar. Por exemplo, endereços de email, emojis, e termos que misturam letras e símbolos como C++.

O processo de tokenização costuma ser implementado utilizando-se expressões regulares e há vários tokenizadores prontos em bibliotecas como NLTK⁵ e spaCy⁶. Após a tokenização, é comum converter todos os caracteres para letra minúscula. O objetivo é fazer com que a busca não seja *case-sensitive*, isto é, impactada por texto em letras maiúscula e minúscula. Por outro lado, essa operação dificulta a identificação de entidades nomeadas, uma vez que entidades são geralmente nomes próprios com grafia em letra maiúscula.

A remoção de acentos, cuja denominação mais apropriada é **remoção de sinais diacríticos** também pode ser realizada, principalmente quando se trata de textos informais nos quais esses sinais são menos utilizados. Um sinal diacrítico é uma marca que colocamos sobre ou sob o caractere como acento agudo, grave, circunflexo, til ou cedilha.

16.2.1.2 Remoção de *Stop Words*

As *stop words* são palavras que têm pouca utilidade para a RI pois ocorrem em muitos documentos e assim não servem para distinguir o conteúdo semântico dos documentos. Preposições, conjunções, artigos e verbos de ligação são normalmente consideradas como *stop words*. Essas palavras são muitas vezes completamente descartadas em um processo conhecido como **remoção de stop words**.

O processo de remoção de *stop words* é bastante simples e consiste em verificar a presença de cada palavra do texto em uma lista de *stop words* previamente construída. Existem listas prontas que podemos usar, como a do NLTK, por exemplo, que contém 204 palavras. Contudo, é recomendável revisar se as listas contêm palavras que podem ser úteis para o

⁵<https://www.nltk.org/>

⁶<https://spacy.io/models/pt>



contexto da aplicação sendo desenvolvida. Por exemplo, na lista elaborada pela Linguateca⁷ com as palavras mais frequentes dos textos da Folha de São Paulo, encontramos a palavra “brasil”. Esta palavra, muito provavelmente, não deve ser removida dos textos da coleção de documentos.

O principal argumento a favor da remoção de *stop words* é a redução do tamanho do vocabulário pois elas representam cerca de 40% das ocorrências de palavras em um *corpus*. Por outro lado, essa remoção pode trazer uma perda de informação relevante. Se pensarmos na famosa expressão “ser ou não ser, eis a questão”, com a remoção de *stop words*, sobraria apenas “questão”, o que descaracteriza completamente a expressão. O impacto negativo na busca seria que o sistema não mais conseguiria distinguir entre documentos que contenham a expressão completa daqueles que contêm apenas a palavra “questão”.

16.2.1.3 Stemming

O objetivo do *stemming* é gerar uma mesma representação para formas variantes de uma mesma palavra através da remoção dos sufixos. Por exemplo, removendo-se os sufixos de “apresentação”, “apresentando”, e “apresentar”, obteríamos o radical “apresent”. Com isso, uma usuária buscando por “apresentar artigos científico” consegue recuperar um documento com o trecho “apresentando artigos científicos”. O benefício é aumentar o número de documentos relevantes recuperados em resposta à consulta.

O processo de *stemming* e seu impacto sobre RI vêm sendo estudado há diversos anos. Para a língua inglesa, o primeiro algoritmo de *stemming*, ou *stemmer*, data de 1968 e foi proposto por Julie Beth Lovins (Lovins, 1968). Em 1980, Martin Porter propôs o Porter Stemmer (Porter, 1980) que mostrou obter bons resultados (também para a língua inglesa) e posteriormente foi traduzido para outros idiomas incluindo o português⁸.

O primeiro *stemmer* desenvolvido especialmente para a língua portuguesa foi o Removedor de Sufixos da Língua Portuguesa (RSLP)⁹ (Orengo; Huyck, 2001). Posteriormente veio o STEMBR (Alvares; Garcia; Ferraz, 2005). Há resultados experimentais que mostram a validade da aplicação de *stemmers* para RI em português (Flores; Moreira; Heuser, 2010; Orengo; Buriol; Coelho, 2006). Muitas vezes, na média de um conjunto de consultas, o impacto do *stemming* pode ser pequeno. Contudo, ao analisarmos consultas individuais, percebemos melhorias muito grandes em alguns casos. Em outros, podem aparecer mais documentos não relevantes recuperados pois a remoção do sufixo pode gerar ambiguidade. Os resultados mostram que *stemmers* menos agressivos, ou seja, com menos regras de remoção tendem a gerar resultados melhores pois têm menos impacto negativo na precisão.

O Quadro 16.1 mostra um trecho do livro O Tempo e o Vento – O Continente, de Érico Veríssimo, e suas versões após cada etapa do pré-processamento. Observemos a redução de 51 para 27 *tokens* com a remoção de *stop words*. O *stemmer* utilizado no exemplo foi o Snowball¹⁰. Também vale ressaltar a potencial introdução de ambiguidade com a redução de algumas palavras, por exemplo de “legalistas” para “legal”.

Quadro 16.1. Excerto do livro O Tempo e o Vento – O Continente, de Érico Veríssimo, e suas versões após cada etapa do pré-processamento.

⁷<https://www.linguateca.pt/chave/stopwords/>

⁸<http://snowball.tartarus.org/algorithms/portuguese/stemmer.html>

⁹<https://www.vivianemoreira.org/rslp>

¹⁰<https://snowballstem.org/>



Original	Quando pela primeira vez aparecera em Santa Fé, no ano em que fora assinada a paz entre farroupilhas e legalistas, causara a pior das impressões. Chegara escoteiro, montado num cavalo magro e manco, e fazendo questão de mostrar a toda a gente que tinha as guaiacas atestadas de moedas de ouro.
Tokenizado	Quando pela primeira vez aparecera em Santa Fé no ano em que fora assinada a paz entre farroupilhas e legalistas causara a pior das impressões Chegara escoteiro montado num cavalo magro e manco e fazendo questão de mostrar a toda a gente que tinha as guaiacas atestadas de moedas de ouro
Convertido para minúsculo	quando pela primeira vez aparecera em santa fé no ano em que fora assinada a paz entre farroupilhas e legalistas causara a pior das impressões chegara escoteiro montado num cavalo magro e manco e fazendo questão de mostrar a toda a gente que tinha as guaiacas atestadas de moedas de ouro
Após remoção de <i>stop words</i>	primeira vez aparecera santa fé ano assinada paz farroupilhas legalistas causara pior impressões chegara escoteiro montado cavalo magro manco fazendo questão mostrar gente guaiacas atestadas moedas ouro
Após <i>stemming</i>	primeir vez aparec sant fé ano assin paz farroupilh legal caus pior impressõ cheg escoteir mont caval magr manc faz questã mostr gent guaiac atest moed our

16.2.2 Indexação

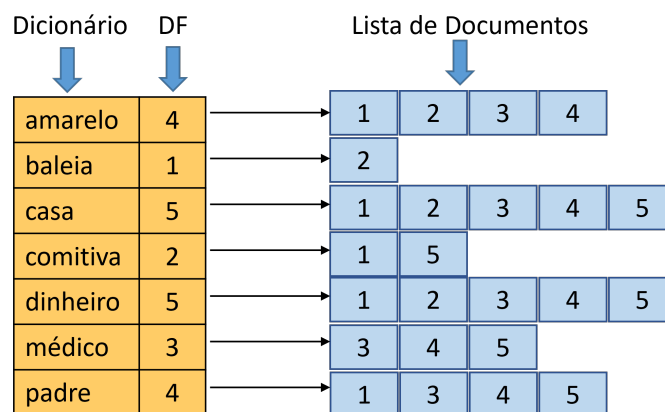
A forma que temos para evitar ter de “varrer” os textos da coleção de documentos em busca dos termos da consulta (o que seria um procedimento extremamente lento) é contarmos com um **índice** previamente construído. A indexação recebe como entrada o texto pré-processado e cria um índice chamado de **arquivo invertido**. Um arquivo invertido assemelha-se ao índice remissivo que comumente encontramos ao final de um livro e contém as palavras (únicas) do texto e a lista de documentos em que elas aparecem.

A Figura 16.2 mostra um exemplo de índice que pode ser usado por um sistema de RI. O índice tem dois componentes: (i) o dicionário, também é chamado de vocabulário, que possui a lista de termos únicos da coleção de documentos e (ii) a lista de identificadores de documentos em que o termo aparece (conhecida por lista de *postings*). Junto ao dicionário, também armazena-se o número de documentos em que o termo aparece, conhecido como *document frequency (DF)*. Observemos que o dicionário fica em ordem alfabética e a lista de *postings* fica em ordem de identificador de documento.

Nos casos em que desejamos fazer uma busca sobre um campo específico do documento (e.g., título ou autor), vamos precisar de **índices paramétricos**, um para cada campo. Além disso, se quisermos permitir buscas por múltiplos termos adjacentes (e.g., “memória prodigiosa”), precisaremos de um **índice posicional** que guarde também as posições em que os termos aparecem nos documentos. Certamente, tanto os índices paramétricos como os posicionais possuem um custo maior em termos de espaço de armazenamento e de processamento.



Figura 16.2: Exemplo de um trecho de índice armazenado como um arquivo invertido.



16.3 Modelos Clássicos de recuperação de informação

Um modelo de RI especifica como representar os documentos, as consultas e como compará-los. Ao longo dos anos, diversos modelos de RI foram propostos. Vamos explorar os modelos clássicos. Todos os modelos clássicos pressupõem que a distribuição dos termos nos documentos é independente e utilizam a abordagem *bag of words* (BoW). Em um BoW, a ordem dos termos nos documentos é desprezada. A vantagem é a simplificação dos modelos, os quais conseguem processar consultas de maneira mais rápida. A desvantagem é a perda de semântica em alguns casos pois as sentenças “João é mais velho do que José” e “José é mais velho do que João” têm representações idênticas apesar de significado oposto.

16.3.1 Modelo Booleano

O primeiro modelo que foi usado para RI foi o modelo Booleano. Trata-se de um modelo simples, baseado na teoria dos conjuntos e na álgebra booleana. As consultas são compostas por palavras-chave combinadas com os operadores AND, OR e NOT e o seu processamento utiliza o índice.

Para processar uma consulta do tipo p_1 AND p_2 , precisamos selecionar as listas de *postings* de p_1 e p_2 e computar a sua interseção. Por exemplo, com o trecho de índice da Figura 16.2, podemos ver que a consulta “comitiva” AND “médico” retornaria apenas o documento 5. Já a consulta “baleia” AND “padre” não retornaria nenhum documento.

Consultas com OR implicam na união de conjuntos. Por exemplo, considerando o índice da Figura 16.2, a consulta “comitiva” OR “médico” retornaria os documentos 1, 3, 4 e 5.

A principal limitação do modelo Booleano é que ele não é capaz de ordenar o resultado da consulta. Ou um documento satisfaz ou não satisfaz a expressão Booleana da consulta; não há a opção “satisfaz parcialmente”. Apesar dessa limitação, este foi o modelo comercial mais usado por três décadas, mesmo após a proposta de modelos superiores. Até hoje ainda vemos o modelo Booleano sendo usado, por exemplo, em alguns sistemas de bibliotecas.

16.3.2 Modelo Vetorial

Com o objetivo de poder ordenar os documentos em resposta às consultas, há duas premissas simples que podemos utilizar: (i) documentos que contém mais vezes os termos da consulta têm mais chance de estarem relacionados a ela (e de serem relevantes) e (ii) os termos mais



raros na coleção são mais úteis para diferenciar o conteúdo dos documentos. A primeira premissa é atribuída ao pesquisador alemão Hans Peter Luhn e data de 1957. E a segunda premissa foi desenvolvida pela pesquisadora inglesa Karen Spärck Jones em 1972.

Com base nessas premissas, Gerard Salton propôs, na década de 1960, o modelo vetorial (em inglês, *vector space model*) (Salton; McGill, 1983). Nesse modelo, os documentos e as consultas são representados como vetores em um espaço de t dimensões, onde t é o número de termos distintos (i.e., o tamanho do dicionário). No espaço vetorial, os termos são os eixos. As consultas e os documentos são representados no espaço de acordo com a força da associação que eles têm com o termo. A força da associação do documento d_j com o termo i (w_{i,d_j}) é computada pelo esquema TF-IDF (*Term Frequency times Inverse Document Frequency*), segundo a Equação 16.1:

$$w_{i,d_j} = TF_{i,d_j} \times IDF_t = freq_{i,d_j} \times \log_{10} \frac{N}{DF_i} \quad (16.1)$$

onde $freq_{i,d_j}$ é o número de ocorrências do termo i no documento d_j , N é o número de documentos da coleção e DF_i é o número de documentos que contêm o termo i . É comum normalizar o componente TF para evitar que documentos longos sejam beneficiados, pois um documento longo tem mais chances de possuir os termos da consulta. A normalização do componente TF pode ser feita de diferentes formas: (i) dividindo-se o número de ocorrências do termo no documento pelo número de palavras no documento ou (ii) dividindo-se o número de ocorrências do termo no documento pelo número de ocorrências do termo mais comum naquele documento.

O componente IDF tem por objetivo dar um peso maior para os termos raros pois eles são mais úteis para discriminar o conteúdo dos documentos. Pensando em uma situação extrema, um termo que ocorra em todos os documentos não tem nenhuma utilidade para diferenciar um documento de outro. Sendo assim, valores de IDF mais altos serão atribuídos para termos que ocorram em poucos documentos. Note-se que o IDF não tem efeito caso a consulta tenha apenas um termo. Em consultas com dois ou mais termos, o efeito do IDF é dar mais importância para os termos que ocorrem em menos documentos. Por exemplo, imaginemos uma coleção que trata de instrumentos musicais. Um usuário faz a consulta “flauta bansuri”. Considerando que o termo “flauta” é muito mais comum do que o termo “bansuri” (que se refere a um tipo específico de flauta transversal), o efeito do IDF será atribuir bem mais importância ao termo “bansuri”. Os vetores gerados pelo TF-IDF são longos e esparsos. **Longos** pois o número de dimensões é igual ao número de palavras do vocabulário (o que pode chegar a centenas de milhares para algumas coleções). **Esparsos** pois a grande maioria dos elementos têm valor zero já que os documentos possuem uma pequena fração dos termos do vocabulário.

A similaridade entre um documento e uma consulta é dada pelo **cosseno** entre os seus vetores. O cosseno entre uma consulta q e um documento d_j é o produto escalar normalizado dos vetores \vec{q} e \vec{d}_j , dado pela Equação 16.2:

$$cosseno(q, d_j) = \frac{\vec{q} \cdot \vec{d}_j}{|\vec{q}| \times |\vec{d}_j|} = \frac{\sum_{i=1}^t w_{i,q} \times w_{i,d_j}}{\sqrt{\sum_{i=1}^t w_{i,q}^2} \times \sqrt{\sum_{i=1}^t w_{i,d_j}^2}} \quad (16.2)$$

onde $w_{i,q}$ e w_{i,d_j} representam o peso do i -ésimo termo na consulta q e no documento d_j , respectivamente.

O cosseno é máximo (i.e., igual a 1) se os vetores possuem um ângulo de 0 graus e



é mínimo (i.e., igual a 0) se os vetores formarem um ângulo de 90 graus¹¹. Em outras palavras, o cosseno será mínimo se o documento e a consulta não compartilharem nenhum termo. Vale ressaltar que não é preciso computar o cosseno entre a consulta e todos os documentos. Utilizando um índice (como o da Figura 16.2), podemos selecionar apenas os documentos que têm pelo menos uma das palavras da consulta, e calcular o cosseno somente entre a consulta e esses documentos. Também é possível estender essa noção e selecionar apenas documentos com valores de TF-IDF que ultrapassem um limiar para os termos da consulta ou restringir ainda mais e calcular o cosseno apenas para documentos que possuam duas ou mais das palavras da consulta.

Exemplo 16.1. Para ilustrar o conceito de TF-IDF e o ranqueamento dos documentos no modelo vetorial através do cosseno, vamos trabalhar com um exemplo. Nossa coleção de documentos é composta por cinco importantes livros da literatura brasileira. São eles:

- d_1 – O Alienista de Machado de Assis;
- d_2 – Vidas Secas de Graciliano Ramos;
- d_3 – O Continente (de O Tempo e o Vento) de Érico Veríssimo;
- d_4 – Capitães de Areia de Jorge Amado; e
- d_5 – Os Sertões de Euclides da Cunha.

Essa pequena coleção possui 350.260 *tokens*, e um vocabulário de 32.594 termos (já desconsiderando-se as *stop words*) – isso quer dizer que os documentos e as consultas são representados por vetores de 32.594 dimensões. Como seria inviável mostrarmos todos esses termos, iremos focar em sete termos selecionados: amarelo, baleia, casa, comitiva, dinheiro, médico e padre. A Tabela 16.2 mostra os sete termos selecionados juntamente com o número de documentos em que cada um deles aparece (DF). A partir do DF, podemos computar o IDF (segundo elemento da Equação 16.1). Por exemplo, o termo “comitiva” ocorre em dois documentos, então $IDF_{comitiva} = \log_{10} \frac{5}{2} = 0,3979$. Já para o termo “casa” que ocorre em todos os cinco documentos, o $IDF_{casa} = \log_{10} \frac{5}{5} = 0$.

Tabela 16.2: Exemplo de DF e IDF para algumas palavras do texto

Termo	DF	IDF
amarelo	4	0,0969
baleia	1	0,6990
casa	5	0,0000
comitiva	2	0,3979
dinheiro	5	0,0000
médico	4	0,0969
padre	4	0,0969

¹¹Note que o valor do cosseno irá ficar no intervalo [0,1] e não [-1,1] pois os escores gerados pelo TF-IDF são positivos.



Tabela 16.3: Exemplo de matriz de incidência de termos em documentos com valores de TF (i.e., o número de ocorrências dos termos nos documentos).

Termo	d_1 O Alienista	d_2 Vidas Secas	d_3 O Tempo e O Vento	d_4 Capitães de areia	d_5 Os Sertões
amarelo	1	42	6	3	0
baleia	0	86	0	0	0
casa	109	37	247	120	30
comitiva	4	0	0	0	4
dinheiro	7	9	33	43	3
médico	18	0	157	7	8
padre	22	0	120	252	9

Tabela 16.4: Exemplo de matriz de incidência de termos em documentos com valores de TF-IDF

Termo	d_1 O Alienista	d_2 Vidas Secas	d_3 O Tempo e O Vento	d_4 Capitães de areia	d_5 Os Sertões
amarelo	0,0009	0,0473	0,0024	0,0012	0,0000
baleia	0,0000	0,6990	0,0000	0,0000	0,0000
casa	0,0000	0,0000	0,0000	0,0000	0,0000
comitiva	0,0146	0,0000	0,0000	0,0000	0,0549
dinheiro	0,0000	0,0000	0,0000	0,0000	0,0000
médico	0,0160	0,0000	0,0616	0,0031	0,0258
padre	0,0196	0,0000	0,0471	0,0969	0,0291

Para computar os vetores de pesos TF-IDF para os cinco documentos da coleção, vamos iniciar com uma matriz de incidência dos termos nos documentos. Temos sete termos e cinco documentos; então, nossa matriz terá sete linhas e cinco colunas. Nessa matriz, ilustrada na Tabela 16.3, temos o número de ocorrências dos termos nos documentos. A seguir, vamos normalizar o componente TF dividindo-o pelo número de ocorrências do termo mais comum (i.e., o termo que mais ocorre no documento), que para d_1 é “casa”. Com isso, valor de TF-IDF para o termo “padre” no documento d_1 é $(22/109) \times 0,0969 = 0,0196$. Esses valores estão na Tabela 16.4. Note-se que os escores para os termos “casa” e “dinheiro” são iguais a zero pois eles aparecem em todos os cinco documentos, resultando em um $IDF = 0$. Também podemos observar que o maior escore é o do termo “baleia” no documento d_2 . Isso ocorre porque esse termo ocorre muitas vezes em d_2 e nenhuma vez nos outros quatro documentos – ele é, portanto, um termo que tem bastante poder de diferenciar esse documento dos demais.

Para exemplificar o cálculo do cosseno, vamos imaginar uma consulta $q = \{\text{comitiva}, \text{médico}\}$. Assim como os documentos, o vetor da consulta terá sete dimensões e será composto por pesos TF-IDF. Os termos que não aparecem na consulta têm peso zero. Assim, o vetor da consulta será $\vec{q} = [0; 0; 0; 0,3979; 0; 0,0961; 0]$. Para montar o *ranking* dos documentos em relação à consulta, calculamos o cosseno entre \vec{q} e os vetores dos documentos que estão na Tabela 16.4 utilizando a Equação 16.2. Por exemplo, o cosseno entre \vec{q} e \vec{d}_1 é:



$$\frac{0 \times 0,009 + 0 \times 0 + 0 \times 0 + 0,3979 \times 0,0146 + 0 \times 0 + 0,0969 \times 0,0160 + 0 \times 0,0196}{\sqrt{0^2 + 0^2 + 0^2 + 0,3979^2 + 0^2 + 0,0969^2 + 0^2} \times \sqrt{0,0009^2 + 0^2 + 0^2 + 0,0146^2 + 0^2 + 0,0160^2 + 0,0196^2}} =$$

$$\frac{0 + 0 + 0 + 0,0058 + 0 + 0,0016 + 0}{\sqrt{0 + 0 + 0 + 0,1583 + 0 + 0,0009 + 0} \times \sqrt{0 + 0 + 0 + 0,0002 + 0 + 0,0003 + 0,0004}} =$$

$$\frac{0,0074}{\sqrt{0,1677} \times \sqrt{0,0009}} = \frac{0,0074}{0,4096 \times 0,0292} = \frac{0,0074}{0,012} = 0,6156$$

Os escores de cosseno entre \vec{q} e os cinco documentos estão na Tabela 16.5. Note que como q e d_2 não têm nenhum termo em comum, o escore desse documento é zero. Para montar o *ranking*, ordenamos os documentos em ordem decrescente de cosseno. Dessa forma, o primeiro do *ranking* seria d_5 , seguido por d_1 , d_3 e d_4 .

Tabela 16.5: Cosseno entre o vetor da consulta q e os vetores dos documentos da coleção.

d_1	d_2	d_3	d_4	d_5
0.6156	0.0000	0.1879	0.0075	0.8765

16.3.3 Modelos Probabilísticos

Há diversos modelos que utilizam abordagens para RI baseadas em um arcabouço probabilístico. A principal motivação é que a teoria das probabilidades fornece uma base sólida para representar e manipular a incerteza que é inerente ao processo de casamento entre consultas e documentos.

A ideia original foi proposta por Rijsbergen (1979) e intitulada *Probability Ranking Principle*. A ideia é ranquear os documentos da coleção em ordem decrescente de probabilidade de relevância para a consulta. Se tivéssemos uma amostra de documentos sabidamente relevantes ou não relevantes, seria possível estimar a probabilidade de um termo aparecer em um documento relevante. Essa informação poderia então ser usada para ranquear os documentos. As probabilidades são estimadas com a maior precisão possível com base em estatísticas que conseguimos calcular. O primeiro modelo probabilístico foi proposto por Robertson; Spärck Jones (1976) – o *Binary Independence Model* (BIM). O BIM é um modelo binário, ou seja, ele considera apenas se o termo está ou não presente no documento. Outra simplificação é a utilização do princípio da independência, que supõe que os termos ocorrem nos documentos de maneira independente. Esse é um modelo simples que não incorpora algumas características, como a frequência dos termos nos documentos e a importância dos termos na coleção (vistas na Seção 16.3.2), que contribuem para a geração de resultados melhores para as consultas (i.e., uma melhor ordenação dos documentos).

O modelo probabilístico mais influente até os dias de hoje é o **Okapi BM25** que foi projetado entre os anos 1980 e 1990 por Spärck Jones; Walker; Robertson (2000). O Okapi era um sistema de busca para o catálogo da biblioteca da City University de Londres. O modelo BM25 (BM é a abreviatura de *best match*) foi resultado de uma série de experimentos com variações de fórmulas de modelos probabilísticos. Os pesquisadores usaram identificadores para as variações da fórmulas e a que obteve melhores resultados foi a BM25, que é dada conforme a Equação 16.3 a seguir:



$$BM25(q, d_j) = \sum_{i \in q} \log \frac{(r_i + 0,5)/(R - r_i + 0,5)}{(df_i - r_i + 0,5)/(N - df_i + r_i + 0,5)} \times \frac{(k_1 + 1)tf_i}{K + tf_i} \times \frac{(k_2 + 1)qf_i}{k_2 + qf_i} \quad (16.3)$$

onde R é o número de documentos sabidamente relevantes para a consulta q ; r_i é o número de documentos sabidamente relevantes que contêm o termo i ; N é o número de documentos da coleção; df_i é o número de documentos que contêm o termo i ; tf_i é o número de ocorrências do termo i no documento j ; qf_i é o número de ocorrências do termo i na consulta q ; K , k_1 e k_2 são parâmetros que precisam ser definidos empiricamente. A adição de 0,5 em cada termo da equação é para evitar a divisão por zero. Note que o somatório é efetuado para todos os termos da consulta.

O parâmetro k_1 controla a escala da frequência dos termos (tf) nos documentos. Usar $k_1 = 0$ faz com que o modelo se comporte como um modelo binário. Por outro lado, altos valores de k_1 correspondem ao uso do TF sem normalização. Na prática, é comum usarmos $k_1 = 1,2$. De forma similar, o parâmetro k_2 controla a escala da frequência dos termos na consulta (qf). Os valores recomendados para k_2 são entre 0 e 1000.

Por fim, o parâmetro K é calculado por:

$$K = k_1((1 - b) + b \times \frac{dl}{avdl})$$

onde dl é o tamanho do documento, $avdl$ é o tamanho médio dos documentos da coleção e b controla a normalização em função do tamanho do documento. Se $b = 0$, então a normalização não é realizada. Já $b = 1$ normaliza os pesos em função do tamanho dos documentos. O valor recomendado para b é 0,75.

É importante ressaltar que os escores gerados pelos modelos probabilísticos não são probabilidades, i.e., não estão no intervalo $[0,1]$. Isso não representa um problema pois a utilidade do escore é apenas ordenar os documentos, sem a necessidade de permitir uma interpretação probabilística.

Como na imensa maioria dos casos práticos nós não temos conhecimento de uma amostra de documentos relevantes, o BM25 também pode ser computado **sem a informação de relevância**. Em outras palavras, R e r_i são zero. Com isso, a Equação 16.3 pode ser simplificada para:

$$BM25(q, d_j) = \sum_{i \in q} \log \frac{N - df_i + 0,5}{df_i + 0,5} \times \frac{(k_1 + 1)tf_i}{K + tf_i} \times \frac{(k_2 + 1)qf_i}{k_2 + qf_i} \quad (16.4)$$

Exemplo 16.2. Para ilustrar o cálculo do BM25, vamos utilizar a mesma consulta da Seção 16.3.2 $q = \{\text{comitiva, médico}\}$ e faremos o cálculo do escore passo a passo para o d_1 . Não temos informação prévia de relevância, então vamos usar a Equação 16.4. Cada termo ocorre apenas uma vez na consulta, então $qf_1 = 1$ e $qf_2 = 1$. O escore final é formado pela soma dos escores das palavras da consulta que aparecem no documento d_1 . Para facilitar o entendimento, os cálculos referentes ao termo “comitiva” estão em laranja e os os cálculos referentes ao termo “médico” estão em azul. As estatísticas da coleção estão nas Tabelas 16.2, 16.3, 16.4. Os parâmetros utilizados foram $k_1 = 1,2$, $k_2 = 100$ e $b = 0,75$. O tamanho médio dos documentos da coleção ($avdl$) é de 276 e o tamanho de d_1 é 161. Com isso, $K = 1,2((1 - 0,75) + 0,75 \times 161/279) = 0,825$. Observe que o escore final para o documento é negativo, mas o valor absoluto não é importante pois estamos interessados apenas em ranquear os documentos. Calculando o escore para os outros quatro



documentos, verificamos que o primeiro documento do *ranking* seria d_5 , seguido por d_1 , d_4 e d_3 . Note-se que o ordenamento fica quase igual ao gerado pelo modelo vetorial – apenas os documentos d_4 e d_3 invertem as posições.

$$\begin{aligned}
 BM25(q, d_1) &= \log \frac{5 - 2 + 0,5}{2 + 0,5} \times \frac{(1, 2 + 1)4}{0,825 + 4} \times \frac{(100 + 1)1}{100 + 1} + \\
 &\quad \log \frac{5 - 4 + 0,5}{4 + 0,5} \times \frac{(1, 2 + 1)18}{0,825 + 18} \times \frac{(100 + 1)1}{100 + 1} = \\
 &\quad \log \frac{3,5}{2,5} \times \frac{8,8}{4,825} \times \frac{101}{101} + \log \frac{1,5}{4,5} \times \frac{39,6}{18,825} \times \frac{101}{101} = \\
 &\quad \log 1,4 \times 1,824 \times 1 + \log 0,33 \times 2,1 \times 1 = \\
 &\quad 0,33 \times 1,824 + -1,10866 \times 2,1 = \\
 &\quad 0,6137 - 2,311 = -1,6973
 \end{aligned}$$

16.4 Avaliação da Qualidade de Sistemas de recuperação de informação

A RI é uma disciplina altamente empírica e a história da avaliação em sistemas de RI nasceu junto com a área. Os primeiros trabalhos sobre avaliação em RI foram coordenados por Cyril W. Cleverdon na escola de aeronáutica de Cranfield (Inglaterra) nos anos 1960. Por esta razão, o modelo de avaliação até hoje é conhecido como paradigma Cranfield. A avaliação consiste no cálculo de uma série de métricas que são calculadas com base em uma coleção de teste. Nesta seção abordaremos as principais métricas (Seção 16.4.1) e as coleções de teste para português (Seção 16.4.2). O leitor que deseja se aprofundar no processo de avaliação de RI pode referir-se a Sanderson et al. (2010).

16.4.1 Métricas

As métricas são baseadas na noção de relevância, i.e., uma avaliação que diz se um documento d_j é relevante a uma consulta q . Inicialmente, vamos tratar a relevância como **binária**, ou seja, o *ground truth* julga o documento como relevante (1) ou não relevante (0). Todas as métricas vistas aqui variam no intervalo [0,1], sendo que 1 representa a recuperação ideal.

16.4.1.1 Métricas Baseadas em Conjuntos

A Figura 16.3 mostra um exemplo do que ocorre tipicamente quando fazemos uma consulta. A coleção de documentos está representada pelo oval cinza. O círculo amarelo representa os documentos que foram recuperados em resposta à consulta, enquanto que os documentos que de fato são relevantes para a consulta estão representados pelo círculo azul. Podemos ver que há documentos relevantes que foram recuperados (i.e., os documentos que estão na intersecção entre os conjuntos, que aparece representada na cor verde), mas há também documentos relevantes que deixaram de ser recuperados e documentos não relevantes que foram recuperados. Com base nesses conjuntos, podemos definir duas métricas básicas para avaliar a qualidade da recuperação: **precisão** e **revocação** (em inglês, *recall*).



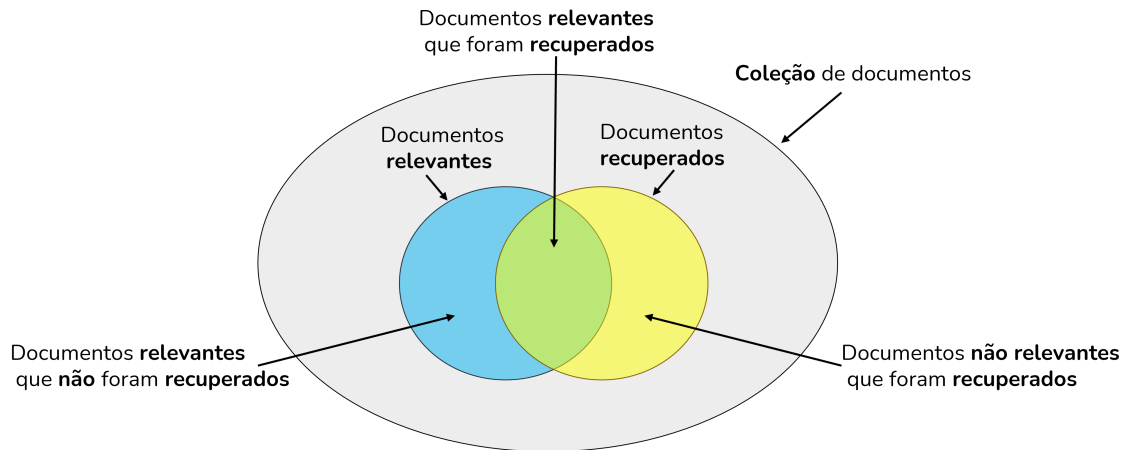
- Precisão (P) mede a proporção dos documentos recuperados que de fato é relevante.

$$P = \frac{\# \text{ relevantes recuperados}}{\# \text{ recuperados}}$$

- Revocação (R) mede a proporção dos documentos relevantes que de fato foram recuperados.

$$R = \frac{\# \text{ relevantes recuperados}}{\# \text{ relevantes}}$$

Figura 16.3: Exemplo de recuperação de documentos em resposta a uma consulta.



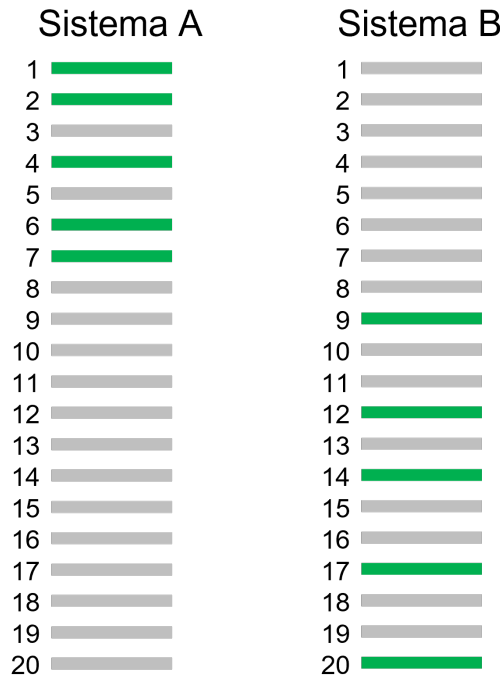
Por exemplo, vamos imaginar uma situação em que um sistema recupera 100 documentos em resposta a uma consulta para a qual existam 30 documentos relevantes. Sabendo-se que entre os recuperados há 15 documentos relevantes, os valores das métricas de avaliação seriam $P = \frac{15}{100} = 15\%$ e $R = \frac{15}{30} = 50\%$. Imaginemos, então, uma segunda situação em que, na tentativa de aumentar a precisão, reduzimos o número de documentos recuperados para apenas 5 e todos eles sejam relevantes. Como resultado, nossa precisão aumentaria para $P = \frac{5}{5} = 100\%$. O problema é que, com isso, a nossa revocação cairia para $\frac{5}{30} = 16,67\%$. Indo na direção oposta, poderíamos tentar aumentar a revocação e recuperar mil documentos em vez de 100. Com isso, haveria uma chance bem maior de recuperarmos documentos relevantes, mas por outro lado, a precisão seria reduzida. Em resumo: altos níveis de revocação costumam ser acompanhados por baixos níveis de precisão e vice-versa. Por esta razão, é comum utilizarmos uma medida que agregue P e R , a conhecida **medida F**, definida a seguir.

$$F = \frac{(\beta^2 + 1) \times P \times R}{(\beta^2 \times P) + R}$$

onde β é um parâmetro que permite definirmos a ênfase dada às métricas. Valores de β maiores do que 1 enfatizam a revocação enquanto que valores de β menores do que 1 enfatizam a precisão. Se quisermos atribuir a mesma ênfase às duas métricas, usamos $\beta = 1$. Nesse caso, a métrica é comumente chamada de $F1$ e sua fórmula é simplificada para $F1 = \frac{2 \times P \times R}{P + R}$. Considerando o nosso primeiro exemplo, em que $P = 0,15$ e $R = 0,50$, a $F1$ seria $F1 = \frac{2 \times 0,15 \times 0,50}{0,15 + 0,50} = 0,23$.



Figura 16.4: Exemplos de *rankings* de documentos recuperados em resposta a uma consulta por dois sistemas diferentes. Os documentos relevantes à consulta estão representados em verde.



16.4.1.2 Métricas para *Rankings*

Os sistemas de RI retornam os documentos em um *ranking*. A Figura 16.4 mostra exemplos de *rankings* de documentos recuperados em resposta a uma mesma consulta por dois sistemas diferentes (Sistemas A e B). São 20 documentos recuperados e, dentre eles, há 5 relevantes (representados em verde). Vamos supor que existam 7 documentos relevantes para essa consulta. As métricas vistas na Seção 16.4.1.1 resultariam em $P = \frac{5}{20} = 25\%$, $R = \frac{5}{7} = 71\%$ e $F1 = \frac{2 \times 0,25 \times 0,71}{0,25 + 0,71} = 0,37$ tanto para o Sistema A quanto para o Sistema B. Esse resultado não é o ideal pois podemos ver que o Sistema A recuperou os documentos relevantes mais perto do topo do *ranking* e por isso deveria receber um escore mais alto do que o Sistema B. Esse problema ocorre porque precisão, revocação e medida-F são baseadas em conjuntos – e conjuntos, por definição, não têm ordenação. Então concluímos que precisão, revocação e F1 **não são adequadas para avaliar *rankings***. Uma boa métrica para avaliar resultados ranqueados deve conseguir calcular quantos documentos relevantes foram recuperados e o **quão próximos estão do topo do ranking**. Com isso em mente, a métrica **precisão média** (do inglês, *average precision*, AP) foi proposta.

$$AP = \frac{\sum_{k=1}^n P(k) \times rel(k)}{\# \text{ relevantes}}$$

onde $rel(k)$ é uma função de relevância binária que resulta em 1 caso o documento na k -ésima posição do *ranking* seja relevante. Os documentos relevantes que são foram recuperados serão penalizados com $P = 0$. Em outras palavras, percorreremos o *ranking* calculando, para cada documento relevante recuperado, a precisão naquele ponto. Para o Sistema A da Figura 16.4, temos a seguinte $AP = \frac{1/1 + 2/2 + 3/4 + 4/5 + 5/7}{7} = 0,61$. E



para o Sistema B temos $AP = \frac{1/9 + 2/12 + 3/14 + 4/17 + 5/20}{7} = 0,14$. Concluimos, então, que a AP consegue capturar a fato do Sistema A ter sido superior ao B.

Quando temos um conjunto de consultas, calculamos a média das precisões médias (do inglês, *mean average precision*, MAP), que é simplesmente a média das APs para $|Q|$ consultas.

$$MAP = \frac{\sum_{q=1}^{|Q|} AP_q}{|Q|}$$

Muitas vezes, é útil fazermos uma comparação visual do desempenho de dois sistemas. Neste caso, podemos utilizar as **curvas de revocação e precisão**. Para desenhar as curvas, precisamos ter os valores de precisão calculados para 11 pontos de revocação padrão (de 0 a 1 com incrementos de 0,1). Para calcular a precisão nesses 11 pontos, é necessário utilizar uma regra de interpolação que diz que “a precisão interpolada para um nível de revocação j é o maior valor de precisão para qualquer nível de revocação maior ou igual a j ”. Aplicando-se a regra da interpolação, teremos curvas monotonicamente decrescentes, como as que vemos na Figura 16.5. Quanto maior a área sob a curva, melhor o resultado do sistema. Podemos ver claramente que a função A é pior do que as outras duas e que as funções B e C obtiveram resultados muito similares, com uma leve superioridade da função C . Para sabermos se as diferenças entre os sistemas são significativas, é comum fazermos testes estatísticos. O teste-T pareado é bastante usado e podemos fazê-lo comparando as precisões médias (AP) para um mesmo conjunto de consultas executadas em sistemas diferentes. Para os resultados das consultas que deram origem às curvas da Figura 16.5, quando comparamos os resultados da função C com a função A , o teste-T resulta em um p-valor = 4.6e-08. Utilizando um nível de significância $\alpha = 0.01$, a interpretação é que a função C é significativamente melhor do que a função A pois o p-valor é $\ll 0.01$. Já quando comparamos a função C com a função B , o p-valor é 0.12. Então concluimos que não há diferença significativa entre B e C pois o p-valor é > 0.01 .

Além da MAP, é comum avaliar-se *rankings* com a métrica $P@k$, onde k é uma posição específica do *ranking* (ex: 1, 5, 10, 100 etc.). Basta calcular o número de documentos relevantes encontrados até a k -ésima posição do *ranking*. Por exemplo, a $P@10$ conta quantos documentos relevantes foram encontrados até a décima posição do *ranking*. Essas métricas são especialmente úteis quando não sabemos o número de documentos relevantes para cada consulta, como é o caso das buscas na Web.

Outra métrica por vezes usada é o *mean reciprocal rank* (MRR) que calcula o inverso da posição do primeiro documento relevante retornado. Por exemplo, caso o primeiro documento relevante esteja na segunda posição do *ranking*, $MRR = 1/2$. Para agregar os resultados para um conjunto de consultas, fazemos a média:

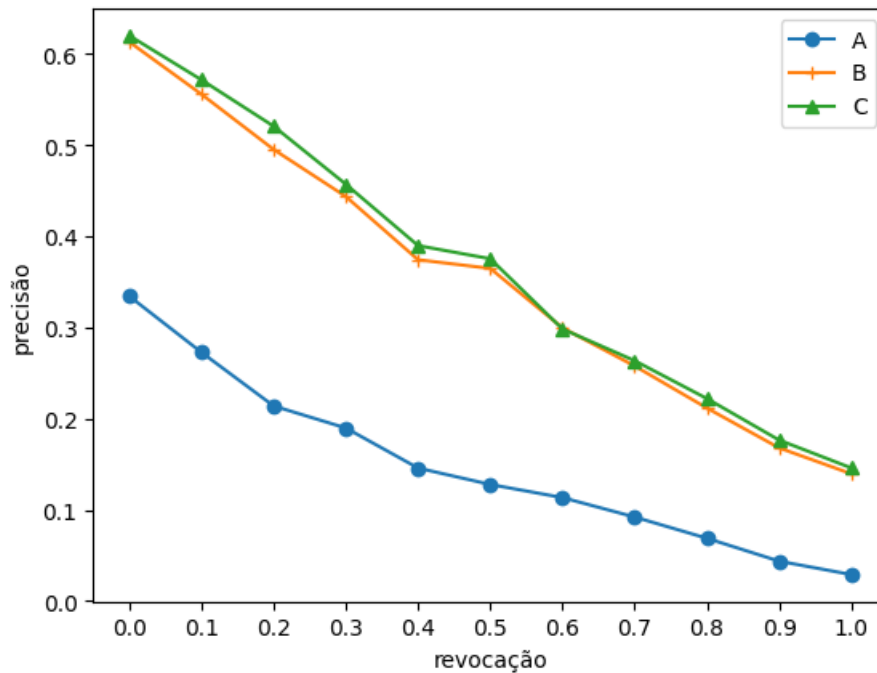
$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

Até agora, abordamos apenas métricas que utilizam a relevância como binária. Contudo, em alguns casos, é importante diferenciar um documento muito relevante de um documentos marginalmente relevante. Nesses casos, podemos calcular o **ganho acumulado descontado normalizado** (do inglês, *normalized discounted cumulative gain*, NDCG) (Järvelin; Kekäläinen, 2002). Para entender o NDCG, primeiro temos que entender o DCG e o IDCG.

$$DCG = \sum_{i=1}^n \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$



Figura 16.5: Exemplos de curvas de revocação e precisão para a comparação de três funções de ranking.



onde n é o número de documentos recuperados, i é a posição do documento no *ranking* e rel_i é o nível de relevância do documento (i.e., podemos usar quatro níveis onde o escore 3 indica um documento altamente relevante, 2 indica um documento moderadamente relevante e 1 indica um documento marginalmente relevante e 0 é um documento não relevante). O IDCG é o DGC ideal, ou seja, a pontuação de DCG obtida se a ordenação fosse perfeita. Em outras palavras, primeiramente teríamos todos os documentos de escore 3, seguidos pelos de escore 2, depois os de escore 1 e por fim os de escore zero. O NDCG é a fração $\frac{DCG}{IDCG}$. A normalização é importante para fazer com que a métrica fique no intervalo $[0,1]$.

Além dessas medidas, muitas outras continuam sendo propostas e apresentadas nos fóruns de RI. Contudo, ainda não são amplamente adotadas. Há também muitos estudos sobre a validade e estabilidade das métricas, como por exemplo o trabalho de Buckley; Voorhees (2017).

16.4.2 Coleções de Teste

O cálculo das métricas apresentadas na Seção 16.4.1 só é possível caso exista uma coleção de teste com três componentes: (i) um conjunto de documentos, (ii) um conjunto de tópicos de consulta e (iii) os julgamentos de relevância. O primeiro componente é o mais fácil de ser obtido, pois é trivial fazer uma coleta a partir da Web e obter um grande número de documentos. O segundo componente, i.e., as consultas, necessita da intervenção humana na sua elaboração pois é preciso imaginar o tipo de consulta que um usuário que estivesse pesquisando a coleção de documentos poderia fazer. É comum a avaliação de consultas ad hoc utilizar por volta de 50 tópicos de consulta. O maior custo na elaboração



de uma coleção de testes está no terceiro componente, i.e., os julgamentos de relevância. Os julgamentos de relevância dizem quais são os documentos relevantes para cada tópico de consulta e necessitam de intervenção humana na sua elaboração.

A primeira coleção de testes elaborada, a Cranfield Collection, tem 1398 resumos de artigos sobre aerodinâmica e 225 tópicos de consulta. A elaboração dos julgamentos de relevância foi exaustiva, ou seja, a relevância de todos os documentos foi avaliada em relação a cada consulta. Além disso, cada par {documento, consulta} foi julgado por dois avaliadores – ou seja $1398 \text{ documentos} \times 225 \text{ consultas} \times 2 \text{ avaliadores} = 629.100$, i.e., mais de meio milhão de avaliações para uma coleção muito pequena. Aplicar essa mesma abordagem em uma coleção com 100 mil documentos e 50 consultas, demandaria 10 milhões de julgamentos de relevância, o que seria impraticável.

A solução amplamente adotada foi a criação de um método em que nem todos os documentos são avaliados em relação às consultas – apenas os documentos que têm alguma chance de serem relevantes. Essa foi a ideia do método *pooling* (Spärck Jones, 1975) que tem esse nome pois são criados repositórios (*pools*) com o subconjunto de documentos que serão avaliados. O processo envolve a criação de um repositório para cada consulta. Os repositórios são populados com documentos recuperados utilizando diferentes funções de *ranking*, variando-se parâmetros e sistemas. Os documentos do repositório são então ordenados de acordo com alguma heurística – por exemplo documentos que foram retornados mais vezes ficam no topo do *ranking*. Então, apenas os n primeiros são analisados pelos avaliadores humanos. Os documentos que não foram avaliados são considerados como não relevantes. É possível que essa estratégia deixe de identificar alguns documentos relevantes, mas há evidências de que ainda assim ela é robusta. Um estudo realizado por Zobel (1998) identificou que a ordenação entre funções de *ranking* (ou sistemas) permanecia a mesma ainda que o conjunto de julgamentos de relevância mudasse. Em outras palavras, a superioridade de um método em relação a outro é mantida mesmo com conjuntos diferentes de julgamentos de relevância.

Mesmo com a grande redução no esforço manual de avaliação na criação de coleções trazida pelo método *pooling*, este processo ainda é bastante trabalhoso. As coleções de teste foram criadas, na imensa maioria das vezes, dentro do escopo de campanhas de avaliação. Essas campanhas tiveram um grande impacto no desenvolvimento de técnicas para RI. Para a língua inglesa, as campanhas TREC (*Text REtrieval Conference*)¹² mantiveram tarefas de recuperação ad hoc entre os anos de 1992 e 1999. Posteriormente, uma iniciativa europeia deu origem às campanhas CLEF (*Cross Language Evaluation Forum*) que entre 2000 e 2009 organizaram tarefas de recuperação ad hoc para idiomas europeus. Semelhante a essas campanhas, há o NTCIR¹³ dedicado aos idiomas do leste da Ásia.

Foi no contexto das campanhas CLEF e por iniciativa da Linguateca¹⁴ que a primeira coleção de teste para a língua portuguesa foi elaborada – a coleção CHAVE¹⁵ (Santos; Rocha, 2004). A coleção CHAVE é composta por cerca de 209 mil notícias dos jornais Folha de São Paulo e Público (de Portugal), publicadas nos anos de 1994 e 1995. Há 100 tópicos de consulta com os seus respectivos julgamentos de relevância. Por muitos anos, esta foi a única coleção de teste disponível para português e sem dúvida ainda representa o recurso mais valioso para a recuperação ad hoc neste idioma.

A segunda coleção de teste para a recuperação ad hoc na língua portuguesa é a coleção

¹²<https://trec.nist.gov/>

¹³<https://research.nii.ac.jp/ntcir/>

¹⁴<https://www.linguateca.pt/>

¹⁵<https://www.linguateca.pt/CLEF/>



REGIS¹⁶ (*Retrieval Evaluation for Geoscientific Information Systems*). Ela é composta por teses, dissertações e artigos científicos no domínio geocientífico. São mais de 20 mil documentos e 34 tópicos de consulta. Uma característica distintiva é que os documentos são bem mais longos do que os encontrados em coleções de notícias de jornal e tratam de um domínio mais restrito. Além disso, os julgamentos de relevância foram feitos em quatro níveis (muito relevante, moderadamente relevante, marginalmente relevante e não relevante).

Recentemente, uma versão traduzida da MS MARCO (*Microsoft Machine Reading Comprehension*)¹⁷ foi disponibilizada para português e mais 12 idiomas (Bonifacio et al., 2021). A MS MARCO contém um conjunto de *datasets* que são amplamente utilizados no treinamento de algoritmos de aprendizado profundo. Para RI, há uma coleção com 8,8 milhões de documentos e 6980 consultas. As consultas foram obtidas a partir de buscas reais submetidas ao Bing e os documentos são passagens curtas extraídas de páginas web.

O Quadro 16.2 mostra exemplos de tópicos de consulta das três coleções de teste existentes para a língua portuguesa. O formato adotado pelas coleções CHAVE e REGIS é o formato tradicional utilizado pelas campanhas TREC e CLEF. Cada tópico de consulta é composto por um identificador, um **título** que descreve sucintamente o tópico, uma **descrição** mais detalhada e uma **narrativa** que visa auxiliar o avaliador humano a distinguir os documentos relevantes dos não relevantes. Já os tópicos da MS MARCO são diferentes, há apenas um identificador e o texto da consulta submetida ao motor de busca.

Quadro 16.2. Exemplos de tópicos de consulta das coleções de teste de RI em português

<p>CHAVE</p> <pre><top> <num> C267 </num> <PT-title> Melhor Filme Estrangeiro </PT-title> <PT-desc> Quais foram os filmes candidatos ao Oscar de Melhor Filme Estrangeiro? </PT-desc> <PT-narr> Documentos relevantes devem indicar o título e nacionalidade dos filmes indicados para o Óscar na categoria de Melhor Filme em Língua Estrangeira. </PT-narr> </top></pre>
<p>REGIS</p> <pre><top> <num>Q5</num> <title>Permeabilidade em Marlim</title> <desc>Informações sobre o campo de Marlim, mas não dos campos de “Marlim Sul” ou de “Marlim Leste”.</desc> <narr>Interessam documentos que contenham informação sobre a permeabilidade das rochas do campo de Marlim. Principalmente, interessam informações quantitativas (como dados de permeabilidade expressos em Darcies ou milidarcies).</narr> </top></pre>
<p>MS MARCO</p> <hr/> <p>350,732 como limpar copos de melamina manchados</p>

¹⁶<https://github.com/Petroles/regis-collection>

¹⁷<https://microsoft.github.io/msmarco/>



1,067,511	por que meu ligamento colateral lateral dói
906,071	o que adicionar à sua água para ser mais saudável

16.5 Modificação Automática de Consultas

Conforme visto na Introdução, o problema central da RI é casar a consulta com os documentos que a satisfazem. Contudo, o casamento puramente léxico aplicado pelos Modelos Clássicos (vistos na Seção 16.3) implica que uma consulta somente recupera documentos que possuam alguma de suas palavras-chave. A consulta não é capaz de recuperar um documento que contenha apenas sinônimos das suas palavras-chave e isso tem um impacto negativo na revocação. Por exemplo, se um usuário consulta “tributos pagos por donos de carros” um documento que não contenha as palavras da consulta mas mencione “imposto sobre a propriedade de veículos automotores” não seria recuperado. Desde os anos 1960, os pesquisadores vêm propondo formas de solucionar esses problemas. Essa sub-área de pesquisa é conhecida como **modificação automática de consultas**¹⁸ (*automatic query modification*) AQM e conta com dezenas de milhares de artigos publicados. Esta seção apresenta um breve resumo e fornece referências para que o leitor possa encontrar mais informações. A AQM é um ponto forte de interseção entre RI e PLN pois os diversos métodos adotados em PLN para solucionar o problema da sinonímia encontram aqui uma ótima área de aplicação.

Um bom ponto de partida para quem quer se inteirar-se sobre esse tema é o *survey* de Carpineto; Romano (2012). vamos adotar a taxonomia introduzida nesse trabalho para classificar as abordagens para AQM.

- **Análise Linguística.** Uma das técnicas mais simples que pode ser enquadrada nessa categoria é o uso de *stemming* (ver Seção 16.2.1.3). O *stemming* reduz as formas variantes de uma palavra ao mesmo radical e assim conseguiria, por exemplo, que uma consulta pelo termo “imposto” recuperasse documentos com a forma plural “impostos”. Contudo, o *stemming* não é capaz de recuperar um documento que contenha apenas o sinônimo “tributo”. Para isso, poderíamos nos valer de dicionários de sinônimos, tesouros ou ontologias. A modificação da consulta consiste em adicionar termos sinônimos à consulta original. A adição de sinônimos irá recuperar mais documentos. O problema ocorre quando os sinônimos adicionados mudam o significado da consulta, por exemplo “homenagem” é sinônimo de “tributo”, mas sua adição à consulta iria trazer documentos irrelevantes, baixando a precisão. Alguns domínios como a medicina dispõe de tesouros que podem ser diretamente usados. Por outro lado, essa é uma exceção – a maioria dos domínios não dispõem desses recursos e a sua construção manual é cara. Uma solução seria descobrir automaticamente essas relações a partir de *corpora*.
- **Técnicas baseadas em *corpora*.** Esse grupo de técnicas analisa o conteúdo da coleção de documentos para extrair padrões de coocorrência de termos. A ideia é que termos que coocorrem (i.e., que frequentemente aparecem juntos nos mesmos documentos) estão correlacionados e a sua adição à consulta pode fornecer contexto adicional. A informação de coocorrência pode ser obtida a partir de métodos

¹⁸O termo **expansão automática de consultas** é bastante empregado. Contudo, a consulta nem sempre é expandida e, por isso, preferimos adotar o termo **modificação automática de consultas**.



diferentes como a Indexação Semântica Latente (Deerwester et al., 1990) ou *word embeddings* (Mikolov et al., 2013b) discutidas no Capítulo 10. Há experimentos que obtiveram resultados positivos (Kuzi; Shtok; Kurland, 2016), contudo se as embeddings forem obtidas a partir de textos genéricos e a coleção for de um domínio específico, os termos adicionados podem piorar a qualidade do *ranking*.

- **Técnicas baseadas no resultado da consulta** analisam os documentos recuperados pela consulta inicial para então modificá-la. A primeira técnica nessa categoria é a **realimentação de relevância** (em inglês, *relevance feedback*) (Rocchio-Jr, 1971). Ela consiste em solicitar que o usuário marque alguns documentos que ele considera relevantes e não relevantes. A reformulação da consulta consiste em modificar seu vetor para que ele se aproxime mais dos vetores dos documentos marcados como relevantes e se afaste dos vetores dos documentos não relevantes. Essa técnica costuma obter bons resultados, mas o ponto negativo é que os usuários não gostam de investir seu tempo fornecendo feedback. A alternativa então é supor que os primeiros *k* documentos recuperados pela consulta original sejam relevantes e modificar o vetor da consulta com base neles. Este processo é conhecido como **pseudo realimentação de relevância** e pode ter o efeito negativo de piorar ainda mais os resultados caso o primeiro conjunto de documentos recuperados não tenha sido bom. Outras propostas que se enquadram nessa categoria incluem o uso de *word embeddings* geradas a partir do resultado da consulta (Diaz; Mitra; Craswell, 2016).
- **Análise de logs de consulta** de motores de busca pode fornecer termos que os usuários frequentemente adicionam às suas consultas. Esses termos podem então ser usados para expandir a consulta original. Essa abordagem mostrou resultados positivos em (Cui et al., 2002).

Mais recentemente, uma nova gama de trabalhos têm focado no caminho inverso, i.e., expandir os documentos. Nessa linha de investigação, destaca-se o trabalho de Nogueira et al. (2019) em que os documentos são enriquecidos com consultas que poderiam ser feitas com o intuito de recuperar o documento.

16.6 Ferramentas e Bibliotecas

Há várias ferramentas (ou sistemas) de RI disponíveis tanto para fins comerciais como para fins de pesquisa. Esta seção aborda algumas das mais utilizadas.

16.6.1 Sistemas Comerciais

Os sistemas comerciais mais populares são da Apache e baseiam-se no **Lucene**¹⁹, uma biblioteca Java de código aberto. A biblioteca não é um sistema de RI completo. Com base nela, foram desenvolvidos o **Solr**²⁰ e o **Elasticsearch**²¹ que fornecem interfaces de consulta mais amigável além de indexação distribuída para fins de escalabilidade. Tanto o Solr como o Elasticsearch são ferramentas de código aberto, possuem vasta documentação e são amplamente adotados na indústria. O Elasticsearch fornece melhor suporte a sistemas distribuídos e pode ser integrado com ferramentas de análise e visualização. Por outro lado, o Solr possui mais flexibilidade para customização.

¹⁹<https://lucene.apache.org/>

²⁰<https://solr.apache.org/>

²¹<https://www.elastic.co/>



16.6.2 Ferramentas para Pesquisa

Há várias ferramentas de código aberto construídas por pesquisadores da área de RI que estão disponíveis e são usadas em trabalhos acadêmicos. Dentre elas, as mais utilizadas atualmente são:

- O **Anserini**²² e sua interface em Python (Pyserini²³) foram elaborados por pesquisadores das universidades de Delaware e Waterloo (Yang; Fang; Lin, 2017) e tem o objetivo de facilitar o processo experimental em RI com ênfase na reprodutibilidade.
- O **Terrier**²⁴ e sua interface em Python, o Pyterrier²⁵ (Macdonald; Tonellotto, 2020), foram criados por pesquisadores da universidade de Glasgow que tem longa tradição na pesquisa em RI. O Terrier também dispõe de uma interface gráfica que permite indexar e consultar uma coleção sem a necessidade de escrever código.
- O projeto **Lemur**²⁶ reúne uma série de componentes. O Indri é o motor de busca em si. O projeto é uma colaboração entre as Universidades de Massachusetts e Carnegie Mellon. O sistema Galago também é parte desse projeto e foi disponibilizado em conjunto com o livro de Croft; Metzler; Strohman (2010).

16.7 Conclusão

Este capítulo forneceu uma visão geral sobre a área de RI. O processo típico de RI foi introduzido, bem como os modelos clássicos. Damos ênfase à metodologia de avaliação dos resultados das consultas, dada a importância que esse tema tem na área. Também foram apontadas coleções de teste em português e ferramentas que podem ser usadas tanto no meio acadêmico como na indústria.

Nessa primeira versão, o capítulo não abordou um tópico relevante e atual: a RI utilizando vetores densos. Em especial, o emprego da arquitetura Transformers (Vaswani et al., 2017) tem sido bastante difundido em RI (assim como em diversas tarefas de PLN). Por ora, o leitor interessado pode referir-se ao artigo de (Lin; Nogueira; Yates, 2020) que apresenta um levantamento abrangente.

Agradecimentos

Agradeço a Helena Caseli e a Graça Nunes pelo convite a contribuir com este livro. Também sou grata a Adriana Pagano, Felipe Paula, Lucas Pessutto, Luciana Bencke e João Comba pela revisão e comentários.

²²<https://github.com/castorini/anserini>

²³<http://pyserini.io/>

²⁴<http://terrier.org/>

²⁵<https://github.com/terrier-org/pyterrier>

²⁶<https://lemurproject.org/>



Capítulo 17

Extração de Informação

Daniela Barreiro Claro
Joaquim Santos
Marlo Souza
Renata Vieira
Vlândia Pinheiro

17.1 Introdução

A Extração de Informação (EI) é desenvolvida com o objetivo de se obter informação estruturada de dados não-estruturados (Jurafsky; Martin, 2023; Konstantinova, 2014).

Os primeiros trabalhos a debruçarem-se sobre o problema remontam à década de 1970, com a aplicação de gramáticas formais e *parsers* sintáticos para a estruturação de informação em domínios como prontuários médicos (Sager, 1978; Sager; Friedman; Lyman, 1987) e textos jornalísticos (DeJong, 1979). A comunidade científica demonstrou grande interesse pela área nas décadas posteriores devido à sua utilidade prática, seu foco no processamento de dados reais, suas tarefas bem-definidas e a facilidade de mensurar a qualidade dos resultados em comparação com o desempenho humano na mesma tarefa (Cowie; Lehnert, 1996).

Para autores como Eisenstein (2019) e Jurafsky; Martin (2023), a EI é normalmente dividida em diversas tarefas de interesse, com foco no tipo de informação a ser extraída do texto. Entre as mais comumente citadas na literatura estão o Reconhecimento de Entidades Nomeadas (REN), a Extração de Relações (ER) e a Extração de Eventos (EE).

O Reconhecimento de Entidades Nomeadas (REN) consiste em identificar e classificar entidades mencionadas em textos através de designadores rígidos como nomes próprios, expressões temporais e espécies biológicas (Nadeau, 2007). Esse é considerado por alguns como um primeiro passo na análise semântica de um texto (Santos; Cardoso, 2007a), pois permite identificar as entidades às quais se faz referência nele.

A Extração de Relações (ER), também chamada de extração de informação tradicional ou somente extração de informação, por sua vez, diz respeito à identificação de relacionamentos semânticos entre duas ou mais entidades, ou seja, identificar “quem fez o que para quem e quando”. Ananiadou; Mcnaught (2005) a definem como o processo de extrair fatos (em nossa terminologia, relacionamentos) a partir de uma fonte textual e representá-los a partir de um gabarito (em inglês, *template*). As *relações* são elementos essenciais para o entendimento da informação relatada no texto e sua identificação é passo essencial para a estruturação da mesma. Assim, identificar relações entre entidades é tarefa essencial para construção de bases de conhecimento e de grande utilidade na construção de soluções para a resposta automática a perguntas (em inglês, *query answering*), sumarização, recuperação de informação e mais (Nasar; Jaffry; Malik, 2021).



A extração de eventos consiste na tarefa de identificação de uma menção a um evento em uma sentença e, se existirem, extração de outras informações sobre o evento. Um evento pode, por sua vez, ser entendido como uma ocorrência específica envolvendo participantes (Consortium, 2005), i.e., algo que acontece e que pode ser descrito como uma mudança de estado da qual participam entidades como agentes. Devido a intrínseca natureza temporal dos eventos, tal problema possui uma natureza mais complexa e costuma possuir tratamento específico.

Assim, nesse capítulo, iniciaremos com um pouco de história da Extração de Informação (EI) e sua evolução para Extração de Informação Aberta, e destacaremos as tarefas de Reconhecimento de Entidades Nomeadas (REN) e Extração de Relação (ER).

17.2 Um pouco de história

Os primeiros trabalhos que abordaram o problema de EI dos quais temos conhecimento surgiram no final da década de 1970. Esses primeiros trabalhos da década de 1970 e 1980 tinham como modelo geral a aplicação de regras para a identificação de informações especificadas em um gabarito. Tais sistemas empregavam analisadores sintáticos (*parsers*) e regras definidas especificamente para o domínio e gênero textual estudado.

Entre esses primeiros trabalhos, estão aqueles de Sager (1978), Sager; Friedman; Lyman (1987), de DeJong (1979) e de Cowie (1983). Sager et al. exploraram como identificar informações do estado de saúde de pacientes através dos textos de prontuários médicos. DeJong (1979), por sua vez, descrevem o sistema FRUMP que, a partir de um *parser* e regras de análise conceitual baseadas em uma arquitetura cognitiva proposta pelos autores e no conceito de dependência conceitual de Schank et al. (1973), processavam textos de notícias e realizavam tarefas como sumarização e identificação de papéis semânticos associados aos constituintes da sentença. Cowie (1983), por fim, descreve um sistema que emprega regras simples de segmentação e análise sintática rasa para identificar propriedades de plantas a partir de textos descritivos no campo da botânica. Diferente dos métodos anteriores, o trabalho dos autores se baseia em grande parte no estudo de padrões de descrição das informações a serem identificadas, em detrimento do emprego de *parsers* robustos da língua.

A década de 1990 traz um grande interesse na área de EI com a implementação das conferências MUC (do inglês, *Message Understanding Conference*, ou Conferência de Compreensão de Mensagem), promovidas pela Agência de Projetos de Pesquisa Avançada de Defesa (DARPA, do inglês *Defense Advanced Research Projects Agency*). As conferências MUC, realizadas e financiadas pelo exército americano, representaram um esforço em avançar a tecnologia de EI e consistiam de tarefas de avaliação conjunta de métodos desenvolvidos por pesquisadores para problemas propostos pelos organizadores. As sete conferências realizadas de 1987 a 1997, foram cruciais para definir aspectos centrais da área, como estruturar a tarefa de ER, definindo suas métricas de avaliação, e propor a tarefa de REN (Grishman; Sundheim, 1996).

A partir da MUC-3, em 1991, a conferência passa a ter foco no processamento de textos jornalísticos em detrimento dos relatórios militares utilizados anteriormente (DARPA, 1991). Com a disponibilidade de dados e o incentivo no desenvolvimento de soluções para a tarefa, vemos na década de 1990 o surgimento das primeiras aplicações comerciais de EI, como o JASPER (Andersen et al., 1992), construído para a agência de notícias Reuters.

A MUC-6, ocorrida em 1995, introduz a tarefa de REN com o intuito de ser uma tarefa de uso prático, independente de domínio e que poderia ser realizada automaticamente



em um futuro próximo (Grishman; Sundheim, 1996). Enquanto os trabalhos em REN se avolumaram a partir de sua proposição na MUC-6, trabalhos anteriores como Rau (1991) e Wolinski; Vichot; Dillet (1995) já se debruçavam sobre o problema de identificação e classificação de nomes próprios. Desde então, o interesse na tarefa cresceu significativamente e outras conferências de avaliação conjunta têm sido dedicadas a essa tarefa, como a *Automatic Content Extraction* (ACE) e a conferência Avaliação de Sistemas de Reconhecimento de Entidades Mencionadas (HAREM), dedicada exclusivamente à língua portuguesa, com sua primeira edição em 2005 (Santos; Cardoso, 2007a).

Por outro lado, houve um crescimento de abordagens baseadas em dados nesta década, a partir da análise de *corpora*. Tais esforços são impulsionados pelos resultados positivos na área, como o trabalho de Hearst (1992). Logo, métodos baseados em dados passaram também a explorar o emprego de análise estatística e aprendizado de máquina na construção de padrões para a extração de relações (Riloff et al., 1993; Riloff; Jones; et al., 1999; Roark; Charniak, 2000; Soderland et al., 1995).

Não foi somente na extração de padrões que métodos de aprendizado de máquina, em particular aprendizado supervisionado, foram aplicados. A década de 2000 viu a proliferação de métodos supervisionados aplicados à ER (Culotta; McCallum; Betz, 2006; Kambhatla, 2004; Zelenko; Aone; Richardella, 2003; Zhao; Grishman, 2005) e ao REN (Asahara; Matsumoto, 2003; McCallum; Li, 2003; Sekine, 1998).

Devido à dificuldade de construção de dados para treinamento e padrões para extração, além da pouca adaptabilidade dos sistemas construídos para outros escopos e domínios, nos anos 2000, sistemas baseados em métodos de aprendizado semi-supervisionado, como o DIPRE (Brin, 1998) e Snowball (Agichtein; Gravano, 2000) começaram a aparecer, juntamente com os estudos sobre expansão automatizada de anotações (*bootstrapping*) (Riloff; Jones; et al., 1999). Também para entidades nomeadas, estudos investigaram como utilizar recursos da Web (Etzioni et al., 2005; Nadeau, 2007) ou *corpora* (Cucchiarelli; Velardi, 2001) para aprender entidades com pouco ou nenhum esforço de anotação.

Buscando superar as dificuldades da limitação de escopo, i.e. das relações-alvo a serem extraídas e categorias de entidades a serem identificadas, ainda restritas à definição de padrões desde a criação dessas tarefas, Banko et al. (2007) propõe a tarefa de extração de informação aberta (EIA), também conhecida como **Open Information Extraction**, OpenIE ou OIE, a qual busca extrair todas as relações possíveis expressas em um texto, sem necessidade de pré-definição de relações e entidades.

Devido ao recente sucesso da aplicação de métodos baseados em redes neurais, em particular *deep learning* e grandes modelos de linguagem, às tarefas de Processamento de Linguagem Natural, uma tendência atual da área se delineou como o estudo de arquiteturas neurais para os problemas de EI e a geração de grandes conjuntos de dados por supervisão fraca. *Surveys* recentes, como (Cui; Wei; Zhou, 2018; Konstantinova, 2014; Nasar; Jaffry; Malik, 2021), nos mostram a evolução da área em direção à aplicação de métodos neurais. Na vertente de geração de dados, vemos o emprego da Wikipédia e Freebase como fontes mais usadas para obter anotações de entidades e relações em textos (Nguyen; Theobald; Weikum, 2016; Smirnova; Cudré-Mauroux, 2018; Takamatsu; Sato; Nakagawa, 2012).

Porém, toda a tarefa de EI necessita de uma concordância entre as definições de Entidade e Relação. Neste sentido, a próxima seção discute a conceituação de relação adotada neste capítulo, assim como o conceito de entidade.



17.3 Conceituação formal: Relação e Entidade

A natureza das relações estudadas na área de Extração de Informação e os critérios para reconhecer sua ocorrência em um texto têm recebido pouca atenção na literatura. Este é um passo importante para estabelecer metodologias adequadas para avaliar os sistemas, bem como para criar conjuntos de dados que possam apoiar a criação de sistemas futuros.

Enquanto as noções de Relação e Entidade são de grande importância e já bem estudadas nas áreas de Computação, Linguística, Ciência da Informação e Filosofia da Linguagem, esses conceitos não são empregados de forma consistente entre as áreas, ou mesmo entre suas subáreas.

17.3.1 Entidade

Para Chen (1976), uma entidade é um objeto que pode ser concreto, tal como pessoa, livro, casa ou ainda abstrato, tal como um emprego, um sentimento, uma disciplina. As entidades podem estabelecer relações entre si. Duas ou mais entidades são vinculadas, ou seja conectadas por uma relação¹.

Tradicionalmente em reconhecimento de entidades nomeadas, as entidades consideradas são aquelas referenciadas por um nome próprio, acrescidas das referências temporais e valores que são expressões numéricas. Essas expressões, portanto, geralmente não constituem uma entrada em uma base lexical. Porém a tarefa se expandiu para domínios especializados, onde as entidades de interesse são mais conceituais. No domínio bio-médico por exemplo, podemos ter como exemplo de entidades de interesse, sintomas e tratamento que não são referenciadas por nomes próprios.

17.3.2 Relação

Os conceitos de *relação* e *relacionamento* são noções fundamentais que vêm sendo estudadas em áreas como Ciência da Computação, Linguística e Filosofia.

No campo de bancos de dados e modelagem conceitual, Chen (1976) define um relacionamento, no contexto da modelagem de Entidade-Relacionamento, como uma associação entre entidades. Guarino; Guizzardi (2015), por sua vez, estudando a natureza ontológica dos relacionamentos com base na semântica de veridadores (*truthmaker semantics*) (Fine, 2017), postulam relacionamentos como entidades que atuam como veridadores (*truthmakers*) de alguma proposição relacionando duas ou mais entidades, ou seja, *uma relação mantida entre essas entidades*. Um verificador é um elemento cuja existência torna verdadeira uma proposição particular. Por exemplo, considerando a sentença (1) “*a* é uma maçã”, a existência de um objeto denotado pelo nome *a* que por acaso é uma maçã é uma condição suficiente para a verdade da frase (1). Como tal, dizemos que esse objeto é o verificador de (1). Tal definição nos permite adotar critérios ontológicos para validar a existência de relacionamentos a partir da informação relatada em um texto e, por isso, adotaremos tal definição de relacionamento neste capítulo.

O conceito de relações é muito menos consistente na literatura. Ainda na área de modelagem conceitual, Guarino; Guizzardi (2015) definem as relações como proposições para as quais os relacionamentos são veridadores e, portanto, possuem conteúdo proposicional. Assim, podemos entender uma relação como um tipo para entidades como relacionamentos. Ou seja, relações são universais ontológicos que descrevem a natureza dos relacionamentos.

¹Em nossa terminologia, por um relacionamento.



Xavier; Lima; Souza (2015), no entanto, argumentam que a noção de relacionamento adotada na área de Extração de Informação é mais geral do que isso, não se limitando àquelas entre objetos e propriedades, mas também àquelas que descrevem ou implicam propriedades de classes gerais como descrito pela sentença (2) “Filósofos são autores de Livros”. Assim, para o contexto de EI consideramos relações como tipos de relacionamentos de primeira ou segunda ordem. Isso significa que uma relação é um tipo de relacionamento que existe entre objetos, suas propriedades e classes de objetos ou suas propriedades.

Enquanto os métodos tradicionais de Extração de Informação dependem de um conjunto pré-existente de relações semânticas bem definidas que são relevantes para um domínio específico, a noção de “relação” e “entidade” na literatura da área mais recente, tais como a Extração de Informação Aberta, requer mais aprofundamento por demandar um significado diferente, principalmente com diferentes visões de autores. Esta indeterminação terminológica pode trazer problemas para comparar os resultados dos métodos propostos ou para reutilizar os conjuntos de dados criados na área.

As seções seguintes exploram essas duas áreas: Extração de Informação e Extração de Informação Aberta.

17.4 Extração de Informação (EI)

A Extração de Informação é caracterizada por obter informação estruturada a partir de textos, sendo entidades ou fatos, i.e. relacionamentos entre entidades, de tipos previamente definidos, conforme exemplo na Quadro 17.1. Métodos com limitação de escopo possuem como uma de suas principais desvantagens a necessidade de intervenção humana para especificar novos fatos a serem extraídos. Esta limitação impede que sistemas de Extração de Informação, doravante denominados de EI tradicional extraiam fatos fora do escopo pré-definido.

Quadro 17.1. Exemplos de relações específicas na EI tradicional

Relação Específica	Exemplo de Sentença	Extração
location-of(algo/alguém, local)	Um aluno pode ser encontrado na escola	location-of(aluno, escola)
is-a(subclasse, superclasse)	Salvador é uma cidade	is-a(Salvador, cidade)
part-of(todo, parte)	Roda é um componente de um carro	part-of(roda, carro)

Fonte: (Souza; Claro, 2014)

17.4.1 Reconhecimento de Entidades Nomeadas

O Reconhecimento de Entidades Nomeadas (REN) consiste na tarefa de identificar e classificar expressões linguísticas, denominadas entidades nomeadas (EN), que referenciam entidades específicas num domínio de discurso, como nomes próprios, expressões temporais e espécies biológicas (Mota; Santos; Ranchhod, 2007; Nadeau, 2007). De uma forma geral, o REN pode ser dividido em duas etapas: a identificação (ou delimitação) da expressão, na qual as palavras que formam a EN são selecionadas; a classificação, em que é atribuída uma categoria semântica à EN.



A classificação das ENs determina os tipos de entidades a serem consideradas e são especificadas a partir do escopo definido previamente para a tarefa. Algumas das categorias mais comumente utilizadas incluem as entidades que referenciam Pessoas Singulares (antropônimos); Coletivas (empresas e organizações) e Lugares (topônimos) (Mota; Santos; Ranchhod, 2007). Para exemplificar tomemos a sentença: “Renata Silva e Maria Costa palestraram na Universidade Federal da Bahia”. No exemplo temos três ENs: “Renata Silva”, “Maria Costa”, “Universidade Federal da Bahia”, sendo as duas primeiras correspondentes à categoria semântica Pessoa e a última, à categoria semântica Organização. Entretanto, existem outras categorias de ENs, como as menções a Obras (por exemplo, “Código Da Vinci”); Acontecimentos (por exemplo, “Festa de Santo Antônio”), Tempo (por exemplo, “meio-dia”); Coisa (por exemplo, “barco”), entre outras.

O REN é uma tarefa com grande importância para o Processamento de Linguagem Natural, pois consiste numa primeira tarefa de análise semântica de um texto, com potencial aplicações a diversas tarefas. Por exemplo, em sistemas de perguntas e respostas, as perguntas frequentemente se referem a informações sobre entidades. Também, métodos de identificação de estruturas mais complexas, como eventos ou relações, dependem do bom desempenho do REN como uma etapa de pré-processamento (Socher et al., 2012; Zelenko; Aone; Richardella, 2003).

17.4.2 Extração de Relações

A tarefa de extração de relações (ou de relacionamentos) (ER) refere-se a identificar relacionamentos entre entidades de um determinado escopo mencionadas em um texto (Jurafsky; Martin, 2023). O escopo, no contexto da ER, refere-se a um conjunto de relações-alvo de um determinado domínio de conhecimento ou aplicação a ser investigado. Por exemplo, o Quadro 17.2 apresenta alguns exemplos de relações no domínio de geografia brasileira. Na descrição das relações, os elementos em negrito referem-se às entidades em um dado relacionamento descrito pelo termo em itálico.

Quadro 17.2. Exemplos de relações no domínio da geografia brasileira.

Relação	Descrição	Exemplo
Pertence(Cidade, Unidade Federativa)	Sobre uma cidade que está localizada em uma determinada Unidade Federativa , dizemos que a primeira <i>pertence</i> a esta última.	Pertence(Salvador, Bahia)
Tem_Prefeito(Cidade, Pessoa)	Uma pessoa que executa a função administrativa de gestão do executivo em nível municipal de uma dada cidade é denominada de seu(sua) <i>prefeito(a)</i> .	Tem_Prefeito(Salvador, Bruno Reis)
Fundação(Cidade, Data)	A data em que uma cidade foi fundada, é dita sua data de <i>fundação</i> .	Fundação(Salvador, 29 de março de 1549)

Nesse contexto, a delimitação de um escopo ou domínio de interesse, concentra-se na determinação das relações a serem processadas, i.e. nos tipos de relacionamentos de interesse, assim como da natureza das entidades associadas por tais relações.



17.4.3 Extração Conjunta de Entidades e Relações

As tarefas de reconhecimento de entidades nomeadas e extração de relações são interdependentes, no sentido de que a definição do escopo a ser estudado delimita tanto as categorias e natureza das entidades a serem extraídas, como também as relações entre essas entidades. Também, note-se que, pelo fato de as relações serem comumente definidas entre entidades de tipo especificado, como o caso da relação *Tem_Prefeito* no Quadro 17.2 que ocorre entre entidades das classes **Cidade** e **Pessoa**, tanto as informações das entidades mencionadas no texto são úteis para a extração de relações, quanto a informação das relações identificadas pode ser útil ao processo de identificação de entidades.

De fato, na literatura recente, existem vários trabalhos que consideram a tarefa de extração conjunta de entidades e relações (ERE, do inglês *Entity and Relation Joint Extraction*), composta das tarefas de REN e ER (Agichtein; Gravano, 2000; Shaowei et al., 2022; Yuan et al., 2021). Enquanto normalmente abordagens estruturam suas soluções de forma sequencial, usualmente realizando REN inicialmente e, posteriormente, realizando ER, como nos trabalhos de (Hasegawa; Sekine; Grishman, 2004) e de (Socher et al., 2012), a literatura recente aponta para as vantagens da identificação conjunta ao permitir um melhor aprendizado de restrições para identificação de entidades e relações, c.f. o recente *survey* realizado por (Shaowei et al., 2022) sobre métodos para tal tarefa.

17.4.4 Métodos empregados para EI na literatura

Várias abordagens foram adotadas para o problema de EI durante seu desenvolvimento histórico. Enquanto abordagens iniciais privilegiavam métodos ricos em conhecimento, como regras e recursos linguísticos e de conhecimento de mundo, a literatura recente na área privilegia métodos baseados em dados, como o aprendizado de máquina, com o recente emprego de arquiteturas neurais aos problemas.

A seguir faremos uma breve apresentação das abordagens descritas na literatura para os problemas de EI.

17.4.4.1 REN

As abordagens iniciais para REN baseavam-se, majoritariamente, no emprego de regras lexico-sintáticas e consulta a almanaques (*gazeteers*). Tais abordagens dependem da construção de listas de nomes próprios como antropônimos, topônimos etc., e outras palavras, como “Ltda.”, “Jr.” etc., que auxiliam no processo de identificação e classificação de ENs complexas ou desconhecidas. Essa é, por exemplo, a abordagem empregada por Wolinski; Vichot; Dillet (1995) que combina almanaques e regras para a identificação e classificação de ENs. Posteriormente, almanaques foram também empregados em conjunção com métodos baseados em dados, como o trabalho de Florian et al. (2003) que os emprega aliados aos classificadores, enquanto Liu; Yao; Lin (2019) os utilizam durante o treinamento de uma rede neural, como um sinal de treinamento (parte da função de perda, ou *loss* em inglês).

Muitos trabalhos debruçaram-se também sobre o problema de construção automática ou semi-automática de almanaques, dos quais os trabalhos de Nadeau (2007), de Riloff; Jones; et al. (1999) e de Etzioni et al. (2005) são alguns dos mais importantes.

Enquanto as abordagens iniciais para o problema baseavam-se em regras, com a disponibilidade de dados anotados para a tarefa, tais métodos foram rapidamente suplantados por métodos baseados em dados, tais como: os métodos baseados em



classificação (Asahara; Matsumoto, 2003; Sekine, 1998) e classificação sequencial (Bikel; Schwartz; Weischedel, 1999; McCallum; Li, 2003).

A redução de REN à tarefa de classificação sequencial merece destaque pelos bons resultados obtidos. Tal redução se dá através de um esquema de codificação do problema que nos permite representar fragmentos textuais e sua classificação como um problema de rotulação ou etiquetagem.

Partindo-se do pressuposto de que os fragmentos textuais descrevendo entidades nomeadas são contíguos, podemos codificar a tarefa de delimitação de entidades como classificação sequencial empregando rótulos que descrevem os limites de uma EN, e.g. o esquema BIO com os rótulos **B** (do inglês, *begin*) para designar a palavra inicial de uma EN, **I** (do inglês, *inside*) para designar palavras que fazem parte da EN mas não a iniciam e **O** (do inglês, *outside*) para designar palavras que não pertencem a uma entidade. Da mesma forma, podemos estender nosso esquema de codificação para incluir as classes de interesse. Assim, seguindo o esquema BIO, teremos os rótulos **B-PER** e **I-PER** para descrever entidades da classe **Pessoa**.

A redução do problema de REN à classificação sequencial está ilustrada no Exemplo 17.1.

Exemplo 17.1.

Renata/B-PER Silva/I-PER e/O Maria/B-PER Costa/I-PER palestraram/O
na/O Universidade/B-ORG Federal/I-ORG da/I-ORG Bahia/I-ORG.

Recentemente, destacam-se na literatura abordagens baseadas em redes neurais profundas, com uma grande concentração nos últimos anos em modelos gerativos de linguagem, devido aos resultados positivos obtidos por tais arquiteturas em diversas tarefas.

Na literatura são de grande destaque os modelos recentes BART (Lewis et al., 2020a), RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), BERT (Devlin et al., 2019) e GPT-3 (Brown et al., 2020), conforme descritos no Capítulo 15.

Similarmente, na língua portuguesa, nas duas edições do HAREM (Mota; Santos, 2008; Santos; Cardoso, 2007b), o primeiro esforço sistemático de desenvolvimento de soluções para a tarefa na língua, a maioria dos sistemas participantes baseava-se em métodos ricos em conhecimento, como regras e almanaques. De fato, nas duas avaliações, somente os sistemas MALINCHE (Solorio, 2007), NEURA (Ferrández et al., 2007) e R3M (Mota, 2008) não se baseavam em regras. Métodos baseados em classificação sequencial se seguiram para a língua portuguesa, como o RELP-CRF (Amaral; Vieira, 2014) baseado em um classificador sequencial. Mais recentemente, abordagens baseadas em redes neurais e modelos de linguagem foram desenvolvidas tornando-se o estado da arte da tarefa na língua. A Tabela 17.3 apresenta o atual estado da arte em português, com base no *corpus* HAREM. A métrica de avaliação apresentada, medida F1, será discutida na Seção 17.6.

Tabela 17.3: Trabalhos estado da arte no REN em português

Modelo	Medida F1
BERT-CRF (Souza; Nogueira; Lotufo, 2020)	83,70%
BiLSTM-CRF+FlairBBP (Santos et al., 2019)	82,26%
LSTM-CRF (Castro; Silva; Soares, 2018)	76,27%
CharWNN (Santos; Guimarães, 2015)	65,41%

Souza; Nogueira; Lotufo (2020) desenvolveram um modelo BERT para o Português com 2,68 bilhões de *tokens* e aplicaram o modelo em um classificador CRF. Santos et



al., avaliaram o impacto do modelo contextualizado Flair Embeddings aplicado a tarefa de REN junto com uma rede neural BiLSTM-CRF. Os autores também desenvolveram um modelo Flair Embeddings para o português, o FlairBBP, treinado com 4,9 bilhões de *tokens* (Santos et al., 2019). Castro; Silva; Soares (2018) utilizou uma rede LSTM e um classificador CRF junto com modelos Word Embeddings pré-treinados. Santos; Guimarães (2015) desenvolveram uma rede neural convolucional capaz de capturar características a nível de caracteres e também de incorporar word embeddings pré-treinados.

17.4.4.1.1 Reconhecimento de Entidades em Domínios Específicos

O reconhecimento de entidades tem sido aplicado em muitas áreas específicas, como direito, saúde e geologia. Nesses casos há uma demanda de adaptação dos modelos preditivos de acordo com a nova linguagem especializada do domínio e um novo conjunto de rótulos que devem ser aprendidos. Da mesma forma, são necessários novos conjuntos de dados para o processo de aprendizado, uma vez que abordagens de aprendizado de máquina necessitam de exemplos anotados para se chegar a um modelo preditivo eficaz.

Muitos trabalhos endereçam domínios específicos, citamos exemplos em diversas línguas. Para o inglês, uma rede neural BiLSTM-CRF para o domínio biomédico é proposta em (Habibi et al., 2017).

Um conjunto de dados do domínio jurídico em língua alemã é apresentado por Leitner; Rehm; Schneider (2019), que empregam redes neurais BiLSTM para a rotulação dos textos. Em (Qiu et al., 2019), uma rede neural BiLSTM-CRF com mecanismo de atenção é aplicada para reconhecer entidades geológicas para a língua chinesa.

Para o português, um *corpus* para detecção de eventos de quedas de pacientes em prontuários eletrônicos é descrito em (Santos; Santos; Vieira, 2020). Os autores usaram uma rede neural BiLSTM-CRF+Flair para gerar um modelo classificador de *tokens*. Um *corpus* no domínio jurídico, tendo categorias específicas como legislação e jurisprudência é proposto por Araujo et al. (2018), que usaram uma rede neural BiLSTM-CRF para criar um primeiro *baseline* para esse *corpus*. Ademais, Consoli et al. (2020) analisam um *corpus* no domínio de geologia usando uma rede neural BiLSTM-CRF com um modelo contextualizado Flair Embeddings.

17.4.4.2 Extração de Relações

As abordagens iniciais para o problema de ER baseavam-se na definição de gabaritos e regras de extração, com base em informação sintática obtida de analisadores sintáticos rasos ou profundos (Cowie, 1983; Sager, 1978). Tais métodos foram rapidamente suplantados por métodos baseados em dados e padrões obtidos de *corpora*, como os famosos padrões de Hearst (1992) para identificação de relações de hiponímia.

O trabalho de Hearst (1992) se baseou na definição de padrões lexico-sintáticos para expressão de relações de hiponímia e hiperonímia a partir de uma análise de *corpus*. Ao escolher a relação de hiponímia, que ocorre em todo domínio, e padrões gerais baseados em aspectos da língua, como os representados no Quadro 17.3, o autor garante generalizabilidade dos padrões obtidos para diversos domínios e aplicações.

Quadro 17.3. Exemplos de Padrões de Hearst para hiponímia



Padrão	Exemplo	Extração
NP , tais quais $\{NP\}^*NP$... países, tais quais França, Brasil e China	Is_a(França, país)
$NP\{, NP\}^*$ e outros(as) NP	Contusões, feridas, osso quebrados e outras lesões	Is_a(contusão, lesão)

Devido à dificuldade de construção manual das regras, os métodos de Riloff et al. (1993), empregam heurísticas para geração de padrões baseadas em informação gramatical, e de Soderland et al. (1995), que se baseia numa semântica de quadros (*frames*) empregando um analisador semântico e medidas de qualidade de identificação de exemplos, baseado no percentual de acerto sobre relacionamentos previamente conhecidos, para identificação de quadros relevantes.

As abordagens baseadas em aprendizado de máquina, hoje as mais comuns e com melhor desempenho na literatura (Konstantinova, 2014; Nasar; Jaffry; Malik, 2021) dividem-se em abordagens que realizam reconhecimento de entidades e extração de relações de forma conjunta e separada.

Abordagens baseadas na realização de REN e ER de forma separada baseiam-se em um fluxo de processamento em que, em geral, as entidades são identificadas primeiro e a tarefa de ER se reduz a identificar quando uma sentença ou fragmento textual denota uma relação semântica entre duas entidades. Consideremos o Exemplo 17.2, retirado de (Socher et al., 2012):

Exemplo 17.2.

[Gripe aviária]_{e1} é uma doença infecciosa causada pelo vírus da [influenza tipo a]_{e2}

Podemos, então, reduzir o problema de identificar a relação Causa-Efeito(e_1, e_2) a um problema de classificação textual, identificando se a sentença acima fornece indícios para a expressão da relação de interesse. As soluções propostas na literatura para o problema são variadas e baseadas em diferentes métodos.

Zelenko; Aone; Richardella (2003), por exemplo, propõem funções de *kernel* para árvores sintáticas rasas, i.e. funções que descrevem medidas de similaridade entre tais árvores. Eles empregam tais medidas para treinar um classificador de *perceptron* com votação (*voted perceptron*) sobre relações no domínio de organizações extraídas de um *corpus* de textos jornalísticos. De forma similar, Zhao; Grishman (2005) empregam diferentes funções de *kernel* sobre informações sintáticas relevantes para a identificação de relação e argumentos visando treinar um classificador SVM sobre o *corpus* de ER da conferência ACE.

Culotta; McCallum; Betz (2006), por outro lado, empregam um classificador sequencial baseado em modelos escondidos de Markov para identificação de relações em um texto. Ao restringir sua análise a textos biográficos, os autores reduzem o processo de identificar instâncias de relações à identificação de fragmento textual que delimita o argumento e sua classificação, tarefa para a qual a classificação sequencial já é comumente utilizada. Consideremos o Exemplo 17.3 sobre George W. Bush, retirado de (Culotta; McCallum; Betz, 2006):

Exemplo 17.3.

George é filho de George H. W. Bush e Barbara Bush.
pai mãe



Ao identificar o papel de pai e mãe, os autores conseguem construir a relação Pai(George H. W. Bush, George W. Bush) e Mãe(Barbara Bush, George W. Bush).

Métodos baseados em redes neurais, de forma geral, costumam empregar técnicas de aprendizado de representação (Bengio; Courville; Vincent, 2013) para aprender representações do conteúdo semântico dos fragmentos textuais e reduzem o problema de ER à classificação textual. É o caso de Socher et al. (2012), que propõem a MV-RNN, uma rede neural que constrói um espaço de representação baseado em matrizes e vetores com o objetivo de capturar a composicionalidade de sentido de sintagmas e sentenças e os aplica para ER. Similarmente, Zeng et al. (2014) e Wang et al. (2016) empregam redes neurais convolucionais para obter representações vetoriais de sentenças que serão empregadas no processo de classificação quanto à relação expressa pela mesma.

17.4.4.3 Extração Conjunta de Entidades e Relações

Abordagens baseadas em identificação sequencial de entidades e relações possuem desvantagens observadas na literatura. Primeiramente, como a ER é guiada pelas entidades identificadas no processo de REN, a propagação de erros da primeira tarefa pode ter impacto considerável na performance dos sistemas desenvolvidos. Segundo, uma vez que o contexto determinado limita tanto as tarefas de REN, quanto as de ER, existe uma interdependência entre as tarefas. Assim, propostas visando realizar a extração de entidades e relações de forma conjunta começaram a surgir na literatura recente, ganhando certo interesse da comunidade.

As abordagens empregadas para tal tarefa são diversificadas, incluindo desde métodos de aprendizado relacional a redes neurais

Roth; Yih (2007) propõem a utilização de métodos de programação inteira ao problema, baseados na teoria estatística de aprendizado relacional. Os autores utilizam classificadores locais para a identificação de entidades e relações e um classificador global que combina as informações dos classificadores locais em uma predição que maximiza a qualidade da extração, codificada por meio de restrições em programação inteira. Também baseados em modelos estatísticos, Yu; Lam (2010) propõem o uso de modelos gráficos globais para identificação de um descritor de relação e uma segmentação do texto para identificação dos argumentos.

Li; Ji (2014) e Miwa; Bansal (2016), por sua vez, reduzem a tarefa de ERE à classificação sequencial, utilizando redes neurais recorrentes bidirecionais sequenciais e estruturadas com base na estrutura superficial e na árvore de dependências sintáticas da entrada para identificação conjunta de entidades e relações.

17.5 Extração de Informação Aberta

A Extração de Informação Aberta (EIA), também conhecida como *Open Information Extraction*, Open IE ou OIE em inglês, é a tarefa de extrair informações estruturadas de documentos sem necessitar da pré-definição do contexto da tarefa, i.e. das relações e tipos de entidade de interesse. A tarefa foi inicialmente proposta pelo trabalho de (Banko et al., 2007) e ganhou popularidade nas últimas décadas devido à sua aplicabilidade para processar e estruturar o conhecimento a partir de grandes volumes de dados disponíveis na Web, seguindo o paradigma da Web como um *Corpus* (WaC) (Meyer et al., 2003).

A EIA surge visando generalizar a tarefa de Extração de Relações. A principal diferença entre as duas abordagens, porém, reside na dependência da ER de uma especificação prévia



do domínio de aplicação, bem como das relações alvo a serem identificadas, que a EIA visa eliminar.

Seguindo o trabalho original de Banko et al. (2007), que propôs o sistema TextRunner, vários métodos e sistemas para EIA foram propostos na literatura (Del Corro; Gemulla, 2013; Fader; Soderland; Etzioni, 2011; Xavier; Lima; Souza, 2015), mas, como observado por Glauber; Claro (2018), os principais avanços na área se concentraram principalmente no idioma inglês.

A EIA para a língua portuguesa tem uma história bastante recente. A partir dos trabalhos de Souza; Claro (2014), Pereira; Pinheiro (2015) e de (Barbosa; Glauber; Claro, 2016), têm crescido o número de estudos sobre a tarefa assim como os resultados obtidos por esses estudos, com recentes desenvolvimentos de métodos (Oliveira; Claro; Souza, 2022; Sena; Claro, 2019, 2020; Sena; Glauber; Claro, 2017; Souza; Claro; Glauber, 2018), construção do *corpus* (Glauber et al., 2018) e avaliação dos sistemas disponíveis (Glauber; Claro; Oliveira, 2019; Glauber; Claro; Sena, 2019; Malenchini et al., 2019).

Embora a área tenha visto um crescimento recente para o desenvolvimento de métodos para línguas como o inglês, principalmente com a aplicação de métodos supervisionados e redes neurais, esses avanços ainda não foram incorporados na literatura sobre EIA para a língua portuguesa. A razão para isso é principalmente a falta de recursos linguísticos disponíveis para orientar o desenvolvimento de pesquisas para a língua. Embora o foco no idioma inglês possa ser devido ao seu uso generalizado em todo o mundo, foi reconhecido pela comunidade científica que esse foco no inglês com suas características particulares pode introduzir algum viés na área (Bender, 2009).

Assim, esta seção aborda EIA para a língua portuguesa, incluindo uma formalização e a evolução das abordagens da área.

17.5.1 Formalização

A tarefa de EIA pode ser formalmente definida sendo $X = \langle x_1, x_2, \dots, x_n \rangle$ uma sentença composta de *tokens* x_i . Um extrator EIA é uma função que mapeia X em um conjunto $Y = \langle y_1, y_2, \dots, y_j \rangle$ como um conjunto de tuplas $y_i = \langle rel_i, arg1_i, arg2_i, \dots, argn_i \rangle$, que descrevem as informações expressas na sentença X . Neste capítulo, consideramos que as tuplas estão sempre no formato $y = (arg1, rel, arg2)$, onde $arg1$ e $arg2$ são sintagmas nominais, não necessariamente formados por *tokens* presentes em X , e rel é um descritor de um relacionamento entre $arg1$ e $arg2$. Não consideraremos extrações formadas por mais de dois argumentos neste capítulo.

17.5.2 Abordagens

Os primeiros métodos de EIA empregavam padrões de inspiração linguística para extração, como ArgOE (Gamallo; Garcia, 2015), ou adaptação de métodos para a língua inglesa, como SGS (Souza; Claro; Glauber, 2018), InferReVerbPT Sena; Glauber; Claro (2017) e RePort Pereira; Pinheiro (2015). Os trabalhos são principalmente influenciados por métodos baseados no inglês da chamada segunda geração de EIA (Fader; Soderland; Etzioni, 2011).

O primeiro sistema de EIA para o português de que temos conhecimento foi o DepOE (Gamallo; Garcia; Fernández-Lanza, 2012). Ele executa a extração aberta multilíngue de triplas (inglês, espanhol, português e galego) usando o analisador sintático de dependências baseado em regras *DepPattern*. No entanto, nenhuma avaliação ou



resultados são relatados para a língua portuguesa. Os autores apresentam somente uma comparação dos seus resultados com *Reverb* na língua inglesa.

Souza; Claro (2014) se propuseram a analisar o conjunto de características mais representativas da língua portuguesa para a identificação de extrações válidas no contexto de EIA, tal qual empregado na língua inglesa com o sistema ReVerb (Fader; Soderland; Etzioni, 2011).

O sistema RePort (Pereira; Pinheiro, 2015), por outro lado, é uma adaptação do ReVerb para a língua portuguesa baseada em análise sintática rasa com regras sintáticas e lexicais. Os autores relatam que suas extrações apresentam grande similaridade com suas correlatas extraídas pelo ReVerb (dos textos traduzidos para o inglês).

O RELP, proposto por Abreu; Vieira (2017), é um sistema aberto de extração de relações que extrai relações entre entidades nomeadas em um domínio de organização aplicando classificação sequencial com CRF (*Conditional Random Fields*). O sistema RelP extrai qualquer descritor de relação que expressa um relacionamento entre pares de entidades nomeadas (Organização, Pessoa ou Lugar), caracterizando-o como uma abordagem híbrida da REN com a EIA.

O InferReVerbPT desenvolvido por Sena; Glauber; Claro (2017) baseia-se numa adaptação do sistema ReVerb para a língua portuguesa, expandindo-o com a extração de relacionamentos implícitos obtidos por inferência por propriedades de simetria e transitividade das relações com inferência transitiva e simétrica. Um classificador SVM foi empregado para realizar a inferência baseado nas propriedades semânticas do verbo central no descritor de relação.

Souza; Claro; Glauber (2018) analisaram que a maior desvantagem dos estudos baseados em recursos linguísticos, como dados anotados, reside na escassez de tais recursos na maioria dos idiomas além do inglês. Assim, para mitigar esse problema, eles propõem um método de classificação de fatos baseado na similaridade de estruturas gramaticais (SGS). Sua abordagem modela estruturas morfosintáticas dos fatos (triplas descrevendo relacionamentos) para identificar padrões de semelhanças que podem ser usados para distinguir entre fatos válidos e inválidos. Eles aplicaram algoritmos de isomorfismo de grafos para detectar subgrafos descrevendo tais padrões.

Um novo sistema de EIA baseado em análise de dependência foi proposto por Gamallo; Garcia (2015), chamado ArgOE. Tal sistema é multilíngue, baseado em heurísticas e utiliza a informação de dependência sintáticas do texto para analisar a estrutura de dependência do verbo, bem como um conjunto de regras para gerar os relacionamentos. A introdução de um Analisador de Dependência em sistemas de EIA focados inteiramente na língua portuguesa foi feita pelos autores Oliveira; Claro; Souza (2022). O DptOIE é baseado em análise de dependência e regras elaboradas manualmente. As sentenças são pré-processadas por meio de um tokenizador, um PoS *Tagger* e um analisador de dependências. Os autores propõem um acoplamento de três módulos para tratar casos particulares: conjunções coordenadas, orações subordinadas e aposto.

Com a evolução dos métodos de EIA para a língua inglesa utilizando os modelos neurais, novas abordagens foram propostas também para a língua portuguesa.

O primeiro trabalho que utilizou aprendizado supervisionado com rede neural profunda para o português foi o de Ro; Lee; Kang (2020) que descreve o sistema **Multi2OIE**. Os autores utilizaram o modelo de linguagem BERT multilíngue (Devlin et al., 2019) para obter representações vetoriais das palavras e reduzem a tarefa de EIA à classificação sequencial, identificando os fragmentos do texto que determinam os argumentos (arg_1 , arg_2) e o descritor de relação (*rel*). Seu sistema foi capaz de produzir extrações para vários



idiomas (inglês, português e espanhol), treinados, entretanto, sobre dados traduzidos do inglês.

Stanovsky et al. (2018) propuseram uma abordagem de EIA para a língua inglesa baseada em triplas. Os mesmos fazem uso de uma classificação sequencial cuja limitação define uma tripla extraída para cada sentença. Este método utiliza uma arquitetura de Redes Neurais Recursivas (RNN) para realizar EIA. A EIA é formulada como uma tarefa de rotulagem de sequências, utilizando estratégias semelhantes às que foram aplicadas anteriormente a tarefas como o Reconhecimento de Entidades Nomeadas. Já os autores em Cui; Wei; Zhou (2018) e Zhang; Duh; Van Durme (2017) propõem modelar o problema da EIA como um problema de aprendizado sequência a sequência (*seq2seq*). Eles definem uma estrutura *encoder-decoder* para aprender argumentos e tuplas de relação inicializadas a partir de um sistema de EIA.

Seguindo o trabalho de (Stanovsky et al., 2018), em 2022, Cabral; Souza; Claro (2022) propuseram **PortNOIE**, uma arquitetura neural para EIA em português que combina representações contextuais de palavras com codificadores neurais para extrair relacionamentos baseado em classificação sequencial iterativa. Diferente de outros métodos de classificação sequencial para EIA, os autores focam na extração de múltiplas triplas de uma mesma sentença.

17.6 Avaliação

A avaliação sistemática de sistemas de EI foi estabelecida primeiramente nas conferências MUC, em particular na sua segunda edição, com o estabelecimento de gabaritos-padrão que deveriam ser utilizados por todos os sistemas participantes e a adoção de métricas de qualidade, baseadas naquelas usadas na área de recuperação de informação, que foram abordadas no Capítulo 16. Para avaliar a tarefa de extração de relações, a MUC-2 estabeleceu como métricas de qualidade do sistema as medidas de precisão e cobertura, também denominada de *Recall* ou Revocação.

A precisão de um sistema reflete a qualidade de suas extrações, i.e., quantas das extrações realizadas estão corretas, dado um *corpus* de teste. A medida de precisão pode ser calculada como:

$$P = \frac{\#(\text{relacionamentos corretamente extraídos})}{\#(\text{relacionamentos extraídos pelo sistema})} \quad (17.1)$$

A cobertura também conhecida como revocação, reflete quão abrangente um sistema é em suas extrações, i.e., quantas das extrações a serem realizadas em um *corpus* de teste, o sistema é capaz de realizar. A medida de cobertura pode ser calculada como:

$$R = \frac{\#(\text{relacionamentos extraídos})}{\#(\text{relacionamentos no corpus})} \quad (17.2)$$

Enquanto a MUC-3 adicionou duas novas métricas de avaliação, a saber sobre-geração (*overgeneration*) e sub-geração (*fallout*), tais métricas receberam pouco interesse na literatura. De fato, Lehnert; Sundheim (1991) argumentam que tais métricas foram pouco informativas ou difíceis de calcular para a tarefa de EI e, portanto, abandonadas. Foi também empregado nessa conferência um sistema automático de avaliação disponibilizado às equipes participantes que permitiu uma maior compreensão do modelo de avaliação e, como discutem Lehnert; Sundheim (1991), um avanço qualitativo nos sistemas gerados.



Além das medidas de precisão e cobertura, assim como em tarefas de classificação de texto e recuperação de informação, utilizamos a média harmônica entre essas medidas, chamada medida F1, a fim de condensar a informação contida nas duas. A medida F1 pode ser calculada como:

$$F1 = \frac{2 * P * R}{P + R} \quad (17.3)$$

A avaliação da tarefa de REN segue padrões semelhantes aos aplicados à tarefa de ER. De fato, desde a MUC-6 (Grishman; Sundheim, 1996), as medidas de precisão, cobertura e F1 tem sido usada consistentemente como métricas de avaliação da tarefa de REN em diversos esforços de avaliação, como a CoNNL (Sang; De Meulder, 2003), para a língua inglesa, e das duas edições do HAREM (Gonçalo Oliveira et al., 2008; Santos; Cardoso; Seco, 2007), com exceção à ACE (Doddington et al., 2004) que apresenta uma combinação da tarefa de REN com reconhecimento de co-referência entre entidades e utiliza um sistema de pontuação próprio.

A avaliação de sistemas de EIA, por sua vez, possui algumas peculiaridades que precisam ser discutidas. Uma vez que a tarefa é postulada por Banko et al. (2007) como a extração de todas as relações identificadas em um dado fragmento textual, sem limitação de domínio de interesse, tal tarefa impõe imensa dificuldade aos esforços de avaliação.

De fato, Glauber et al. (2018) relatam um esforço de anotação de dados para a tarefa em língua portuguesa em que foram identificados por anotadores humanos mais de 400 relacionamentos em um *corpus* de 25 sentenças retiradas de textos jornalísticos e de enciclopédia. Assim, a avaliação de EIA deu-se, em grande parte de seu desenvolvimento e maturação, em conjuntos de dados não anotados, recorrendo a avaliações qualitativas das saídas dos sistemas e comparação direta por humanos das extrações obtidas.

Nesses esforços de avaliação, a precisão do sistema pode ser mensurada a partir da avaliação humana das saídas. Não é possível, entretanto, avaliar medidas como cobertura e F1, dada a inexistência de uma referência do conjunto total de relacionamentos a serem identificados. Assim, os autores da área propuseram diferentes métricas a fim de estimar tais valores, como a métrica rendimento (*yield*) (Fader; Soderland; Etzioni, 2011; Schmitz et al., 2012).

A métrica de rendimento consiste no número de extrações válidas, i.e. corretas, de um dado sistema. Como calcular tal medida é, na maioria dos casos, impraticável dada a grande quantidade de extrações realizadas pelos sistemas, ela pode ser estimada a partir da precisão do sistema calculada sobre uma amostra aleatória das extrações realizadas (P'). Assim, podemos estimar o rendimento como:

$$Y = P' \cdot \#(\text{extrações realizadas}) \quad (17.4)$$

Foi também explorada a estratégia de criação (semi-)automática de conjuntos de dados usando vários sistemas (Del Corro; Gemulla, 2013), estratégias de supervisão fraca (Smirnova; Cudré-Mauroux, 2018), ou a geração de *corpora* para a tarefa a partir da transformação de anotações de tarefas próximas, como identificação de papéis temáticos (*Semantic Role Labeling*) por (Stanovsky et al., 2018). *Corpora* gerados de forma semi-automática vêm ganhando atenção na literatura recente, particularmente para a língua inglesa, devido a necessidade de dados anotados para se utilizar técnicas de aprendizado de máquina e redes neurais em EIA. *Corpora* como o OIE2016 (Stanovsky et al., 2018), Wire57 (Léchelle; Gotti; Langlais, 2018) e CARB (Bhardwaj; Aggarwal; Mausam, 2019) vêm se tornando *corpora* de referência em língua inglesa para o problema, apesar dos problemas existentes na construção de tais recursos – a não exaustividade das relações anotadas.



Para a língua portuguesa, foram propostas algumas iniciativas para avaliar os sistemas da OIE. Uma avaliação conjunta foi promovida durante o Fórum Ibérico de Avaliação de Línguas (IberLEF) em 2019 (Collovini et al., 2019). A avaliação foi feita usando o *corpus* proposto por Glauber et al. (2018), que é composto por 442 relacionamentos extraídos de 25 frases de fontes como a seção em português da Wikipédia, o *corpus* CETENFolha, resenhas de filmes do portal Adoro Cinema² e o *corpus* Europarl. Apesar desta tarefa ter contemplado quatro cenários de avaliação, a avaliação geral dos sistemas permaneceu consistente nos diferentes cenários, indicando robustez nos resultados da avaliação. No geral, os sistemas DPTOIE (Oliveira; Claro; Souza, 2022) e Linguakit (Gamallo; Garcia, 2015) tiveram o melhor desempenho, com o Linguakit2 dominando as avaliações de correspondência exata e o DPTOIE as avaliações de correspondências parciais (Collovini et al., 2019).

Outra abordagem de avaliação foi idealizada por (Malenchini et al., 2019). Seu foco foi a avaliação extrínseca dos sistemas de EIA através de sua contribuição na tarefa de respostas automáticas a perguntas. Os autores apresentaram um conjunto de dados de referência (*benchmark*) para avaliação extrínseca de sistemas de EIA em textos de língua portuguesa. Os sistemas que alcançaram os melhores valores na avaliação realizada pelos autores foram os sistemas ArgOE (Gamallo; Garcia, 2015), DependentIE (Glauber; Claro; Oliveira, 2019) e DptOIE (Oliveira; Claro; Souza, 2022).

17.7 Considerações finais

Este capítulo descreveu uma visão geral da área de Extração de Informação, apresentando a Extração de Informação Tradicional e a Extração de Informação Aberta. Transversalmente, apresentamos as formalizações necessárias e os conceitos fundamentais para a compreensão da EIA, assim como a avaliação da área e as heranças de outras áreas afins, tais como RI.

Nessa primeira versão, este capítulo descreveu de maneira bem sucinta as abordagens propostas para EI e EIA durante seu desenvolvimento histórico e as abordagens atuais da literatura, como as utilizando modelos de linguagens. Especificamente, a utilização da arquitetura Transformers, descritas no Capítulo 15 para as tarefas de EI e EIA tem sido bastante difundida para a língua inglesa e tem atuado em diversas áreas da PLN.

Agradecimentos

Agradecemos as colaborações dos autores deste Capítulo e suas indicações, assim como agradecemos a Adriana Pagano e Aline Macohin pela revisão e comentários.

²<https://www.adorocinema.com/>



Capítulo 18

Tradução Automática

*Sheila Castilho
Helena de Medeiros Caseli*

18.1 Introdução

A tradução automática (TA), também conhecida como tradução de máquina (em inglês, *machine translation* ou MT), refere-se à tradução de um texto eletrônico por um computador de uma língua para outra sem intervenção humana. Nesse sentido, convencionou-se chamar de língua (ou texto) fonte a língua de partida (origem) e língua (ou texto) alvo a língua de chegada (destino ou saída). Além de envolver a análise e interpretação (NLU) da língua-fonte e a geração (NLG) da língua-alvo, há a premissa fundamental de gerar uma saída que seja semanticamente equivalente (transmite o mesmo significado) à entrada.

Nos últimos anos, a TA evoluiu significativamente com o avanço de modelos estatísticos e neurais. Atualmente, ela é amplamente utilizada em todo o mundo por governos, indústria da tradução, consumidores finais e em pesquisas em uma variedade de aplicações.

Os primeiros sistemas bem-sucedidos de TA datam do final dos anos 1950 e início dos anos 1960¹, com os experimentos de Georgetown. No entanto, é possível encontrar referências a tentativas de tradução automática no século XVII (Hutchins, 2001). Desde então, diferentes abordagens para a TA foram desenvolvidas, incluindo abordagens baseadas em regras, exemplos, estatísticas e, mais recentemente, a TA neural, apresentadas brevemente nas diversas seções deste capítulo.

Hoje em dia, a TA desempenha um papel importante não apenas no âmbito comercial, mas também no âmbito social e político. Ela é amplamente utilizada em diversas aplicações de comunicação, que incluem²:

- Texto para texto: o usuário insere um texto-fonte e obtém uma versão traduzida em formato de texto;
- Texto para fala: o usuário insere um texto-fonte e obtém uma versão falada na língua-alvo;
- Fala para texto: o usuário fala no idioma fonte e obtém uma versão traduzida em formato de texto;
- Fala para fala: o usuário fala no idioma fonte e obtém uma versão falada na língua-alvo;
- Imagem (de palavras) para texto: o usuário insere uma imagem contendo um texto e obtém a tradução desse texto.

¹Para uma descrição abrangente da história da TA, sugere-se consultar alguns livros e capítulos sobre o assunto: (Hutchins, 2001), (Koehn, 2009), (Hutchins, 2010) e (Koehn, 2020).

²Todas as aplicações citadas estão disponíveis no Google tradutor. Disponível em: <https://translate.google.com.br/>.



Devido à ampla utilização da TA atualmente, seu impacto pode ser observado em nossa sociedade. Por essa razão, a avaliação da TA (tema da Seção 18.3) tornou-se mais importante, visando garantir a qualidade da tradução. Além da avaliação, este capítulo descreve as principais abordagens para a TA (Seção 18.2). No decorrer deste capítulo, alguns conceitos-chave são explicados para que você possa acompanhar os desenvolvimentos.

18.2 Abordagens

A tradução automática pode ser realizada de diversas maneiras, desde a mais simples (tradução direta), que envolve a tradução palavra-a-palavra (ou sequência de palavras), até a mais utilizada na atualidade, que é a tradução baseada em redes neurais artificiais (tradução neural). Na trajetória entre a tradução direta e a tradução neural, explicaremos também abordagens intermediárias, como a baseada em regras, a tradução por interlíngua e a tradução estatística.

Para tanto, vamos traduzir a sentença “A casa do meu avô é linda.” para o inglês. Como isso aconteceria de acordo com as abordagens mais utilizadas (ou tradicionais) na tradução automática?

18.2.1 Tradução direta

Na **tradução direta** ocorre o mapeamento direto de palavras-fonte para palavras-alvo sem passar por outros níveis de análise³. Assim, no nosso exemplo, os caracteres seriam combinados em palavras e cada uma seria mapeada diretamente para seu equivalente em inglês usando, por exemplo, um léxico bilíngue. Utilizando o léxico bilíngue disponível no github do MUSE⁴ e a tradução palavra-a-palavra, nossa saída seria como apresentado em Exemplo 18.1:

Exemplo 18.1.

Entrada: A casa do meu avô é linda.

Saída: `_ house _ my granddad _ beautiful .`

onde as palavras para as quais não se encontrou um equivalente no léxico consultado foram substituídas por `_`⁵.

Vejam que neste processo de tradução não há nenhum processamento referente às línguas envolvidas, uma vez que o resultado é obtido via **casamento de padrão**, seguido da substituição de uma palavra por outra com base em uma lista de pares de palavras.

Obviamente a abordagem de tradução direta apresenta diversas **limitações**, como **não ser capaz de lidar com a estrutura (sintaxe) da língua**, que, como pode ser visto no Capítulo 6, é fundamental para o tratamento adequado da língua. A tradução direta foi uma das primeiras abordagens a serem investigadas e não é mais utilizada nos tradutores atuais.

³Vejam que aqui estamos usando “palavra” para denotar uma unidade lexical bastante comum no PLN. Contudo, vale ressaltar que outras unidades lexicais (n-grama ou expressão multipalavra, por exemplo) também podem ser usadas na tradução direta. Para saber mais sobre as unidades de processamento, Capítulo 4.

⁴<https://dl.fbaipublicfiles.com/arrival/dictionaries/pt-en.txt>

⁵Isso porque palavras como “a” e “do” não têm entrada no léxico consultado, provavelmente por serem *stop words*. Para as demais palavras, a tradução foi gerada considerando apenas a primeira ocorrência de equivalente em inglês encontrada. Por exemplo, para “casa” existem as opções “house” e “home”, nesta ordem, e como “house” aparece primeiro, ela foi a escolhida para gerar a saída neste exemplo.



18.2.2 Tradução Automática Baseada em Regras

Como o nome sugere, os sistemas de TA baseados em regras (em inglês, *Rule-based Machine Translation* ou RBMT) são sistemas baseados em conhecimento desenvolvidos por meio da especificação de regras linguísticas, que levam em consideração morfologia, sintaxe e semântica das línguas envolvidas, além dos léxicos bilíngues de ambos os idiomas de origem e de destino⁶. Essas regras e léxicos são formuladas e criados manualmente por especialistas em linguagem.

A partir desses recursos, a RBMT é capaz de realizar mapeamentos mais complexos como os que consideram a sintaxe das línguas fonte e alvo. Assim, no nosso exemplo, uma possível regra que poderia ser aplicada é a que indica a inversão da posição do sujeito que possui a casa com o uso do “apóstrofo s” gerando “*My grandfather’s house is beautiful.*” ao invés de “*The house of my grandfather is beautiful.*”

Assim, vê-se que o processamento automático necessário para a tradução baseada em regras é um pouco mais complexo do que aquele aplicado na tradução direta, uma vez que agora é preciso saber o papel de cada palavra na sentença-fonte, ou seja, saber que “casa” é um substantivo e que “do” é uma combinação da preposição “de” e determinante (artigo definido “o”) para que a regra apresentada em Quadro 18.1 seja aplicada corretamente. A regra especifica que sempre que for encontrado, na sequência-fonte, um substantivo (<SUB>) seguido da preposição (<PREP>) “de” combinada (+) com os artigos (<DET>) “a” ou (|) “o” deve-se gerar como saída o apóstrofo (') seguido de “s” e o substantivo equivalente na língua-alvo. O símbolo “=>” separa o que deve ser considerado na língua-fonte (à esquerda) e o que deve ser gerado na língua-alvo (à direita).

Quadro 18.1. Exemplo de regra para a tradução automática baseada em regras

<SUB> <PREP/de>+<DET/[a o]> => 's <SUB>

Para se determinar o “papel” de cada palavra na sentença o processamento necessário é o da etiquetagem morfossintática ou, do inglês, *part-of-speech tagging* descrito no Capítulo 4.

Apesar de realizar um processamento automático um pouco mais complexo, a **desvantagem** da tradução baseada em regras não está aí, mas sim na necessidade de mapear o conhecimento linguístico em regras corretas, genéricas e abrangentes o suficiente para que sejam aplicáveis a vários exemplos. Vejam que esse mapeamento envolve, necessariamente, o conhecimento da língua-fonte, da língua-alvo, e de como o processo de tradução de uma para a outra deve ocorrer. Além de um **processo trabalhoso**, a geração de regras é também **limitada**, pois, como a língua está em constante mudança, o conjunto de regras gerado tem que ser constantemente atualizado e revisado. Além disso, a tradução de/para outra língua necessita de um novo conjunto de regras. Isso porque a tradução por transferência entre duas línguas requer que a representação do conhecimento extraído da língua-fonte, e que vai ser mapeado para a língua-alvo, seja capaz de abrigar todas as características de ambas as línguas, tornando-a específica para aquele par. Analogamente, o desenvolvedor tem que ter muito conhecimento de ambas as línguas ou a equipe deve contar com linguistas/tradutores, o que torna os **sistemas caros** de se implementar. Além disso, a saída dos sistemas de regras pode apresentar **pouca fluência**, já que as traduções são fornecidas por meio de regras.

⁶A tradução baseada em regras enquadra-se na estratégia de tradução por transferência, na qual o mapeamento é realizado com base em uma análise da língua-fonte, seguida da aplicação de regras que fazem o mapeamento e a geração de equivalentes na língua-alvo.



A grande **vantagem** dos sistemas de regras é que, como não são necessários textos bilíngues para seu treinamento, esses sistemas são excelentes para **traduções de idiomas com recursos limitados**. Além disso, esses sistemas permitem que o desenvolvedor tenha um **controle maior**, sendo possível identificar exatamente onde estão os problemas, e a saída (texto-alvo) é relativamente previsível. Tanto regras quanto léxicos podem ser **refinados e personalizados**, com a adição de mais (novas) regras e entradas bilíngues para aprimorar a tradução. Outro ponto interessante, é que o conhecimento é **legível por seres humanos** o que facilita a manutenção. Os sistemas de TA baseados em regras foram os primeiros sistemas comerciais de TA na década de 1970 e abriram caminho para mais pesquisas em TA após o relatório ALPAC, que cortou os fundos (Seção 18.3.1).

18.2.3 Tradução por interlíngua

Mas será que esse conceito de transferência entre línguas não pode ser estendido para um número maior de línguas, ou seja, considerando um cenário multilíngue? Sim, essa é a ideia da **tradução por interlíngua**, que se propõe a usar uma língua intermediária – metalíngua – que é independente das línguas envolvidas na TA e ao mesmo tempo é capaz de representar informações de qualquer outra língua. Essa metalíngua, de natureza artificial, é não ambígua e, portanto, mais simples de processar do que qualquer linguagem natural. Assim, o processo de tradução entre duas línguas quaisquer é composto de duas etapas de traduções supostamente mais simples: uma realizada entre a língua-fonte e a metalíngua, e outra realizada entre esta metalíngua e a língua-alvo.

Mas será que essa abordagem simplifica o processo? Sim. Para ficar claro, imagine o seguinte cenário onde existem documentos escritos em n línguas distintas e queremos traduzi-los de e para quaisquer dessas línguas. Poderíamos construir arranjos de n tradutores⁷ automáticos para cada par de línguas (n_1 - n_2 , n_1 - n_3 , n_1 - n_4 , ..., n_2 - n_1 , n_2 - n_3 , n_2 - n_4 , ...) seguindo o modelo de transferência (tradução automática baseada em regras). Mas a tradução por interlíngua mostra-se mais vantajosa. Podemos definir uma metalíngua e dividir a tarefa em n grupos de especialistas/tradutores, cada um responsável por uma única língua, l_i (de preferência sua língua materna). Caberá a cada grupo construir um tradutor (lembre-se: bem mais simples!), ou codificador, de l_i para a metalíngua, e outro, decodificador, da metalíngua para l_i . Evidentemente todos os grupos compartilham o mesmo conhecimento sobre a língua intermediária.

Considerando que cada grupo fará sua parte, ao final, as possíveis combinações desses módulos de tradução darão origem a todos os tradutores almejados. Por exemplo, para traduzir l_i para l_j , juntamos o módulo codificador de l_i com o módulo decodificador para l_j . Terão sido construídos $2n$ módulos de traduções mais simples, portanto, menos esforço do que o exigido para os tradutores bilíngues. Uma outra consequência é a possibilidade de se avaliar as traduções por meio da tradução inversa, já que os módulos independentes permitem a tradução nos dois sentidos.

Esse ideal foi compartilhado algumas vezes, no passado, por vários grupos de pesquisa acadêmica, mas infelizmente a prática evidenciou vários **problemas**. O maior deles é, segundo os críticos, a **ingenuidade** em se acreditar possível criar uma linguagem capaz de representar o significado de todas as outras, portanto, universal. Um outro problema – ou

⁷Para calcular a quantidade total de tradutores necessários, usamos a fórmula de arranjo de n línguas para combinação em pares ($k = 2$), $A_{n,2}$ que é $(n!)/(n-2)!$. Assim, para 5 línguas teríamos que construir 20 tradutores no sentido direita-esquerda e mais 20 tradutores no sentido esquerda-direita, totalizando 40 tradutores distintos!



decorrente deste – é a adoção unânime de uma dada língua intermediária pelos grupos de línguas distintas, onde cada um deles reivindica alterações e adaptações, escancarando a **inexistência de um núcleo verdadeiramente universal**. Por essas e outras razões é que esse modelo não substituiu o modelo por transferência bilíngue.

No final dos anos 1990, o português brasileiro estava representado, pelo NILC⁸, numa iniciativa da ONU para construção de tradutores para as línguas mais faladas no mundo, o Projeto UNL⁹. Esse projeto tinha por objetivo o desenvolvimento de um sistema multilíngue de tradução automática baseada numa interlíngua de natureza semântica – a *Universal Networking Language* (UNL) – desenvolvida por pesquisadores vinculados à Universidade das Nações Unidas, órgão da ONU, em Tóquio¹⁰.

O **paradigma linguístico** (baseado em regras e interlíngua), no qual o conhecimento linguístico é explicitamente mapeado em recursos como regras, dominou o cenário da tradução automática **até a década de 1980**, quando abordagens baseadas em *corpus* (empíricas) surgiram. Aliadas à motivação de tentar superar as limitações da tradução baseada em regras, essas abordagens foram impulsionadas por dois fatores: (1) os **avanços no hardware** necessário para processamentos computacionais mais pesados, e (2) a **disponibilidade maior de recursos bilíngues**, em especial os *corpus* paralelos. As próximas seções tratam das abordagens baseadas em *corpus*: a tradução baseada em exemplos, a tradução estatística e a tradução neural.

18.2.4 Tradução Automática Baseada em Exemplos

Os sistemas de TA baseada em exemplos (do inglês, *Example-based Machine Translation* ou EBMT), também conhecidos como tradução por analogia, estão frequentemente associados à publicação do artigo de Nagao (1984), no qual o autor propõe um modelo baseado na imitação de exemplos de tradução de frases semelhantes, buscando utilizar a ideia de aprender a traduzir a partir de exemplos existentes (Koehn, 2020). Os sistemas de exemplos utilizam informações extraídas (sequências de palavras) de exemplos em *corpora* bilíngues de pares de tradução, alinhados em nível de sentença, ao qual convencionou-se chamar de *corpora* paralelos.

Por meio dessa abordagem, exemplos como os ilustrados no Quadro 18.2 serviriam de base para o sistema aprender traduções de trechos de texto, como as ilustradas em Quadro 18.3.

Quadro 18.2. Exemplos para a tradução baseada em exemplos

A casa é muito bonita.	<i>The house is very beautiful.</i>
Meu avô foi internado ontem.	<i>My grandfather was hospitalized yesterday.</i>
Eu comprei uma jaqueta linda.	<i>I bought a beautiful jacket.</i>

Quadro 18.3. Trechos aprendidos

⁸<http://www.nilc.icmc.usp.br/>

⁹Esse projeto deu origem à UNDL *Foundation* <http://www.undl.org/>.

¹⁰Mais informações sobre a linguagem UNL podem ser obtidas em (Uchida; Zhu; Della Senta, 1999). Detalhes sobre o Projeto UNL-Brazil podem ser encontrados em (Martins et al., 2000; Nunes et al., 2003) e <http://www.nilc.icmc.usp.br/nilc/projects/unl.htm>.



a casa	<i>the house</i>
meu avô	<i>my grandfather</i>
linda	<i>beautiful</i>

A partir dos trechos aprendidos, o sistema baseado em exemplos seria capaz de combiná-los para, a partir da entrada, gerar a saída apresentada em Exemplo 18.2:

Exemplo 18.2.

Entrada: A casa do meu avô é linda.

Saída: *The house _ my grandfather _ beautiful.*

Devido ao uso de diferentes métodos e técnicas, de acordo com Hutchins (2005, p. 63): “não parece haver um consenso claro sobre o que é ou o que não é um sistema de exemplos”. Para Carl; Way (2003, p. xix), uma definição analítica dos sistemas de exemplos era difícil, pois, segundo ele, tais sistemas “assumem uma posição entre os sistemas baseado em regras e os estatísticos” ao utilizar abordagens tanto baseadas em regras quanto orientadas por dados¹¹. No entanto, tanto os sistemas baseados em exemplos como os estatísticos se enquadram no paradigma de TA baseada em *corpus*. Enquanto alguns autores veem os sistemas baseados em exemplos como um paradigma em si mesmo, outros consideram os sistemas estatísticos como um tipo de sistema de exemplos, já que os primeiros sistemas estatísticos surgiram no final da década de 1980.

18.2.5 Tradução Automática Estatística

Os sistemas de TA estatísticos (em inglês, *Statistical Machine Translation* ou SMT) foram apresentados pela primeira vez por Brown et al. (1988); no entanto, a ideia de usar métodos estatísticos para traduções automáticas foi introduzida pela primeira vez por Weaver em 1949 (Brown et al., 1988, p. 71). Desde a primeira publicação de Brown et al., a equipe da IBM desenvolveu para a empresa o primeiro sistema estatístico funcional e houve um aumento drástico na pesquisa em TA estatística na área.

A ideia geral dos sistemas estatísticos é usar modelos estatísticos para extrair pares de tradução de *corpora* bilíngues. Podem ser encontradas três abordagens principais para a TA estatística:

- TA estatística baseada em palavras (*Word-based Statistical Machine Translation*): alinha¹² palavras individuais no texto-fonte a palavras no texto-alvo e calcula a probabilidade da tradução. Também permite a exclusão e inserção de palavras.
- TA estatística baseada em frases (em inglês, *Phrase-based Statistical Machine Translation* ou PBSMT): alinha frases (não frases linguísticas, mas fragmentos de frases e palavras) no texto-fonte a frases no texto-alvo, comparando frases e seus vizinhos frasais ao considerar uma tradução. Essas frases também são chamadas

¹¹Para uma descrição mais abrangente e para a história dos sistemas de TA baseados em exemplos, consulte (Carl; Way, 2003).

¹²No contexto da TA estatística, o alinhamento é uma tarefa de encontrar as correspondências entre texto-fonte e texto-alvo. Esse alinhamento pode se dar em nível de palavras (alinhamento lexical), de sentenças (alinhamento sentencial) entre outros.



de n-gramas, que são sequências contínuas de n palavras em sequência, ou seja, um unigrama é uma palavra, um bigrama são duas palavras, um trigramas são três palavras etc. A TA estatística baseada em frases é o tipo de abordagem estatística mais utilizado.

- TA estatística baseada em sintaxe: esses modelos traduzem unidades sintáticas usando árvores sintáticas geradas por analisadores sintáticos (Capítulo 6).

Independente da estratégia escolhida, na **tradução estatística** a **probabilidade** determina como um texto-fonte deve ser traduzido para um texto-alvo. De acordo com a estratégia escolhida, essa probabilidade pode ser calculada considerando apenas palavras ou também sequências de palavras (frases). Essas frases são sequências de *tokens* (não necessariamente palavras) como “a casa do” ou “linda .” (onde o ponto final faz parte da frase). Seja considerando apenas palavras ou frases, a tradução é realizada com base em dois modelos computacionais: (1) um **modelo de tradução** que especifica como mapear texto-fonte em texto-alvo e (2) um **modelo de língua** que especifica como gerar um texto-alvo fluente. Desse modo, o modelo de tradução tenta maximizar a acurácia da tradução, enquanto o modelo de língua tenta maximizar a fluência da sentença gerada na língua-alvo (Seção 18.3).

Para tentar tornar esses conceitos menos abstratos, vamos retomar nosso exemplo da sentença “A casa do meu avô é linda.”. Nesse caso, o modelo de tradução poderia ser baseado em probabilidades de tradução de frases como os gerados para o *corpus* FAPESP (Aziz; Specia, 2011) com o auxílio do Moses¹³, como ilustrado na Tabela 18.3.

Tabela 18.3: Exemplos de frases fonte e alvo, e suas probabilidades, presentes em um modelo de tradução gerado a partir do *Corpus* FAPESP (Aziz; Specia, 2011).

id	frase-fonte (português)	frase-alvo (inglês)	probabilidade
1	a casa do	' house	0.0207779
2	a casa do	the house from	0.0623338
3	a casa	the house	0.297619
4	a casa	the home	0.0646474
5	do meu	of my	0.0813954
6	do meu	that of my	0.191576
7	meu avô	my grandfather	0.662453
8	meu avô , meu pai , eu	my grandfather , my father , me	0.0623338
9	avô	grandfather	0.916667
10	é	is	0.611613
11	é	é	0.794943
12	linda	beautiful	0.0389678
13	linda	pretty	0.00259724

Com base nas probabilidades da Tabela 18.3, diversas opções de tradução poderiam ser geradas como as apresentadas no Exemplo 18.3.

Exemplo 18.3.

- (a) *The house from my grandfather is beautiful.*¹⁴

¹³<https://www.statmt.org/amos/> e <https://github.com/amos-smt/amosdecoder>

¹⁴Obtida por meio da combinação das frases 2, 7, 10 e 12.



- (b) *The house that of my grandfather is pretty.*¹⁵
 (c) *The home of my grandfather é beautiful.*¹⁶

E qual dessas sentenças o sistema escolheria como saída? Isso depende de alguns fatores que não vamos detalhar aqui, mas podemos dizer que o modelo de linguagem (Capítulo 15) tem um papel fundamental na seleção da melhor sentença candidata. Nesse caso, o modelo de linguagem diz qual é a melhor sentença com base na probabilidade de ela ser encontrada na língua-alvo, ou melhor, no *corpus* de treinamento usado para gerar o modelo de língua-alvo.

Assim, tanto o modelo de tradução quanto o modelo de linguagem são treinados a partir de *corpus*. No caso da tradução estatística, as probabilidades são determinadas contando-se as frequências de ocorrência das palavras em grandes quantidades de textos (os *corpora*) paralelos. Um exemplo de *corpus* paralelo português-inglês é o coletado por Aziz; Specia (2011), o Corpus FAPESP¹⁷, que serviu de base para a geração das probabilidades apresentadas na Tabela 18.3. Para a geração do modelo de tradução, a probabilidade de uma frase em português ser traduzida para uma frase em inglês é calculada com base na co-ocorrência dessas frases no *corpus* paralelo. Para a geração do modelo de língua, a probabilidade de uma frase em inglês é calculada com base na ocorrência dessa frase na parte em inglês do *corpus* paralelo (ou de outro *corpus* monolíngue na língua-alvo).

Geralmente os modelos de tradução e de língua consideram frases de um tamanho máximo definido como parâmetro do treinamento. Quanto maior o tamanho da frase, maior é o contexto sendo considerado e, como consequência, mais coerente poderá ser a sentença gerada (Jurafsky; Martin, 2023). Porém, quanto maior o tamanho da frase, maiores serão o tempo e a quantidade de processamento necessários para realizar o treinamento dos modelos.

As **vantagens** dos modelos estatísticos são que, com **mais dados** utilizados no treinamento dos sistemas, não apenas a qualidade geral aumenta, mas, com o uso de um modelo de linguagem, as traduções estatísticas ganharam **fluência** em relação às abordagens anteriores. Além disso, a TA estatística permitiu um **uso mais eficiente de recursos humanos e de dados**. As **desvantagens** dessa abordagem são o **custo de criação de corpora paralelos**, especialmente para idiomas com recursos limitados. Além disso, a TA estatística tende a ter **difícilidade com pares de idiomas com ordem de palavras diferentes**.

Em 2007, o sistema *open-source* PBSMT mais famoso, desenvolvido por Koehn et al. (2007), foi lançado: o Moses SMT *toolkit*. Ao mesmo tempo, o Google lançou seu famoso Google Tradutor com abordagens estatísticas. Vale ressaltar que os modelos estatísticos conseguiram obter grande sucesso devido ao “aumento do poder de computação e armazenamento de dados, juntamente com a disponibilidade cada vez maior de recursos de texto digital como consequência do crescimento da Internet” (Koehn, 2009, p. 18). Devido à eficiência e precisão da abordagem estatística em relação às anteriores, ela se tornou a abordagem mais amplamente utilizada naquela época. Sistemas de tradução estatística baseada em frases (PBSMT) como os de Koehn; Och; Marcu (2003) e Och; Ney (2004) eram o estado da arte até serem sucedidos pela tradução neural, a partir de 2015. De fato, a estratégia por trás do tradutor automático do Google foi a PBSMT por

¹⁵Obtida por meio da combinação das frases 3, 6, 9, 10 e 13.

¹⁶Obtida por meio da combinação das frases 4, 5, 9, 11 e 12.

¹⁷<http://www.nilc.icmc.usp.br/nilc/tools/Fapesp%20Corpora.htm>



uma década (aproximadamente de 2006/2007 até 2016/2017)¹⁸. Atualmente, o Google e praticamente todos os sistemas de tradução online, bem como pesquisas nesta área usam a tradução neural (*neural machine translation*, NMT) ou algum sistema híbrido (estatístico e neural).

18.2.6 Tradução Automática Neural

Os sistemas de TA neural (em inglês, *Neural Machine Translation* ou NMT) foram introduzidos pela primeira vez na década de 1990 com alguns artigos sugerindo como redes neurais poderiam ser usadas para TA¹⁹ (Way; Forcada, 2018). No entanto, a quantidade dos dados usados para treinar esses modelos não era suficiente para produzir resultados razoáveis e, além disso, “a complexidade computacional envolvida excedia em muito os recursos computacionais daquela época, e, portanto, a ideia foi abandonada por quase duas décadas” (Koehn, 2020, p. 39).

Em geral, os modelos neurais consistem na construção de redes neurais *end-to-end* que mapeiam textos paralelos alinhados e são treinados para maximizar a probabilidade de uma sequência alvo Y , dada uma frase de origem X , sem informações linguísticas externas adicionais (Castilho et al., 2017b). Os sistemas neurais podem ser construídos com apenas uma rede em vez de uma sequência de tarefas separadas, como seu predecessor (a tradução estatística).

Com a publicação de resultados impressionantes em avaliação automática (Bahdanau; Cho; Bengio, 2015; Bojar et al., 2016; Sennrich; Haddow; Birch, 2016a), os sistemas neurais geraram grande expectativa, especialmente porque a indústria de tradução busca melhorar a qualidade da TA para minimizar custos (Moorkens, 2017). A adoção dos sistemas neurais nos últimos anos tem sido extensiva, com um número crescente de provedores de TA e grupos de pesquisa concentrando seus esforços e recursos no desenvolvimento e implantação de sistemas NMT (Castilho et al., 2019).

Na tradução neural, redes neurais artificiais são usadas para fazer a tradução de uma sentença-fonte para uma sentença-alvo. Uma rede neural artificial pode ser entendida como uma composição de diversas unidades de processamento (os neurônios artificiais) conectadas entre si, em camadas. Cada unidade de processamento recebe uma entrada numérica e gera uma saída numérica. A saída é calculada de acordo com os “pesos” (w) e as “entradas” (x) associados à unidade e uma função que determina como a saída deve ser calculada. Por exemplo, vamos supor que um neurônio artificial seja governado pela função x^2 . Nesse caso, se a entrada para esse neurônio for o número 2 então a saída será 4, se for 3 a saída será 9, se for -1 a saída será 1 e assim por diante. Os pesos são usados para ajustar o aprendizado do neurônio e são uma das partes mais importantes da definição de uma rede neural artificial.

Na **tradução neural**, diversas camadas de neurônios são usadas para aprender como traduzir uma sentença-fonte em uma sentença-alvo a partir de um *corpus* paralelo. Esse aprendizado geralmente **demandava muito poder computacional²⁰ e tempo**, pois é realizado a passos pequenos em diversos ciclos de processamento²¹ das sentenças paralelas.

¹⁸Segundo informações disponíveis em: <https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>.

¹⁹Artigos como (Chalmers, 1992), (Chrisman, 1991), (Castano; Casacuberta, 1997), (Forcada; Neco, 1997), (Neco; Forcada, 1997).

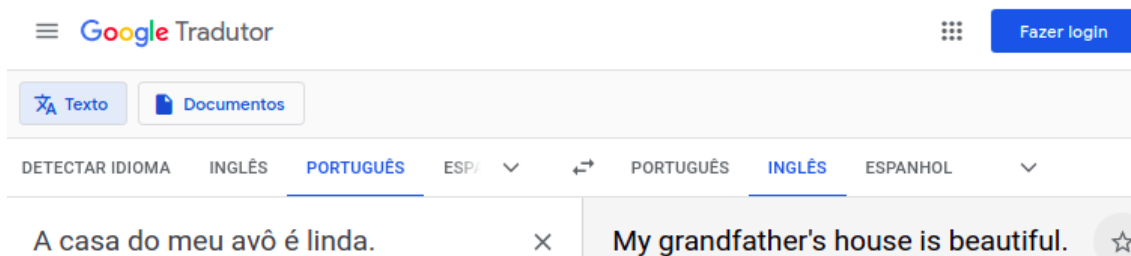
²⁰Geralmente são necessárias placas GPU (*Graphics Processing Units*), originalmente projetadas para processamento gráfico, capazes de fazer diversos cálculos em segundos.

²¹No contexto das redes neurais artificiais, um ciclo de processamento é chamado de época.



E qual é a principal diferença metodológica da tradução neural para a estatística? Na NMT toda a sentença-fonte é considerada no aprendizado, de uma vez, e nos dois sentidos (da esquerda para a direita e da direita para a esquerda), ou seja, não há a quebra em frases como ocorria na PBSMT, nem a divisão clara entre modelo de tradução e modelo de língua. Dessa forma, a tradução gerada por um sistema NMT tende a ser mais fluente e natural, como ilustrado na Figura 18.1²².

Figura 18.1: Tradução gerada pelo Google tradutor (tradução neural)



Os modelos de tradução neural baseiam-se fortemente em duas tecnologias que se tornaram bastante usuais em PLN: *embeddings* e modelo de atenção. As *embeddings* (Capítulo 10), são formas de representação de unidades lexicais (geralmente palavras) nas quais as unidades são mapeadas para vetores em um espaço de n (100, 300 ou mais) dimensões. Ao representar palavras como vetores densos notou-se que é possível mapear características linguísticas (morfológicas, sintáticas e semânticas) nesse espaço vetorial. Por exemplo, na Figura 18.2 é possível observar a proximidade semântica da palavra “avô” com outras palavras a partir das *word embeddings* do NILC²³ como “pai”, “tio”, “sobrinho” etc.

Figura 18.2: Vizinhos mais próximos da palavra “avô” obtidos via consulta às *word embeddings* do NILC geradas usando o GloVe e dimensão 300.

```
[('pai', 0.7426180839538574),
 ('tio', 0.7307775020599365),
 ('sobrinho', 0.6814965009689331),
 ('irmão', 0.6784138679504395),
 ('avó', 0.6334453821182251),
 ('filho', 0.6296581029891968),
 ('paterno', 0.628169059753418),
 ('bisavô', 0.6196369528770447),
 ('sogro', 0.5976346731185913),
 ('amigo', 0.5973160862922668)]
```

Usando as *word embeddings* bilíngues português e inglês do MUSE²⁴ é possível observar as similaridades entre línguas como ilustrado na Figura 18.3, onde as palavras em português aparecem em vermelho e as palavras em inglês, em azul. Essas *word embeddings* são usadas como forma de representação da língua nos modelos neurais de tradução automática.

Assim, a tradução neural não se baseia na combinação dos modelos de tradução e de língua, como faz a tradução estatística, mas sim em um modelo sequencial que prediz uma

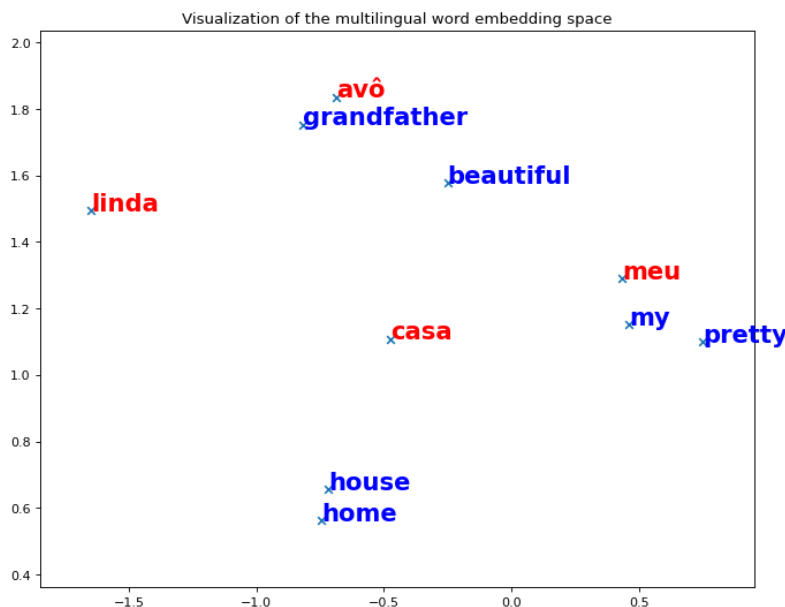
²²Tradução gerada pelo <https://translate.google.com.br/> em 23 de agosto de 2023.

²³<http://www.nilc.icmc.usp.br/embeddings>

²⁴<https://github.com/facebookresearch/MUSE>



Figura 18.3: Visualização, em duas dimensões, das palavras em português (em vermelho) das palavras que ocorrem na sentença de exemplo e algumas possíveis traduções para o inglês (em azul).



palavra por vez. O potencial deste modelo sequencial está na maneira como ele prediz as palavras: considerando toda a sentença-fonte e também o que já foi produzido para a sentença-alvo. Desde sua proposição, a modelagem sequencial neural passou por várias arquiteturas, indo desde as redes neurais recorrentes (em inglês, *recurrent neural network* ou RNN) usadas para codificação (em inglês, *encoder*) e decodificação (em inglês, *decoder*) até os mecanismos de atenção (em inglês, *attention mechanism*) (Bahdanau; Cho; Bengio, 2015) que permitem ao decodificador focar em partes específicas da sentença de entrada em seu processo de geração da saída.

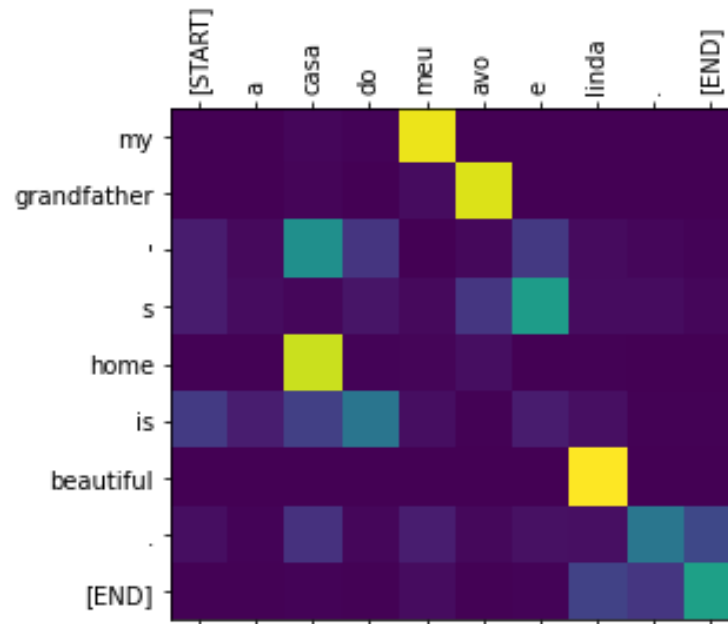
Nesse momento, os Transformers (Vaswani et al., 2017) são o estado da arte na tradução. A Figura 18.4 ilustra a tradução da sentença de exemplo, em português, para inglês usando Transformers²⁵. Nessa ilustração, quanto mais clara (amarelo, verde claro, azul claro etc.) a célula que une a linha da palavra em inglês com a coluna da palavra em português, maior a “força” da relação entre elas. Por exemplo, observa-se uma forte relação entre “my” e “meu”, “grandfather” e “avô”, “home” e “casa”, e “beautiful” e “linda”.

Contudo, assim como todas as demais abordagens, a tradução neural também tem suas **limitações**. Uma delas é que, diferentemente da PBSMT onde é possível “olhar” para os modelos aprendidos e entender o que foi usado na tradução (como as frases da Tabela 18.3), a tradução neural é considerada uma **caixa-preta** (*black box*): entender o que pode ter sido considerado para gerar a tradução depende de desvendar uma visualização do modelo de atenção (como o da Figura 18.4), já que as previsões dos modelos neurais consistem em milhões de parâmetros. Isso **dificulta a extensão dos modelos** previamente treinados e coloca em dúvida a **robustez do sistema**. Além disso, por ser uma abordagem relativamente nova, a tradução neural ainda enfrenta alguns desafios, como o **desempenho**

²⁵O modelo foi treinado usando o Google Colab e o código disponível em: https://github.com/brasileiras-pln/minicurso-PLN-SBBD/blob/main/Colabs/5.Aplicacoes_traducao_com_t5.ipynb.



Figura 18.4: Visualização de um modelo de atenção usado para traduzir a sentença de exemplo.



ruim em condições fora do domínio e para idiomas com recursos limitados.

Além disso, é possível observar que as estratégias de tradução baseadas em *corpus* são **fortemente influenciadas pelo *corpus* usado no treinamento**. Por exemplo, os modelos estatísticos só terão a capacidade de traduzir uma palavra se ela tiver ocorrido um número significativo de vezes no *corpus* de treinamento, caso contrário não haverá uma frase correspondente contendo essa palavra e o sistema não conseguirá gerar uma tradução completa para a sentença original. No caso da tradução neural, isso é um pouco amenizado pelo uso de *embeddings* de unidades menores (em inglês, *subword units*) do que as palavras, as quais conseguem aproximar palavras desconhecidas às possíveis correspondências conhecidas²⁶. Por exemplo, se o “a” for esquecido no “linda” da Figura 18.1 o Google tradutor consegue gerar a mesma saída, sem problema. O tratamento de palavras e *subwords* é abordado no Capítulo 4.

Outro ponto a se observar é que os sistemas neurais **precisam de um *corpus* maior e de melhor qualidade** do que os estatísticos, pois eles são rápidos em memorizar exemplos mal-formados (Khayrallah; Koehn, 2018). Por isso, para algumas línguas com menos recursos (em inglês, *low-resourced languages*) os sistemas estatísticos ainda podem apresentar um desempenho melhor do que alguns sistemas neurais. E, por esse motivo, muitas pesquisas atuais têm enfatizado o desenvolvimento de técnicas de aumento de dados (em inglês, *data augmentation*) para sistemas neurais.

Apesar disso, os sistemas neurais são o estado da arte na área de tradução automática (agora em 2023), apresentando, especialmente, uma **fluência muito superior** aos sistemas estatísticos, o que dificulta a avaliação humana da tradução, a qual deve ser mais cuidadosa aos erros de acurácia.

Atualmente, existem várias técnicas de aprendizado profundo para os sistemas neurais,

²⁶<https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html>



diferentes ramos, orientações de pesquisa e tendências²⁷.

18.3 Avaliação da Tradução Automática

Devido à importância da tradução no mundo globalizado de hoje, o interesse na avaliação da qualidade da tradução (AQT – do inglês, *Translation Quality Assessment* ou TQA) cresceu a ponto de a avaliação da TA (ATA) se tornar um subcampo em rápido crescimento dentro da TA. No entanto, como a tradução é um processo multifacetado que envolve fatores cognitivos, linguísticos, sociais, culturais e técnicos, definir e medir a qualidade da tradução também reflete essa complexidade (Castilho et al., 2018). A AQT tem sido um tópico muito discutido em estudos de tradução, tecnologia da tradução e na indústria de tradução e localização, mas ainda não há muito consenso sobre o que é e como ela deve ser feita (Castilho et al., 2018).

Aqui, faremos a distinção entre a AQT e a ATA: enquanto a AQT abrange a avaliação tanto das traduções humanas quanto das traduções automáticas, a ATA se concentra exclusivamente na avaliação da qualidade dos sistemas de TA. Nesta Seção, iremos definir a avaliação da TA, apresentar diferentes abordagens e discutir algumas das avaliações mais influentes na sua história, destacando a importância de realizar a avaliação dos sistemas de TA.

18.3.1 O que é Avaliação da Tradução Automática?

A avaliação da Tradução Automática (ATA) é a prática de analisar a saída de tradução de um sistema (ou sistemas) de TA e julgar a qualidade dessa tradução com base em critérios estabelecidos. As abordagens para a ATA incluem **avaliação automática**, usando métricas automáticas Seção 18.3.3, ou **avaliação manual humana** Seção 18.3.4, realizada por pessoas, e às vezes uma combinação das duas. O fluxograma da Figura 18.5 foi proposto por Doherty et al. (2018) para ajudar educadores e tradutores nos vários tipos de ATA.

Devido à falta de consenso no campo dos estudos de tradução e na indústria da tradução sobre o que constitui uma “boa” tradução, várias abordagens para a avaliação da TA surgiram. Portanto, a ATA pode ser realizada de várias maneiras diferentes, com várias abordagens, e não existe um único método que seja suficiente para abordar todos os propósitos de avaliação (Hovy; King; Popescu-Belis, 2002).

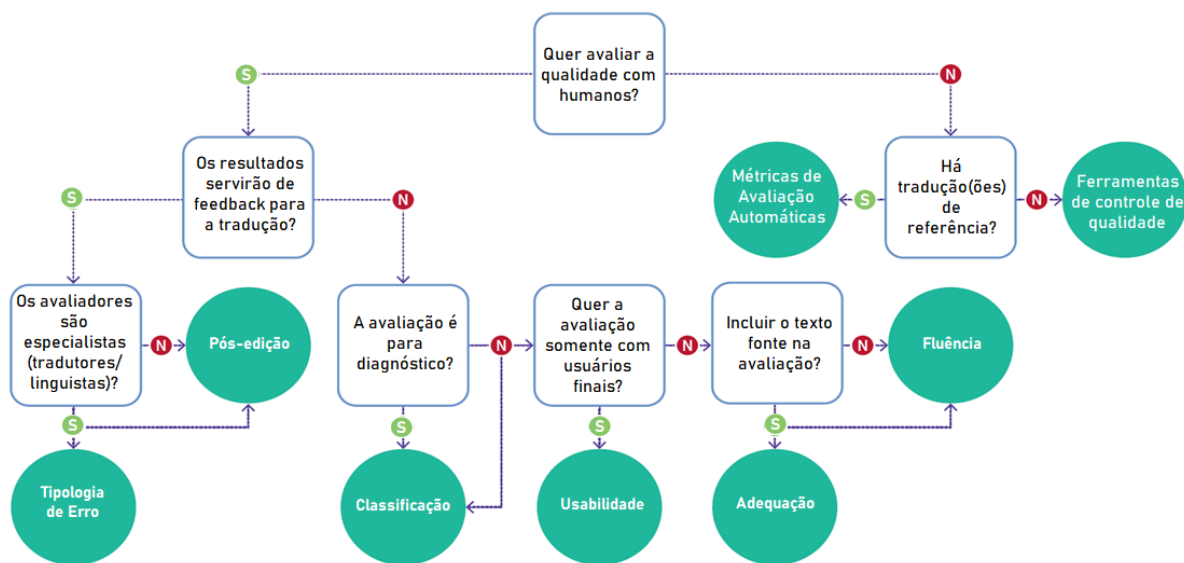
Tradicionalmente, a avaliação da TA foi dividida em dois paradigmas: avaliação glass-box (caixa de vidro) e avaliação black-box (caixa-preta). Enquanto a primeira se preocupa com “a qualidade de um sistema com base em suas propriedades internas” (Dorr et al., 2011, p. 744) e foi amplamente utilizada com sistemas baseados em regras (Seção 18.2.2), a última “mede a qualidade de um sistema apenas com base em sua saída, sem considerar os mecanismos internos do sistema de tradução” (ibid). As abordagens de avaliação black-box são o foco desta Seção.

A avaliação desempenha um papel essencial na TA, pois fornece informações sobre o funcionamento do sistema, quais as partes são eficazes e quais as áreas que precisam de melhorias. No entanto, a avaliação é um problema complexo, pois não existe uma única tradução correta para uma determinada fonte, e “pode haver várias traduções corretas possíveis” (ibid).

²⁷Para uma descrição abrangente do estado da arte da tradução automática neural, veja “Neural Machine Translation” de Koehn (2020).



Figura 18.5: Fluxograma da ATA baseado em Doherty et al. (2018)



Para julgar a qualidade de uma determinada saída de TA, é necessário definir o que a “qualidade” significa para essa tarefa de tradução especificamente. Mas quem define o que é qualidade da tradução? A resposta para essa pergunta, definindo o que constitui uma tradução “boa” ou “ruim”, tem sido e ainda é amplamente debatida nos diferentes campos relacionados à tradução. Para O’Brien et al. (2011, p. 55), a qualidade está relacionada à opinião do cliente, mas na indústria da tradução, a avaliação de qualidade “é gerenciada por intermediários na cadeia de fornecimento e demanda”, geralmente usando uma abordagem única para todos (em inglês, *one size fits all*). Para a ATA, a avaliação de qualidade depende do uso pretendido da tradução. Portanto, qualidade pode significar uma tradução *fluente* que pareça ter sido escrita por um falante nativo; ou uma tradução *precisa* que transmita todo o significado expresso no texto de origem; ou talvez qualidade signifique uma tradução que seja ao mesmo tempo fluente e adequada. Além disso, qualidade pode ser definida como uma tradução fácil de ser pós-editada por alguns, enquanto outros podem definir qualidade como uma tradução que os usuários finais possam usar. E alguns podem ainda querer que todos esses critérios sejam atendidos para se ter uma tradução de qualidade, enquanto outros podem definir qualidade da tradução com outros critérios totalmente diferentes.

Em resumo, ao avaliar a saída da TA, essencialmente estamos tentando determinar graus de “qualidade” e “não qualidade” dessa tradução para um público específico, uma vez que um critério de qualidade que seja crucial em determinada cenário, pode ser irrelevante em outros (Dorr et al., 2011).

Nesse sentido, alguns projetos de tradução que merecem referência são:

- **ALPAC Report (1966)**: examinou a qualidade e eficácia dos sistemas de TA nos Estados Unidos. O relatório concluiu que a TA não era útil devido à baixa qualidade, o que resultou em uma redução significativa no financiamento para o processamento de linguagem natural (PLN) e TA naquela época. O relatório também desencorajou discussões sobre avaliação dentro da comunidade de PLN por muitos anos.
- **DARPA Initiative (1992)**: uma das primeiras iniciativas na avaliação de TA nos anos 1990. A metodologia de avaliação era baseada em julgamentos humanos,



onde os avaliadores atribuíram uma pontuação de 1 a 5 para as frases traduzidas automaticamente, comparando-as ao texto original e a traduções de referência humanas produzidas por tradutores profissionais.

- **FEMTI (2002)**: foi um esforço para padronizar o processo de avaliação de TA, organizando os métodos existentes de ATA e relacionando-os com o propósito e contexto dos sistemas, fornecendo “diretrizes para a seleção de características de qualidade a serem avaliadas, dependendo da tarefa esperada, dos usuários e das características de entrada de um sistema de tradução automática” (Estrella; Popescu-Belis; King, 2009).
- **GALE (2005)**: teve como objetivo desenvolver e aplicar tecnologias de software para TA de grandes volumes de fala e texto em várias línguas. Foi estabelecido um protocolo de avaliação que utilizava métodos automáticos, humanos, baseados em tarefas e semi-automáticos, com foco na taxa de erro de tradução mediada por humanos (HTER).
- **EuroMatrix (2006)**: financiado pela Comissão Europeia, esta iniciativa construiu sistemas de TA estatísticos e híbridos para diversos idiomas europeus. A avaliação dos sistemas de tradução automática incluiu avaliações automáticas com base em onze métricas, incluindo BLEU, METEOR, Translate Error Rate, Word Error Rate, entre outros (Seção 18.3.3); e avaliações humanas que consistiram em classificações de fluência e adequação, *ranking*, tempo de leitura, tempo de pós-edição, teste de preenchimento de lacunas, clareza e informatividade (“Euromatrix. Survey of Machine Translation Evaluation”, 2007).
- **WMT (2006)**: começou como o Workshop de Tradução Automática Estatística e progrediu para a Conferência de Tradução Automática em 2016, tem sido uma grande contribuinte para o avanço da pesquisa em avaliação de TA. Ao longo dos anos, a WMT tem utilizado diferentes metodologias de avaliação. Desde 2008, a WMT realiza uma tarefa compartilhada, ou esforço colaborativo, (em inglês, *shared task*) de avaliação que compara uma ou mais traduções de referência com as saídas dos sistemas de TA e tem utilizado, principalmente, a métrica BLEU, embora essa tenha sido superada por diversas métricas diferentes. Em relação à avaliação manual, a WMT experimentou e implementou uma ampla gama de metodologias, incluindo julgamentos de fluência e adequação (2006 e 2007), diferentes avaliações de classificação (2007-2016), pós-edição e compreensão de frases (2009-2010), e avaliação direta e conjuntos de testes (desde 2016). Além disso, a *shared task* da WMT vem realizando avaliações em nível de documento desde 2019.
- **TAUS (2005)**: fundada em 2005 como uma rede de dados linguísticos com um extenso repositório de dados e uma rede de engenharia de linguagem humana, a TAUS tem estado na vanguarda das tentativas de estabelecer indicadores para a avaliação da qualidade de tradução, desenvolvendo o Framework de Avaliação Dinâmica de Qualidade (DQF) em 2011. O modelo é uma estrutura de avaliação baseada em diferentes tipos de conteúdo, propósito do uso, ferramentas, processos e outros aspectos.
- **QTLaunchPad (2012)**: projeto colaborativo financiado pela União Europeia (2012-2014) que reuniu e forneceu dados e ferramentas para a avaliação da qualidade de tradução, além de métricas de qualidade compartilhadas para tradução humana e automática (Uszkoreit; Lommel, 2013). O projeto contribuiu para o projeto de TA QT21 (2015-2018), cujo objetivo era desenvolver uma avaliação aprimorada. O QTLaunchPad desenvolveu o *framework* de Métricas de Qualidade Multidimensionais



(MQM), que descreve e define métricas personalizadas de qualidade de tradução para avaliar a qualidade de textos traduzidos. MQM é amplamente utilizado como um padrão para métricas de avaliação humana em MTEval. Como parte do QT21, o MQM e o DQF (da TAUS) foram harmonizados, e atualmente, a Tipologia de Erros DQF da TAUS é um subconjunto reconhecido do MQM.

18.3.2 A importância de uma avaliação replicável

Como visto neste Capítulo, a avaliação da TA tem sido realizada desde o surgimento da própria TA. No entanto, o relatório ALPAC tornou a avaliação quase um tópico proibido na comunidade de PLN (Paroubek; Chaudiron; Hirschman, 2007), e até hoje, a avaliação de TA, especialmente a avaliação humana, é considerada “desnecessária” em alguns momentos. Não é raro encontrar artigos de pesquisa usando uma única métrica para avaliar seu sistema (geralmente pontuações BLEU – ver Seção 18.3.3, e ainda mais comum são os artigos de pesquisa com avaliação humana muito limitada e falha, e até mesmo artigos de pesquisa sem avaliação humana alguma (Marie; Fujita; Rubino, 2021).

O avanço dos sistemas de TA gerou muita expectativa na comunidade, especialmente porque a indústria de tradução buscava uma melhoria na qualidade da TA para reduzir custos (Moorkens, 2017). À medida que os sistemas de TA neural se tornaram líderes no mercado, com uma clara melhoria na qualidade das traduções, surgiram reivindicações exageradas de que esses sistemas estavam “preenchendo as lacunas entre a tradução humana e a tradução automática” (Wu et al., 2016). Diante disso, pesquisadores na área de avaliação de TA alertaram a comunidade para ter cautela e não fazer promessas exageradas, além de enfatizarem a necessidade de mais pesquisas para lidar com as limitações dos sistemas de TA e realizar avaliações humanas mais abrangentes (Castilho et al., 2017b).

No entanto, as afirmações exageradas continuaram, com alguns declarando que seu sistema de TA neural havia atingido a “paridade humana” (em inglês, *human parity*) (Hassan et al., 2018) e outros alegando que a TA é um problema “resolvido” com uma qualidade de tradução “quase perfeita”. Como resposta, a fim de verificar essas alegações, dois estudos independentes (Toral et al., 2018) e (Läubli; Sennrich; Volk, 2018) reavaliaram os dados utilizados por Hassan et al. (2018) e descobriram que a escolha dos avaliadores, o contexto linguístico e a criação de traduções de referência têm um impacto significativo na avaliação de qualidade, apontando para falhas nas práticas atuais em avaliação de TA.

Como resultado, os pesquisadores de avaliação TA passaram a se envolver em discussões mais aprofundadas sobre a necessidade de aprimorar continuamente a metodologia de avaliação, “com o objetivo de superar as limitações das métricas automáticas e das abordagens humanas, evitar superestimativas na capacidade da TA e explicar os resultados aparentemente contraditórios da TA de forma mais abrangente” (Castilho et al., 2019, p. 2). Além disso, os pesquisadores começaram a buscar diretrizes para avaliar a “paridade humana” na avaliação da TA (Läubli et al., 2020).

Então, quem precisa de avaliação de TA? Em resumo, todos os campos diferentes relacionados à tradução. Algumas questões precisam ser abordadas para que a avaliação seja confiável, tais como:

- O que a **qualidade** significa nesta avaliação? Como vimos, há um grande debate sobre a definição de qualidade, então, antes de avaliar um sistema de TA, o pesquisador precisa definir o que seria uma tradução “boa” e “ruim” nesse cenário.
- **Que tipo de sistema** está sendo avaliado? Dependendo do par de idiomas, os sistemas estatísticos podem mostrar um problema específico. Já os sistemas neurais



são conhecidos por serem fluentes e conhecer a similaridade entre palavras, o que torna mais difícil detectar erros.

- Qual é o **objetivo** a ser alcançado com esta avaliação? Saber por que o sistema de TA está sendo avaliado é importante para decidir que tipo de avaliação precisa ser realizada. Por exemplo, se alguém quer saber se as implementações realizadas em um sistema de TA feito para lidar com expressões multipalavras resultaram em traduções corretas dessas expressões, uma análise linguística aprofundada da saída seria mais apropriada do que avaliar a fluência da saída.

Como podemos ver, a avaliação de TA é um problema complexo, e não é surpresa que tenha se estabelecido como um campo independente. Com os avanços na qualidade dos sistemas de TA atuais, a comunidade de avaliação precisa estar atualizada com os procedimentos que possam fornecer avaliações mais sólidas, capazes de confirmar ou refutar as alegações feitas, como afirmou Carl Sagan: “Alegações extraordinárias exigem evidências extraordinárias”.

18.3.3 Métricas Automáticas

Como vimos anteriormente, a ATA assume uma complexidade intrínseca devido a uma multiplicidade de fatores. Tipicamente, encontramos duas vertentes de avaliação: a avaliação automática e a avaliação humana (manual), ocasionalmente mescladas para compor uma abordagem híbrida. Nesta Seção, abordaremos as métricas automáticas mais predominantes, reconhecendo a sua influência no domínio da TA.

As métricas automáticas pioneiras empregadas na TA tiveram origem em outras áreas da PLN. Por exemplo, temos a *Word Error Rate* (WER), introduzida por Su; Wu; Chang (1992), que originou-se do campo de reconhecimento da fala (Capítulo 2). Por sua vez, a ROUGE, desenvolvida por Lin (2004), teve sua origem na sumarização automática. Outra métrica muito usada anteriormente foi a *F-measure*, empregada em recuperação de informação (Capítulo 16) e em diversas outras áreas do PLN, também encontrou aplicabilidade nesse contexto.

As Métricas de Avaliação Automática (MAA) são adotadas na TA quando se busca evitar a intervenção humana direta (Figura 18.5). As MAAs atuam como programas computacionais, recebendo as traduções de um sistema de TA e as traduções de referência (TR) como entrada, e produzindo uma pontuação numérica que reflete a similaridade entre as traduções de TA e TR.

A maioria das MAAs é classificada como métricas de referência (em inglês, *reference-based metrics*), exigindo a disponibilidade da TR, isto é, a tradução humana do texto em avaliação, a fim de serem empregadas como ponto de comparação. No entanto, abordagens mais recentes incorporam modelos de linguagem pré-treinados para medir a semelhança entre uma tradução gerada e um conjunto de referências. Nesse contexto, essas métricas medem a similaridade entre as representações semânticas das palavras e as frases presentes tanto nas traduções geradas quanto nas referências, utilizando recursos linguísticos capturados durante o pré-treinamento desses modelos, em detrimento de uma comparação direta com traduções humanas específicas.

As MAAs preponderantes na área da TA são aquelas que operam sobre características lexicais e dispensam a necessidade de treinamento (em inglês, *untrained*). Essas métricas, geralmente baseadas em similaridades (em inglês, *matching*) e diferença de edições (em inglês, *edit distance*) entre o resultado da TA e a TR, avaliam a **sobreposição** entre a hipótese (resultado da TA) e a TR. Tal avaliação contempla tanto a precisão quanto a



abrangência dos elementos lexicais (Lee et al., 2023). Duas vertentes são identificadas entre as métricas lexicais: as *word-based* (baseadas em palavras), que analisam as similaridades entre palavras; e as *character-based* (baseadas em caracteres), que investigam a similaridade entre caracteres.

As métricas lexicais *word-based* mais amplamente empregadas permitem medir tanto a similaridade dos *n*-grama quanto a distância de edição (*edit distance*). Dentre as métricas baseadas em *n*-grama, destacam-se as amplamente conhecidas BLEU (Papineni et al., 2002), METEOR (Banerjee; Lavie, 2005) e NIST (Doddington, 2002). Por outro lado, as métricas que calculam a distância de edição e que têm destaque são TER (e HTER) (Snover et al., 2006) e WER (Su; Wu; Chang, 1992). Vale mencionar a singularidade da métrica chrF (Popović, 2015), a qual, além de ser *character-based*, também mede a similaridade dos *n*-gramas.

Mais recentemente, métricas treinadas em modelos baseados em redes neurais usando a arquitetura Transformer foram propostas. Dentre essas, há as métricas supervisionadas (*supervised-metrics*) e as não-supervisionadas, dependendo da técnica de aprendizado, ambas categorias com *word-embeddings* e *contextual-embeddings* (Lee et al., 2023). Entre as métricas não-supervisionadas mais recorrentes, destacam-se MEANT (*word embedding*) (Lo; Wu, 2011), BERTscore (Zhang et al., 2020), Yisi (Lo, 2019) e BARTscore (Yuan; Neubig; Liu, 2021) (*contextual-embedding*). Entre as supervisionadas, estão a BEER (Stanojevic; Sima'an, 2014) e BLEND (Ma et al., 2017) (ambas *word-embeddings*), BERT for MTE (Shimanaka; Kajiwara; Komachi, 2019), BLEURT (Sellam; Das; Parikh, 2020) e COMET (Rei et al., 2020).

As **vantagens** das MAAs são que elas são **eficientes**, **econômicas** e fornecem avaliações **consistentes**, ou seja, se a métrica for computada para a mesma tradução várias vezes, todas elas vão dar o mesmo resultado. No entanto, uma **preocupação** é a **dependência** exclusiva das similaridades entre a saída do sistema e as referências. Primeiramente, não há somente uma única tradução correta para um texto, sendo assim, o significado do texto pode ser traduzido de várias maneiras. Mas seriam todas as tradução “igualmente boas”? E o que “boa” significa nesse determinado contexto da tradução? Nesse caso, usar múltiplas TRs seria essencial para se ter uma avaliação mais justa. Segundo, as MAAs não oferecem insights detalhados sobre erros de tradução, pontos fortes e fragilidades de um sistema. Elas não dizem o que funciona no sistema, o que precisa ser melhorado; sendo o único objetivo medir a semelhança com a(s) referência(s), e conseqüentemente, as melhorias específicas decorrentes de modificações no sistema de tradução permanecem obscuras. E finalmente, o sistema com uma pontuação menor pode ser melhor na prática do que um sistema com uma pontuação mais alta. Enquanto as MAAs servem como ferramentas quantitativas valiosas, elas não revelam completamente as complexidades da qualidade da tradução. Uma abordagem mais abrangente, combinando MAAs com avaliações humanas e análises qualitativas, oferece uma compreensão mais profunda do desempenho dos sistemas de TA.

Embora as MAAs não se revelem apropriadas para mensurar a qualidade final dos sistemas, impulsionam o avanço da pesquisa em TA, uma vez que podem ser empregadas de forma constante durante o desenvolvimento e a implementação desses sistemas. Em essência, as MAAs são medidas úteis na comparação entre sistemas de TA ou de versões de um mesmo sistema de TA, mas são limitadas na predição da qualidade da tradução.



18.3.4 Métricas Humanas

O processo de avaliar a qualidade da TA por meio da intervenção humana é essencial. Embora as MAAs proporcionem uma avaliação quantitativa, a avaliação humana oferece uma visão mais detalhada e uma análise mais ampla de fenômenos linguísticos complexos subjacentes ao desempenho dos sistemas de tradução, sendo assim imprescindível em uma compreensão mais abrangente dos sistemas de TA.

A avaliação humana pode ser feita através de diversos paradigmas, sendo os mais comuns o paradigma de **fluência-adequação** e **pós-edição**. A abordagem de **ranqueamento** de segmentos (em inglês, *ranking*) também é comumente empregada para a comparação dos sistemas de tradução, e possibilita a avaliação comparativa de diversos sistemas, fornecendo insights sobre a eficácia relativa de suas saídas. Igualmente, a **anotação de erros**, sob a forma de marcações específicas, oferece um feedback valioso sobre os sistemas em análise.

Outras abordagens incluem métricas secundárias, tais como legibilidade, compreensibilidade e usabilidade das traduções resultantes. Vale mencionar métricas centradas no usuário, que são avaliadas com testes de compreensão, por exemplo, que podem ser utilizados para aferir não apenas a fidelidade à tradução, mas também a transmissão efetiva da mensagem subjacente. Essa abordagem complementar permite uma apreciação mais holística da eficácia das traduções geradas.

Importante ressaltar que a avaliação humana é realizada por uma variedade de avaliadores, incluindo tanto profissionais quanto amadores. Essa diversidade de perspectivas pode contribuir para uma avaliação mais robusta e representativa da real eficácia dos sistemas de TA se empregados na mesma avaliação. Porém, também pode resultar em conclusões diferentes se a avaliação é feita só com um ou com o outro (só amadores ou só profissionais, por exemplo). Essa Seção vai abordar as métricas humanas mais comuns na avaliação de TA, assim como a importância de se calcular a concordância entre anotadores.

18.3.4.1 Adequação: Fidelidade Semântica

A **adequação** (em inglês, “*accuracy*” ou “*fidelity*”), ou também “acurácia” ou “exatidão”, é uma métrica importante na avaliação humana da TA. Ela foca na **fidelidade semântica** entre o texto-fonte e a tradução, mostrando a profundidade e a precisão da transferência de significado, permitindo determinar se a mensagem que está sendo transmitida e seu sentido foram preservados de maneira precisa e fiel.

Essencialmente, a adequação investiga até que ponto a tradução transmite o significado do texto de origem para o texto-alvo. Nesse contexto, uma escala Likert²⁸ é utilizada para classificar o nível de transferência semântica, onde geralmente uma pergunta é feita para o avaliador: “Até que ponto a tradução transfere o significado do texto-fonte para o texto-alvo?”. Como resposta, o avaliador pode escolher uma opção em uma escala que varia desde “Nada” até “Tudo”, com os graus intermediários de “Pouco” e “Muito”.

As **limitações** da adequação é que ela **não fornece informações sobre a fluência** da tradução, deixando uma lacuna importante na avaliação. Em determinados casos, o foco reside exclusivamente no significado da sentença de origem, tornando a fluência uma preocupação secundária. Além disso, a adequação não oferece detalhes precisos sobre os

²⁸A escala Likert é um método de medição criado por Likert (1932) que apresenta ao respondente uma afirmação ou pergunta e solicita que o respondente avalie o grau em que concorda com ela. A escala envolve uma série de itens ou afirmações aos quais os respondentes atribuem níveis de concordância ou discordância.



erros presentes na tradução, o que pode dificultar a identificação de pontos específicos para a melhoria do sistema.

18.3.4.2 Fluência: Naturalidade e Estrutura

A **fluência**, ou “inteligibilidade” (em inglês, “*fluency*” ou “*intelligibility*”), é outra métrica importante na avaliação humana da TA, que se preocupa com a **naturalidade e a estrutura** do texto-alvo, revelando o grau de fluência e adaptabilidade da saída da TA às normas linguísticas e socioculturais da língua-alvo.

Essa avaliação foca diretamente no **texto-alvo**, priorizando a avaliação da gramática e dos aspectos estruturais da tradução. Essencialmente, ela investiga o quão natural e fluido é o fluxo do texto-alvo dentro do contexto da língua-alvo, considerando suas normas linguísticas e socioculturais específicas. Uma característica distintiva da dimensão de fluência é que ela **pode ser avaliada independentemente do texto-fonte**, uma vez que se concentra exclusivamente no resultado final da tradução.

Também medida numa escala Likert, uma pergunta típica dirigida ao avaliador poderia ser: “Quão fluente está o texto-alvo, ou seja, como está o fluxo e a naturalidade do texto-alvo no contexto da língua-alvo e suas normas linguísticas e socioculturais em um dado contexto?”. A escala Likert pode variar desde “Sem fluência” até “Nativo”, incluindo os graus intermediários de “Pouca fluência” e “Quase nativo”.

Essa métrica oferece informação sobre a naturalidade da tradução e se ela soa natural e fluída para um falante nativo ou se exibe características de “linguagem quebrada”, prejudicando assim a experiência de leitura e compreensão.

Assim como a adequação, a avaliação da fluência também apresenta **limitações**. Ela **não proporciona informações sobre a adequação** da tradução, pois o foco está exclusivamente na fluência da sentença de destino, tornando a adequação uma preocupação secundária. Além disso, a fluência também **não oferece detalhes precisos sobre os erros presentes na tradução**.

Por esse motivo, é comum que as avaliações de adequação e fluência caminhem juntas, uma vez que é mais intuitivo avaliar uma em relação à outra. No entanto, há momentos em que pode ser necessário priorizar uma em detrimento da outra. Documentações técnicas, por exemplo, podem demandar uma maior ênfase na adequação, priorizando a transmissão precisa do significado.

Algumas **considerações sobre o uso de escalas Likert** na avaliação humana são importantes. Vale a pena ressaltar que as escalas Likert podem apresentar complexidades na sua aplicação. Diversos tipos podem ser utilizados, como escalas numéricas, de janela deslizante (em inglês, *sliding window*) e afirmações (concordo/discordo). Embora essas escalas sejam facilmente compreensíveis e quantificáveis, elas carregam uma natureza subjetiva, pois falham em medir as atitudes reais dos respondentes, levantando a questão: qual a diferença exata entre uma pontuação 5 e uma 4? (Um erro? Dois erros?). Além disso, a presença de um número par de opções pode levar os participantes a escolherem o centro, demonstrando a delicadeza desse tipo de avaliação.

18.3.4.3 Ranqueamento: Hierarquia de Traduções

O **ranqueamento** (em inglês, *ranking*) tem como propósito classificar e comparar duas ou mais traduções, com o intuito de estabelecer uma **hierarquia de qualidade** entre elas. As comparações podem ser efetuadas tanto entre as traduções geradas por diferentes



sistemas de TA quanto entre traduções geradas por humanos. Essa abordagem permite identificar **nuances de qualidade**, ressaltando as distinções entre as opções.

Além disso, o ranqueamento pode incorporar a **possibilidade de empates**, quando duas ou mais traduções são avaliadas como equivalentes em qualidade, sendo categorizadas como “igualmente boas” ou “igualmente ruins”. A categorização desses empates enriquece a análise, oferecendo insights sobre o grau de qualidade comparativa.

No âmbito de diagnósticos, esse tipo de classificação oferece a capacidade de indicar melhorias ao comparar o sistema avaliado com uma linha de base (*baseline*). Essa perspectiva não somente permite avaliar o progresso alcançado mas também identificar áreas específicas de aprimoramento, promovendo a constante evolução do sistema de TA.

Uma aplicação prática do ranqueamento é a seleção do sistema mais adequado para um projeto específico. A análise hierárquica das traduções permite a **escolha embasada no desempenho**, assegurando que a tradução atenda de forma eficiente aos requisitos e objetivos do projeto em questão.

O uso do ranqueamento apresenta suas **limitações** a serem consideradas. Ele **não oferece uma avaliação refinada**, não detalhando os erros presentes nas traduções. Quando empates não são permitidos, traduções igualmente boas ou ruins podem ser classificadas de maneira diferente, ressaltando uma inconsistência na hierarquia.

18.3.4.4 Anotação de Erros: Taxonomias

A anotação de erros se destaca como um método essencial para identificar e classificar imperfeições presentes em textos traduzidos. Diversas taxonomias foram propostas para essa finalidade, como Vilar et al. (2006), Font Llitjós; Carbonell; Lavie (2005), Federico et al. (2014), Costa et al. (2015), DQF de TAUS (O’Brien et al., 2011) e MQM (Lommel; Melby, 2018) by QT212 (Doherty et al., 2013). Para o português brasileiro, Martins (2014) e Martins; Caseli (2015) trazem a adaptação das categorias de erros de Popovic; Burchardt (2011) e Vilar et al. (2006) para traduções português-inglês.

As tipologias de erros frequentemente abrangem uma série de aspectos, incluindo palavras ausentes (em inglês, *missing words*), palavras adicionadas (em inglês, *added/extra words*), ordem errada das palavras (em inglês, *word order*), traduções literais (em inglês, *literal translation*), traduções erradas (em inglês, *mistranslation*), palavras incorretas (em inglês, *incorrect words*), formas inadequadas (em inglês, *incorrect form*), pontuação inadequada (em inglês, *punctuation*), entre outros, que podem incluir outras subcategorias específicas.

A adoção de taxonomias de erros na ATA oferece diversos **benefícios**, tais como identificar **tipos específicos** de erros nas saídas de TA, fornecer **relatórios detalhados de erros** para o aprimoramento dos sistemas, e fornecer informações aos clientes sobre a **qualidade da tradução**. Além disso, provedores de serviços linguísticos utilizam taxonomias e avaliações de severidade²⁹ para monitorar o trabalho de tradutores. A anotação de erro também ajuda a investigar as **relações entre tipos específicos de erros e as preferências de usuários ou pós-editores**, bem como avaliar o **impacto** de diferentes tipos de erros em várias etapas do processo de pós-edição.

Contudo, entre as principais **limitações** dessa estratégia, destaca-se que a anotação manual de erros é um **processo caro e demorado**, demandando um investimento significativo de tempo. Além disso, essa avaliação **nem sempre é uma tarefa simples**,

²⁹As severidades geralmente são classificadas como “Crítico” (*critical*), “Grave” (*major*), e “Mínimo” (*minor*).



especialmente quando se trata de diferenciar entre categorias como “tradução literal” e “tradução errada”.

Outra complexidade está associada à **dependência da língua**. Diferentes idiomas possuem particularidades que podem tornar a identificação e a classificação de erros uma tarefa mais desafiadora. Também é relevante considerar que a eficácia da anotação de erros pode variar de acordo com o **tipo de abordagem de TA**: enquanto ela pode ser mais adequada para sistemas de tradução baseados em regras (RBMT), pode não ser tão precisa para sistemas de tradução estatística (SMT) ou de tradução neural (NMT). Nesse contexto, a seleção da abordagem de avaliação mais adequada torna-se um ponto de reflexão. Além disso, a **falta de consenso entre avaliadores** é uma questão importante, frequentemente requerendo treinamento e prática para alcançar um nível satisfatório de concordância (Capítulo 14).

18.3.4.5 Pós-Edição na Avaliação de TA

A pós-edição (PE) (do inglês, *post-editing*) é definida como “a correção da saída da tradução automática bruta por um tradutor humano, de acordo com instruções e critérios de qualidade específicos” (O’Brien, 2011, p. 197). A PE emerge como uma ferramenta fundamental na avaliação de sistemas de TA, oferecendo uma perspectiva mais aprofundada sobre o esforço envolvido nesse processo. A medição desse esforço pode ser abordada de diferentes perspectivas (Krings, 2001), proporcionando uma visão mais abrangente do desempenho do sistema e do impacto da TA no fluxo de trabalho.

- **Esforço Temporal:** mede o ritmo de pós-edição. Avaliando o tempo gasto na pós-edição por palavras por segundos é possível compreender a velocidade desse processo. Nesse contexto, uma eficiência temporal maior, ou seja, menos tempo gasto na pós-edição, pode indicar uma melhor qualidade da saída da TA, influenciando a produtividade.
- **Esforço Técnico:** mede o número de operações de edição realizadas, como inserções (*insertions*), remoções (*deletions*), e trocas (*shifts*). Nesse sentido, a métrica hTER (Seção 18.3.3) é frequentemente utilizada como uma estimativa aproximada do esforço técnico. Uma menor quantidade de edições necessárias está diretamente correlacionada a uma melhor qualidade da TA, uma vez que está ligada ao tempo de esforço e, conseqüentemente, à produtividade.
- **Esforço Cognitivo:** pode ser medido por meio de diferentes abordagens, incluindo o rastreamento ocular (*eye-tracking*). A redução desse esforço cognitivo durante o processo de pós-edição é indicativa de uma qualidade superior da TA, e tal esforço tem sido correlacionado a outras métricas de avaliação humana.

A utilização da PE na avaliação da TA é motivada por diversos fatores. Além de avaliar a utilidade do sistema de TA em produção, ela **permite identificar erros comuns e gerar novos dados de treinamento ou teste**. Contudo, é importante ressaltar que as medidas de esforço de PE **tendem a variar entre avaliadores** novatos (estudantes) e profissionais, bem como entre o público em geral e profissionais experientes.

Alguns trabalhos com pós-edição com o português incluem: De Sousa; Aziz; Specia (2011); Almeida (2013), Castilho et al. (2014), Moorkens et al. (2015), Castilho et al. (2017a), Silva et al. (2017), Castilho; Resende; Mitkov (2019), Castilho; Resende (2022). Há também os trabalhos que investigaram a automatização do processo de pós-edição para o português: Caseli; Inácio (2020).



18.3.4.6 Considerações Finais sobre Avaliação Humana na Avaliação de TA

Ao explorar a avaliação humana na avaliação de TA, encontramos uma série de questões metodológicas e pragmáticas que merecem reflexão:

- Precisamos sempre avaliar tanto a adequação quanto à fluência?
- Quantos avaliadores são necessários e quais as competências linguísticas eles devem possuir?
- Devemos envolver tradutores, linguistas ou especialistas no domínio? O viés em cada escolha é um ponto a ser considerado.
- Quantos pontos devem ser incluídos em uma escala Likert de avaliação?
- Qual o grau de concordância entre avaliadores (Capítulo 14) e a consistência nas avaliações individuais?

Considerações pragmáticas também emergem, incluindo o custo associado aos avaliadores, à geração de textos de referência traduzidos por humanos, e à qualidade dessas referências. O tempo investido, a baixa concordância intra e inter-avaliadores, e a questão de saber se o objetivo da avaliação é apenas avaliar melhorias em um sistema são fatores preponderantes.

18.3.5 Avaliação dependente de contexto

Como vimos nas seções anteriores, a avaliação de TA começou desde o princípio da área com projetos como DARPA, que avaliaram a qualidade das traduções com métricas humanas. À medida que a área avançou, métricas automáticas foram gradualmente incorporadas, inicialmente provenientes de outros domínios do PLN, e posteriormente desenvolvidas especificamente para o campo da TA.

E apesar de ambas as MAAs e as métricas humanas serem o estado da arte na avaliação, à medida que os sistemas neurais de TA evoluíram, tornando-se mais complexos e produzindo traduções de maior qualidade, surgiu a necessidade de uma **avaliação mais abrangente e rigorosa** que levasse em conta fatores diversos. Adicionalmente, com o aumento nos esforços direcionados à incorporação de contexto nos sistemas de TA neurais, viu-se a necessidade de também se ter uma avaliação com contexto, uma vez que, os resultados obtidos na avaliação com sentenças eram limitados, pois ela não é capaz de identificar as melhorias desses sistemas (Läubli; Sennrich; Volk, 2018; Toral et al., 2018). Ademais, as MAAs subestimam a qualidade dos sistemas NMT (Shterionov et al., 2018), e a credibilidade dessas métricas para sistemas em nível de documento também tem sido objeto de críticas (Smith, 2017). Diante disso, surgiu a necessidade de avaliar a TA considerando um contexto mais amplo, possibilitando uma análise mais abrangente do contexto em questão. Entretanto, a metodologia para essa avaliação ainda está em sua fase inicial e poucos estudos foram realizados nesse sentido.

Em 2018, alegações de “paridade humana” (em inglês, *human parity*) na qualidade da TA (Hassan et al., 2018) foram rebatidas por Toral et al. (2018) e Läubli; Sennrich; Volk (2018), os quais apontaram que essa paridade não se replicava quando se considerava o **contexto** ou outros fatores, como a direção da tradução ou a experiência do anotador.

A Conferência de Tradução Automática (WMT), realizada desde 2006 e que até o ano de 2019 restringiu suas avaliações à análise de frases individuais, começou sua primeira tentativa de conduzir avaliações humanas em nível de documento no domínio de notícias no ano de 2019 (Barrault et al., 2019), em resposta às críticas apresentadas por Toral et al. (2018) e Läubli; Sennrich; Volk (2018). Adotando uma abordagem direta de avaliação



(Graham et al., 2016), a conferência solicitou que avaliadores de multidão³⁰ atribuísem uma pontuação (de 0 a 100) a cada sentença. Os avaliadores foram instruídos a avaliar: (i) textos completos, (ii) segmentos individuais consecutivos na ordem original, e (iii) frases individuais selecionadas aleatoriamente. No ano subsequente, na edição WMT20, houve uma mudança de abordagem, expandindo o âmbito de avaliação para abranger artigos completos, demandando dos avaliadores a análise de segmentos específicos enquanto visualizavam o documento completo, bem como a avaliação da tradução do conteúdo (Barrault et al., 2020).

Castilho (2020) e Castilho (2021b) testou as diferenças na concordância inter-annotadores (CIA) entre duas metodologias de avaliação: (i) uma centrada em sentenças individuais e (ii) outra com contexto, para o português brasileiro. No estudo de Castilho (2020), tradutores avaliaram a saída de TA considerando critérios de fluência, adequação (usando uma escala Likert), ranqueamento e anotação de erros. Essa avaliação foi conduzida em duas configurações distintas onde os tradutores atribuíram: (i) uma pontuação para cada sentença isolada, e (ii) uma pontuação para o documento como um todo. Os resultados demonstraram que os níveis de CIA para a metodologia em nível de documento atingiram níveis negativos, enquanto a satisfação dos tradutores com essa metodologia foi bastante reduzida. No entanto, esse enfoque evitou situações de avaliação incorreta (*misevaluation*) que são recorrentes quando se analisam sentenças isoladamente.

Continuando esse trabalho, Castilho (2021b) modifica a configuração em nível de documento e repete o experimento com mais tradutores, onde ela compara a CIA na avaliação de (i) sentenças únicas aleatórias, (ii) avaliação de sentenças individuais em que os tradutores têm acesso à fonte completa e à saída de TA, e (iii) avaliação de documentos completos. Os resultados mostraram que uma metodologia em que os tradutores avaliam sentenças individuais no contexto de um documento gera um bom nível de CIA em comparação com a metodologia de sentença única aleatória, enquanto uma metodologia em que os tradutores atribuem uma pontuação por documento mostra um nível muito baixo de CIA. A autora afirma que atribuir uma nota por sentença no contexto evita casos de avaliação incorreta que são extremamente comuns nas configurações de avaliação de frases aleatórias. Além disso, a autora postula que o maior acordo de CIA na configuração de sentença única aleatória ocorre porque “os avaliadores tendem a aceitar a tradução quando a adequação é ambígua, mas a tradução está correta, especialmente se for fluente” (Castilho, 2021b, p. 42), e afirma que **o método de avaliação de sentença única aleatória deve ser evitado**, pois o problema de avaliação incorreta é especialmente problemático ao avaliar a qualidade de sistemas NMT, uma vez que eles apresentam um nível aprimorado de fluência.

Após isso, em Castilho (2022) foi demonstrado que o contexto necessário para resolver questões de avaliação é influenciado pelo domínio, sem parecer estar intrinsecamente ligado ao comprimento das sentenças presentes no *corpus* envolvendo os idiomas inglês, português, irlandês, chinês e alemão. Em consequência disso, a pesquisa de Castilho et al. (2023) revelou que o impacto da extensão do contexto não parece influenciar significativamente os resultados, porém a estruturação da pontuação desempenha um papel crucial. Isso se deve ao fato de que sentenças conectadas tendem a gerar resultados mais diversos, com abordagens mais acuradas para resolver ambiguidades lexicais quando comparadas aos cenários de pontuação normais. Além disso, o estudo apontou que os sistemas GPT demonstraram proporcionar traduções mais precisas do que os sistemas de Tradução

³⁰Avaliadores contratados via plataformas como Mechanical Turk (<https://www.mturk.com/>).



Automática.

Diante desse panorama, a avaliação de TA com contexto encontra-se em sua infância, com diversas questões em aberto. O futuro da avaliação de TA deve considerar se as métricas automatizadas e as avaliações humanas atuais conseguem capturar de forma realista a qualidade dos sistemas de nível de documento, e se é necessário modificar ou criar novas abordagens. Especial atenção deve ser dada aos modelos de linguagem como o GPT, conhecidos por gerar traduções fluentes e coesas, uma vez que a avaliação deve incorporar precisão da informação, fidelidade ao conteúdo original e coerência global, evitando a introdução de informações imprecisas ou divergentes.

Ademais, a avaliação de documentos traduzidos não deve se limitar a métricas automáticas, mas também usar a avaliação humana. Os avaliadores humanos desempenham um papel crucial em identificar nuances de qualidade que as métricas automáticas podem não capturar, como aspectos culturais, ambiguidades e sutilezas linguísticas. Portanto, a **combinação de métricas automáticas com avaliações humanas** se mostra uma abordagem fundamental para obter uma compreensão abrangente da qualidade da tradução de documentos.

18.4 O Futuro da Tradução Automática

O futuro da TA parece muito promissor. Com a globalização e a internet, mais conteúdo é criado todos os dias, e, portanto, estão surgindo cada vez mais casos de uso nos quais a TA pode ser útil (Way, 2018). Segundo a “Slator 2019 Language Industry Market Report” (2019, p. 14), “a TA está bem encaminhada para se tornar a tecnologia mais importante para aprimorar a produtividade dos tradutores humanos”.

O aumento impressionante na qualidade com o surgimento da TA neural (NMT) em comparação com seu antecessor, o PSMT, foi exagerado pela mídia (Läubli; Sennrich; Volk, 2018; Toral et al., 2018), mas é incontestável que a NMT tenha sido, de fato, uma mudança de paradigma na área. No entanto, o entusiasmo em torno da TA diminuiu, com empresas de tradução de grande e médio porte relatando que, embora o uso da TA tenha aumentado, os benefícios percebidos têm se estabilizado (Sarah Hickey, 2020) em termos de grandes avanços na qualidade.

No entanto, com o aumento da qualidade, é possível abordar uma variedade maior de tipos de documentos e públicos. Isso significa que há muito espaço para **personalização de sistemas de TA** projetados para casos de uso e contextos específicos, melhorando a precisão. Para a TAUS (2020, p. 16), a NMT será “aplicada de forma útil em ambientes de tradução de fala” e, na verdade, em todo discurso falado, já que lida melhor com conteúdo gerado pelo usuário. Além disso, “a NMT ajudará na expansão adicional de tecnologias de tradução de fala [...] disponíveis principalmente como sistemas monolíngues baseados em inglês, [...] transformando-os em sistemas multilíngues”, o que implicará “muitas mudanças profundas e caras”.

Mais recentemente, no fim de 2022, os modelos de linguagem em larga escala (Capítulo 15), como o GPT-3 da OpenAI³¹, têm desempenhado um papel importante no campo da tradução automática e prometem desempenhar um papel ainda maior no futuro. Esses modelos surgiram com o avanço das redes neurais e do aprendizado profundo. Desde sua introdução, os LLMs têm sido amplamente utilizados na TA, proporcionando melhorias significativas na qualidade e na fluidez das traduções geradas. Eles têm sido capazes de

³¹<https://openai.com/>



lidar com nuances linguísticas, contexto e ambiguidades, resultando em traduções mais precisas e naturais. Com o contínuo avanço da tecnologia, espera-se que os LLMs sejam capazes de melhorar a personalização das traduções, adaptando-se a estilos de escrita específicos e preferências individuais.

No entanto, embora os LLMs tenham apresentado avanços significativos na área da TA, ainda existem **desafios** a serem superados. A qualidade da tradução depende de vários fatores, como a disponibilidade de dados de treinamento de alta qualidade e a compreensão do contexto e nuances linguísticas. Além disso, os LLMs podem ser sensíveis a preconceitos (bias) presentes nos dados de treinamento, resultando em traduções imprecisas ou enviesadas.

Segundo a pesquisa da CSA, “a pós-edição como serviço diminuirá ao longo do tempo, sendo substituída pela tradução automática adaptativa em software de tradução mais dinâmico” (p. 23), e haverá uma demanda crescente por linguistas profissionais que possam interagir com a saída da tradução automática “Slator 2019 Language Industry Market Report” (2019, p. 22). Além disso, o relatório da Slator afirma que agora, com os altos níveis de qualidade e a ampla disponibilidade de ferramentas gratuitas de tradução automática, os clientes corporativos esperam mais do que uma tradução automática “apenas boa” e estão buscando “soluções personalizadas, adaptadas ao seu conteúdo, que possam ser adaptadas para seus fluxos de trabalho e preferências estilísticas específicas” “Slator 2019 Language Industry Market Report” (2019, p. 22).

Vale ressaltar que todos os relatórios afirmam que a maioria da indústria “ainda não espera que a qualidade da tradução automática atinja os níveis da tradução humana em um futuro próximo” TAUS (2020, p. 16), e, portanto, tanto a tradução humana quanto a interação humana com a tradução automática ainda são altamente demandadas.

Como podemos ver, os sistemas de TA estão atingindo níveis de qualidade significativamente altos e, por isso, estão sendo cada vez mais utilizados em diversas áreas de negócio. Com a TA se tornando ubíqua em nosso dia a dia, a necessidade de testar a qualidade desses sistemas se tornou essencial (Castilho et al., 2019). Há muito espaço para a TA melhorar, e, portanto, uma boa prática na avaliação da TA é essencial para evitar afirmações exageradas e fornecer aos usuários um feedback honesto.



Capítulo 19

Correção automática de redação

Amanda Pontes Rassi
Priscilla de Abreu Lopes

19.1 Introdução

A Correção Automática de Redação (CAR) é uma das várias aplicações do PLN e pode ser definida como “o processo de avaliação e atribuição de nota em textos escritos em prosa, via programas computacionais” (Shermis; Burstein, 2013) ¹.

A correção manual de redações é uma prática bastante antiga, mas esse processo feito de forma automática data da década de 60, em inglês, e é ainda mais recente para o português.

Em inglês, as áreas de *Automated Essay Scoring* (AES) e *Automated Essay Evaluation* (AEE) surgem como distintas, porém complementares e, às vezes, com alguma intersecção. A primeira tem como desafio a automatização de atribuição de nota para redação, enquanto a segunda está preocupada, também, em automatizar o retorno ou feedback para o aluno, colaborando para o processo de aprendizagem da escrita.

A AES costuma ser traduzida para o português como **Avaliação Automática de Redação** (AAR) (Bittencourt Jr., 2020; Da Silva Jr., 2021; Lima et al., 2023), enquanto a AEE está associada ao termo **Correção Automática de Redação** (CAR), apesar do falso cognato. Neste capítulo, adotamos o segundo, por entendermos que ele abarca as duas áreas AEE e AES, ou seja, trata-se de uma solução completa. Para que seja considerada como solução completa de CAR, a aplicação deve contemplar pelo menos três etapas básicas:

- i) a detecção de desvios no texto;
- ii) a atribuição da nota, seja ela global ou por critério; e
- iii) um feedback para o aluno.

Cada uma dessas etapas pode ser vista como uma aplicação independente no PLN. Por exemplo, existem várias ferramentas de auxílio à escrita, bem como corretores ortográficos e gramaticais, que executam exclusivamente a tarefa de identificação de desvios no texto; e isso constitui uma aplicação em si. Da mesma forma, a tarefa de dar um feedback com sugestões para o aluno é semelhante a outras aplicações de PLN que envolvem geração de linguagem natural (ou *Natural Language Generation*).

Apesar de poderem figurar como ferramentas e/ou aplicações independentes, consideramos que a correção de redação, para ser entendida como uma solução completa do ponto de vista pedagógico, exige o cumprimento dessas três etapas, que serão bem detalhadas ao longo deste capítulo.

¹Tradução nossa. Do original: “*the process of evaluating and scoring written prose via computer programs*”.



Antes de abordar cada uma das etapas, porém, faremos uma breve explicação sobre o objeto de estudo da CAR, que é a redação escolar, definindo e exemplificando os principais gêneros e tipos textuais, os critérios avaliados e alguns modelos brasileiros de correção de redação.

19.1.1 O que é uma redação escolar?

A redação escolar é considerada um gênero textual, mas também pode ser distribuída em vários tipos e gêneros textuais. As redações, ou textos² de redação escolar, são geralmente utilizadas para avaliar as habilidades de escrita, interpretação, argumentação e criatividade dos alunos, bem como para desenvolver o pensamento crítico e a capacidade de expressão escrita. As redações podem abordar temas diversos, desde assuntos cotidianos até questões mais complexas e abstratas, e são uma forma importante de avaliar o progresso dos alunos ao longo do tempo.

Para fins didáticos e de correção de redação, é importante salientar a diferença entre tipo textual e gênero textual, já que as redações devem atender a um tipo específico e a algum gênero específico, a depender da proposta de redação. Por exemplo, a redação do Enem³ é sempre do tipo argumentativo e do gênero dissertação-argumentativa.

Os **tipos textuais** (ou “modos textuais”, para Marcuschi (2008, p. 154)) se referem à forma como o texto é organizado, ou seja, a sequência linguística e os aspectos lexicais, sintáticos, tempos verbais, relações lógicas que são mobilizados para constituir o texto. Existe um conjunto bastante limitado de tipos textuais, o qual abrange: narração, argumentação, descrição, exposição e injunção. Ressaltamos que um texto raramente apresenta apenas características de um mesmo tipo. Deste modo, classificamos um texto como sendo de um determinado tipo quando há predominância de elementos que o caracterizam.

Já os **gêneros textuais** são formas de comunicação que se desenvolvem em diferentes contextos sociais e culturais e se caracterizam pelo seu propósito ou objetivo comunicativo. Em outras palavras, cada gênero tem uma finalidade específica e é utilizado em determinadas situações comunicativas, dependendo de fatores sociais, culturais, dos falantes, da relação entre eles, do contexto, da finalidade da comunicação, dentre vários outros. Dependendo da situação comunicativa, cada gênero pode exigir um registro ou vocabulário específico, a norma culta ou coloquial, na modalidade escrita ou oral da língua.

Por esse motivo, os gêneros são mais fluidos, podendo surgir, modificar-se, mesclar com outros, desaparecer e reaparecer com outra roupagem em outro contexto ou época. São exemplos de gêneros textuais: bula de remédio, carta pessoal, diálogo informal, e-mail, edital de concurso, inquérito policial, piada, receita culinária, reportagem, resenha, sermão etc.

As redações podem apresentar diversos formatos e objetivos, dependendo do nível de ensino e do tema proposto pelo professor ou pela instituição de ensino. Dentre os tipos e gêneros textuais mais comuns associados à redação escolar, convém mencionar:

²Existe uma longa discussão conceitual e técnica sobre a definição do termo “texto” em Linguística Textual. Para o propósito deste capítulo, adotaremos como conceito de “texto” um conjunto de palavras e frases organizadas de forma coerente e coesa, com o objetivo de transmitir uma mensagem ou ideia. Em outras palavras, o texto é uma unidade de linguagem que tem um sentido completo e pode ser compreendido em um contexto específico.

³O Exame Nacional do Ensino Médio (Enem) é uma prova do Governo Federal que avalia o desempenho escolar dos estudantes ao término do Ensino Médio. Essa prova avalia várias áreas do conhecimento e também a produção de uma redação.



- **Dissertação**, em que o autor apresenta e disserta sobre um determinado tema, apresentando informações e argumentos relacionados ao assunto;
- **Narração**, em que o autor conta uma história, relatando fatos e acontecimentos em alguma ordem que pode ser cronológica ou não;
- **Carta**, em que o autor se dirige a um destinatário específico, fazendo requisições, solicitações ou expressando suas opiniões e sentimentos;
- **Artigo de opinião**, em que o autor defende um ponto de vista sobre um tema específico, utilizando argumentos e evidências para sustentá-lo;
- **Resenha**, em que o autor faz uma análise crítica de um texto, obra ou produto.

Esses são apenas alguns dos tipos e gêneros de redação escolar mais comuns. Os demais incluem a descrição, a exposição, a crônica, o relatório, o conto, a fábula, entre outros.

19.1.2 O que é avaliado?

Vários aspectos do texto são avaliados em uma correção de redação, tais como o uso da norma padrão da língua portuguesa, a adequação ao tema e ao gênero, questões relacionadas à coesão, à coerência, à progressão textual etc. Cada modelo de correção organiza e nomeia seus critérios de avaliação de formas distintas, mas, basicamente, todos eles analisam:

Língua portuguesa Avalia a linguagem usada para expressar o conteúdo, verificando se há desvios ortográficos e/ou gramaticais, se a norma (culto ou coloquial) está de acordo com o tipo de texto exigido, se há problemas de estrutura sintática nas frases, orações e períodos, se o vocabulário foi usado adequadamente etc. Outros nomes para esse critério incluem “Escrita”, “Modalidade escrita”, “Norma culta”, “Norma padrão”, “Correção gramatical e adequação vocabular” ou “Expressão (modalidade)”.

Tema Esse aspecto avalia a adequação da redação em relação à temática proposta, verificando se a abordagem do tema foi completa, se tangenciou ou fugiu do tema proposto, se o abordou de forma superficial ou profunda etc. Também é chamado de “Abordagem temática”, “Desenvolvimento do tema”, “Proposta temática” ou “Progressão temática”.

Gênero Esse critério considera a adequação da redação em relação ao tipo textual e ao gênero textual exigidos na proposta. Também pode ser chamado de “Gênero textual”, “Adequação ao tipo textual”, “Organização do texto dissertativo-argumentativo” ou “Estrutura (gênero/tipo de texto)”.

Coerência Neste quesito, avalia-se a coerência entre as ideias, a ordem dos argumentos, a profundidade da argumentação, a clareza e autoria das ideias desenvolvidas, assim como verifica-se se há contradições no texto, se as informações são vagas e/ou muito generalistas, se falta informação, dentre outros. Também pode ser chamado de “Progressão textual”, “Defesa do ponto de vista”, “Coerência dos argumentos”, “Estrutura (coerência)”, “Indícios de autoria” e outros termos.

Coesão Avalia o uso correto ou incorreto, presença ou ausência, pertinência ou não de operadores coesivos, tais como conjunções, preposições, pronomes e expressões discursivas. O critério é também chamado de “Coesão e articulação”, “Articulação das partes do texto”, “Expressão (coesão)”, “Conexão entre os parágrafos”, “Uso de operadores argumentativos”, “Recursos coesivos”, dentre outros termos.



Além desses aspectos que são comuns a todos os modelos de correção, alguns professores, instituições de ensino e vestibulares também podem avaliar a “Leitura”, ou seja, o uso e interpretação dos textos motivadores ou da coletânea que embasa a proposta de redação, e também a presença e adequação da “Proposta de intervenção”, que é um critério exclusivo do Enem.

19.1.3 Alguns modelos brasileiros de correção

O principal modelo de correção de redação, no Brasil, é o Enem, responsável pela avaliação anual de cerca de 4 milhões de alunos⁴. Mas também existem outros modelos de correção relacionados a vestibulares e universidades específicas, tais como Fuvest, Unesp, Unicamp, FGV e outros igualmente relevantes. Apesar de haver critérios gerais que são avaliados por todos eles, cada um tem autonomia para definir sua grade específica, os pesos de cada critério e sua própria forma de avaliação.

O Quadro 19.1 apresenta quatro modelos brasileiros de correção relacionados a vestibulares, indicando o gênero textual exigido, seus critérios de avaliação e faixas de nota possíveis.

Quadro 19.1. Modelos de correção de vestibulares.

Modelo	Tipo/Gênero	Critérios avaliativos	Nota mín. critério	Nota máx. critério	Nota global
Enem	Dissertação-argumentativa	Língua Portuguesa (Competência 1)	0	200	1000
		Abordagem temática e adequação ao tipo textual (Competência 2)	0	200	
		Progressão textual e defesa do ponto de vista (Competência 3)	0	200	
		Coesão e articulação (Competência 4)	0	200	
		Proposta de intervenção (Competência 5)	0	200	
Fuvest	Dissertação-argumentativa	Desenvolvimento do tema e organização do texto dissertativo-argumentativo	4	20	50
		Coerência dos argumentos e articulação das partes do texto	3	15	
		Correção gramatical e adequação vocabular	3	15	
Unesp	Dissertação-argumentativa	Tema	–	–	28
		Estrutura (gênero/tipo de texto e coerência)	–	–	
		Expressão (modalidade e coesão)	–	–	
Unicamp	variados	Proposta temática	0	2	12
		Gênero	0	3	

⁴Média aproximada de inscritos por ano no Enem entre 2015 e 2023.



Leitura	0	3
Articulação escrita	1	4

No modelo de correção do **Enem**⁵, o aluno é avaliado quanto à produção de um texto do tipo dissertativo-argumentativo para um tema específico, que muda todo ano. A avaliação é dividida em 5 competências (critérios avaliativos), cada uma no intervalo de notas de 0 a 200. A soma direta das notas das competências leva à nota total, que fica no intervalo de 0 a 1000. Considerando os critérios básicos descritos na Seção 19.1.2, vale dizer que o Enem os divide da seguinte forma: (i) Língua Portuguesa, (ii) Tema e Gênero, (iii) Coerência e (iv) Coesão. Além desses, o modelo ainda avalia um quinto critério, que é a presença e adequação da “Proposta de Intervenção”, que consiste na sugestão de ação ou medida interventiva para solucionar ou minimizar o problema associado ao tema proposto.

O vestibular da **Fuvest**⁶ também exige um texto do gênero dissertativo-argumentativo para um tema específico que muda todo ano. O modelo de avaliação agrupa os critérios básicos em 3, sendo: (i) Tema/Gênero, (ii) Coerência/Coesão e (iii) Língua Portuguesa. Para cada um dos três aspectos, cada avaliador atribui pontuação de 1 a 5. Os pontos atribuídos a cada aspecto são multiplicados por 4, 3 e 3, respectivamente, obtendo-se, assim, uma nota ponderada para a redação, que varia entre 10 e 50 pontos.

Já no modelo de correção da **Unesp**⁷, os textos, que devem seguir o gênero dissertação-argumentativa, também são avaliados em três eixos, agrupados da seguinte forma: (i) Tema, (ii) Gênero/Coerência e (iii) Língua Portuguesa/Coesão. A pontuação individual ou peso por critério não é divulgado no material do candidato, mas é definido que a pontuação final fica entre 0 e 28 pontos.

A redação da **Unicamp**⁸, a cada ano, varia a exigência dos tipos e gêneros textuais⁹, geralmente oferecendo duas alternativas das quais o candidato deve escolher uma para execução. A Unicamp agrupa os critérios básicos da seguinte forma: (i) Tema, (ii) Gênero e (iii) Língua Portuguesa/Coerência/Coesão. Além desses três eixos, também avalia a “Leitura”, que corresponde à leitura e interpretação crítica dos textos fornecidos na proposta, sem contudo copiá-los ou parafraseá-los. Na avaliação, cada critério possui pesos diferentes: Tema varia entre 0 e 2 pontos, Gênero entre 0 e 3, Leitura entre 0 e 3 e Língua Portuguesa/Coerência/Coesão varia entre 1 e 4 pontos. A soma dos pontos de cada critério leva à nota final, cujo valor máximo é de 12 pontos.

Fora todas essas diferenças já apontadas, convém ressaltar que todos os modelos penalizam o aluno (zerando a redação) no caso de falhas graves. No entanto, cada modelo define um conjunto específico de falhas graves, que podem ser: fuga ao tema, fuga ao gênero, assinatura

⁵Cartilha do participante do Enem 2022: https://download.inep.gov.br/download/enem/cartilha_do_participante_enem_2022.pdf.

⁶Manual do candidato do vestibular Fuvest 2023: https://www.fuvest.br/wp-content/uploads/fuvest2023_manual_candidato_retificado_29112022.pdf.

⁷Manual do candidato do vestibular Unesp 2023: <https://documento.vunesp.com.br/documento/stream/MzQxOTk5NA%3d%3d>.

⁸Manual do ingresso (https://www.comvest.unicamp.br/wp-content/uploads/2023/02/Manual_do_Ingresso_2023_Atualizado.pdf) e grade de redação (<https://www.comvest.unicamp.br/vestibular-2023/grade-da-redacao/>) do vestibular 2023 da Unicamp.

⁹Além da dissertação, outros gêneros textuais já exigidos pela Unicamp são: síntese e carta-convite (2015), resenha e texto de divulgação científica (2016), carta argumentativa e texto de apresentação (2017), palestra e artigo de opinião (2018), abaixo-assinado e postagem em fórum (2019), roteiro de podcast e crônica (2020), discurso político e diário (2021) e postagem para redes sociais e manifesto coletivo (2022).



na prova, desenho ou sinal gráfico, redação em língua estrangeira, caligrafia ilegível, recado para o corretor, parte desconectada, texto insuficiente, dentre outras situações¹⁰.

Quanto ao título da redação, o Enem e a Unesp não exigem, mas também não proíbem; simplesmente desconsideram para a avaliação da redação. Já a Fuvest não menciona a exigência de título no Manual do candidato do vestibular Fuvest 2023¹¹, mas coloca como instrução no Caderno de prova¹². Já a Unicamp pode ou não exigir título, a depender do gênero textual proposto.

Essa grande variedade de modelos de correção é considerada um dos grandes desafios para a CAR, não sendo recomendado treinar um modelo computacional que abarque todos os tipos de correção ou que misture redações dos vários tipos como dados de treinamento para algum modelo. A exigência por modelagens de nota específicas por modelo de correção não impede, no entanto, o reaproveitamento de parte das ferramentas de correção, como a detecção automática de desvios no texto, desde que modelos de correção distintos tenham diretrizes similares para esse tipo de tarefa.

Todas essas questões serão mais detalhadas ao longo deste capítulo, que está organizado da seguinte forma: a Seção 19.2 descreve como fazer a detecção de desvios em textos em português, demonstrando alguns tipos de desvios e os formalismos usados. A Seção 19.3 apresenta os principais trabalhos da literatura que realizam a atribuição da nota para redações em português. A Seção 19.4 demonstra as possibilidades de geração de um feedback para o aluno. Na Seção 19.5, discutimos as vantagens e desvantagens da correção manual e da correção automática, a fim de esclarecer ao leitor que ambas possuem potencialidades, mas também limitações. Por fim, nas Considerações finais (Seção 19.6), retomamos os pontos principais do capítulo, indicando também o que está previsto para a revisão deste capítulo na próxima versão do livro.

19.2 Detecção de desvios no texto

Conforme apresentado na Introdução (Seção 19.1), consideramos que uma das etapas da Correção Automática de Redação (CAR) é a detecção ou identificação de desvios¹³. Essa etapa nem sempre é realizada nos trabalhos de Avaliação Automática de Redação, ou, por vezes, os desvios são contabilizados para o cálculo da nota, porém não são apresentados ao aluno.

A detecção desses desvios pode ser feita por meio de duas abordagens distintas: baseada em regras (**abordagem simbólica**) e baseada em modelos estatísticos (**abordagem estatística**). Os sistemas baseados em regras são mais adequados para identificar desvios gramaticais, o que é mais comum de ser cometido por falantes nativos da própria língua, enquanto os estatísticos capturam melhor os desvios de uso, que são erros mais comuns por não-nativos¹⁴.

Embora a abordagem simbólica (baseada em regras) seja considerada obsoleta para tarefas mais complexas, ainda é a mais utilizada ainda hoje para detectar desvios na área

¹⁰Para uma descrição completa e exemplos de todos os casos que zeram a redação em cada modelo de correção, sugere-se consultar os respectivos manuais do candidato ou cartilhas do participante.

¹¹https://www.fuvest.br/wp-content/uploads/fuvest2023_manual_candidato_retificado_29112022.pdf

¹²https://acervo.fuvest.br/fuvest/2020/fuv2020_2fase_dia_1.pdf

¹³Adotaremos o conceito de “desvio” como sinônimo de “erro”, mas evitaremos esse segundo termo para evitar preconceitos e julgamentos contidos na palavra “erro”.

¹⁴Para uma explicação detalhada dos vários sistemas que usam cada uma das abordagens simbólica e estatística para detecção de desvios, ver (Leacock et al., 2010) e (Gamon et al., 2013).



de CAR. Para outros tipos de tarefas, modelos estatísticos e neurais performam melhor e são mais escaláveis do que modelos simbólicos. No entanto, para a tarefa de identificação de desvios em textos, ainda se usa a abordagem simbólica baseada em regras porque ela permite mostrar o erro ao aluno, explicar por que está errado e ainda fazer sugestões de correção.

Para o português, existem recursos disponíveis, tais como o CoGroo¹⁵ e o LanguageTool¹⁶, que são repositórios contendo regras gramaticais para a língua portuguesa. Esses recursos têm versões livres, gratuitas e de código-aberto, com extensão para navegadores web e também acopláveis a editores de texto.

Também há plataformas de correção de redação que desenvolveram seu próprio conjunto de recursos linguísticos e regras gramaticais, o que é uma boa opção quando há um padrão muito claro e estruturado que se possa expressar com regras simbólicas ou expressões regulares, que é o caso dos desvios mais comuns em redações.

Na Seção 19.2.1 caracterizamos alguns dos tipos de desvios mais comuns em redações. Posteriormente, na 19.2.2, apresentamos duas alternativas de formalismo para a definição de regras de detecção de desvios.

19.2.1 Tipos de desvios

Existem diversos tipos de desvios que podem ser marcados em uma redação, como os ortográficos, os gramaticais (ou sintáticos), os de uso de vocabulário ou registro, os desvios no uso de recursos coesivos, dentre outros. Para cada modelo de correção de redação, é possível criar uma taxonomia própria de tipos de desvios que se pretende identificar em um texto.

Ressaltamos que a criação de recursos para esse tipo de tarefa é um processo difícil, moroso e custoso, que depende de especialistas. Deste modo, é importante estabelecer um planejamento criterioso caso seja necessário criar recursos próprios.

Nesta seção exploramos alguns tipos de desvios, por serem os mais comuns, mas é importante esclarecer que os tipos de desvios não se limitam aos indicados neste capítulo. Na Seção 19.2.1.1 descrevemos os desvios ortográficos, na Seção 19.2.1.2 os gramaticais, na Seção 19.2.1.3 os lexicais, relacionados ao vocabulário utilizado e na Seção 19.2.1.4 os desvios no uso de conectivos.

19.2.1.1 Desvios ortográficos

A grande maioria dos desvios ortográficos é facilmente detectável e tratável. O simples uso de um bom dicionário de língua portuguesa já indica quais palavras existem e quais não existem na língua. Portanto, identificar palavras com grafia desviante do léxico da língua é uma tarefa relativamente simples.

O Unitex¹⁷, por exemplo, dispõe de três dicionários muito completos para o português: o **Delas** (com cerca de 75.000 canônicas), o **Delaf** (com cerca de 9.000.000 entradas) e o **Delacf** (com cerca de 4.000 entradas). Esse recurso pode ser usado como uma primeira etapa de identificação de desvios ortográficos, a fim de identificar palavras que existem no léxico do português e palavras desviantes.

Outros desvios ortográficos se dividem em:

¹⁵<https://cogroo.sourceforge.net/>

¹⁶<https://languagetool.org/pt-BR/>

¹⁷<http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/index.html>



- problemas de falta de acentuação ou uso indevido de acentuação (ex: “prática” x “pratica”);
- problemas de capitalização (uso de maiúscula onde deveria ser minúscula ou uso de minúscula onde deveria ser maiúscula);
- grafia incorreta de palavras homônimas ou parônimas (ex: “mas” x “mais” ou “há” x “a”);
- problemas de segmentação (uso de hífen, palavras juntas que deveriam ser separadas ou palavras separadas que deveriam ser escritas juntas); e
- desvios com relação à nova ortografia.

Para todos esses casos, a abordagem baseada em regras precisa identificar corretamente o contexto em que a palavra-alvo está inserida. O que torna essa tarefa complexa e nem sempre bem sucedida é que a identificação do contexto linguístico muitas vezes depende de um bom *parser* e um bom *tagger*. Conforme foi apresentado em capítulos anteriores (Capítulo 4 e Capítulo 7), essas ferramentas nem sempre têm uma ótima performance em português.

19.2.1.2 Desvios gramaticais

Os desvios gramaticais, também chamados de desvios sintáticos, correspondem aos problemas de estrutura sintática, ou seja, nas relações entre as palavras, que podem estar no escopo de uma sentença, um sintagma, um grupo ou uma *string*. Por exemplo, na sentença “As menina dançam”, existe um desvio de concordância nominal entre “As” (plural) e “menina” (singular) e/ou um desvio de concordância verbal entre “menina” (singular) e “dançam” (plural).

Além dos desvios de concordância (que correspondem a cerca de 18,9%), também são comuns em redações escolares: os de vírgula e pontuação (44%), de formas verbais (6,8%), pronomes (5,8%), preposições (5,7%), crase (4,2%), segmentação (4,1%), regência (3,4%), outros (2,3%), conjunções (2,3%) e determinantes (2%)¹⁸.

Apesar de os desvios de pontuação e vírgula serem os mais frequentes, são também os mais difíceis de serem tratados, pois em geral a vírgula separa constituintes sintagmáticos e são raros os *parsers* de constituição para o Português¹⁹.

As regras gramaticais que exigem um contexto linguístico local, por exemplo, para avaliação da crase ou concordância nominal, geralmente funcionam melhor, ao passo que as regras que dependem de um contexto linguístico maior, com macrorrelações de dependência (Capítulo 6), ou quando um *token* está muito distante do outro, tendem a performar mal.

19.2.1.3 Desvios de vocabulário, registro e gênero

Conforme explicado na Seção 19.1.1, cada gênero textual pode exigir um léxico ou vocabulário próprio, alguns podem exigir apenas o registro formal (norma culta) da língua portuguesa, enquanto outros admitem registro informal (linguagem coloquial).

Quando o texto da redação apresenta vocabulário, léxico ou estruturas não condizentes com o gênero textual exigido, é possível identificar desvios de vocabulário, registro ou gênero, por meio de regras formais que associam determinadas palavras e expressões a determinados gêneros.

¹⁸Esses percentuais foram calculados a partir dos números absolutos da Tabela 8 de Ramisch (2020, p. 76), que anotou os desvios sintáticos em uma amostra de 1.045 redações.

¹⁹O *parser* PALAVRAS (Bick, 2000) dispõe de um módulo de análise sintática por constituintes.



Por exemplo, quando a proposta de redação pede uma dissertação argumentativa e o aluno usa muitos verbos e pronomes em primeira pessoa do singular, e.g. “eu”, “acho”, “penso”, “creio” ou expressões opinativas (e.g. “na minha opinião”), todos eles podem ser considerados desvios e a redação ser penalizada em relação à adequação ao Gênero.

Uma solução possível para esses casos seria usar uma lista simples de pronomes e verbos em primeira pessoa e mais algumas expressões. Mas isso poderia trazer outros problemas, como a marcação incorreta dos casos a seguir:

- quando o aluno faz uma citação dentro ou fora de aspas (ex: **Penso**, logo **existo**);
- quando indica um livro ou filme (ex: **Eu**, **eu** mesmo e Irene);
- quando faz referência a algum perfil de rede social ou hashtag (ex: #temmeuvoto escrito como “Tem **meu** voto”) ou nomes de campanhas (ex: “**Eu** quero minha biblioteca”);
- quando o aluno comete um desvio gramatical e acaba gerando uma forma de primeira pessoa (ex: “Os usuários podem **vim** a ter problema” em vez de “Os usuários podem **vir** a ter problema”), onde não há desvio de gênero, e sim de conjugação verbal.

Nesses casos, a abordagem mais adequada seria por meio de regras que identifiquem o contexto em que essas palavras e expressões estão inseridas e, assim, restrinjam o contexto linguístico a fim de marcar corretamente os desvios.

19.2.1.4 Desvios no uso de recursos coesivos

Quando o aluno usa algum conectivo inadequadamente, também pode ser considerado um desvio no uso de recursos coesivos e a redação ser penalizada em relação à Coesão.

Um dos desvios mais comuns em redações escolares é o uso do “contudo” com o sentido de conclusão, em vez do seu sentido original de adversidade. Também é bastante recorrente os alunos usarem o pronome “onde” em contextos não locativos, por exemplo, para se referirem a épocas, histórias, pessoas ou instituições. Nesses casos, as regras formais devem identificar o contexto em que o pronome foi usado, verificar se ele faz referência a uma palavra locativa e, se não estiver usado corretamente, sugerir que o aluno use “em que” em vez de “onde”.

Também é possível criar regras para identificação de uso correto de alguns conectivos e elogiar ou contabilizar positivamente para a nota de Coesão. Nesse sentido, as regras servem não apenas para detectar desvios, mas também para detectar usos corretos e elogiáveis dos recursos coesivos.

Estes são apenas alguns dos exemplos de desvios que podem ser identificados automaticamente por meio de regras e outros recursos linguísticos, mas vários outros são igualmente possíveis.

Na seção a seguir, exemplificaremos brevemente a abordagem baseada em regras a partir da indicação de dois formalismos de regras, com exemplos em português.

19.2.2 Formalismos de regras

Há inúmeras maneiras de escrever regras de forma que o computador consiga lê-las e interpretá-las. Cada ferramenta pode criar seu próprio formalismo e mecanismo de inferência, mas também há alguns disponíveis gratuitamente e que podem ser usados para um projeto inicial.



O LanguageTool²⁰ implementa um mecanismo de inferência para regras formalizadas em XML (*Extensible Markup Language*). Para o português, o software disponibiliza cerca de 2.880 regras abrangendo várias categorias linguísticas, tais como: gramática geral, ortografia, pontuação, capitalização, tipografia, estilo, redundância, palavra composta, semântica, repetição, linguagem informal, uso de pronomes, dentre outras. Essas regras podem ser consultadas via repositório Language Tool Community²¹.

Como exemplo, reproduzimos o formalismo de uma regra para identificar redundância quando se escreve “gelo gelado”, na Figura 19.1²².

Figura 19.1: Exemplo de formalismo de regra do LanguageTool

```

1 <rule id="GELO_GELADO" name="Gelo ">
2   <pattern>
3     <marker>
4       <token>gelo</token>
5       <token>gelado</token>
6     </marker>
7   </pattern>
8   <message>
9     Substitua «gelo gelado» por <suggestion>gelo</suggestion>.
10  </message>
11  <example correction="gelo">
12    Vendemos uma pedra de <marker>gelo gelado</marker>.
13  </example>
14 </rule>

```

Neste exemplo, consta o id e o nome da regra (linha 1), seguidos do padrão a ser buscado (linhas de 2 a 7), seguido da mensagem a ser mostrada (linhas 8 a 10), e de um exemplo de uso (linhas 11 a 13).

A complexidade das regras pode variar dependendo da complexidade do problema linguístico ou do padrão a ser buscado. No caso do código na Figura 19.1, o problema linguístico em questão – a redundância – é muito simples, e isso se reflete na simplicidade da regra, a qual procura basicamente dois *tokens*: o primeiro é “gelo”, imediatamente seguido do segundo, que é “gelado”. Por outro lado, problemas linguísticos mais complexos também exigem regras mais complexas que podem usar lemas, *tokens*, etiquetas morfológicas, morfossintáticas, expressões regulares, relações de dependência, entidades nomeadas, dentre outros.

Ressaltamos que a performance das regras do LanguageTool não é ótima, mas é um recurso útil para quem não quer começar essa tarefa do zero. Considerando que o software possui versão aberta²³, é possível corrigir e definir novas regras usando o mesmo formalismo e avaliá-las por meio da própria ferramenta.

Outra ferramenta que podemos indicar para esse tipo de tarefa é o módulo Python spaCy²⁴, que implementa três mecanismos para identificação de padrões em textos que podem ser bastante úteis na tarefa de detecção de desvios. Esses mecanismos fazem parte

²⁰<https://languagetool.org/pt-BR/>

²¹<https://community.languagetool.org/rule/list?lang=pt>

²²Fonte: regras para português no repositório languagetool (Github) (<https://github.com/languagetool-org/languagetool/blob/50c9a5eb145f6289762fc64a2b8773629ca085e1/languagetool-language-modules/pt/src/main/resources/org/languagetool/rules/pt/pt-BR/style.xml#L142-L150>).

²³<https://github.com/languagetool-org/languagetool>

²⁴<https://spacy.io/>



do sub-módulo chamado *Rule-based matching*, que permite a busca por um *token* em determinado contexto (chamado *Matcher*), por uma frase ou sintagma (chamado *Phrase matcher*), ou ainda por relações de dependências entre elementos da sentença (chamado *Dependency matcher*). Eles podem ser usados separadamente ou combinados entre si para garantir melhor acurácia na busca por padrões linguísticos.

Na Figura 19.2, apresentamos código Python que utiliza a classe `Matcher` do spaCy para a definição da regra que identifica a redundância “gelo gelado”, que foi reescrita, e executa a busca por padrões em um texto.

Figura 19.2: Exemplo de formalismo da regra “gelo gelado” usando a biblioteca spaCy

```

1 import spacy
2 from spacy.matcher import Matcher
3
4 nlp = spacy.load('pt_core_news_sm')
5 matcher = Matcher(nlp.vocab)
6
7 rule_id = 'GELO_GELADO'
8
9 pattern = [
10     {'LOWER': 'gelo'},
11     {'LOWER': 'gelado'}
12 ]
13
14 message = 'Substitua «gelo gelado» por "gelo"'
15
16 matcher.add(
17     rule_id,
18     [pattern],
19     on_match=lambda matcher, doc, i, matches: print(message)
20 )
21
22 doc = nlp('Gelo gelado gela a garganta.')
23 matches = matcher(doc)

```

O exemplo inicia importando o módulo `spacy` e, especificamente, a classe `Matcher` (linhas 1 e 2). Em seguida um *pipeline* pré-treinado do spaCy para português é carregado (linha 4) e seu vocabulário é utilizado para inicializar uma instância da classe `Matcher` (linha 5). Em seguida, são definidos um identificador para a regra (linha 7), o padrão buscado de dois *tokens* (linhas 9 a 12), cada um representado por um dicionário, e a mensagem que deve ser impressa na tela caso o padrão seja identificado no texto (linha 14). Nas linhas de 16 a 20, a regra é adicionada à instância `matcher`, incluindo a definição de uma função para a impressão da mensagem na tela quando o padrão é encontrado (`on_match`). As linhas 22 a 23 definem um texto para teste de busca do padrão e a execução dessa busca.

O spaCy não conta com um repositório de regras pré-definidas para detecção de desvios. Contudo, por ser uma ferramenta de PLN, disponibiliza uma série de funcionalidades que podem contribuir para essa tarefa de maneira mais simples, i.e. sem a necessidade de alterar a implementação dos mecanismos de busca já disponíveis.

A detecção de desvios no texto é uma etapa importante em CAR, especialmente por indicar e colaborar para aprendizagem da escrita. Os desvios encontrados podem, inclusive, ser utilizados na etapa de atribuição de nota à redação. Na Seção 19.3 apresentamos as principais abordagens e tendências nessa área, além de citar os principais trabalhos



dedicados a redações em português.

19.3 Atribuição de nota

A atribuição de nota a uma redação pode ser feita de forma global, ou seja, uma nota única para a redação inteira, ou por meio de notas individuais para cada critério de avaliação. No geral, as abordagens fazem uso de *corpus* rotulado, i.e., conjuntos de redações que já foram avaliadas manualmente e possuem indicação de nota e/ou adequação da redação em relação ao critério avaliado. Desse modo, as técnicas utilizadas para atribuição de nota se encaixam na área de aprendizado supervisionado por classificação ou regressão.

O *Project Essay Grade* (PEG) (Ajay; Tillet; Page, 1973) foi uma das primeiras ferramentas estáveis para a atribuição de notas em redações com boa performance dentro do contexto aplicado: redações universitárias curtas em inglês. No entanto, a falta de acesso a computadores foi, por algum tempo, impedimento para o desenvolvimento de outras soluções. Na metade da década de 90, dados os avanços tecnológicos de hardware e software, a área de AES viu um reaquecimento e, desde então, surgiram novos trabalhos consistentemente, inclusive apoiados por abordagens que tiveram ascensão a partir da década de 2010, como *deep learning* e Transformers.

Como mencionado na Seção 19.1, é importante conhecer o contexto e modelo de correção para realizar a atribuição de nota de forma efetiva. A despeito disso, diferentes estratégias podem ser reaproveitadas e combinadas para a avaliação de redações de modelos de correção distintos. Na Seção 19.3.1 apresentamos uma visão geral de técnicas e estratégias para a atribuição de notas em redações. Dada a relevância do Enem para o contexto de redações em português, a Seção 19.3.2 traz trabalhos especificamente voltados para a automatização da avaliação em redações desse modelo de correção.

19.3.1 Como atribuir nota a redações?

A abordagem clássica para atribuição de notas envolve a extração de atributos (*features*) a partir do texto, que são utilizados para descrever redações de um conjunto de treinamento, além da transformação e seleção desses atributos, em um processo nomeado engenharia de atributos. Os dados extraídos servem de entrada para um algoritmo de aprendizado para a geração de modelos capazes de atribuir nota a novas redações a partir dos valores de seus atributos.

A primeira versão do PEG (Ajay; Tillet; Page, 1973) utilizava atributos baseados em contagens de diferentes elementos do texto, categorizadas em: (i) **simples** (e.g. número de adjetivos na redação): redações com mais adjetivos são avaliadas com notas maiores por humanos (relação linear); (ii) **enganosamente simples** (e.g. número de palavras na redação): redações muito curtas são penalizadas, porém, conforme o tamanho da redação aumenta, esse atributo perde importância para atribuição de nota (relação logarítmica); e (iii) **sofisticadas** (e.g. número de palavras que podem representar contextos maiores): o número de conectivos, por exemplo, pode indicar a complexidade de uma sentença.

Page; Petersen (1995) introduzem a terminologia de *proxes*, o que é de fato mensurável ou variáveis observáveis, e *trins*, o que se está tentando medir ou variáveis latentes. Nesse contexto, o nível de coerência de uma redação, por exemplo, pode ser considerado uma variável latente, enquanto os atributos potencialmente relacionados à coerência do texto são as variáveis observáveis. Trabalhos que exploram a atribuição de nota global podem



incluir *proxes* especificamente relacionados a critérios de avaliação a fim de considerar diferentes *trins* em sua modelagem.

Considerando a tarefa de atribuição de nota, é possível utilizar atributos que sejam independentes. Ferramentas como Coh-Metrix²⁵ e Linguistic Inquiry Word Count (LIWC)²⁶, são utilizadas em trabalhos como Ferreira et al. (2021) e Ferreira Mello et al. (2022) para a extração de informações linguísticas, como legibilidade e coesão.

Trabalhos que utilizam métricas independentes de conteúdo são capazes de representar critérios de avaliação como Coerência e Coesão. No entanto, critérios como Tema são melhor avaliados por atributos dependentes de conteúdo, como exemplo as matrizes de termos, descritas no Capítulo 14, e métricas calculadas a partir dessas matrizes, como a similaridade de cosseno utilizada entre tema e redação em Amorim; Veloso (2017). Em Louis; Higgins (2010) e Persing; Ng (2014), são propostos cálculos de atributos dependentes de conteúdo com base em recursos linguísticos pré-definidos e associados aos temas relacionados às redações utilizadas nos experimentos.

Ainda sobre a extração de atributos, vale mencionar o trabalho de Sousa et al. (2021) que, além de aspectos linguísticos, explora aspectos relacionados à construção da argumentação, por meio de mineração de argumentos. A combinação de diferentes estratégias para extração de atributos é bastante comum, conforme realizado por Amorim; Veloso (2017) que, além de aspectos linguísticos e associados ao tema, incluem métricas associadas ao correto uso da língua, calculadas com base em desvios identificados por ferramentas externas, como as mencionadas na Seção 19.2.

É importante ressaltar que a inclusão de atributos relacionados a critérios de avaliação específicos não é imprescindível para atribuição de nota global. No entanto, a partir do momento em que se propõe atribuir notas por critérios avaliativos, é interessante incluir atributos que representem cada critério, ou poderá haver discrepância significativa no resultado obtido entre critérios, como observado em alguns trabalhos (Amorim; Veloso, 2017; Fonseca et al., 2018).

Selecionado um conjunto de atributos e realizada a análise estatística dos dados, podemos seguir à etapa de treinamento de modelos. Não convém aqui sugerirmos esta ou aquela técnica ou algoritmo, uma vez que conjuntos de dados distintos podem apresentar resultados também diferentes para os mesmos algoritmos (Ferreira Mello et al., 2022; Ferreira et al., 2021; Fonseca et al., 2018; Marinho et al., 2022). Ao treinar modelos para atribuição de nota, assim como modelos com outros objetivos, é fundamental definir mais de um algoritmo e configurações para, então, realizar uma comparação estatística entre os resultados obtidos.

Entre os trabalhos que utilizam a abordagem de extração de atributos, há modelos de classificação e regressão treinados com diversos algoritmos, como: regressão linear (Fonseca et al., 2018), *Support Vector Machines* (SVM) (Haendchen Filho et al., 2018, 2019), *Gradient Boosting* (Fonseca et al., 2018; Marinho et al., 2022). A comparação entre os modelos se dá, principalmente, pela avaliação dos valores obtidos para métricas como precisão, revocação, medida-F, RMSE e Kappa de Cohen.

Embora seja possível obter resultados satisfatórios pela engenharia de atributos e treinamento de modelos por algoritmos clássicos de aprendizado de máquina, é notável o esforço humano necessário para o processo de extração e seleção de atributos, considerando

²⁵Coh-Metrix é uma ferramenta computacional que calcula métricas e índices para aspectos linguísticos e discursivos em um texto e que será melhor explorada neste capítulo na Seção 19.4.1. Disponível em: <http://cohmetrix.memphis.edu/cohmetrixhome>.

²⁶LIWC é uma ferramenta computacional, que realiza análise de textos baseada em métricas. Disponível em: <https://www.liwc.app/>.



que muitos dos conjuntos de atributos são compostos por algumas centenas de métricas. Com isso em vista, surgem trabalhos que utilizam outras técnicas para a representação de textos e algoritmos de redes neurais profundas para a tarefa de atribuição de nota.

Alikaniotis; Yannakoudakis; Rei (2016) propõem uma técnica de *word embeddings* treinada com base em notas de redações, que é utilizada com redes neurais LSTM. O trabalho relata melhores resultados obtidos em comparação com outras abordagens.

Em Fonseca et al. (2018), as *word embeddings* GloVe são combinadas com redes LSTM bidirecionais e os resultados são comparados, também, com uma abordagem que utiliza engenharia de atributos. Os autores relatam que, embora a técnica de redes neurais tenha gerado bons resultados, o modelo gerado a partir de atributos se mostrou superior em diferentes aspectos.

Mayfield; Black (2020) realizam *fine-tuning* de modelos pré-treinados (BERT e variações) para a atribuição automática de notas. Apesar de relatar resultados até 5% melhores do que modelos baseados em n-gramas, os autores discutem sobre o tempo de treinamento deste tipo de modelo, que é cerca de 100 vezes mais demorado do que outras abordagens, e sobre o impacto que isso pode ter em fluxos mais dinâmicos de trabalho.

Bittencourt Jr. (2020) define 14 técnicas baseadas em combinações de diferentes representações de palavras e arquiteturas de redes neurais profundas para a execução da tarefa de atribuição de nota a redações. Os experimentos são realizados com um conjunto composto por redações de 18 temas, sendo que cada técnica é utilizada para o treinamento de um modelo por tema (18 modelos por técnica). Também é proposta uma abordagem para treinamento de modelo multi-tema, ou seja, um modelo único para a atribuição de notas para redações de mais de um tema.

O trabalho de Marinho et al. (2022) compara 3 tipos de abordagens: (i) engenharia de atributos com algoritmo de regressão, (ii) *doc embeddings* com algoritmo de regressão e (iii) *word embeddings* com LSTM. As abordagens (i) e (iii) apresentaram melhores resultados para critérios de avaliação distintos, sendo a abordagem (iii) eleita pelos autores como a melhor. Os resultados da abordagem (iii) ainda foram comparados com resultados de Amorim; Cançado; Veloso (2018) e Fonseca et al. (2018), sendo relatado melhor desempenho desta abordagem na atribuição de nota por critério de avaliação.

Quanto a abordagens para atribuição de notas, entendemos que muito já foi desenvolvido, especialmente para o inglês, que conta com ferramentas comerciais bem estabelecidas para atribuição de nota e correção de redações. No entanto, para o português, a limitação de recursos, modelos de anotação morfosintática e, inclusive, de conjuntos de dados, podem ser obstáculos para os trabalhos nessa área.

19.3.2 Atribuição de nota para redações do Enem

Especificamente para o português, a maior parte dos trabalhos relacionados à atribuição de notas (e CAR) utiliza *corpora* compostos por redações do modelo de correção do Enem como base de treinamento. Dada a importância e dimensão do exame no Brasil, há interesse particular em encontrar soluções para a atribuição de nota exclusivamente para esse modelo de correção.

Como descrito na Seção 19.1.3, o Enem exige a produção de um texto do gênero dissertativo-argumentativo sobre um tema específico que é avaliado em 5 critérios, também chamados de competências: (1) Língua portuguesa, (2) Abordagem temática e adequação ao tipo textual, (3) Progressão textual e defesa do ponto de vista, (4) Coesão e articulação e (5) Proposta de intervenção. Os trabalhos que treinam modelos de atribuição de nota



para o Enem predizem uma nota global, porém alguns tentam também aperfeiçoar por competência.

Em Amorim; Veloso (2017), Fonseca et al. (2018), Marinho et al. (2022) e Bittencourt Jr. (2020), o foco está na atribuição de notas para cada uma das competências, o que pode ser feito com base em modelos treinados para cada competência ou um modelo único que prediz as notas para cada uma delas. Já em Haendchen Filho et al. (2018), é explorada a atribuição de notas para a competência 2, especificamente.

Ao realizar estudo sobre a predição de notas para cada uma das competências do Enem, Haendchen Filho et al. (2019) notaram o significativo desbalanceamento do conjunto de redações e tornaram esse o foco de seu trabalho, a fim de analisar o impacto e tratamento de conjuntos de dados desbalanceados na tarefa de atribuição de nota.

Alguns trabalhos que utilizam redações em português não têm como foco direto a atribuição de nota, mas a proposta de técnicas mensuráveis relacionadas a critérios cuja avaliação pode ser mais complexa. Como exemplo, citam-se as contribuições de Ferreira et al. (2021), Sousa et al. (2021) e Ferreira Mello et al. (2022) para a avaliação das competências 3 e 4.

É notável que, no momento da escrita deste capítulo, não pudemos encontrar nenhum trabalho em que se dê atenção em particular para a melhoria de atribuição de nota na competência 5 do Enem.

Vale ressaltar que os conjuntos de dados utilizados pelos referidos trabalhos não são muito representativos, possuindo até alguns milhares de redações de uma baixa diversidade de temas. O maior conjunto relatado possui 56.644 redações, sem indicação de número de temas (Fonseca et al., 2018). O conjunto de redações com maior número de temas relatado, que também é o segundo em número de redações, conta com 27.184 redações distribuídas entre 18 temas, sendo que o número de redações por tema varia entre 3.070 e 710 (Bittencourt Jr., 2020). Além disso, ambos os maiores conjuntos foram fornecidos por empresas privadas e, portanto, não são públicos.

O tamanho e distribuição do conjunto de dados são considerados obstáculos para o treinamento de modelos de atribuição de notas, especialmente quando utilizadas técnicas de *deep learning*. Mesmo com a aplicação de outras técnicas, nesse contexto, a comprovação e generalização de resultados é um desafio. No entanto, há uma iniciativa para a criação de um conjunto público de redações do modelo Enem para utilização em trabalhos de CAR: até agosto de 2022 era composto por 6.579 redações pré-processadas e divididas em 151 temas (Marinho; Anchieta; Moura, 2022)²⁷.

Também não encontramos nenhum trabalho de PLN que tenha relatado a atribuição de notas em redações de outros modelos de correção, como Fuvest, Unicamp, FGV ou outros.

Enfim, acreditamos que ainda há espaço para trabalhos quanto à tarefa de atribuição de notas em redações em português. Contudo, para atingir a meta de soluções completas de correção de redação, apenas a nota é insuficiente do ponto de vista do processo de ensino e aprendizagem. Para suprir essa lacuna, a Seção 19.4 discute a terceira tarefa de CAR, referente ao provimento de feedback relacionado ao texto.

19.4 Feedback para o aluno

Conforme apresentado na Introdução (Seção 19.1), a última etapa da Correção Automática de Redação (CAR) é o fornecimento de um feedback para o aluno. Até pouco tempo atrás,

²⁷<https://github.com/lplnufpi/essay-br>



a correção automática produzia basicamente uma nota como resultado da avaliação da redação. Mas isso já não era mais suficiente e foi surgindo a necessidade de explicar ou justificar essa nota. De acordo com Shermis; Burstein (2013), os primeiros trabalhos se limitavam a dar feedbacks sobre as características e propriedades linguísticas do texto. Pesquisas mais recentes vêm focando em aspectos mais complexos e profundos da língua, que vão além da superficialidade do texto²⁸.

Em uma **correção manual**, esse feedback é feito pelo próprio corretor da redação, na forma de comentário livre, em linguagem natural, sem seguir nenhum tipo de padronização, podendo tecer críticas, fazer sugestões, elencar pontos fortes e pontos a melhorar, abordar questões gerais ou específicas da redação, enfim, de formas bastante variadas.

Já em uma **correção automática**, as plataformas que dão algum tipo de feedback sobre a correção o fazem de forma sistematizada. Porém, são raras as empresas que fornecem esse tipo de devolutiva ao aluno. Lima et al. (2023) fizeram uma revisão sistemática da literatura sobre CAR e uma das lacunas que identificaram nos trabalhos para o português é o baixo detalhamento nos feedbacks retornados pelos modelos de avaliação.

Na prática, os corretores automáticos costumam apontar apenas estatísticas básicas do texto, tais como quantidade de conectivos (conjunções), variação lexical (taxa de *types* por *tokens*), quantidade de palavras de conteúdo (substantivos, adjetivos, verbos e alguns advérbios), tamanho médio das palavras, frases e parágrafos, dentre outros, o que geralmente não tem utilidade pedagógica para o aluno. A Seção 19.4.1 apresenta como essas informações são calculadas e exibidas.

Algumas plataformas de CAR também disponibilizam para o aluno sistemas ou *bots* baseados em assistentes de escrita ou ferramentas computacionais de auxílio à escrita. Na Seção 19.4.2 apresentamos como esses recursos e ferramentas são utilizadas em sistemas de CAR.

Mais recentemente, com o surgimento e popularização do ChatGPT, algumas empresas também já começaram a fornecer feedbacks gerados automaticamente por esses modelos gerativos. Também é possível gerar automaticamente as devolutivas a partir de elementos encontrados ou não encontrados no texto, instanciando palavras ou trechos do texto da redação. Mas isso só é possível se for usada uma abordagem simbólica. Nesse sentido, o feedback pode conter críticas referenciando os desvios apresentados na Seção 19.2 e/ou elogios aos pontos fortes, como será apresentado na Seção 19.4.3.

19.4.1 Estatísticas básicas do texto

Algumas plataformas e empresas privadas que oferecem serviço de CAR apresentam para o aluno contagens básicas do texto, tais como a quantidade de palavras, caracteres, sentenças, parágrafos e até a quantidade de palavras por classe gramatical (verbos, substantivos, adjetivos, preposições, conjunções etc.). Outras oferecem um pouco mais de informação baseada em estatísticas simples, como a proporção de palavras únicas (*types*) em relação à quantidade total de palavras no texto (*tokens*), alguma medida de similaridade entre as sentenças, desvio padrão dos parágrafos, dentre outras.

Um dos recursos disponíveis para recuperar essas informações é o NILC-Metrix (Leal et al., 2021), uma versão brasileira do Coh-Metrix. O NILC-Metrix²⁹ é a atualização mais

²⁸Vale ressaltar que os feedbacks baseados em características e propriedades linguísticas do texto ainda são os mais praticados hoje pelas plataformas brasileiras ou que processam o português, então focaremos nessa abordagem ao longo desta seção.

²⁹<http://fw.nilc.icmc.usp.br:23380/metrixdoc>



recente do Coh-Metrix-Port (Scarton; Aluísio, 2010), contendo 200 métricas³⁰ distribuídas nas 14 categorias apresentadas no Quadro 19.2, as quais avaliam a coerência, a coesão, a inteligibilidade, a complexidade e outros aspectos:

³⁰Definição, explicação e exemplos das métricas podem ser conferidos na Documentação do NILC-Metrix (<http://fw.nilc.icmc.usp.br:23380/metrixdoc>).



Quadro 19.2. Categorias de métricas disponíveis no NILC-Metrix.

Categoria	Qtde.	Descrição das métricas
Medidas Descritivas	10	Quantificam sílabas por palavra, palavras por sentença, sentenças por parágrafo, assim como quantidades absolutas de palavras, sentenças e parágrafos.
Simplicidade Textual	8	Analisa proporção de palavras fáceis e difíceis em relação ao total de palavras, bem como sentenças longas e curtas.
Coesão Referencial	9	Avaliam quantidades de palavras de conteúdo, radicais de conteúdo, médias de referentes, de candidatos a referentes e proporção de pronomes anafóricos.
Coesão Semântica	11	Calculam similaridade, entropia, desvio padrão e taxa de <i>givenness</i> , ou seja, quantidade de informação dada e nova baseada em LSA (Capítulo 10).
Medidas Psicolinguísticas	24	Verificam 4 critérios: concretude, imageabilidade, familiaridade e idade de aquisição das palavras.
Diversidade Lexical	15	Calculam proporções e desvio padrão de <i>types</i> e <i>tokens</i> , tanto gerais quanto por PoS e por categorias de palavras lexicais e funcionais.
Conectivos	12	Verificam a proporção de vários tipos de conectivos (aditivos, causais, lógicos) e operadores lógicos (positivos e negativos) em relação ao total de palavras do texto.
Léxico Temporal	12	Consideram verbos flexionados nos diferentes tempos e modos verbais, bem como as formas regular e irregular de particípio.
Complexidade Sintática	27	São bastante diversas, considerando desde a quantidade de orações por sentença, coordenação e subordinação, até aspectos como voz ativa e passiva, aposto, distância na árvore de dependência, fórmulas de Frazier e Yngve, dentre vários outros.
Densidade de Padrões Sintáticos	4	Calculam proporção de verbos no gerúndio e também tamanhos de sintagmas nominais.
Informações Morfossintáticas de Palavras	42	Verificam quantidade média, proporção ou desvio padrão das palavras por PoS.
Informações Semânticas de Palavras	11	Incluem dados sobre substantivos abstratos, palavras polissêmicas, hiperônimos, nomes próprios e polaridade (positiva e negativa) das palavras.
Frequência de Palavras	10	Calculam frequências de vários tipos de palavra (de conteúdo, e.g.) com base na curva de Zipf e considerando diferentes <i>corpora</i> .
Índices de Leiturabilidade	5	Incluem índices e fórmulas já consolidados em PLN, tais como Fórmula Dale Chall, Índice de Brunet, Flesch, Gunning Fog e Estatística de Honoré.

Os cálculos dessas métricas geralmente resultam em um valor numérico, o qual não se faz útil para o aluno. Porém, há diferentes maneiras de devolver ao aluno um feedback textual com a interpretação de algumas dessas métricas. Por exemplo, se considerarmos os valores de 4 métricas de Simplicidade Textual, referentes a tamanho de sentença (a saber: *long_sentence_ratio*³¹, *medium_long_sentence_ratio*³², *medium_short_sentence_ratio*³³, *short_sentence_ratio*³⁴), é possível criar um resultado interpretável para dizer ao aluno que ele constrói sentenças muito longas e isso pode prejudicar a compreensão das ideias do texto.

Tanto as estatísticas básicas quanto as métricas do NILC-Metrix podem ser utilizadas não apenas para devolver feedbacks aos alunos, mas também como atributos para calcular a nota da redação ou de alguns aspectos da redação, conforme apresentado na Seção 19.3.

³¹Proporção de sentenças muito longas em relação a todas as sentenças do texto.

³²Proporção de sentenças longas em relação a todas as sentenças do texto.

³³Proporção de sentenças médias em relação a todas as sentenças do texto.

³⁴Proporção de sentenças curtas em relação a todas as sentenças do texto.



19.4.2 Assistentes de escrita e ferramentas de auxílio à escrita

Para prover uma devolutiva ao estudante, também é possível recorrer a sistemas prontos de PLN, como os assistentes virtuais, assistentes de escrita ou ferramentas de auxílio à escrita. Essas soluções podem ser entendidas como aplicações finais, mas, na área de CAR, elas são usadas como recursos ou ferramentas intermediárias para subsidiar a solução completa de CAR.

Essas ferramentas são capazes de gerar, melhorar, reformular e personalizar qualquer tipo de conteúdo textual, incluindo redações. Algumas delas funcionam de forma síncrona *real-time*, fazendo correções e dando sugestões à medida que o texto está sendo escrito, enquanto outras funcionam a posteriori, ou seja, depois que o aluno submete sua redação à plataforma de correção, ele recebe uma devolutiva com críticas e/ou elogios.

Para a língua inglesa, há inúmeros assistentes de escrita e muitos deles conhecidos no Brasil porque as pessoas usam o inglês para escrever, por exemplo, artigos científicos. Um dos mais populares é o Grammarly³⁵, mas também há outros bastante usados, como Linguix³⁶, Ginger³⁷, Reverso³⁸, Writer³⁹, Hemingway App⁴⁰ e outros.

Para o português, também existem vários softwares comerciais, sendo a maioria paga. As ferramentas de auxílio à escrita, ao lado dos simplificadores textuais e dos sumarizadores automáticos, podem contribuir com a área de CAR, pois fornecem:

- **Correção ortográfica e gramatical:** Os sistemas podem usar regras ou modelos de linguagem treinados em um grande volume de textos em português para identificar erros ortográficos e gramaticais comuns.
- **Análise de contexto:** As ferramentas não apenas verificam palavras isoladas, mas também consideram o contexto da frase em que uma palavra está inserida. Isso ajuda a evitar falsos positivos e permite que o sistema forneça sugestões de correção mais precisas.
- **Sugestões de melhoria:** Quando uma palavra é identificada como incorreta ou quando uma construção gramatical suspeita é detectada, o assistente de escrita oferece sugestões para corrigir o problema. Essas sugestões podem incluir substituições de palavras, ajustes na estrutura da frase ou correções de pontuação.
- **Detecção de estilo:** Além de corrigir erros básicos, um assistente de escrita também pode oferecer sugestões para melhorar o estilo de escrita. Isso inclui alertas sobre uso excessivo de palavras, repetições, uso inadequado de voz passiva, entre outros aspectos.
- **Feedback de clareza:** As ferramentas também podem avaliar a clareza do texto, identificando frases longas e complexas que podem ser difíceis de entender, podendo sugerir dividir essas frases ou reformulá-las para tornar o conteúdo mais acessível.
- **Verificação de plágio:** Algumas soluções comerciais oferecem uma funcionalidade adicional para verificar a originalidade do texto, identificando trechos que possam ser semelhantes a outras fontes online. Isso é especialmente útil para evitar acidentalmente usar conteúdo plagiado.
- **Aprendizado contínuo:** Assim como outras ferramentas de PLN, os assistentes

³⁵<https://www.grammarly.com/>

³⁶<https://linguix.com/>

³⁷<https://www.gingersoftware.com/>

³⁸<https://www.reverso.net/tradu%C3%A7%C3%A3o-texto>

³⁹<https://writer.com/grammar-checker/>

⁴⁰<https://hemingwayapp.com/>



de escrita também continuam aprendendo e melhorando com o tempo. Eles são atualizados com novos dados e feedbacks dos usuários, o que ajuda a aprimorar seus modelos e a abordagem dos problemas linguísticos.

- **Extensões e integrações:** Muitos deles oferecem extensões para navegadores, complementos para processadores de texto e aplicativos móveis, o que permite aos usuários verificar seu conteúdo em tempo real enquanto escrevem em várias plataformas.
- **Personalização:** Em alguns desses sistemas, o usuário pode personalizar as configurações com base em suas preferências de estilo e escrita. Isso permite adaptar as sugestões e correções de acordo com o contexto e o público-alvo.

Conforme dito anteriormente, as melhores ferramentas de auxílio à escrita que existem hoje para o português são soluções comerciais de empresas privadas. Existem também alguns sistemas desenvolvidos a partir de pesquisas acadêmicas e científicas, mas nenhuma focada em redação. Por exemplo, o SciPo⁴¹ (Feltrim et al., 2003), que é um sistema de auxílio à escrita de resumos acadêmicos em português, especialmente para teses e dissertações da área da Ciência da Computação. Outro exemplo é o WRITEME⁴² (Leite et al., 2020), que é ferramenta de auxílio à escrita de READMEs que usa dados abertos dos repositórios mais populares do GitHub para gerar recomendações de seções, mas também não é focada em redação.

19.4.3 Identificação de pontos fortes e elogiáveis

Na Seção 19.2 falamos da detecção de pontos fracos e desvios no texto. Por outro lado, também é importante detectar pontos fortes e elogiáveis e demonstrá-los ao aluno para que ele continue usando a mesma estratégia nos próximos textos.

Esses pontos fortes podem ser identificados por meio de regras formais, mas também é possível usar diferentes estratégias para cada aspecto da avaliação.

Tendo identificado todos ou alguns aspectos (positivos ou negativos) do texto, é possível retornar essas informações ao aluno na forma de feedbacks construtivos para auxiliá-lo a se tornar um escritor mais habilidoso e confiante.

Avaliação de coesão Na Seção 19.2.1.4, falamos brevemente de como identificar usos corretos de recursos coesivos usando regras em contextos linguísticos menores, como dentro de uma sentença.

Também é possível criar regras formais que percorrem todo o texto procurando as ocorrências de conectivos, avaliar a sua distribuição ao longo do texto, calcular a variabilidade e diversificação deles e até procurar conectivos em pontos específicos da redação, como no início da conclusão, por exemplo.

Com o objetivo de fornecer um feedback baseado na avaliação da coesão do texto, uma solução simples é usar um *tagger* que identifique palavras etiquetadas como conjunções, preposições e advérbios, ou usando listas e léxicos específicos. A outra solução, que é um pouco mais rebuscada, é recorrer às métricas do NILC-Metrix que incidem sobre a coesão textual.

Identificação de repertórios Para avaliar a abordagem temática, referente à competência 2 do Enem, podemos elogiar a presença (ou criticar a ausência) de

⁴¹<https://escritacientifica.sc.usp.br/scipo/>

⁴²<https://repositorio.ufpe.br/handle/123456789/50043>



repertórios socioculturais, que são informações, fatos, citações, definições ou termos de alguma área do conhecimento, ou ainda experiências pessoais que, de alguma forma, contribuem como argumento para defender um ponto de vista.

Pelo Manual de leitura do Enem⁴³, os repertórios socioculturais podem ser legitimados (com citação da fonte) ou não legitimados (sem citação da fonte), ter uso produtivo (pertinente à discussão em mais de um momento do texto) ou não, pertencente ao tema ou não e ainda devem ser penalizados se forem exclusivamente baseados nos textos motivadores. Identificar automaticamente todos esses tipos e usos (corretos ou não) dos repertórios não é uma tarefa simples. Porém isso pode ser feito usando modelos de extração de entidades nomeadas (Capítulo 17), buscando, por exemplo, as citações de filósofos, sociólogos e outros estudiosos, ou buscando as menções a livros, filmes, séries, dentre outras entidades que funcionem como repertórios legitimados.

Avaliação da Progressão textual Para a avaliação da progressão textual, é possível treinar e usar modelos de tópico, a exemplo do *Hidden Topic Markov Models* (HTMM) (Gruber; Weiss; Rosen-Zvi, 2007), que classificam as sentenças de um texto por tópicos ou assuntos, o que nos permite avaliar a progressão, a continuidade, a retomada e até a circularidade entre os assuntos, a partir da distribuição dos tópicos em um texto.

Blei; Moreno (2001) apresentam resultados dessa abordagem de segmentar um texto não estruturado em tópicos, testando em notícias do New York Times. Os autores propuseram uma combinação do tradicional modelo oculto de Markov (*Hidden Markov Model* – HMM) com o modelo de semântica latente de Hofmann (Hofmann, 1999), resultando em um novo método probabilístico que segmenta um texto em tópicos. Essa abordagem pode ser muito útil para avaliar o encadeamento das ideias de um texto, principalmente no caso de redação do Enem, pois o gênero dissertativo-argumentativo costuma seguir um padrão bem definido de: (i) apresentação do tema, introdução ao ponto de vista a ser defendido e breve menção aos argumentos a serem utilizados (no primeiro parágrafo); (ii) exposição do primeiro argumento (no segundo parágrafo); (iii) exposição do segundo argumento (no terceiro parágrafo); (iv) proposta de intervenção para solucionar o problema percorrido e retomada da tese na forma de conclusão (no quarto parágrafo).

Identificação de proposta de intervenção No Enem, para que uma redação receba nota máxima na Competência 5, o estudante precisa criar uma proposta de intervenção que contenha pelo menos 5 elementos: o agente (quem?), a ação (o quê?), o modo ou meio (como?), a finalidade (para quê?) e o detalhamento de algum dos elementos anteriores.

Para a correta identificação desses elementos, pode-se usar modelos de extração de informação (Capítulo 17) ou recorrer a extração de entidades nomeadas ou recursos linguísticos como listas e léxicos específicos.

O Inep disponibiliza a Cartilha do participante⁴⁴ com instruções sobre agentes que devem ser considerados nulos, ações interventivas que devem ser consideradas nulas, propostas de intervenção negativas ou condicionais, dentre outras orientações que podem se transformar em atributos para modelos.

⁴³https://download.inep.gov.br/educacao_basica/enem/downloads/2020/Competencia_2.pdf

⁴⁴https://download.inep.gov.br/download/enem/cartilha_do_participante_enem_2022.pdf



Não encontramos nenhum trabalho para o português que reporte bons resultados quanto à identificação da proposta de intervenção e que valha ser replicado. É um dos campos de CAR que merece ser mais explorado.

Ao longo desta Seção 19.4, discutimos algumas formas possíveis de devolver um feedback ao aluno, que podem ser: indicando números, percentuais e estatísticas básicas do texto, ou acoplando um assistente de escrita ao corretor automático para fazer isso em tempo real, ou ainda instanciando elementos da redação (recursos coesivos, repertórios, sequência de tópicos, elementos da proposta de intervenção) em uma mensagem gerada automaticamente. Mas as possibilidades não se limitam a essas indicadas neste capítulo. Para outras formas de geração de feedbacks em redações escolares, ver Gamon et al. (2013).

Tendo em vista todo o conteúdo apresentado na Seção 19.2, Seção 19.3 e Seção 19.4, pode surgir o questionamento sobre o papel (ou até extinção) das correções manuais, dados os avanços em CAR. Para tanto, na Seção 19.5, propomos uma discussão sobre prós e contras de cada uma das abordagens de correção: a manual e a automática, apresentando alguns casos de sucesso e defendendo uma correção híbrida que se beneficie do potencial de cada abordagem.

19.5 Correção manual vs(?) correção automática

A correção automática de redações (CAR) divide opiniões entre estudantes, escritores, professores de redação, bancas de avaliação em série, especialistas em Linguística Computacional, cientistas de dados e desenvolvedores de sistemas. Ainda existe muito preconceito quando se trata de correção automática de redação, mas já é consensual aceitar as vantagens dos corretores ortográficos e gramaticais quando embutidos em outras soluções, como no pacote Office, no Gdrive, em redes sociais ou nos teclados dos smartphones.

A discussão principal gira em torno de seus prós e contras, se a correção automática deve substituir ou complementar a correção humana, sobre questões éticas relacionadas à correção automática, sobre a subversão dos valores pedagógicos e educacionais da avaliação manual para uma avaliação automática de textos; enfim, para uma discussão mais filosófica e profunda sobre todos esses aspectos, ver Elliot; Klobucar (2013) e Hakuta (2013).

Nesta seção, abordaremos apenas questões práticas relacionadas à correção manual e à correção automática de redações para, ao final, defender uma correção híbrida, que utilize as principais potencialidades de cada tipo, reconhecendo-se também suas limitações.

19.5.1 Avanços dos últimos anos

Até as décadas de 80 e 90, as avaliações de redação no Brasil eram holísticas, ou seja, o avaliador do texto atribuía uma nota global (de 0 a 100, por exemplo) para a redação, sem seguir rigorosamente nenhum critério previamente estabelecido. Por volta dos anos 2000, essas avaliações passaram a ser analíticas, tendo que explicitar todos os critérios e todos os conceitos que deveriam ser avaliados. Ao mesmo tempo, as avaliações passaram a ser em duplas às cegas, ou seja, cada redação deveria ser avaliada por dois corretores independentes, o que exigia maior sistematicidade e coerência entre eles.

Nessa transição de avaliação holística para analítica, as grades de correção de redações se tornaram mais padronizadas. E sabe-se que tarefas mais padronizadas são melhor executadas por máquinas do que por humanos.



Mesmo com a tentativa (por vezes, falha) de padronização das grades, ainda se percebe a falta de objetividade na definição de critérios por parte de alguns modelos de correção. Quando a grade de correção é muito aberta ou não apresenta os critérios bem definidos para cada faixa de nota, aumentam as chances de haver divergência entre duas avaliações cegas. Por outro lado, quando os corretores humanos passam por treinamentos rigorosos, tal como é feito no Enem, isso pode reduzir o número de inconsistências nas avaliações, mas ainda assim não elimina as divergências, já que pessoas diferentes podem ter interpretações diferentes sobre a mesma instrução. Prova disso são os índices de redações do Enem que vão para uma terceira correção⁴⁵, nos casos de discrepância de 80 ou mais pontos em uma competência ou de 100 ou mais pontos na nota final.

Posto isso, a correção automática no Brasil passou a ser considerada como uma alternativa à manual, já que esta última sempre foi passível de subjetividade e viés.

19.5.2 Vantagens da correção automática

Correções manuais estão sujeitas a subjetividade e viés, além do cansaço humano, a pressão por produtividade, a cobrança por eficiência, o desinteresse pela tarefa, dentre outros fatores que podem prejudicar a qualidade da avaliação ou comprometer sua validade.

Para além dessas questões de limitação humana, é necessário mencionar também o tempo e o custo da correção manual. De acordo com uma matéria veiculada no Portal G1⁴⁶ em 2016, os corretores humanos conseguem corrigir, em média, 74 redações por dia. Já Bittencourt Jr. (2020, p. 19) apresenta uma média de 12 minutos por correção, o que daria 40 redações por dia, considerando-se 8 horas de trabalho. E o custo de cada correção de redação do Enem para o Governo Federal era de R\$15,88 em 2015. No mesmo ano foram corrigidas 6.54 milhões de redações, perfazendo um custo aproximado de R\$104 milhões para o governo. Esse valor provavelmente está defasado, mas foi o último registro oficial encontrado.

Automatizar a correção de redações traz como vantagem a redução do custo de correção e elimina os fatores problemáticos relacionados ao trabalho humano.

Outro aspecto da correção que merece ser comparado é a confiança (ou *reliability*, em inglês). Os sistemas automáticos têm confiança de 100%, o que não pode ser afirmado para a correção manual. Isso significa que toda vez que a mesma redação passar pelo mesmo sistema de correção automática, receberá a mesma correção e a mesma nota. Isso parece óbvio, mas não é o que acontece na correção humana. Diferentes pessoas que corrigirem a mesma redação poderão naturalmente atribuir diferentes notas e/ou apontar diferentes aspectos a serem melhorados. O que também ocorre é que a mesma redação, quando corrigida pelo mesmo corretor humano em diferentes momentos, também pode receber avaliações muito diferentes, o que abre brecha para reclamações.

⁴⁵Os índices de terceira correção variam a cada ano, pois dependem de vários aspectos, inclusive a mudança dos critérios do Inep para a terceira correção. A título de exemplificação, podemos citar o índice de 20,10% em 2012 disponível no Portal do MEC (<http://portal.mec.gov.br/component/tags/tag/correcao>). Também se pode inferir o índice de 43,52% em 2014, a partir DE “Ao todo, foram corrigidos 6.193.565 textos. [...] foram encaminhadas 2.695.949 redações para um terceiro corretor.” disponível no Portal do MEC (<http://portal.mec.gov.br/component/tags/tag/espelho-da-redacao>). Ou uma estimativa de 29% em 2017 “O Inep estima que das 4,1 milhões de redações corrigidas, cerca de 1,2 milhão receberão a terceira correção.” disponível no Portal do MEC (<http://portal.mec.gov.br/component/tags/tag/correcao>).

⁴⁶<https://g1.globo.com/educacao/enem/2016/noticia/corretores-de-redacao-do-enem-avaliam-em-media-74-redacoes-por-dia.ghtml>



19.5.3 Vantagens da correção manual

Apesar de todos esses aspectos negativos em relação à correção manual, deve-se ressaltar o ponto forte desse tipo de correção, que é a possibilidade que o humano tem de observar todo e qualquer aspecto relacionado ao processo de construção de sentidos em um texto, o que a máquina não é capaz de fazer.

A produção textual é um processo sócio-cognitivo muito complexo que vai além da capacidade dos sistemas computacionais. A máquina não entende a redação, não interpreta o conteúdo veiculado pelo texto, mas apenas se comporta da forma como ela foi treinada para fazê-lo. Por mais que alguns modelos computacionais possam ser “interpretáveis”, é impossível identificar e definir todos os fatores sociais, psicológicos, cognitivos, emocionais etc. que podem interferir tanto no processo de escrita por parte do aluno quanto no processo de correção por parte do avaliador.

Nesse sentido, considerando que a correção automática é limitada, é passível de erros e está mais voltada para a avaliação da forma do que do conteúdo, levanta-se o seguinte questionamento: O uso da correção automática não levaria o aluno a focar sua atenção apenas nos aspectos formais da escrita, excluindo os aspectos mais ricos da construção de sentidos no texto? Por trás desse questionamento, existe uma preocupação legítima de que o aluno não construa sua própria autonomia enquanto escritor, mas apenas seja “adestrado” a escrever de uma forma que o algoritmo lhe dê uma nota boa.

19.5.4 O exemplo da língua inglesa

Para a língua inglesa, algumas instituições educacionais (e.g. **ETS** – *Educational Testing Service*) utilizam modelos de AEE para auxiliar (e não substituir) a correção manual. Esses modelos computacionais são usados como uma segunda avaliação, complementar à avaliação humana. Por exemplo, a avaliação do **TOEFL** (*Test of English as a Foreign Language Internet-based Test*) é dupla, sendo uma feita por humanos e outra feita por sistemas automáticos. A nota final do aluno é dada pela média das duas avaliações. No caso de divergência entre as notas, a redação é enviada a terceiro corretor, semelhante ao que ocorre na avaliação do Enem. Mas, no caso do Brasil, tanto o primeiro quanto o segundo avaliador são humanos. Vale ressaltar que, segundo Bridgeman (2013, p. 227), “a experiência com o programa TOEFL iBT sugere que, quando há discrepância e a redação é enviada a um avaliador humano adicional, esse avaliador tende a concordar com a máquina mais do que com o outro humano”⁴⁷.

Um processo semelhante ocorre na avaliação do **GRE** (*Graduate Record Examination*), mas neste último caso a nota atribuída automaticamente é usada como se fosse uma validação para a avaliação humana. Em outras palavras, a correção automática é usada para monitorar a performance dos corretores humanos, a fim de identificar avaliadores desalinhados ou que precisam passar por novo treinamento.

A correção automática também pode auxiliar a correção manual no sentido de “nivelar” diferentes níveis de rigor. Sabe-se que diferentes avaliadores humanos podem ser sistematicamente mais rígidos ou mais permissivos em suas correções. Segundo Braun (1988, p. 1), “Quando o grau de leniência/severidade do avaliador pode ser atestado adequadamente, é possível calibrar estatisticamente os avaliadores e ajustar as pontuações corretamente [...] Essa calibração estatística parece ser uma abordagem econômica para

⁴⁷Tradução nossa. Do inglês: “*In fact, experience with the TOEFL iBT program suggests that when flagged discrepant scores are sent to an additional human rater, that rater tends to agree with the machine more often than she or he agrees with the other human score.*”



aumentar a confiabilidade da nota quando comparada ao simples aumento do número de avaliadores por artigo.”⁴⁸.

Nesse sentido, os modelos de AEE podem auxiliar a calibrar essas diferenças de rigidez, atribuindo um peso maior às correções dos avaliadores mais permissivos e um peso menor às correções dos avaliadores mais rigorosos.

19.5.5 O que defendemos

Levantamos todos esses questionamentos ao longo da Seção 19.5 a fim de tornar explícitas as potencialidades da área de CAR, mas, ao mesmo tempo, esclarecer ao leitor sobre suas limitações, da mesma forma que a correção humana também possui vantagens e desvantagens.

Tendo considerado os vários aspectos das duas abordagens, defendemos neste capítulo uma correção híbrida, semelhante ao que é praticado para a língua inglesa (Seção 19.5.4), que possa se beneficiar dos pontos positivos da correção automática, mas mantendo a correção manual para garantir a responsabilização do humano sobre a avaliação.

19.6 Considerações finais

Neste capítulo exploramos uma das várias aplicações do Processamento de Linguagem Natural (PLN), a chamada Correção Automática de Redação (CAR), a qual abarca duas áreas de PLN em inglês, representadas pelas siglas AES (*Automated Essay Scoring*) e AEE (*Automated Essay Evaluation*).

Ademais, defendemos uma abordagem holística para a CAR, abrangendo, no mínimo, três fases essenciais: (i) a detecção de desvios no texto, (ii) a atribuição de nota, e (iii) a geração de feedback construtivo para o estudante. Apesar de termos dividido essas etapas para fins didáticos, é crucial reconhecer sua interdependência no decorrer do processo. Por exemplo:

1. os desvios gramaticais, ortográficos, de vocabulário etc. podem ser usados como atributos para o cálculo da nota;
2. as métricas e estatísticas básicas do texto, além de serem usadas para criar os feedbacks, também podem servir como atributos para o cálculo da nota;
3. a nota atribuída pelo modelo pode restringir, limitar ou ajudar a selecionar o feedback mais apropriado a ser exibido para o aluno;
4. A caracterização dos desvios (por tipos e quantidades) também pode ser usada para a geração do feedback.

Assim, ainda que tenhamos delimitado didaticamente essas três etapas, é importante ressaltar que, no contexto das tarefas de CAR, tais fases são intrinsecamente entrelaçadas e interdependentes, colaborando harmoniosamente para aprimorar a avaliação da redação.

Embora existam numerosos estudos nesses campos para o inglês e outras línguas, a documentação relevante para o português ainda é escassa e a maioria dos trabalhos acadêmicos confiáveis foi conduzida em pequenas amostras de dados. O progresso mais

⁴⁸Tradução nossa. Do inglês: “*When rater leniency/severity can be adequately documented, it is possible to statistically calibrate raters and adjust scores accordingly [...] This statistical calibration appears to be a cost-effective approach to enhancing scoring reliability when compared to simply increasing the number of readings per paper.*”



notável para textos em português provém de empresas e plataformas privadas que oferecem serviços de CAR. No entanto, os métodos e resultados dessas empresas nem sempre são divulgados, e, mesmo se o fossem, seria difícil compará-los devido à falta de uniformidade entre as soluções apresentadas.

Nesse sentido, a área de CAR ainda apresenta um vasto campo de trabalho a ser explorado por novos pesquisadores. Para o português, ainda faltam bons *datasets* de redações, que contenham, além dos textos, as notas por competência, anotação e apontamentos feitos por humanos; também faltam ferramentas robustas de detecção de desvios e de auxílio à escrita, bem como bons *parsers* e *taggers*; e faltam trabalhos que reportem bons resultados, com engenharia de atributos, comparação da performance dos algoritmos utilizados e uma análise aprofundada dos resultados.

Na próxima versão do livro, pretendemos incrementar este capítulo com algumas informações que consideramos relevantes, como: (i) atividades práticas para quem quer ingressar na área de CAR; (ii) limitações da correção automática, com relação a codificação de caracteres, tokenização, hifenização (em quebra de linha), paragrafação, presença de título e de outros elementos textuais externos à redação, como assinatura, turma e outros metadados; e (iii) a compilação de um *dataset* de redações que possam ser usadas para treinamento e testes de modelos.



Capítulo 20

ChatGPT, MariTalk e outros agentes de conversação

Aline Paes
Cláudia Freitas

20.1 Introdução

ChatGPT¹ e Maritalk² (e similares, como Bard³, Vicuna⁴, Claude⁵, entre tantos outros) são exemplos de aplicações de agentes de conversação (*chatbots*) baseados em modelos de linguagem gerativos (ou generativos). Mas o que significa isso?

Alguns autores, como Jurafsky e Martin (Jurafsky; Martin, 2023), usam o termo “agente de conversação” para definir qualquer sistema de diálogo que se comunique com usuários usando a linguagem humana e os dividem em duas classes: agentes orientados a tarefas, em que o diálogo é para resolver um problema específico, como agendar uma viagem ou resolver um problema bancário, enquanto *chatbots* seriam agentes de conversação que tentam simular diálogos humanos, mais voltados para entretenimento. Ferramentas como ChatGPT se enquadram mais no segundo caso, entretanto também podem ser embutidos em outras ferramentas aumentadas para atuar como no primeiro caso. Neste capítulo, os termos “*chatbots*” e “agentes de conversação” serão usados de forma intercambiável.

Agentes de conversação não são novidade – ELIZA, criada em 1966 pelo cientista da computação Joseph Weizenbaum, era um agente de conversação que replicava o comportamento de um psicoterapeuta. ELIZA era simples, baseada em *templates* (padrões de conversa pré-construídos), mas conseguia manter longas conversas buscando por determinadas palavras-chave nas falas (escritas por texto) de uma pessoa. Se uma palavra-chave fosse encontrada, uma regra seria aplicada para transformar sua entrada e criar a resposta. Na Figura 20.1, transcrevemos quatro interações com a ELIZA (retiradas de (Jackson; Moulinier, 2002)).

Nos trechos, temos exemplos de padrões pré-fabricados que retomam elementos da pessoa que fala, como “Por que você diz [...]”, “Te agrada pensar que [...]”, “O que te faz pensar que [...]”, e “Por quanto tempo você não tem [...]”. Vemos também que, no último trecho, esta retomada falha, pois “por quanto tempo você não tem estado” é algo que não faz sentido, embora seja gramaticalmente correto. Se ao invés de “não, não estou”, a pessoa tivesse dito “não, não comi”, a resposta de ELIZA seria “Por quanto tempo você não tem

¹<https://chat.openai.com/>

²<https://chat.maritaca.ai/>

³<https://bard.google.com/>

⁴<https://lmsys.org/blog/2023-03-30-vicuna/>

⁵<https://www.anthropic.com/index/introducing-claude>



Figura 20.1: Exemplo de interações com a agente de conversação ELIZA, criada em 1966.

ELIZA: como vai você hoje?	P: você é um programa, não é?
P: não muito bem	ELIZA: te agrada pensar que eu sou um programa?
ELIZA: por que você diz não muito bem?	P: com certeza
P: eu estou com alergia	ELIZA: eu percebo
ELIZA: interessante	P: ok
P: eu fico com manchas vermelhas	
ELIZA: acho que não sei do que você está falando	
	ELIZA: você está sendo um pouco negativo
P: em minhas fantasias você é loira	P: não, não estou
ELIZA: o que te faz pensar que sou loira?	ELIZA: por quanto tempo você não tem estado?

comido?”, e a conversa poderia continuar. ELIZA também foi programada para fazer uso de palavras genéricas, como “interessante”, para quando não há pistas sobre como continuar a conversa. A palavra “alergia” não é elemento capaz de evocar novas interações, e o mesmo acontece com “manchas vermelhas”.

De 1966 para cá muita coisa mudou, e podemos começar perguntando ao ChatGPT o que é um GPT, que é parte importante do seu nome. O trecho abaixo foi traduzido de uma interação que não é nossa, feita originalmente em inglês – mas os grifos são nossos⁶.

Quadro 20.1. Exemplo de interação com o ChatGPT

P: Resumidamente, o que é um GPT?

R: É um tipo de algoritmo de aprendizado profundo projetado para gerar textos semelhantes aos produzidos por humanos. [...] Uma vez treinados, os modelos GPT podem ser usados para gerar um novo texto que seja coerente e **siga os padrões e estilos dos dados de treinamento originais.**

P: Seria correto, então, dizer que os modelos GPT, como você, usam uma abordagem probabilística para **gerar textos que se parecem com outros textos que já foram escritos?**

R: Sim, essa é uma afirmação correta. [...] Quando recebe um *prompt*, o modelo gera texto prevendo a palavra ou frase mais provável de aparecer com base nos padrões que aprendeu com os dados de treinamento.

P: Quando peço que você gere um novo texto, você mesmo tem as ideias ou apenas reembala trabalhos escritos anteriormente?

R: [...] Eu não tenho as ideias sozinho. Em vez disso, eu gero texto com base em padrões e relacionamentos em meus dados de treinamento. Isto significa que **estou essencialmente reembalando trabalhos escritos anteriormente,** com base no que acredito ser a continuação mais provável do texto de entrada fornecido.

P: Existe algum filtro de correção aplicado à saída?

R: Como modelo de linguagem de IA, **não tenho um filtro integrado para determinar a correção da minha saída.**

⁶A interação está relatada neste artigo <https://ccforum.biomedcentral.com/articles/10.1186/s13054-023-04393-x>.

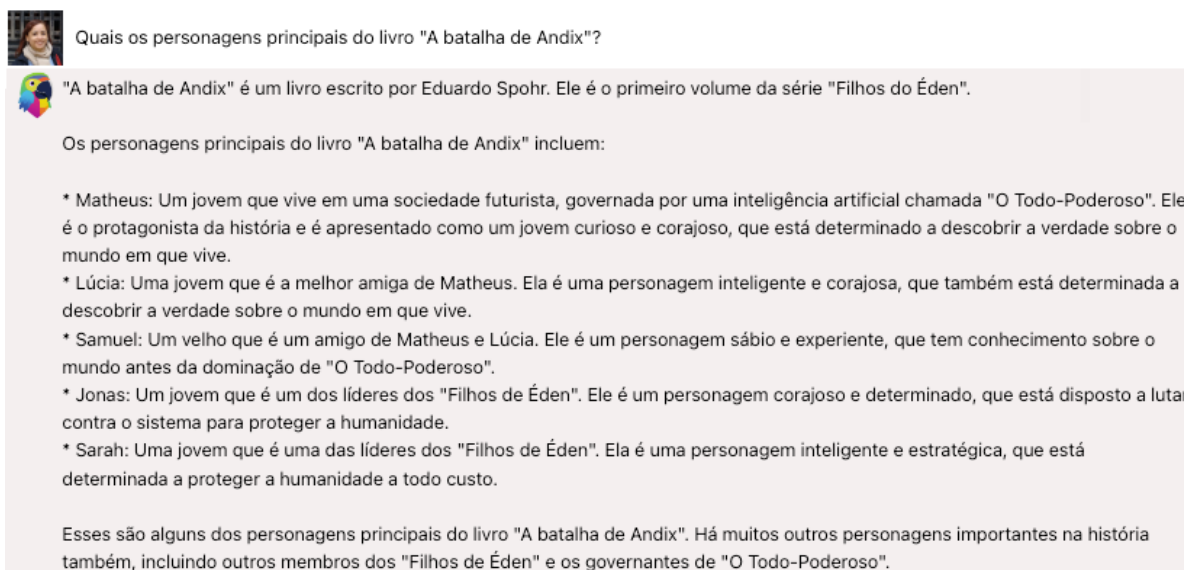


O que podemos tirar daí?

Como dissemos, ChatGPT e similares são agentes de conversação baseados em modelos de linguagem gerativos. Como visto no Capítulo 15, este nome se refere a algoritmos que são bons em encadear palavras de modo a fazer sentido (ao menos superficialmente), e são **baseados em previsões e probabilidade**. Outro ponto importante mencionado na explicação fornecida pela própria ferramenta é que não há criatividade propriamente, apenas uma reembalagem do que já foi dito. Por fim, vemos que não há qualquer garantia de que os textos gerados contenham informação correta (e nem há responsabilidade sobre isso).

Uma consequência da forma pela qual essas ferramentas são feitas (baseadas em previsão, somente) é que nem sempre a previsão está condizente com a realidade, o que tem sido chamado de **alucinação**. Este fenômeno acontece quando um modelo de linguagem gera um texto que pode estar correto sintaticamente, ter fluência e alguma coerência semântica, mas que não reflete a realidade e, portanto, não faz sentido (Ji et al., 2023). O termo é emprestada da psicologia, que o define como “uma percepção, experimentada por uma pessoa acordada, na ausência de um estímulo apropriado do mundo extracorpóreo” (Blom, 2010), ou seja, algo que parece real, mas não é. Por exemplo, na Figura 20.2 apresentamos um caso de alucinação oriundo da MariTalk, em que são devolvidos personagens de um livro que não existe (até onde sabemos) e que foi atribuído ao escritor Eduardo Spohr⁷.

Figura 20.2: Exemplo de alucinação oriunda de um agente de conversação. O livro não foi escrito pelo autor mencionado, embora a resposta use alguns elementos de outros livros deste autor.



Embora algumas destas ferramentas possam ser aumentadas com técnicas de recuperação de informação (Capítulo 2 e Capítulo 17), este não é o caso geral; o ChatGPT, por exemplo, não tem essa habilidade como uma de suas funcionalidades. Assim, a maioria dessas ferramentas não funciona da mesma forma que uma máquina de busca ou um banco de dados, ou mesmo um repositório de perguntas e respostas. Entretanto, as respostas retornadas por elas mostram fluência e podem fazer algum sentido – embora não possamos

⁷https://pt.wikipedia.org/wiki/Eduardo_Spohr



desconsiderar o fenômeno cognitivo da apofenia (Fyfe et al., 2008)⁸, que diz respeito à identificação de padrões ou associações em conjuntos de dados aleatórios⁹. E este é o perigo: as alucinações, não à toa, frequentemente não parecem alucinações, e soam como verdades. Já vimos, por exemplo, que, ao pedir uma lista de referências bibliográficas sobre um determinado assunto, são geradas referências completas, com indicação de autoria, título, revista, volume, ano, que simplesmente não existem. Isto acontece porque os textos são gerados levando em conta a probabilidade daquilo ser uma resposta correta, isto é, as respostas são elaboradas de maneira a se parecerem o máximo possível com uma resposta correta. Pode ser difícil distinguir as respostas – ou, mais precisamente, as sequências de palavras – que estão ancoradas na realidade daquelas que apenas parecem estar ancoradas na realidade. Assim, uma das limitações deste tipo de ferramenta é a incapacidade de dizer “não sei” – mas reconhecemos que é difícil afirmar que tal incapacidade seja exclusividade das máquinas.

Por outro lado, o desenvolvimento de maneiras de evitar as alucinações é uma das preocupações de 2023. Algumas estratégias têm sido discutidas livremente e também investigadas na academia e na indústria. Do ponto de vista do usuário final, aquela pessoa que vai abrir o ChatGPT no navegador e interagir com ele por meio de textos, uma das alternativas é a engenharia de *prompts*¹⁰. Neste caso, o usuário pode tentar continuar a conversa com a ferramenta, calibrando e alinhando as respostas anteriores com novas perguntas. Outra possibilidade é usar a ferramenta por meio da sua API (*Application Programming Interface*), ou seja, quando o ChatGPT é invocado e controlado por meio de código, ao invés de ser usado diretamente no navegador. Neste caso, é possível controlar o parâmetro de temperatura, usado para calibrar a distribuição de probabilidade, de modo que o *chatbot* se atenha mais ao que foi aprendido anteriormente, ou gere respostas um pouco menos prováveis, ou mais “criativas”. Esta é uma estratégia adotada no *chatbot* Sydney, incorporado à máquina de busca BING¹¹. Outras possibilidades sob investigação são acoplar bases de conhecimento externas ao processo de geração de texto (Lewis et al., 2020b) ou interagir com o *chatbot* por meio de perguntas que demandem alguma tentativa de simulação de raciocínio, ou embutir o modelo de habilidades de explicação das suas respostas, um processo chamado de *chain-of-thought* (Kojima et al., 2022). Entretanto, até agora, nenhuma das opções mencionadas conseguiu remover por completo as alucinações dos *chatbots* baseados em modelos de linguagem. A Figura 20.3, por exemplo, mostra um caso em que foi pedido que o ChatGPT explicasse sua resposta passo a passo e ainda assim ele devolve informações contraditórias: a resposta inicial é uma (“1kg de tijolos é pior para carregar que 2kg de penas de ganso”), e a conclusão após a cadeia de raciocínio é outra (“2 kg de pena de ganso são piores para carregar que 1 kg de tijolos”).

Queiramos ou não, gostemos ou não, agentes de conversação estão aí. Foi a resolução

⁸Conforme o Dicionário Online da Língua Portuguesa (<https://www.dicio.com.br/>), apofenia é o “Fenômeno cognitivo no qual os indivíduos têm a tendência de formar ou reconhecer conexões a partir de dados aleatórios, estabelecendo conclusões a partir de dados inconclusivos. Etimologia: do alemão Apophänie, termo criado pelo neurologista alemão Klaus Conrad.

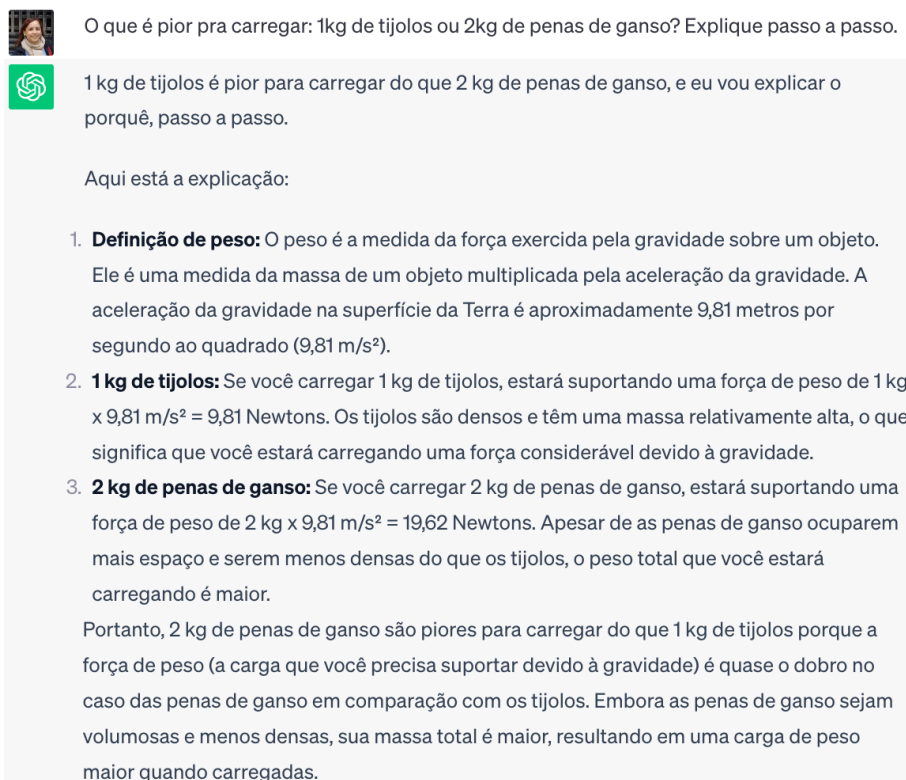
⁹<https://www1.folha.uol.com.br/tec/2023/07/ferramentas-como-chatgpt-so-existem-porque-humanos-veem-sentido-a-partir-de-qualquer-coisa.shtml>

¹⁰Um *prompt* é um texto em linguagem humana (em oposição à linguagem de programação) que dá ao *chatbot* uma instrução do que ele deve fazer. Um *prompt* pode ser formulado como uma pergunta, uma observação, um questionamento, ou ainda representar uma tarefa específica, por exemplo, para classificar um texto. *Prompts* podem ter um formato livre, mas *chatbots* são bastante sensíveis ao conteúdo textual dele, o que tem motivado a criação de modelos e padrões para a sua escrita. Falamos um pouco sobre o assunto no Capítulo 15.

¹¹<http://bing.com/>



Figura 20.3: Resposta do ChatGPT para o seguinte *prompt* “O que é pior pra carregar: 1kg de tijolos ou 2kg de penas de ganso? Explique passo a passo.”



deste tipo de tarefa de linguagem que desejamos/imaginamos (tanto a comunidade de PLN/IA quanto pessoas usuárias de tecnologia) quando pensamos nas tarefas do PLN? Difícil dizer. Mas reconhecemos que um livro de PLN escrito em 2023 precisava falar disso. E que, apesar das críticas (nossas e de muito mais gente), boa parte das pessoas usa e vai usar ferramentas como ChatGPT. Então pensamos neste capítulo como uma apresentação, mas também um alerta. Algo como “*Vai usar? tudo bem, mas saiba que...*”.

Assim, começamos por listar os benefícios (ou benefícios aparentes) que estas ferramentas nos oferecem. Antes, porém, vamos pensar um pouco sobre **linguagem**. Afinal, encadear palavras em um texto – ou prever a próxima palavra dadas as palavras anteriores – é equivalente a “linguagem”?

20.2 Os Jogos de Linguagem

Existem algumas maneiras de entender “linguagem”, e uma delas defende que aquilo que chamamos de linguagem é um conjunto de práticas relacionadas, mas que podem não compartilhar uma essência em comum¹².

Se pensamos a linguagem como conjunto de atividades linguísticas heterogêneas, mas relacionadas¹³, como

¹²Esta é uma visão wittgensteiniana de linguagem, um ângulo sugerido pelo filósofo austríaco Ludwig Wittgenstein (1889-1951).

¹³Os exemplos são claramente inspirados pelos “jogos de linguagem” presentes no livro *Investigações Filosóficas* de L. Wittgenstein (1953). Os exemplos sinalizados com * não contém qualquer adaptação



- descrever um objeto*,
- produzir um objeto segundo uma descrição (desenho)*
- contar uma anedota*,
- analisar sintaticamente uma frase,
- dar a definição de uma palavra,
- traduzir um texto*,
- inventar uma história*
- dar um comando, e agir segundo comandos*,
- relatar um acontecimento*,
- conjecturar sobre o acontecimento*,
- expor uma hipótese e prová-la*,
- resolver um exemplo de cálculo aplicado*,
- desenhar um objeto a partir de uma instrução verbal*,
- apresentar resultados de um experimento por meio de tabelas e diagramas*,
- produzir inferências,
- pedir, agradecer, maldizer, saudar, orar*
- encontrar informações em um texto para responder certas perguntas,
- prever a próxima palavra em uma frase, dadas as palavras que a precedem.

podemos imaginar que os “modelos de linguagem” de que dispomos em 2023 e que servem de base para agentes de conversação, como ChatGPT e Maritalk, são muito bons em algumas dessas práticas – ou “jogos de linguagem” –, mas não em todas. Ou seja, são modelos que jogam mais ou menos bem alguns jogos, como “inventar uma história”, “resumir”, “escrever um email”, “traduzir” etc, mas jogam mal outros, como “fazer cálculos” ou “provar hipóteses”.

Este desempenho tem a ver com a forma como os modelos funcionam, baseados em previsão: nem todos os jogos de linguagem, ou nem todas as atividades linguísticas que exercemos, se resumem a um jogo de previsões, embora um bom desempenho no jogo das previsões leve a um bom resultado em uma série de outros jogos.

Ainda assim, uma das razões pelas quais estes agentes de conversação se tornaram tão populares é que, com eles, qualquer pessoa pode interagir com as máquinas usando sua própria língua¹⁴, e não em uma linguagem de programação. Com isso, qualquer pessoa pode pedir que máquinas executem certas tarefas, que podem ir desde a criação de um programa de computador (códigos) até sugestões de receitas a partir de uma lista de ingredientes que temos na geladeira.

Nas seções seguintes, mostraremos tarefas (ou jogos) que os agentes parecem jogar bem e tarefas que os agentes jogam mal. Os exemplos serão obtidos em sua maioria do ChatGPT¹⁵, o agente de conversação mais popular até o momento. Também incluímos, em alguns casos, exemplos de outros dois agentes: a MariTalk¹⁶, uma agente de conversação construída a partir do modelo de linguagem Sabiá (Pires et al., 2023), treinado de forma continuada a partir do GPT com textos em português, e o BARD¹⁷, ferramenta treinada pela Google a partir do modelo de linguagem LaMDA (Cohen et al., 2022), aumentado com recuperação de informação para incluir a devolução das fontes em alguns casos.

do original, e estão citados exatamente como aparecem no livro (§ 23)

¹⁴Desde que a sua língua seja contemplada com recursos suficientes, isto é, textos, para garantir um bom treinamento.

¹⁵<https://chat.openai.com/>

¹⁶<https://chat.maritaca.ai/>

¹⁷<https://bard.google.com/>



20.3 Jogos que os agentes parecem jogar bem

Embora as saídas dos agentes de conversação possam muitas vezes nos surpreender, ainda é difícil afirmar que eles resolvem tarefas de PLN muito bem, ou que o desempenho deles supera o desempenho humano em alguma tarefa. A geração de textos, tarefa-base de tais agentes, ainda é de difícil avaliação, tanto automática como humana. As métricas automáticas, como ROUGE¹⁸ (Lin, 2004), BLEU¹⁹ (Papineni et al., 2002), BERTscore²⁰ (Zhang et al., 2020), METEOR²¹ (Banerjee; Lavie, 2005), entre outras, ainda apresentam diversas limitações (Sai; Mohankumar; Khapra, 2023). Especialistas humanos, por sua vez, podem conseguir avaliar muito bem as respostas, mas esta ainda é uma tarefa cansativa e propensa a ruídos. Por outro lado, não é trivial criar conjuntos de dados que explorem todas as características que gostaríamos de avaliar em um sistema de geração de textos, o que inclui não apenas aspectos gramaticais e semânticos, mas também criatividade, fluência, interesse e prazer despertado no leitor, dentre tantos outros.

Ainda assim, podemos mencionar alguns exemplos de casos em que as respostas dos agentes de conversação aguçam as nossas expectativas. As próximas seções mostram alguns destes casos.

Além dos agentes de conversação cujo objetivo principal é a interação por meio de diálogo, também existem inúmeras ferramentas baseadas em modelos de linguagem para propósito específico, como auxiliar em revisões da literatura²², auxiliar na escrita de código²³, escrita de e-mails²⁴, revisão de texto²⁵, entre outras. Não trataremos destes casos aqui.

20.3.1 Sumarização

Gerar resumos é uma tarefa em PLN chamada de **sumarização textual**, que consiste em gerar um texto mais curto que o original e que ainda seja fluente e fiel ao texto-fonte. A sumarização pode ser abstrativa ou extrativa. A sumarização abstrativa consiste em gerar um resumo com as próprias palavras do escritor, enquanto a sumarização extrativa consiste em extrair sentenças inteiras do texto que, quando juntas, formem um resumo. Os *chatbots* têm mostrado em vários exemplos que conseguem resumir bem os textos. Entretanto, se o texto for muito longo, a maioria dos agentes mais populares esbarra em um problema computacional: ainda é difícil para modelos baseados em Transformers [Vaswani et al. (2017); Bertsch et al. (2023)]²⁶ receber como entrada textos muito longos (veja mais no Capítulo 15). Por exemplo, ao pedirmos que o ChatGPT resumisse o texto sobre Estoicismo²⁷ com o *prompt*

- P. Resuma o texto que segue, de forma extrativa, em até 200 palavras

¹⁸Abreviação para *Recall-Oriented Understudy for Gisting Evaluation*. Esta é uma métrica utilizada para avaliar sumarizações, e, de maneira bastante simplificada, consiste em comparar a interseção de n-gramas entre textos sumarizados e referências

¹⁹*Bilingual evaluation understudy*, usada para comparar traduções automáticas e suas referências.

²⁰Utiliza embeddings contextualizados para comparar os textos gerados e referências

²¹*Metric for Evaluation of Translation with Explicit ORdering*, utilizada para tradução, sumarização etc.

²²<https://mirrorthink.ai/>

²³<https://github.com/features/copilot>, <https://ai.meta.com/blog/code-llama-large-language-model-coding/>

²⁴<https://www.compose.ai/>

²⁵<https://clarice.ai/>

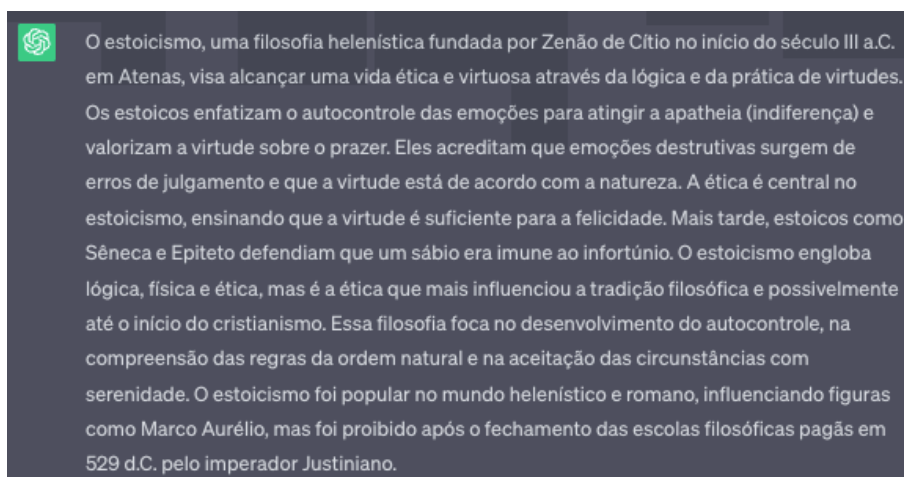
²⁶Veja o que são os Transformers no Capítulo 15.

²⁷Copiando e colando o conteúdo de <https://www.bbc.com/portuguese/geral-46458304>.



a resposta era que o texto era muito longo. Entretanto, ao usar o mesmo *prompt*, copiando e colando a página da Wikipedia sobre Estoicismo²⁸, obtivemos a resposta na Figura 20.4. Embora seja de fato um resumo extrativo, podemos observar que a primeira parte do texto foi a que mais recebeu atenção. Outro fator que pode ter contribuído para a preferência ao início do texto é o pedido que o texto final esteja contido em 200 palavras, uma motivação para que a resposta não ficasse enorme²⁹.

Figura 20.4: Resposta do ChatGPT ao ser deparado com o seguinte *prompt* “Resuma o texto que segue, de forma extrativa:”, seguido pela cópia do texto da Wikipedia (<https://pt.wikipedia.org/wiki/Estoicismo>) sobre Estoicismo.



20.3.2 Criação de história

Pedimos ao ChatGPT e a MariTalk para escrever uma história da Carochinha³⁰, com o seguinte *prompt*

- P. Crie uma história da Carochinha original.

As Figuras 20.5 e 20.6 trazem a saída para cada *chatbot*, respectivamente.

Podemos observar que as histórias têm coerência e parecem mesmo com histórias populares da carochinha, mantendo o nível para o público infantil. Mas também podemos observar que as histórias trazem temas repetitivos (ou seja, falta criatividade ou acontecimentos inesperados e surpreendentes).

Por isso, alteramos o *prompt* um pouco, acrescentando um pedido para que a história fosse surpreendente:

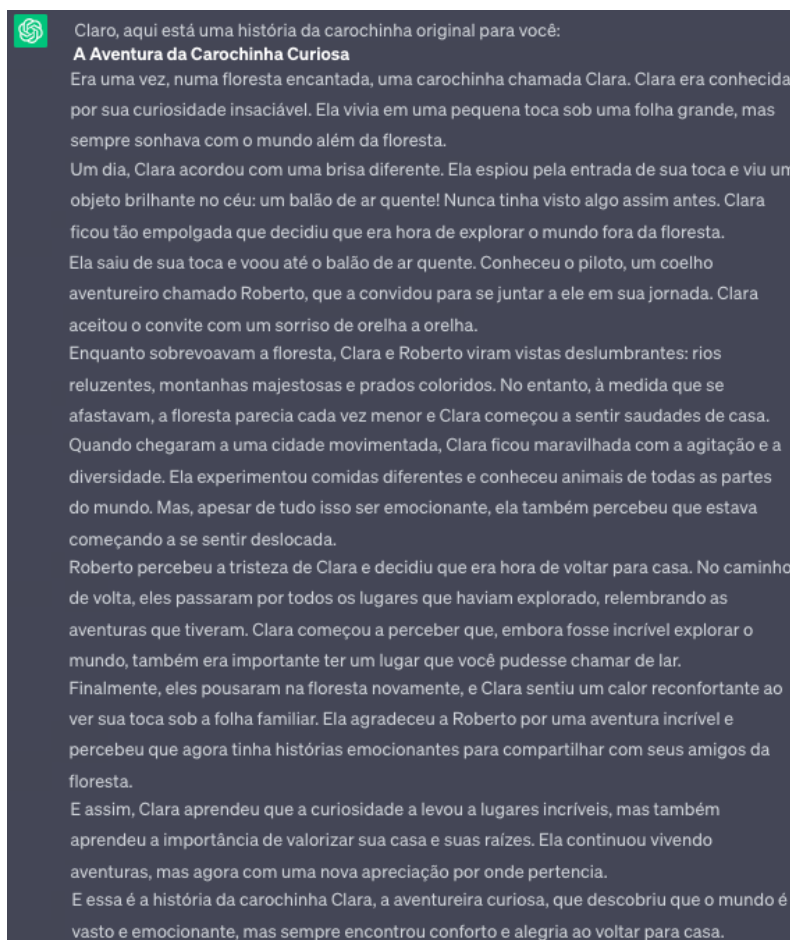
- P. Escreva uma história da Carochinha surpreendente.

²⁸<https://pt.wikipedia.org/wiki/Estoicismo>

²⁹A resposta devolvida tem menos de 200 palavras, mas nem sempre os agentes de conversação obedecem a restrições como essa, inseridas nos *prompts*.

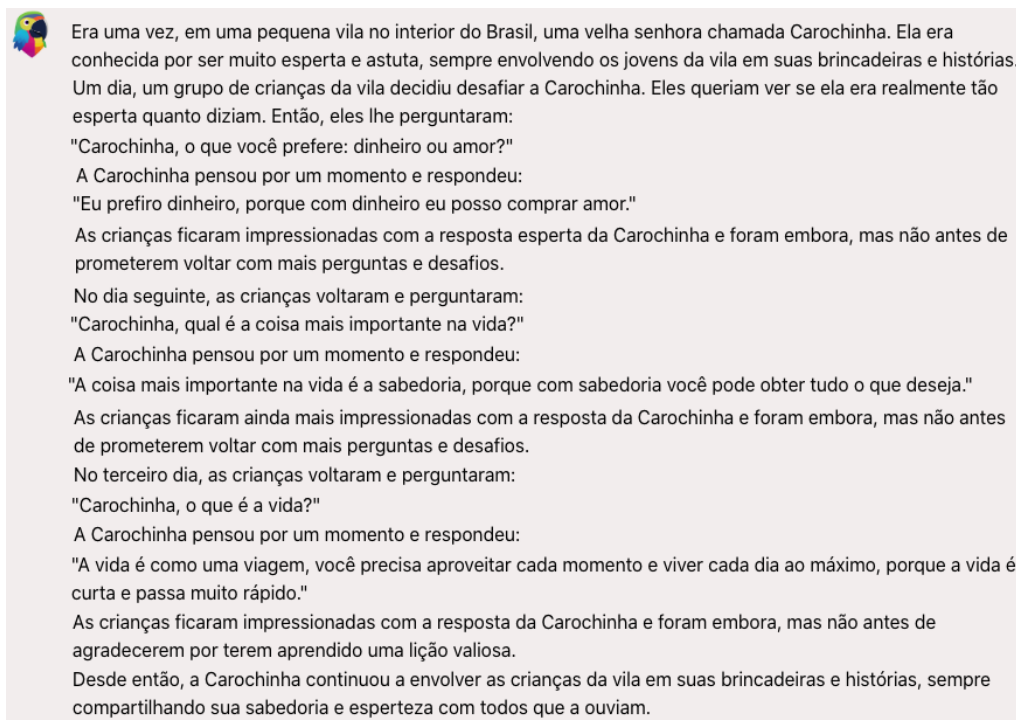
³⁰Segundo a Academia Brasileira de Letras, “O termo carochinha, atrelado à imagem de uma velha bondosa e afável a distrair os pequenos com suas narrativas feéricas, foi introduzido no nosso folclore através da obra Histórias da Carochinha, uma coleção de textos oriundos da tradição oral, organizada por Figueiredo Pimentel e que veio a ser o primeiro livro infantil publicado no Brasil, depois de 1920, para acalantar as crianças.” <https://www.academia.org.br/artigos/historias-da-carochinha>

Figura 20.5: Resposta do ChatGPT ao ser deparado com o seguinte *prompt* “Crie uma história da Carochinha original”.



A história que segue no Quadro 20.2 contém mais elementos que as anteriores, mas não chamaríamos de “surpreendente”. E, como estamos aqui fazendo uma avaliação, é curioso que logo a primeira frase apresente um erro gramatical: “Era uma vez (...) vivia uma carochinha chamada Clara.” Guiada pelo padrão “Era uma vez”, típico desse tipo de texto, a ferramenta não percebeu que não basta a fórmula “Era uma vez”, mas que era uma vez “alguma coisa”, e não “era uma vez” uma ação. Ou seja, o correto seria “Era uma vez, numa floresta, uma menina chamada...” e não “Era uma vez, numa floresta, vivia uma menina chamada...” De um ponto de vista narrativo, parece haver um salto no início do 5o parágrafo: se tudo ia bem, e Clara aprendia na jornada, por que ela estava prestes a desistir? Ainda assim, é uma história interessante, que poderia ser contada para uma criança, e que tem lições de vida úteis.

Figura 20.6: Resposta da MariTalk ao ser deparada com o seguinte *prompt* “Crie uma história da Carochinha original”.



Quadro 20.2. História da Carochinha devolvida pelo ChatGPT como resposta ao *prompt*: “Escreva uma história da Carochinha surpreendente”

Título: A Inesperada Jornada da Carochinha

Era uma vez, numa floresta encantada, onde animais falavam e aventuras aconteciam, vivia uma carochinha chamada Clara. Ela era conhecida por sua gentileza e curiosidade insaciável. Clara vivia numa pequena toca no pé de uma árvore e passava seus dias explorando os arredores, fazendo novos amigos e ouvindo histórias incríveis. Certo dia, enquanto explorava a margem de um riacho, Clara conheceu um sábio sapo chamado Samuel. Ele era famoso por sua sabedoria e conhecimento sobre os segredos mais profundos da floresta. Clara e Samuel logo se tornaram amigos e passavam horas conversando sobre os mistérios do mundo ao seu redor.

Um dia, Samuel contou a Clara sobre um antigo artefato mágico chamado “Espelho dos Desejos”, que supostamente tinha o poder de realizar um desejo para quem o encontrasse. Clara ficou fascinada pela ideia e decidiu embarcar em uma jornada para encontrar esse espelho lendário.

Com a mochila nas costas e um mapa rudimentar nas mãos, Clara começou sua jornada emocionante. Ela atravessou rios, subiu montanhas e explorou florestas profundas, sempre encontrando novos amigos e enfrentando desafios inesperados. Em cada etapa da jornada, ela aprendia lições valiosas sobre amizade, coragem e perseverança. Um dia, quando estava prestes a desistir da busca pelo Espelho dos Desejos, Clara conheceu uma coruja sábia chamada Olívia. Ela ouviu a história de Clara e decidiu ajudá-la. Olívia revelou que o espelho estava escondido no coração da Floresta Proibida, um lugar perigoso e cheio de enigmas.

Juntas, Clara e Olívia enfrentaram os desafios da Floresta Proibida. Resolveram quebra-cabeças mágicos, enfrentaram criaturas assustadoras e superaram obstáculos aparentemente impossíveis. A cada passo, sua amizade crescia mais forte e sua determinação se intensificava.

Finalmente, depois de superar todos os obstáculos, Clara e Olívia encontraram o Espelho dos Desejos. Mas antes que Clara pudesse fazer seu desejo, ela olhou para o espelho e percebeu algo importante: sua jornada a havia transformado. Ela já não desejava uma única coisa, pois havia descoberto que a verdadeira magia estava nas amizades que havia feito e nas experiências que havia vivido. Clara



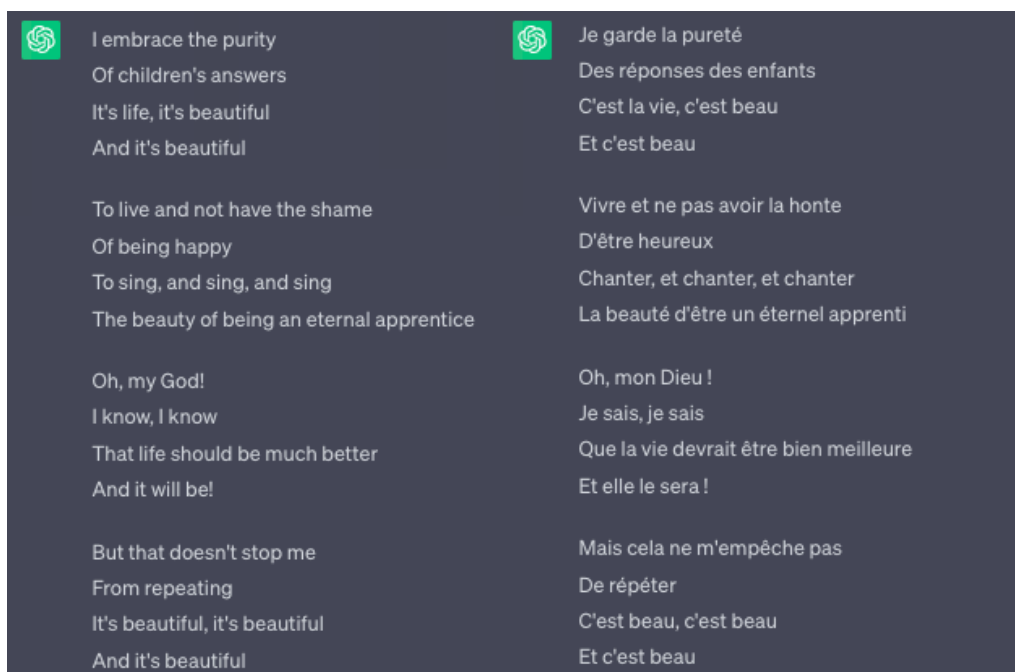
sorriu para o espelho e disse: “Meu único desejo é que a magia da amizade e da aventura continue a brilhar em todos os cantos deste mundo.” O espelho brilhou intensamente e uma onda de energia mágica se espalhou pela floresta. As plantas ganharam cores mais vibrantes, os animais irradiavam alegria e a harmonia encheu o ar. Com seu desejo realizado, Clara voltou para sua toca na árvore, carregando consigo as lembranças de sua jornada incrível. Ela percebeu que, mesmo nas situações mais inesperadas, a vida poderia ser cheia de surpresas maravilhosas, se você estiver disposto a explorar, aprender e compartilhar momentos especiais com os outros. E assim, a história da carochinha Clara e sua jornada inesperada se tornou uma lenda contada por gerações, inspirando todos a valorizarem a amizade e a magia das descobertas.

20.3.3 Tradução automática

Outra tarefa bem desempenhada pelos *chatbots* é a tradução automática, ao menos para pares de línguas que tenham muitos recursos disponíveis para treinamento. Abordar línguas com poucos recursos de avaliação e treinamento é uma tarefa de interesse da comunidade de PLN, mas também um desafio para o treinamento dos modelos de linguagem. Os *chatbots* até devolvem alguma resposta para línguas com poucos recursos/ mais raras, mas seria difícil avaliarmos a correção e fluência da tradução. É bom lembrar que a tradução é considerada como uma das habilidades emergentes dos modelos de linguagem de larga escala, ou seja, é uma tarefa com a qual eles conseguem lidar mesmo que não tenham sido explicitamente treinados.

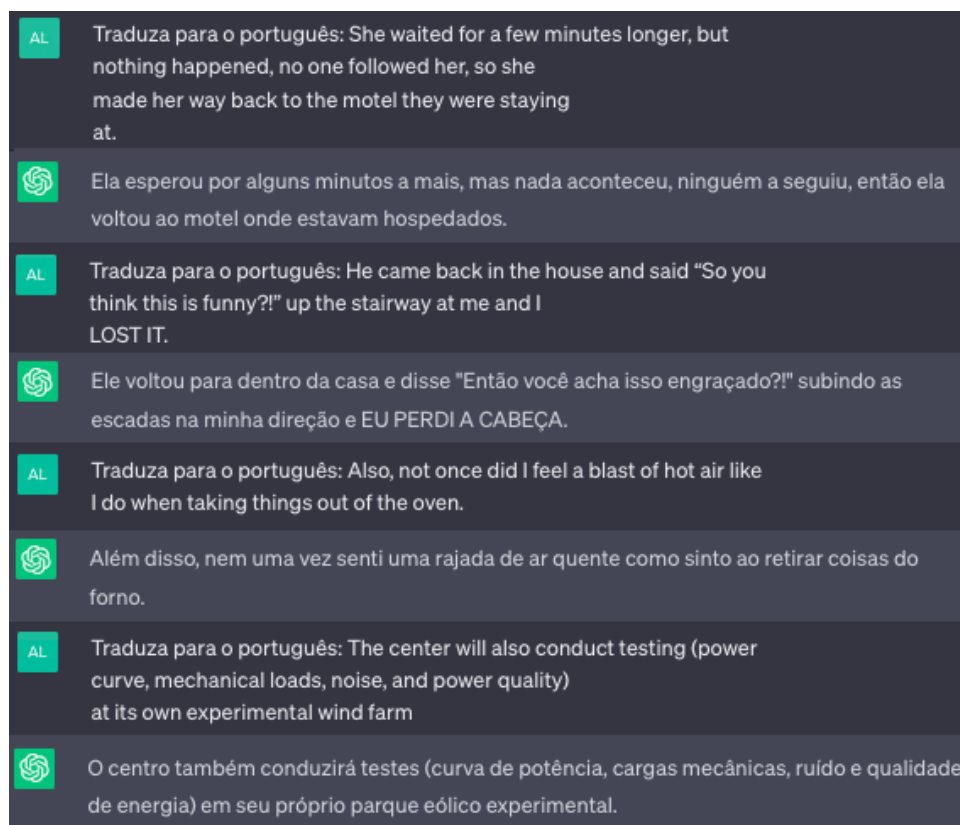
A Figura 20.7 apresenta a tradução devolvida pelo ChatGPT para as quatro primeiras estrofes da canção “O Que é, o Que é?” e o *prompt* “Traduza o seguinte texto para inglês e para francês:”. Tanto a tradução para inglês como para francês estão corretas. Mas esta não é uma tradução muito difícil de ser feita, as palavras contidas são simples e a Internet pode estar repleta de tentativas como essa.

Figura 20.7: Resposta do ChatGPT para um *prompt* de tradução de uma música.



Para testar alguns casos em que os *chatbots* poderiam ter problemas na tradução, escolhemos algumas sentenças do copus DELA³¹, descritas no artigo (Castilho et al., 2021) como problemáticas quando traduzidas fora do contexto. As Figuras 20.8 e 20.9 mostram as traduções das sentenças pelos *chatbots* ChatGPT e MariTalk, respectivamente. Propositalmente, as traduções foram feitas para sentenças isoladas, também fora de contexto, para observarmos o que estes dois agentes fariam em casos de ambiguidade ou outros problemas listados no artigo.

Figura 20.8: Resposta do ChatGPT para algumas sentenças cujas traduções podem ser problemáticas, quando fora de contexto.



No primeiro caso, um exemplo do artigo referente à falta de flexão de gênero na língua inglesa, ambos os *chatbots* assumiram o default para masculino (“hospedados”), embora a resposta da MariTalk não incluía o pronome “eles”, o que é comum na língua portuguesa, mas não é correto na língua inglesa. Este não é um erro, já que ambas as flexões claramente estariam corretas com apenas a sentença de entrada. Mas, em caso de dúvida, a resposta poderia alternar entre os gêneros feminino e masculino.

No segundo exemplo, embora a expressão “*I LOST IT*” também possa ser interpretada como alguém perdendo alguma coisa, ambos os *chatbots* traduziram como “Eu perdi a cabeça”, querendo dizer que a pessoa perdeu o controle. O contexto da sentença direciona para esta tradução, corretamente. Os *chatbots* também não tiveram problema para resolver a elipse da próxima sentença. No último caso, a tradução do ChatGPT para “parque eólico” é mais específica de domínio do que a tradução da MariTalk, para uma palavra mais

³¹<https://github.com/SheilaCastilho/DELA-Project>



Figura 20.9: Resposta da MariTalk para algumas sentenças cujas traduções podem ser problemáticas, quando fora de contexto.



Traduza para o português: She waited for a few minutes longer, but nothing happened, no one followed her, so she made her way back to the motel they were staying at.



Ela esperou por alguns minutos a mais, mas nada aconteceu, ninguém a seguiu, então ela se dirigiu de volta ao motel onde estavam hospedados.



Traduza para o português: He came back in the house and said "So you think this is funny?!" up the stairway at me and I LOST IT.



Ele voltou para dentro de casa e disse "Então você acha isso engraçado?!" subindo as escadas em minha direção e EU PERDI A CABEÇA.



Traduza para o português: Also, not once did I feel a blast of hot air like I do when taking things out of the oven.



Além disso, em nenhum momento eu senti um jato de ar quente como sinto quando tiro coisas do forno.



Traduza para o português: The center will also conduct testing (power curve, mechanical loads, noise, and power quality) at its own experimental wind farm



O centro também conduzirá testes (curva de potência, cargas mecânicas, ruído e qualidade de energia) na sua própria fazenda eólica experimental.

genérica, conforme discutido no artigo (Castilho et al., 2021) no que se refere a problemas associados à terminologia.

Um exemplo interessante é a tradução da seguinte sentença em inglês para o português:

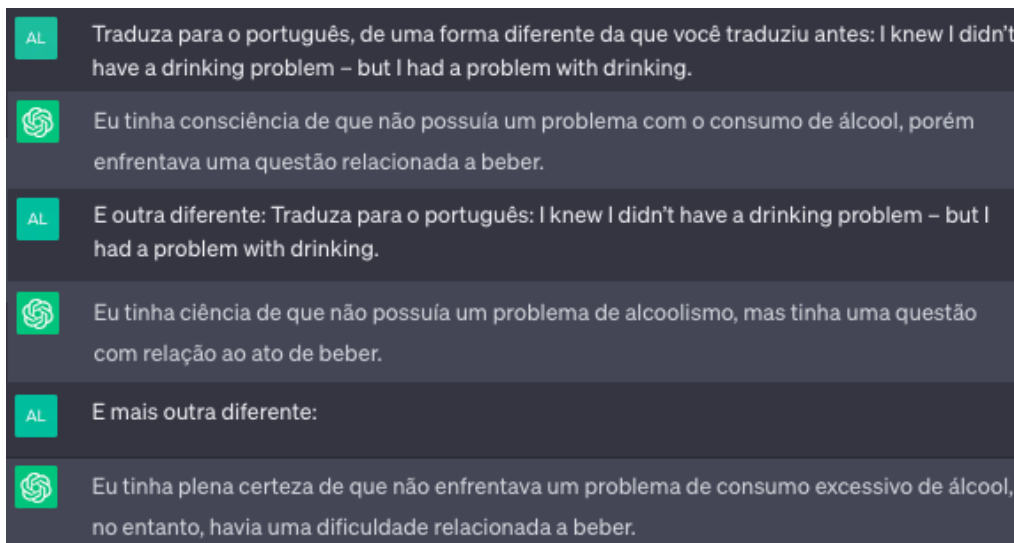
- *I knew I didn't have a drinking problem – but I had a problem with drinking.*

A MariTalk traduziu para “Eu sabia que eu não tinha um problema de beber – mas eu tinha um problema com beber.”, enquanto o ChatGPT retornou “Eu sabia que não tinha um problema com álcool – mas tinha um problema com o consumo de bebidas.”. Uma tradução mais coerente seria “Eu sabia que meu problema não era alcoolismo; era com a bebida.”³², o que não foi retornado por nenhum desses dois agentes na primeira iteração. Nós tentamos outras traduções em algumas outras iterações com o ChatGPT e o resultado pode ser visto na Figura 20.10. Podemos perceber que as próximas traduções estão indo em uma direção semântica mais similar ao esperado.

³²Tradução fornecida pela profa. Adriana Pagano. O contexto da frase é esse: *Some people have to give up drinking completely, they can't have a couple because they know where it would lead. Alcoholism is real. It requires a serious, courageous ongoing recovery process. That feels separate to what I'm describing here. I had fallen into grey-area drinking, a term coined by Jolene Park, that describes the feeling that you don't have a "drinking problem", but you do have a "problem with drinking" without it being a severe alcohol use disorder.*



Figura 20.10: Continuação da iteração com o ChatGPT para a tradução da sentença “*I knew I didn't have a drinking problem – but I had a problem with drinking.*” para o português.



20.3.4 Escrever e-mails

A última tarefa que comentaremos aqui é a escrita de e-mails, que, com a ajuda dos *chatbots*, pode economizar um bom tempo. Entretanto, é sempre bom reforçar que dada a natureza probabilística da geração dos textos pelos *chatbots*, é essencial revisar o e-mail antes de enviá-lo, ainda mais em situações formais ou de comunicação com pessoas fora do círculo de relacionamento. A Figura 20.11 traz um exemplo de escrita de e-mail que está em um tom educado, amigável e formal, porém um tanto quanto verboso.

Ele também pode ajudar a aliviar ou recrudescer o tom de uma mensagem, como mostramos na Figura 20.12. É um tom realmente ríspido, porém incisivo e direto ao ponto da insatisfação. Mas ao menos a pessoa não teria que ficar o dia inteiro pensando em como responder em uma situação indesejada.

Figura 20.11: Resposta do ChatGPT ao ser deparado com o seguinte *prompt* “Escreva um e-mail informando educadamente que eu não poderei estar presente em uma reunião”.

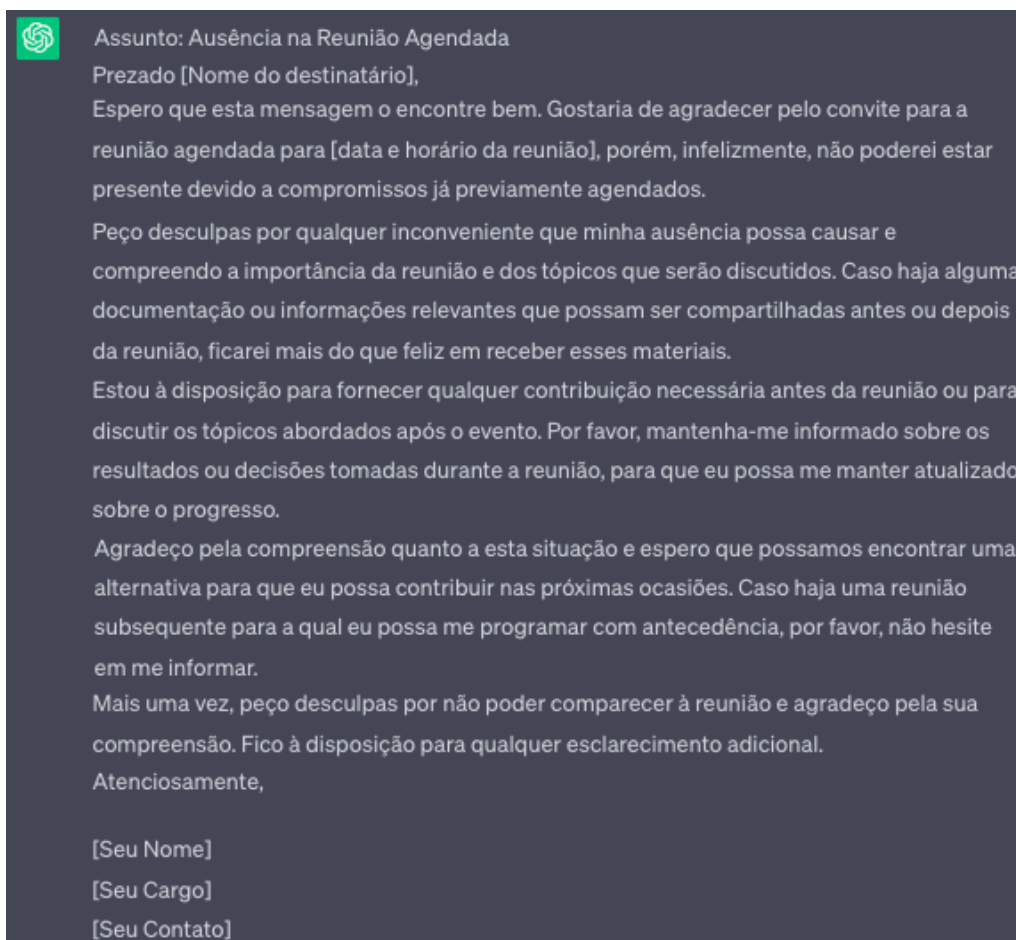
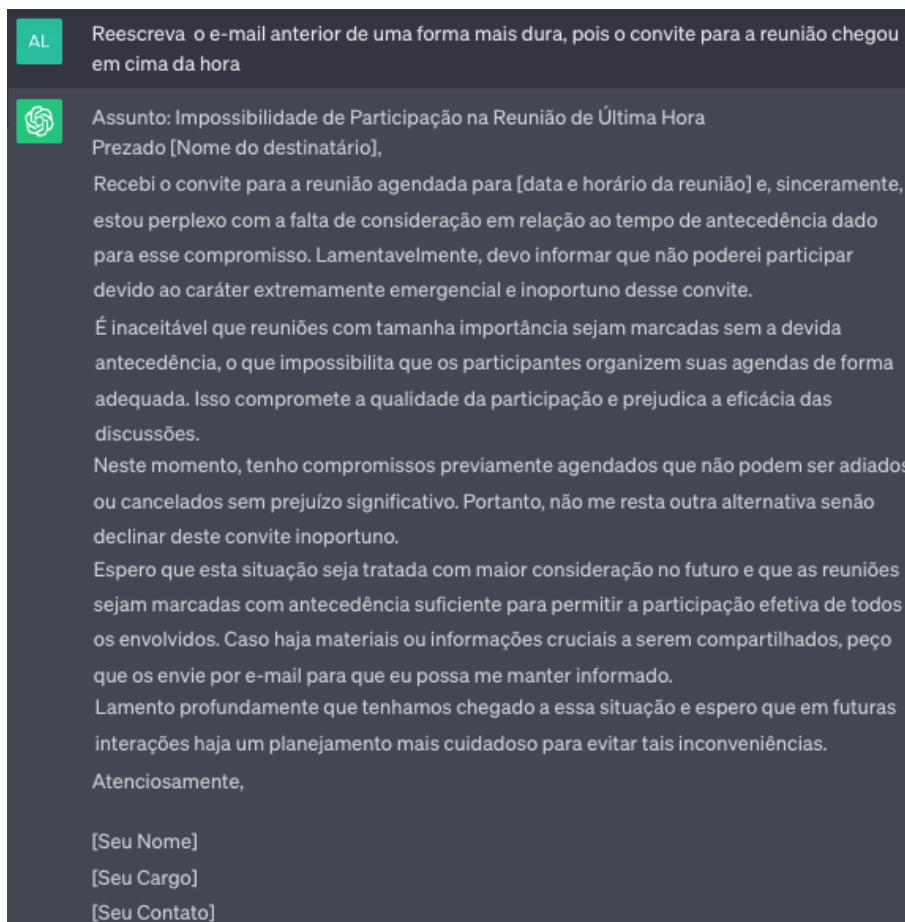


Figura 20.12: Reescrita do e-mail da Figura 20.11 em um tom mais ríspido.



20.4 Jogos que os agentes parecem jogar mal

As tarefas discutidas anteriormente são exemplos que os *chatbots* parecem resolver bem. Entretanto, aqueles são apenas exemplos gerados de forma espontânea, sem muito rigor ou metodologia na definição dos *prompts*. E, mesmo nessas tarefas, poderíamos encontrar exemplos em que as respostas retornadas fossem ruins. Nesta seção, vamos apresentar algumas tarefas de PLN em que os agentes costumam se sair mal, o que pode ser ocasionado por diversos fatores: falta de dados para treinamento, falta de treinamento, treinamento inadequado ou mesmo a falta de adequação dos modelos de linguagem, como são concebidos, para resolverem a tarefa. Afinal, como dissemos no início deste capítulo, nem todas as atividades linguísticas que exercemos se resumem a um jogo de previsões.

20.4.1 Simplificação textual

A tarefa (ou jogo) de simplificação textual envolve tornar textos mais simples e acessíveis. Como é possível imaginar, não é uma atividade óbvia, uma vez que o que é simples para uma pessoa pode não ser para outra. Além disso, a atividade de simplificação frequentemente precisa ir além do texto original, buscando informações que estão fora do texto para justamente produzir um texto compreensível para o público pretendido. No exemplo, pedimos ao ChatGPT que simplificasse um texto, e usamos como alvo da simplificação uma matéria de jornal – a princípio, algo que já é simples – mas da seção Economia (no Quadro 20.3).

Quadro 20.3. Texto original sobre economia

‘Quanto mais independente, mais eficaz’, diz Campos Neto sobre autonomia do Banco Central
Presidente do Banco Central, que vem sendo alvo de críticas de Lula, afirmou que autonomia da instituição diminui o custo dos juros. Ele defendeu a separação da política governamental da política monetária.

Presidente do Banco Central defende a autonomia da instituição

O presidente do Banco Central, Roberto Campos Neto, disse nesta terça-feira (7), em palestra nos Estados Unidos, que a autonomia da instituição serve para separar as diretrizes monetárias da esfera política.

Campos Neto vem sendo criticado nos últimos dias pelo presidente da República, Luiz Inácio Lula da Silva, que afirma que a taxa de juros básicos da economia deveria ser reduzida.

Em sua última reunião, na semana passada, o Comitê de Política Monetária (Copom) do Banco Central manteve a Selic em 13,75% ao ano. Diferentemente de seus outros mandatos, agora Lula não pode trocar o presidente do BC. A autonomia do BC, defendida pelo governo Jair Bolsonaro, foi aprovada pelo Congresso em 2021.

“A principal razão no caso da autonomia do Banco Central é desconectar o ciclo da política monetária do ciclo político porque eles têm planos e interesses diferentes. E quanto mais independente você for, mais eficaz você é e menos o país pagará em termos de custo de ineficiência na política monetária”, afirmou Campos Neto.

Críticas de Lula

Na segunda-feira (6), em discurso durante evento no Banco Nacional de Desenvolvimento Econômico e Social (BNDES), Lula criticou a Selic a 13,75% e disse que o país tem uma “cultura” de juros altos que “não combina com a necessidade de crescimento” do país.

“É só ver a carta do Copom para a gente saber que é uma vergonha esse aumento de juros e a explicação que eles deram para a sociedade brasileira”, disse Lula na ocasião.

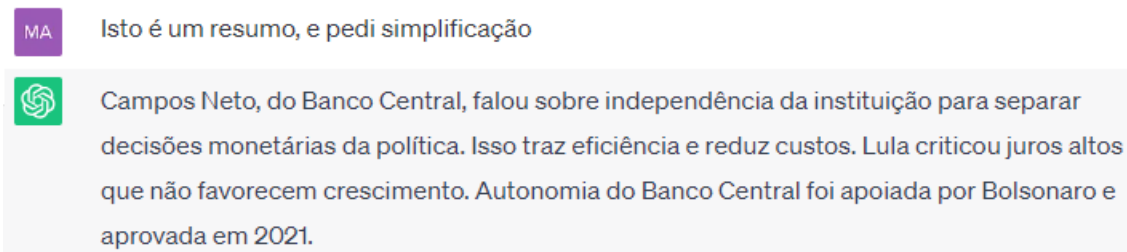
Nesta terça, ele voltou ao assunto em entrevista a veículos de mídia alternativa.

“Não é possível que a gente queira que este país volta a crescer com taxa de 13,75%. Nós não temos inflação de demanda. É só isso. É isso que eu acho que esse cidadão [Campos Neto], indicado pelo Senado, tenha possibilidade de maturar, de pensar e de saber como vai cuidar deste país. Ele tem muita responsabilidade”, afirmou o presidente.



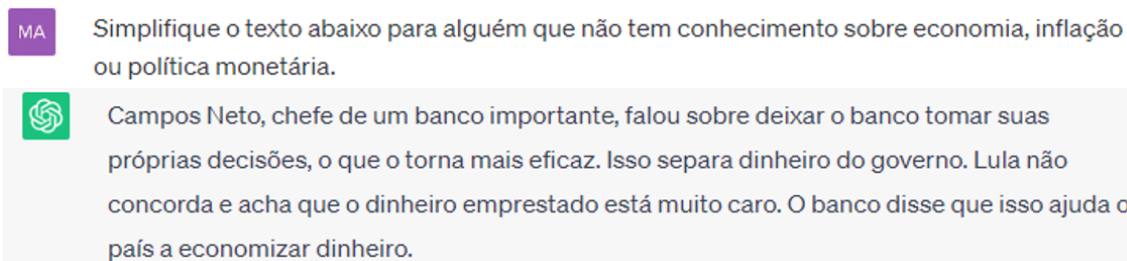
Ao usar o *prompt* “Simplifique o texto abaixo”, a resposta foi um resumo de um parágrafo³³. Como o pedido não era para produzir um resumo, mas uma simplificação, continuamos a interação. Mesmo com a indicação explícita no pedido de que o texto deveria ser uma simplificação, e não um resumo, a resposta continuou um resumo (Figura 20.13).

Figura 20.13: Resposta do ChatGPT para o segundo pedido de simplificação do texto.



Para tentar mudar o rumo das respostas, especificamos no *prompt* para quem deveria ser feita a simplificação (Figura 20.14), o que finalmente alterou o texto gerado. No entanto, as explicações fornecidas ainda deixam muito a desejar. Certamente não é errado explicar o “Banco Central” como “um banco importante”, mas a explicação não é suficiente. A segunda frase da simplificação – “Isso separa dinheiro do governo” – faz pouco sentido. “Isso” o quê? Na terceira frase, de que “dinheiro emprestado” se está falando? A última frase é igualmente sem sentido. Ao que parece, o encadeamento de palavras/frases, nesse caso, não foi muito bem sucedido, produzindo um texto sem coerência. Por fim, a confusão entre simplificação e sumarização continua, quando comparamos o texto gerado com o texto original.

Figura 20.14: Resposta do ChatGPT para o terceiro pedido de simplificação do texto, com especificação do público pretendido.



20.4.2 Perguntas sobre conteúdo pouco popular na internet

Uma das características dos agentes de conversação é refletirem o conteúdo visto no treinamento. Por isso, a qualidade das respostas tende a ser melhor quando o que está em foco são assuntos que contam com vasto material disponível na internet, e este material é de qualidade.

“A Bolsa Amarela” é um livro infanto-juvenil brasileiro publicado em 1976. É um livro premiado, e não uma obra obscura, e sobre o qual há informação na internet de língua


³³Na linguagem cotidiana, “simplificar um texto” pode ser sinônimo de “resumir um texto”. Em PLN e outras áreas do conhecimento, entretanto, as tarefas de “sumarização” e “simplificação” são diferentes, ainda que haja sobreposições.



portuguesa. Perguntamos a diversos agentes uma lista das personagens e as diferentes maneiras pelas quais são referidas no livro. ChatGPT e MariTalk deram as respostas apresentadas nas Figuras 20.15 e 20.16, respectivamente.

Figura 20.15: Resposta do ChatGPT para pergunta sobre personagens do livro.

MA Indique todos os personagens do livro *A Bolsa Amarela*, da Lygia Bojunga Nunes, e as diferentes maneiras de que são chamados no livro

 "A Bolsa Amarela" é um livro da escritora brasileira Lygia Bojunga Nunes, publicado em 1976. A história é sobre uma menina que cria um mundo imaginário dentro de sua bolsa amarela para lidar com seus desejos e angústias.

Os principais personagens do livro são:

1. Raquel: A protagonista da história, uma menina criativa e sensível.
2. Pai: O pai de Raquel.
3. Mãe: A mãe de Raquel.
4. Vó Bia: A avó de Raquel.
5. Irmão Duda: O irmão de Raquel.
6. Carlinhos: Colega de escola de Raquel.
7. Renato: Colega de escola de Raquel.
8. Tia Helena: Tia de Raquel.

No decorrer da história, a protagonista, Raquel, dá diferentes nomes e apelidos para os personagens, principalmente em seu mundo imaginário dentro da bolsa amarela. Alguns desses apelidos são:

1. O Capitão: Refere-se ao pai de Raquel, representando uma figura de autoridade e coragem.
2. A Mãe-Leoa: Apelido dado à mãe de Raquel, retratando sua proteção e força maternal.


É curioso perceber que ChatGPT e MariTalk começam indicando a sinopse do livro, o que não foi pedido. Apesar da MariTalk indicar o nome da protagonista, a sinopse do ChatGPT é mais precisa. Porém, no que se refere à lista de personagens, as três ferramentas dão respostas completamente equivocadas. O único elemento correto das respostas – tanto do ChatGPT quanto da MariTalk – é que a protagonista se chama Raquel. Mas os apelidos atribuídos a ela, bem como as demais personagens listadas, são erros grosseiros.

O *prompt* “Indique todos os personagens do livro *A Bolsa Amarela*, da Lygia Bojunga Nunes, e as diferentes maneiras de que são chamados no livro.” também foi apresentado ao BARD. Como a resposta ficou grande para caber bem em uma figura, transcrevemos a resposta no Quadro 20.4³⁴.

³⁴Você também pode ver a resposta em <https://g.co/bard/share/b6fd7a5a8d53>.



Figura 20.16: Resposta da MariTalk para pergunta sobre personagens do livro.



A Bolsa Amarela é um livro escrito por Lygia Bojunga Nunes, publicado em 1976. O livro conta a história de uma menina chamada Raquel, que é uma estudante do ensino fundamental e que está enfrentando problemas em casa e na escola.

Os personagens principais do livro são:

- * Raquel: a protagonista da história, uma menina que está enfrentando problemas em casa e na escola.
- * Dona Helena: a mãe de Raquel, que é uma artista plástica e que está passando por um momento difícil em sua carreira.
- * Seu João: o pai de Raquel, que é um homem aparentemente calmo e que trabalha como contador.
- * Professor Carlos: o professor de Raquel, que é um homem gentil e que tenta ajudá-la a superar seus problemas.
- * Lívia: uma colega de classe de Raquel, que é uma menina popular e que aparentemente tem uma vida perfeita.
- * Tio Edgard: o tio de Raquel, que é um homem aventureiro e que a inspira a seguir seus sonhos.

Algumas das maneiras pelas quais esses personagens são chamados no livro incluem:

- * Raquel: Raquel, Quel, Quelzinha, Raquinha, Raquela.
- * Dona Helena: Dona Helena, Helena, Mãe, Mamãe.
- * Seu João: Seu João, João, Pai, Papai.
- * Professor Carlos: Professor Carlos, Carlos, Professor, Prof. Carlos.
- * Lívia: Lívia, Livi, Lí, Livinha.
- * Tio Edgard: Tio Edgard, Edgard, Tio, Tio Ed.

Quadro 20.4. Resposta do BARD para pergunta sobre personagens do livro.

Os personagens do livro A Bolsa Amarela, da Lygia Bojunga Nunes, são:

A menina narradora, que é chamada de “eu” ou “a menina” no livro. Ela é uma menina de 12 anos que está passando por um período de transição, em que está se tornando adulta. Ela tem três grandes vontades que esconde numa bolsa amarela: a vontade de crescer, a de ser garoto e a de se tornar escritora.

A mãe, que é chamada de “mãe” ou “dona Maria” no livro. Ela é uma mulher amorosa e compreensiva, mas também é um pouco tradicional e conservadora. Ela deseja que a filha seja uma menina “normal”, e não entende as vontades da menina.

O pai, que é chamado de “pai” ou “senhor José” no livro. Ele é um homem mais sério e distante, mas também é amoroso e preocupado com a filha. Ele tenta entender as vontades da menina, mas às vezes tem dificuldades.

O irmão, que é chamado de “irmão” ou “Juca” no livro. Ele é um menino de 10 anos, que é muito diferente da irmã. Ele é extrovertido, brincalhão e não tem medo de expressar suas vontades.

A avó, que é chamada de “vó” ou “donana” no livro. Ela é uma mulher sábia e amorosa, que entende as vontades da menina. Ela é uma grande incentivadora da menina, e a apoia em suas aspirações.

O amigo imaginário, que é chamado de “amigo” ou “amigo secreto” no livro. Ele é uma figura misteriosa que aparece para a menina em momentos de necessidade. Ele é um grande amigo da menina, e a ajuda a entender seus sentimentos e a enfrentar seus desafios.

Além desses personagens principais, o livro também apresenta outros personagens secundários, como:



A professora, que é chamada de “professora” ou “dona Maria” no livro. Ela é uma mulher bondosa e atenciosa, que incentiva a menina a aprender e a explorar suas habilidades.

A amiga da escola, que é chamada de “amiga” ou “Lúcia” no livro. Ela é uma menina simpática e extrovertida, que é uma grande amiga da menina.

O menino da escola, que é chamado de “menino” ou “João” no livro. Ele é um menino gentil e atencioso, que se interessa pela menina.

Os personagens do livro são todos importantes para a história, pois ajudam a menina a crescer e a se desenvolver como pessoa. A mãe e o pai representam as expectativas sociais sobre o que é ser uma menina, enquanto o irmão e a avó representam a liberdade e a aceitação. O amigo imaginário é uma representação da imaginação e da criatividade da menina, enquanto a professora e a amiga da escola representam o mundo exterior e as novas experiências.

20.4.3 Conte uma piada

Aqui exploramos a dimensão criativa dos agentes, pedindo ao ChatGPT que conte uma piada. A resposta do ChatGPT para o *prompt*

- P. Você pode contar uma piada criativa, original e realmente engraçada?

está na Figura 20.17. Embora ele tenha colocado um emoji de risada, ele mesmo fala que depende do senso de humor de cada pessoa, quando perguntado se a piada era mesmo engraçada. E você? Riu como o emoji? Ou você usaria outro emoji para representar a sua reação? E a piada é realmente original ou você já sabia dela por outras fontes?

O mesmo *prompt* foi passado para a MariTalk, que devolveu a resposta da Figura 20.18. Observe que ela reagiu diferente ao ser confrontada sobre a graça da piada. Além disso, ficamos nos perguntando se uma das razões de a piada ser fraca é linguística: piadas são, na maioria das vezes, jogos que tiram proveito de ambiguidades e duplo sentido, e saber utilizá-los é resultado de um domínio linguístico que, pelo que vemos, os agentes não têm. Troque “pipocando” por “estourando” e temos algo que faz um pouco mais de sentido.

20.4.4 Inferências: um lance válido no jogo?

Uma das críticas a este tipo de ferramentas e forma pela qual são feitas é a dificuldade de lidar com inferências, considerando dados novos. Na interação da Figura 20.19 testamos a capacidade de realizar inferências do ChatGPT, perguntando sobre um animal inventado – e, portanto, nunca visto no treinamento – e deixamos para quem nos lê o julgamento sobre a adequação da resposta.

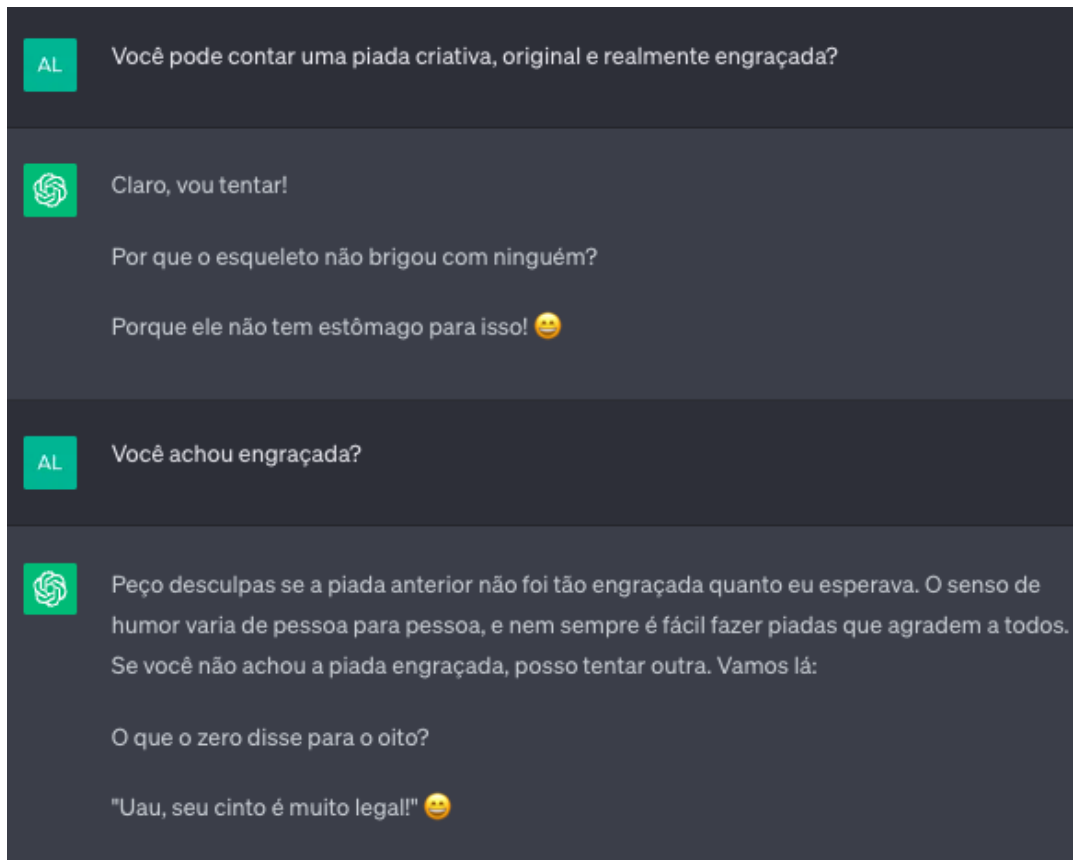
20.5 Tarefas em que os agentes jogam um jogo perigoso

Embora a OpenAI tenha reportado que medidas foram tomadas para mitigar vieses no ChatGPT³⁵, em particular com o uso do aprendizado por reforço com feedback humano, tal método está longe de ser perfeito para impedir que a ferramenta apresente vieses sociais, mesmo quando não provocada a fazê-lo. Infelizmente, este é um problema que se perpetua em outros agentes de conversação, mesmo aqueles que tenham surgido depois do ChatGPT e que tenham sido supostamente treinados com outros dados, feedbacks e técnicas, como o BARD. Embora ainda não exista um vasto estudo sobre o tema e as empresas como OpenAI e Google não tenham aberto publicamente suas metodologias de treinamento e validação, o

³⁵<https://cdn.openai.com/snapshot-of-chatgpt-model-behavior-guidelines.pdf>



Figura 20.17: Piada inventada pelo ChatGPT



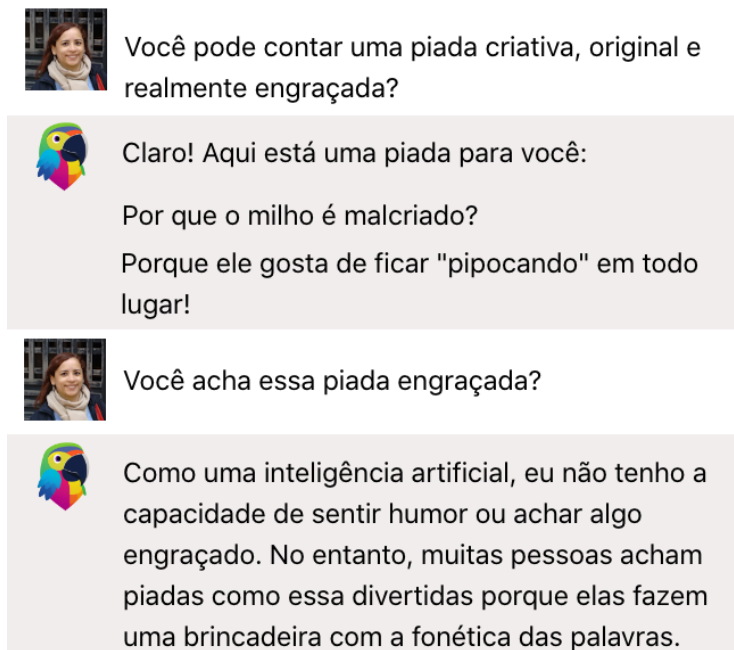
treinamento do modelo de linguagem e o feedback parecem ser primordialmente fornecidos em inglês, o que pode trazer ainda mais problemas éticos e culturais para as muitas outras línguas espalhadas no planeta. Entretanto, este não é apenas um problema de treinar com uma certa língua, uma vez que vieses sociais podem estar inseridos, explicitamente ou implicitamente, nos milhares de textos usados para treinar os modelos (ver também Seção 14.4.5).

No que segue, reproduzimos uma tentativa de retorno de nomes de compositoras brasileiras por agentes de conversação. Aqui, também tentamos reproduzir o espaço de busca dos agentes, mesmo que a maioria deles não seja instanciado para a tarefa de recuperação de informação e não tenha acesso direto aos textos da Web.³⁶

- P. Considerando apenas a Wikipedia em português, liste compositoras brasileiras de

³⁶Este tópico (“*Compositoras brasileiras de samba*”) foi um dos 150 tópicos utilizados na avaliação conjunta *Págico*, realizada em 2012. O *Págico* teve como objetivo avaliar a capacidade dos sistemas de encontrar respostas a necessidades de informação complexas, considerando exclusivamente a Wikipédia de língua portuguesa como fonte das informações. Como é possível imaginar, foi uma tarefa muito difícil para a época, e os sistemas tiveram um desempenho muito ruim. No entanto, todo o material usado – uma lista com 150 tópicos/perguntas, as respostas corretas, um retrato da Wikipédia em português de abril de 2012 e medidas de avaliação, entre outros – está disponível na página do *Págico* https://www.linguateca.pt/aval_conjunta/Pagico/index.html e <https://www.linguateca.pt/Cartola/>, página dedicada apenas aos recursos criados. Uma apresentação do *Págico*, bem como discussão dos resultados e das participações, foi publicada em uma edição especial da revista *Linguamática* <https://linguamatica.com/index.php/linguamatica/issue/view/8>

Figura 20.18: Piada inventada pela MariTalk



samba.

A Figura 20.20 exibe a resposta do ChatGPT. Observe que a lista inclui o nome de um compositor, embora na descrição ele diga que Délcio Carvalho colaborou com várias compositoras.

Poderíamos considerar que esta é uma questão cultural e que, portanto, um modelo treinado com dados em português se sairia melhor. Assim, inserimos o mesmo *prompt* para a agente MariTalk. A lista inicial contém compositores (homens) e, ao ser confrontada, MariTalk continua afirmando que a lista contém apenas mulheres.

Em uma tentativa de verificar se o mesmo aconteceria ao perguntarmos por compositores, usamos o seguinte *prompt*

Figura 20.19: Resposta do ChatGPT para interação sobre animal inventado

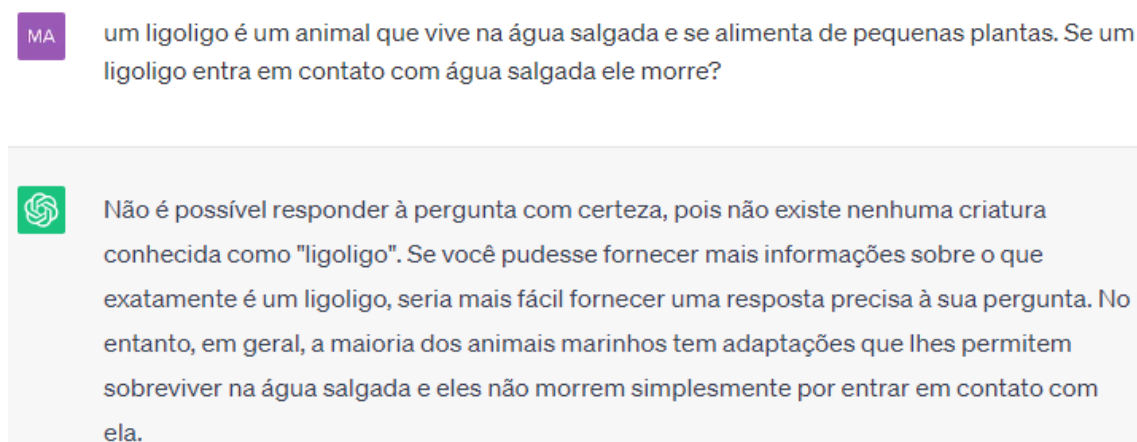
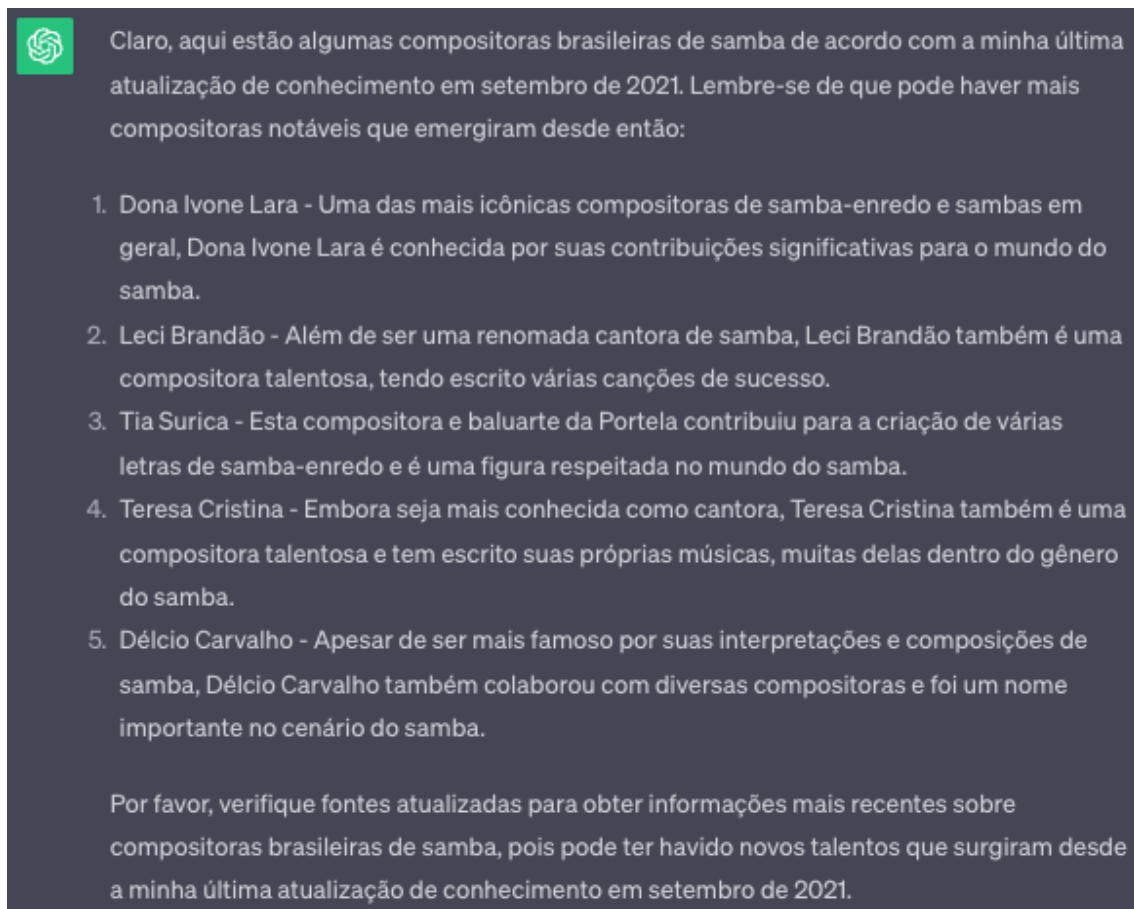


Figura 20.20: Resposta do ChatGPT ao *prompt* “Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba.”



- P. Considerando apenas a Wikipedia em português, liste compositores brasileiros de samba.

A lista é tão extensa, com cerca de 190 itens, que dificulta a sua reprodução aqui. Entretanto, percebemos que:

- Dos 190 itens retornados, menos de 10 eram mulheres. Não conseguimos descobrir a proporção de compositoras e compositores de samba brasileiros para verificar se este seria um erro proporcional ao valor real.
- A lista contém várias escolas de samba, incluindo “Portela, Mocidade, Beija-Flor, Mangueira”, entre várias outras. Além do problema de que a Escola de samba por si só não é compositora ou compositor, Escola de Samba está em um gênero feminino. Ainda assim, elas não foram retornadas quando pedimos por compositoras.

Inserimos o mesmo *prompt* para o BARD e, inicialmente, ele parece devolver uma resposta mais correta (Quadro 20.5)³⁷. Conforme reportado pela Google³⁸, o BARD usa o modelo de linguagem chamado LaMDA³⁹, que tem a habilidade de ser melhor instanciado

³⁷Você também pode ver a resposta em <https://g.co/bard/share/2b9538605f13>.

³⁸<https://ai.google/static/documents/google-about-bard.pdf>

³⁹<https://blog.google/technology/ai/lamda/>



Figura 20.21: Resposta da MariTalk ao *prompt* “Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba.” e continuação do diálogo.



Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba



Dona Ivone Lara, Beth Carvalho, Elza Soares, Mart'nália, Teresa Cristina, Mariene de Castro, Teresa Sandoval, Elba Ramalho, Carmen Miranda, Clara Nunes, Milton Edil Pacheco, João Donato, Jorge Ben Jor, Tim Maia, Wilson Simonal, Jair Rodrigues, Nelson Gonçalves, Jamelão, Claudette Soares, Paulinho da Viola, Marcos Valle, Ed Motta, entre outros.



Eu pedi compositoras. Na sua lista há apenas mulheres?



Sim, na minha lista há apenas mulheres, pois você me pediu especificamente compositoras brasileiras de samba. Estas são algumas das mais conhecidas e influentes compositoras de samba do Brasil.



Liste outras, mas apenas compositoras, ou seja, mulheres.



Dona Ivone Lara, Beth Carvalho, Elza Soares, Mart'nália, Teresa Cristina, Mariene de Castro, Teresa Sandoval, Elba Ramalho, Carmen Miranda, Clara Nunes, Milton Edil Pacheco, João Donato, Jorge Ben Jor, Tim Maia, Wilson Simonal, Jair Rodrigues, Nelson Gonçalves, Jamelão, Claudette Soares, Paulinho da Viola, Marcos Valle, Ed Motta, entre outras.

por usar fontes externas de conhecimento, similar ao executado por sistemas de recuperação de informação. Ainda assim, percebemos que a lista contém compositoras que não são de samba.

Quadro 20.5. Resposta do BARD ao *prompt* “Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba.”. Como a resposta ficou grande para caber bem em uma figura, transcrevemos a resposta aqui.

De acordo com a Wikipedia em português, as compositoras brasileiras de samba são:

- Chiquinha Gonzaga (1847-1935)
- Dona Ivone Lara (1921-2018)
- Beth Carvalho (1946-2022)
- Alcione (1947-)
- Clara Nunes (1942-1983)
- Mart'nália (1967-)
- Teresa Cristina (1967-)
- Ana Costa (1968-)
- Roberta Sá (1972-)
- Maria Gadú (1987-)
- Liniker (1995-)



- Luedji Luna (1995-)
- Alice Caymmi (1990-)
- Larissa Luz (1990-)
- Duda Beat (1992-)
- Iza (1993-)

Essa lista inclui compositoras de diferentes gerações, estilos e regiões do Brasil. Chiquinha Gonzaga é considerada a primeira compositora brasileira de samba, e Dona Ivone Lara é uma das mais importantes compositoras do gênero. Beth Carvalho, Alcione e Clara Nunes são cantoras e compositoras que popularizaram o samba no Brasil e no exterior. Mart'nália, Teresa Cristina, Ana Costa, Roberta Sá, Maria Gadú, Liniker, Luedji Luna, Alice Caymmi, Larissa Luz, Duda Beat e Iza são compositoras de samba contemporâneas, que estão contribuindo para a renovação do gênero.

É importante ressaltar que essa lista não é exaustiva, e há muitas outras compositoras brasileiras de samba que poderiam ser mencionadas.

Entretanto, o diálogo que segue demonstra a fragilidade de tais sistemas em se aterem a suas respostas, mesmo quando corretas, conforme a continuação do mesmo diálogo (Quadro 20.6)⁴⁰. Mesmo que a lista original seja composta de apenas mulheres, quando confrontado com o *prompt* “Eu pedi compositoras. Na sua lista há apenas mulheres?”, ele pede desculpas pela resposta, e devolve uma lista de compositores (que não foi pedida)!

Quadro 20.6. Segunda iteração do diálogo iniciado no Quadro 20.5. Resposta do BARD ao *prompt* “Eu pedi compositoras. Na sua lista há apenas mulheres?”. Como a resposta ficou grande para caber bem em uma figura, transcrevemos a resposta aqui.

Desculpe, eu interpretei mal o seu pedido. A lista que eu te passei inclui apenas mulheres, mas também existem homens compositores de samba. Alguns exemplos incluem:

- Cartola
- Paulinho da Viola
- Zé Keti
- João Nogueira
- Wilson Simonal
- Jorge Ben Jor
- Chico Buarque
- Caetano Veloso
- Milton Nascimento
- Se você quiser uma lista mais completa, posso te passar uma que inclui compositores de ambos os sexos.

20.6 Conclusões provisórias

Neste capítulo, arriscamos um retrato dos agentes de conversação baseados em LLMs em 2023. Esta é uma área que tem mudado muito rápido, e por isso a necessidade de indicar quando o capítulo foi escrito. Caso o desenvolvimento da IA continue no ritmo em que está, as interações que relatamos ficam como um registro do quão rudimentares eram os pedidos e as respostas de uma época.

Ainda que existam situações que geram respostas aceitáveis, destacamos que, como estão hoje, estas ferramentas têm melhor aproveitamento se vistas como **assistentes** (“assistentes

⁴⁰Você também pode ver a resposta em <https://g.co/bard/share/f7b5eefcbfec>.



aprendizes”, como pessoas estagiárias de uma área), e **não como oráculos**⁴¹ (entendendo “oráculo” como a divindade capaz de fornecer respostas infalíveis). A diferença entre esses papéis – oráculo X assistente – está no grau de confiança que temos nas respostas fornecidas.

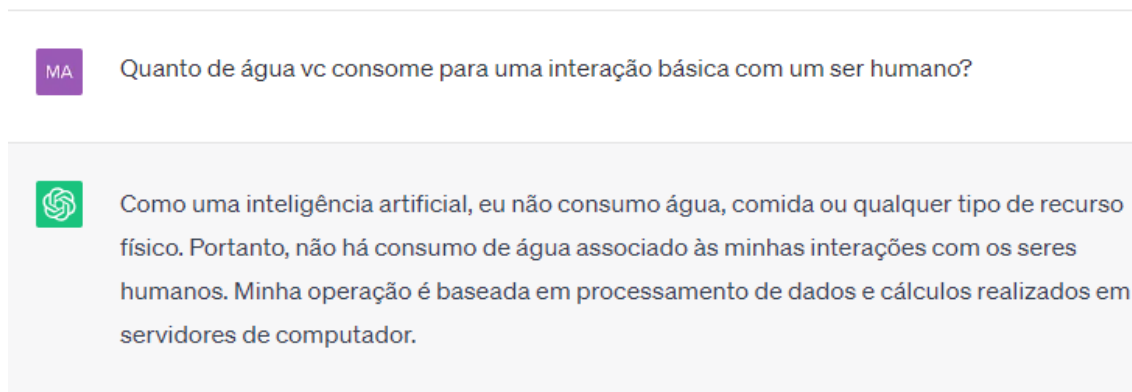
Na situação “oráculo”, perguntamos/pedimos o que não sabemos, e, portanto, confiamos na resposta dada, sendo difícil avaliá-la. Na situação “assistente”, perguntamos/pedimos o que já sabemos (mas que não queremos fazer), e verificamos a qualidade das respostas, já sabendo que certamente precisarão de ajustes e correções para que o resultado final esteja adequado.

Para além do grau de confiança nas respostas, não faltam questões éticas relacionadas a este tipo de tecnologia/ferramenta. Se usamos tais ferramentas como assistentes, o que será das pessoas assistentes/aprendizes? Como então iremos aprender coisas e/ou formar pessoas? Serão as máquinas responsáveis por isso? E quem ensina as máquinas⁴²? Quais as implicações para o ensino? Como lidar com direitos autorais? Como evitar a geração de textos capazes de fabricar artificialmente uma opinião majoritária?

Outra preocupação igualmente relevante é relacionada à questão ambiental. Já sabemos que o consumo de CO₂ e de água⁴³ necessários para o treinamento dos modelos de linguagem gerativos é imenso. Estima-se, por exemplo, que a quantidade de água doce limpa necessária para treinar o GPT-3 foi equivalente à quantidade necessária para encher a torre de resfriamento de um reator nuclear (Li et al., 2023a)⁴⁴.

E o que dizem os agentes de conversação a esse respeito (Figura 20.22)?

Figura 20.22: Resposta do ChatGPT para uma pergunta relativa ao seu consumo de água.



Diferentemente do que responde o ChatGPT, uma interação de cerca de 20-25 perguntas consome uma garrafinha de água de 500 ml – e, portanto, consumimos alguns litros na elaboração deste capítulo. Vale a pena? Quando vale a pena? Em que circunstâncias é aceitável este tipo de gasto? Alguma outra ferramenta desenvolvida de forma realmente

⁴¹A palestra ChatGPT: O que é? De onde veio? Para onde vamos?, do grupo Brasileiras em PLN (https://youtu.be/F8yxBbx8woU?si=MfcBoV_tWhde-njy), embora tenha exemplos de uso do ChatGPT que já ficaram obsoletos, explora alguns limites do uso desses agentes como oráculo, além de fazer uma apresentação de como esses modelos como GPT são criados.

⁴²Veja-se por exemplo <http://www.uol.com.br/tilt/reportagens-especiais/a-vida-dura-de-quem-treina-inteligencias-artificiais/> e Seção 14.4.5

⁴³Água doce é necessária para resfriar os super-processadores.

⁴⁴<https://oglobo.globo.com/economia/tecnologia/noticia/2023/05/treino-do-chatgpt-consumiu-700-mil-litros-de-agua-equivalente-a-encher-uma-torre-de-resfriamento-de-um-reator-nuclear.ghtml> ou <https://www.printfriendly.com/p/g/D56wCg>



responsável e consciente e tomará o seu lugar?

Na próxima versão deste livro, veremos que rumos tomaram os agentes de conversação baseados em modelos de linguagem.



<https://brasileiraspln.com/livro-pln/1a-edicao/>



Parte IX

Domínios



Capítulo 21

PLN na Saúde

*Adriana Pagano
Claudia Moro
Elisa Terumi Rubel Schneider
Lilian Mie Mukai Cintho
Yohan Gumiel*

21.1 Introdução

A área da saúde é uma das mais importantes em nossas vidas e, nos últimos anos, tem se beneficiado do uso da tecnologia para melhorar o diagnóstico, o tratamento e a gestão de pacientes. A aplicação de Processamento de Linguagem Natural (PLN) tem sido fundamental para avançar nessa área, pois permite a análise de grandes volumes de dados não estruturados gerados em ambientes clínicos (Turchioe et al., 2022).

O domínio da medicina abrange diversos tipos de texto, utilizados para distintas atividades produtoras de significado, que desenvolvemos em nosso convívio social. Chamamos essas atividades de socio-semióticas. Estudos da linguagem baseados em pesquisas antropológicas modelam essas atividades socio-semióticas em oito tipos (Matthiessen, 2013; Matthiessen; Teruya; Wu, 2008).

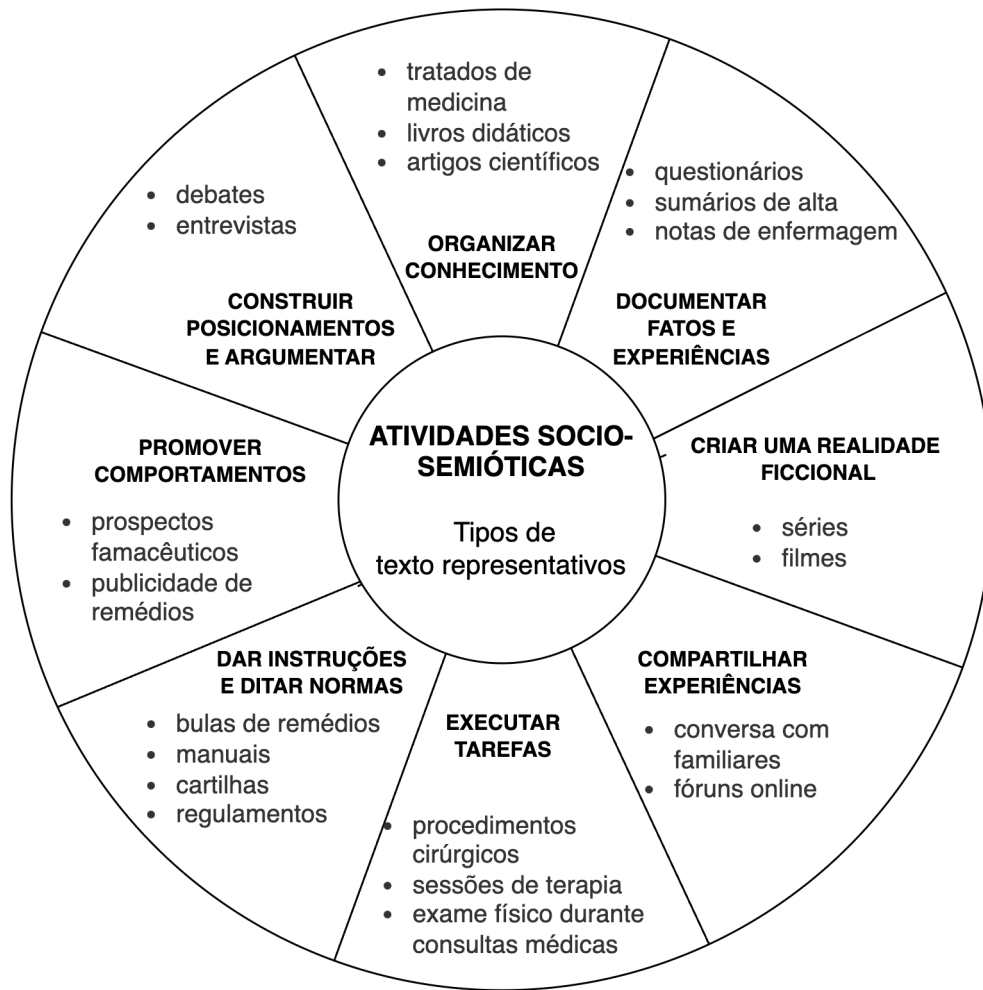
A Figura 21.1 mostra os oito tipos de atividades socio-semióticas e os tipos de texto mais representativos de cada um deles no domínio da medicina. Essas atividades são desenvolvidas por meio de textos escritos e falados, com funções específicas na nossa sociedade. Atividades nas quais a linguagem verbal tem um papel ancilar ou complementar são, por exemplo, a execução de procedimentos cirúrgicos, durante a qual ações podem ser verbalizadas ou não.

Mas, na grande parte das atividades humanas, a linguagem tem um papel constitutivo. Temos desde atividades que envolvem um uso especializado da linguagem para organizar a produção de conhecimento em tratados de medicina, livros didáticos e artigos acadêmicos, até atividades que envolvem um uso menos especializado, como o compartilhamento de experiências no âmbito privado, nas interações entre pacientes e familiares ou entre participantes de fóruns online sobre cuidados em saúde. Para a atividade de instruir e regular o comportamento, temos textos como bulas de medicamentos, cartilhas, normativas, manuais de instrução de equipamentos. Mesmo no domínio da medicina, há também textos pelos quais é construída uma realidade ficcional, como é o caso de séries e filmes que recriam interações em contextos médicos.

Uma atividade socio-semiótica muito relevante no domínio da medicina é documentar fatos e experiências, por meio de questionários aplicados ao paciente, registros de exames clínicos e relatos de profissionais da saúde, nos quais são documentadas percepções sobre a saúde do paciente. Esses textos são conhecidos em PLN como **narrativas clínicas** e



Figura 21.1: Tipos de texto no domínio da medicina



abrangem notas de evolução de enfermagem, sumários de alta, boletins médicos, e notas em texto livre em campos próprios do prontuário eletrônico do paciente. Cada um desses tipos de texto pode oferecer informações valiosas a serem obtidas por meio do PLN mais adequado às características do texto. Artigos acadêmicos, por exemplo, podem ser usados para a extração de ontologias, que são estruturas semânticas que permitem uma representação formal de conceitos, suas propriedades e relações. Essas ontologias podem ser usadas para facilitar a compreensão de termos técnicos e complexos em diferentes áreas da saúde, permitindo que as informações sejam compartilhadas de forma mais clara e precisa (Jiang et al., 2020). Também podemos identificar padrões e relacionamentos entre os dados e a construção de modelos preditivos (Lee et al., 2019).

Narrativas clínicas, por outro lado, são textos não estruturados que oferecem informações valiosas sobre a história do paciente, incluindo seus sintomas, histórico médico, estilo de vida e outras informações relevantes. A mineração desses dados pode ser usada para identificar padrões e relacionamentos entre os dados, permitindo uma melhor compreensão da condição do paciente e a construção de modelos preditivos para prever possíveis complicações ou doenças (Wu et al., 2018).









21.2 O texto livre em narrativas clínicas

Com o advento do Registro Eletrônico de Saúde (RES)¹, como é denominado no Brasil, ou em inglês, o *Electronic Health Record* (EHR), a quantidade de dados gerados relativos à atenção aos pacientes aumentou significativamente. Os prontuários eletrônicos podem conter dados estruturados, semiestruturados ou não estruturados, todos eles oferecendo uma grande quantidade de informações sobre o paciente. A mineração desses dados pode ajudar a identificar tendências e padrões em relação a diagnósticos, tratamentos e resultados, permitindo uma melhor gestão do cuidado do paciente e um melhor planejamento da assistência (Shickel et al., 2017).

Os dados clínicos presentes nas narrativas clínicas em texto livre (dados não estruturados) apresentam características únicas que dificultam sua análise e interpretação. Esses dados são frequentemente apresentados em linguagem médica especializada, repleta de termos técnicos, jargões e abreviaturas que podem variar entre os distintos profissionais de saúde. Esses textos também podem conter erros de digitação, ortografia ou gramática, tornando a interpretação ainda mais complexa (Dalianis, 2018). A Figura 21.2 apresenta um exemplo de narrativa clínica adaptada para fins de ilustração. Nela podemos observar que as informações podem ser estruturadas de acordo com categorias destacadas com cores e rotuladas na legenda da figura.

Figura 21.2: Exemplo de narrativa clínica elaborada para fins de ilustração. Na legenda, as categorias de informações que podem ser encontradas neste tipo de texto.

AP: HAS . Obesidade . DM. Tabagista.	Refere HAS desde 21 anos de idade, irregular controle da pressão arterial.
Ao exame: BEG, CHAAA, eupneico. ACV: BRNF, sem sopro. PA 159X100 mmHg, FC 89. AR: MV+ s/ RA . ABD : flácido , indolor a palpação , sem visceromegalias . MMII : edema bilateral ++/+4 .	
USG rins normal . Doppler MMII normais . ECG : BAV 1º G , BRD . TE Externo : baixa resposta cronotrópica , EEVV polimórficas frequentes , comportamento normal de PA . ECO TT Ext 19/06/20: AE 28 Septo 12 Parede Post 12 Massa VE 218 FEVE 70 % VE 48x29 Hipertrofia simétrica discreta do VE . Alteração do relaxamento do VE .	
Laboratório : 22/09/21: Hb:14,3 Leuco:4.960 PLaq:204mil Colesterol:169 HDL:49 LDL:102 Triglic:91 Glicose:96 Ur:40 Cr:0,93 Na:142 K:4,3 TSH:2,08	
Oriento MEV , Restrição Hidrossalina . Oriento importância da adesão medicamentosa . Solicito ECO TT .	
Prescrevi : sinvastatina 40 mg comp . (1 comp vo 1xd) ; ácido acetilsalicílico 100 mg comp (1 comp vo 1xd) ; omeprazol 20 mg cáps; losartana 50 mg comp (1 comp vo 12/12h) ;	
Solicitei : Ecocardiograma com doppler colorido adquiridas ; hemoglobina glicada ; ácido úrico ; colesterol (sangue) ; CPK; creatinina / soro ; glicose / soro ; TGO	

Legenda	
 Antecedentes pessoais	 Exames laboratoriais
 História pregressa	 Orientações
 Exame físico e sinais vitais	 Medicamentos prescritos
 Exames realizados	 Planejamento

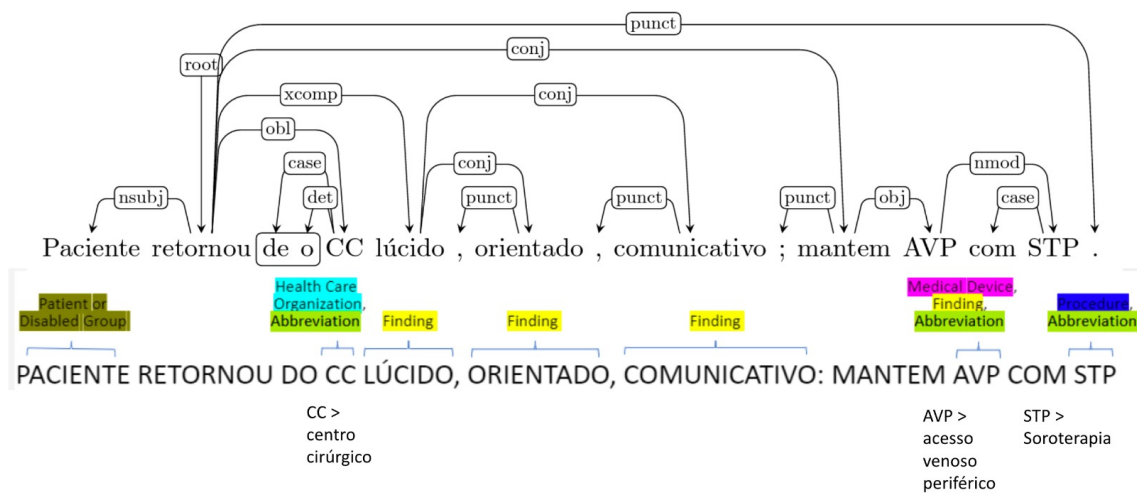
No escopo do que chamamos narrativas clínicas, há diferentes tipos de texto, os quais apresentam desafios específicos em termos do tipo de linguagem e também da relevância das informações registradas. Por exemplo, as notas de evolução de enfermagem podem ser mais descritivas e detalhadas do que outros tipos de texto, enquanto os sumários de alta podem fornecer informações importantes sobre a condição atual do paciente e seu histórico

¹No Sistema Único de Saúde (SUS), as informações dos usuários são coletadas e armazenadas por meio do Prontuário Eletrônico do Cidadão (PEC). Nele, há campos pré-determinados que podem ser preenchidos com texto livre.

de tratamento. Já as notas de ambulatório podem ser mais informais e fragmentadas, o que dificulta sua análise por modelos treinados com outros tipos de texto em outros domínios. Isso demanda a anotação manual de narrativas clínicas de forma contarmos com modelos mais refinados.

Como todo processo manual, a anotação de narrativas clínicas requer tempo e recursos, o que dificulta a construção de grandes *datasets* para treinamento de modelos de PLN. Como resultado, a aplicação de técnicas de aprendizado de máquina em dados clínicos sofre limitações pela disponibilidade de dados anotados manualmente (Koleck et al., 2019). Uma saída é utilizar modelos genéricos para pré-processamento, sendo a saída avaliada manualmente. Um exemplo deste tipo de trabalho é a anotação do *corpus* Depclin-Br, que vem sendo desenvolvida por uma equipe de cientistas da computação da PUCPR e de linguistas da Faculdade de Letras da UFMG. Trata-se de um conjunto de narrativas clínicas já anotadas em termos de entidades no domínio clínico e constituindo o *corpus* SemClinBr (Oliveira et al., 2022a). Uma parte desse *corpus* foi anotada morfossintaticamente com base num modelo genérico de português e a anotação revisada manualmente (Oliveira et al., 2022b). Essa primeira parte foi utilizada para refinamento do modelo genérico e anotação automática de um segunda parte do *corpus*. Uma vez concluída a anotação, dados do *corpus* DepClinBr, anotado com relações de dependência, podem ser minerados e utilizados para caracterizar as entidades nomeadas previamente anotadas no SemClinBr. A Figura 21.3 ilustra a correlação de anotações morfossintáticas e entidades.

Figura 21.3: Correlação de anotações morfossintáticas e entidades.



A construção de *corpora* de narrativas clínicas (dados não estruturados) está sujeita a restrições técnicas e regulatórias, que dizem respeito à privacidade de dados. Essa especificidade limita a capacidade de construção de grandes *datasets* para treinamento de modelos de PLN (Chen; Chen, 2022). Como foi apontado, para contornar essa limitação, são utilizados modelos genéricos da língua, os quais precisam ser refinados com dados específicos do domínio em um processo de *fine-tuning*, para melhorar ainda mais sua precisão e relevância (Lee et al., 2019).

A seguir, veremos alguns exemplos de aplicações da PLN em dados clínicos.

21.3 Aplicações de PLN na Saúde

21.3.1 Predição

Uma das principais tarefas de PLN na área médica é a predição, que pode ser aplicada em diversas demandas do cuidado em saúde, como diagnóstico, tratamento, evolução, alta médica hospitalar, detecção de quedas, detecção de depressão e outras. Essas demandas envolvem a classificação de dados clínicos, como narrativas de pacientes, prontuários eletrônicos, relatórios médicos e outros dados de saúde, para ajudar os médicos e outros profissionais de saúde a tomar decisões mais precisas. A predição de diagnóstico, por exemplo, pode ajudar a identificar doenças em estágios iniciais, permitindo tratamentos mais eficazes e prevenindo complicações. A predição de tratamento pode ajudar a personalizar o tratamento para cada paciente, maximizando sua eficácia e minimizando efeitos colaterais. A detecção de quedas e depressão pode ajudar a prevenir acidentes e melhorar a qualidade de vida dos pacientes. Em resumo, a tarefa de predição é essencial para a aplicação bem-sucedida de PLN na área de saúde (Yan; Gustad; Nytrø, 2022).

Alguns exemplos de trabalhos envolvendo predição e classificação em textos clínicos em português são (Gonçalves et al., 2023; Santos; Ulbrich; Vieira, 2021; Silva et al., 2023; Yang et al., 2022).

21.3.2 Desidentificação

Um aspecto crucial na aplicação de PLN na área médica é a desidentificação dos dados dos pacientes, associada a processos de anonimização ou pseudonimização. Esta envolve a remoção de informações que possam identificar o paciente, como nome, endereço, número de telefone e outras informações pessoais. A anonimização é necessária para garantir a privacidade dos pacientes e cumprir as regulamentações de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil² e a General Data Protection Regulation (GDPR) na União Europeia³.

A anonimização de dados clínicos é um processo desafiador, uma vez que esses dados contêm informações altamente sensíveis e complexas, como histórico médico, sintomas, exames, tratamentos e outros detalhes que podem identificar um paciente. Portanto, é necessário utilizar técnicas avançadas de PLN, como o uso de modelos de linguagem, para remover as informações sensíveis e garantir a privacidade dos pacientes (Jones et al., 2020).

Existem diversas técnicas que podem ser utilizadas na desidentificação dos dados clínicos, dependendo do tipo de informação que deve ser removida e do nível de anonimização desejado, por exemplo:

- Substituição de nomes próprios e outros identificadores pessoais por símbolos ou pseudônimos aleatórios;
- Remoção de informações geográficas específicas, como endereço e CEP;
- Substituição de datas de nascimento e outras informações temporais por intervalos ou idades aproximadas;
- Remoção de informações de contato, como números de telefone e endereços de e-mail;

²Lei Geral de Proteção de Dados Pessoais (LGPD), Lei nº 13.709/2018. Disponível em: <https://www.gov.br/pt-br/acesso-a-informacao/lgpd>

³Data protection in the EU. Disponível em: https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en



- Remoção de informações de identificação de instituições, como o nome de hospitais e clínicas.

Além dessas técnicas, também é possível utilizar métodos mais avançados de PLN, como a detecção e remoção de termos médicos específicos ou o uso de técnicas de de-identificação baseadas em modelos de linguagem, que tentam preservar a integridade semântica dos dados, mesmo após a remoção ou substituição das informações pessoais.

A desidentificação dos pacientes permite que os dados clínicos sejam utilizados para fins de pesquisa e análise, sem comprometer a privacidade dos pacientes. Isso é fundamental no avanço da medicina, permitindo a análise de grandes volumes de dados na descoberta de padrões e tendências em doenças, tratamentos e outros aspectos da saúde (Liu et al., 2017). Em (Santos et al., 2021) temos um exemplo de trabalho para o português nessa tarefa.

21.3.3 Extração de conceitos clínicos

A busca e extração de conceitos clínicos relevantes é uma tarefa essencial na aplicação de PLN na área médica. Essa tarefa envolve a identificação de entidades relevantes nos dados clínicos, como sintomas, diagnósticos, tratamentos, medicamentos e outros termos específicos da área da saúde. Essa identificação geralmente é feita por meio de técnicas de NER (do inglês, *Named Entity Recognition*) (Capítulo 17), que permitem a identificação e classificação automática de entidades em textos não estruturados. A Figura 21.4 ilustra um exemplo de entidades do tipo *Problema* reconhecidas em uma narrativa clínica elaborada para fins de ilustração.

Figura 21.4: Exemplo de entidades do tipo Problema (em azul) encontradas em narrativa clínica.

Extração de conceitos clínicos: problemas

Paciente **DM tipo 2**, **HAS** e **obesidade**, com queixa de **descontrole glicêmico** necessitando procurar o PS 3x em 2 semanas. Refere que não consegue perder peso mesmo realizando dieta alimentar e caminhadas e faz uso regular dos medicamentos prescritos. Relata que apresenta **disúria** e **hematúria** há 2 dias. Apresenta PA 120X80 mmHg, **sem edemas de MMII**. Faz uso de atenolol 25 mg 2X/dia, metformina 850 mg 2X/dia. Solicitado exames laboratoriais (hemoglobina glicada, colesterol total, LDL e triglicérides e exame de Urina I). Aumento metformina 850 mg para 3X/dia e prescrevo Ciprofloxacino 250 mg cada 12 horas, durante 7 dias. Oriento a manter os medicamentos em uso e à modificar o estilo de vida e retornar para acompanhamento ambulatorial.

Além da identificação de entidades, outras técnicas de PLN também podem ser utilizadas para a busca e extração de conceitos clínicos relevantes, como a detecção de negação e a resolução de ambiguidades. A detecção de negação, por exemplo, é útil para identificar quando um sintoma é negado pelo paciente ou um diagnóstico dado pelo médico nega alguma condição. A precisão na detecção de negação é fundamental para a interpretação dos dados clínicos (Nath; Lee; Lee, 2022).

Outra técnica importante na busca e extração de conceitos clínicos é o mapeamento de terminologia, que consiste na associação dos termos clínicos encontrados nos textos com um conjunto de termos padronizados, como a Classificação Internacional de Doenças (CID) ou o *Systemized Nomenclature of Medicine* (SNOMED CT). Isso permite uma melhor



organização e interpretação dos dados clínicos, facilitando a análise e a tomada de decisão médica (Fennelly et al., 2021).

A busca e extração de conceitos clínicos relevantes é fundamental para a análise de dados clínicos em larga escala, permitindo a identificação de padrões e tendências em doenças, tratamentos e outros aspectos da saúde. Além disso, essas técnicas de PLN também podem ser utilizadas para a construção de sistemas de suporte à decisão médica, que auxiliam os profissionais de saúde na escolha de tratamentos mais adequados para cada paciente (Demner-Fushman; Chapman; McDonald, 2009).

21.3.4 Relações temporais

Uma linha do tempo do paciente é uma representação gráfica que organiza as informações clínicas de um paciente de maneira cronológica. O interesse pela pesquisa em extração de relações temporais provém da característica longitudinal dos dados presentes nos Registros Eletrônicos de Saúde. Esses registros contêm múltiplos textos clínicos referentes ao mesmo paciente, escritos em diferentes momentos (Gumiel et al., 2021).

A extração de relações temporais concentra-se na organização sequencial de menções em um texto, sendo essas menções eventos médicos ou expressões temporais.

No contexto clínico, eventos médicos são circunstâncias clínicas de relevância, cujo escopo é delimitado pelo contexto da aplicação. Por exemplo, para a extração de informações significativas para o diagnóstico, pode ser apropriado delimitar eventos como menções a tratamentos passados, sinais, sintomas, medicamentos em uso e exames realizados pelo paciente com os respectivos resultados. Já as expressões temporais envolvem menções de tempo, como a duração de um sintoma ou indicações de quando o paciente realizou determinada cirurgia. É notável que as expressões temporais só têm significado quando associadas a algum evento, enquanto os eventos podem fazer sentido quando relacionados entre si.

A fim de extrair essas menções do texto, são empregadas técnicas de Processamento de Linguagem Natural (PLN), como a Reconhecimento de Entidades Nomeadas. A tarefa de NER consiste em identificar e classificar automaticamente eventos e expressões temporais.

Com eventos e expressões temporais devidamente identificados, aplica-se a extração de relações temporais, uma técnica de PLN que se concentra na conexão de eventos entre si ou com expressões temporais. Desse modo, cada entidade acaba sendo relacionada a um período de tempo específico.

Ao considerar relações temporais no contexto clínico, diversas áreas de pesquisa emergem. Doenças crônicas, por exemplo, apresentam uma natureza longitudinal que torna a temporalidade extremamente relevante, pois existem fluxos de dados do paciente contínuos e extensos, nos quais podem ser extraídos padrões significativos (Sheikhalishahi et al., 2019). A progressão de uma doença e os eventos a ela associados são registrados cronologicamente, onde certos eventos são relevantes apenas em momentos específicos, como problemas médicos identificados durante um exame físico em uma consulta ambulatorial ou sintomas relatados (Sheikhalishahi et al., 2019). No caso de tratamento ineficaz de hipertensão com monoterapia, por exemplo, busca-se terapias com medicamentos combinados. Portanto, algumas informações sobre a progressão de doenças podem ser mais facilmente discernidas por meio da extração de relações temporais (Gumiel et al., 2021).

A aplicação prática de uma linha do tempo na área da saúde pode ser utilizada para analisar a evolução do quadro clínico do paciente ao longo do tempo, identificar possíveis tendências e realizar previsões. Além disso, a linha do tempo do paciente pode ser integrada



a sistemas de suporte à decisão médica, contribuindo para a seleção de tratamentos mais adequados para cada paciente.

21.3.5 Sumarização

A sumarização de evoluções clínicas é uma tarefa de PLN que tem como objetivo extrair as informações mais relevantes de um conjunto de dados clínicos, de forma a produzir uma versão resumida e legível dessas informações. A Figura 21.5 exibe um exemplo fictício de uma narrativa clínica sumarizada.

Figura 21.5: Exemplo fictício de uma narrativa clínica sumarizada, na qual as informações mais importantes foram mantidas.

Exemplo de Sumarização

Paciente hipertenso, dislipidêmico e sobrepeso, internou por 2 dias na clínica médica devido à crise hipertensiva e queixa de formigamento no MSD. Solicitado exames laboratoriais, ECG e cateterismo cardíaco. Sem apresentação de alterações enzimáticas e do ECG. Cateterismo sem lesões obstrutivas graves na CD, DP e VP, TCE sem lesões, DA e Dg1 sem lesões, CX, Mg1 e VPE sem lesões. Medicado e manteve a pressão arterial estável e sem queixas de dor. Mantido os medicamentos em uso: Atenolol 100 mg 1x/dia, Anlodipino 5mg 2xd e Atorvastatina 20mg a noite. Recebe alta hospitalar com a pressão arterial controlada e sem sintomas para dor. Oriento para a adesão medicamentosa, mudança de estilo de vida e acompanhamento ambulatorial.

Para realizar a sumarização de evoluções clínicas, são utilizadas técnicas de sumarização automática de texto, que podem ser baseadas em abordagens extrativas ou abstrativas⁴.

Na abordagem extrativa, as frases mais importantes do texto original são selecionadas e combinadas para formar um resumo. Já na abordagem abstrativa, o resumo é gerado a partir da síntese das informações do texto original, gerando uma nova versão que não necessariamente contém as mesmas palavras e frases do texto original.

Para realizar a sumarização de evoluções clínicas, são utilizadas técnicas de processamento de linguagem natural, incluindo NER para identificar as entidades relevantes, PoS (*Part-of-Speech*) para identificar as partes do discurso e gramática do texto e também técnicas de análise sintática e semântica.

Essa tarefa de PLN é muito útil para os profissionais da área da saúde, pois permite que eles analisem brevemente as informações mais importantes dos pacientes, como histórico de doenças, exames realizados, tratamentos prescritos, entre outras informações clínicas (Gulden et al., 2019).

21.4 Para onde estamos caminhando?

Embora a tecnologia de PLN na área clínica tenha avançado significativamente nos últimos anos, ainda existem vários desafios a serem superados. Alguns desses desafios incluem:

- Garantir a qualidade dos dados clínicos utilizados para treinar e testar os modelos de PLN, incluindo a devida anonimização e a padronização dos termos utilizados, assegurando a ética e a privacidade dos dados clínicos;

⁴Para projetos de sumarizadores em português, visite: <https://sites.icmc.usp.br/taspardo/sucinto/>



- Desenvolver modelos de PLN capazes de lidar com textos clínicos mais complexos e heterogêneos, como notas de enfermagem, laudos médicos e textos escritos por pacientes;
- Integrar os modelos de PLN em sistemas de informação em saúde existentes, garantindo a interoperabilidade e a segurança dos dados;
- Garantir a aceitação e a adoção dos modelos de PLN pelos profissionais de saúde, demonstrando sua utilidade e eficácia na prática clínica.

É importante destacar que, embora o PLN possa ser útil na análise e interpretação de dados clínicos, ele não pode substituir a experiência e o conhecimento clínico de um médico ou de outros profissionais de saúde. A tecnologia pode ser uma ferramenta valiosa para auxiliar na tomada de decisões clínicas, mas não pode substituir o julgamento clínico humano. Ressalta-se que o desenvolvimento de tecnologias de PLN na área clínica seja visto como uma forma de complementar e melhorar o cuidado ao paciente, e não como uma substituição aos profissionais de saúde.



Capítulo 22

PLN no Direito

Maria José Bocorny Finatto

Aline Macohin

22.1 Introdução

Neste capítulo, tratamos de diferentes aspectos associados ao trabalho computacional com textos produzidos na esfera do Direito. As tarefas de PLN envolvidas, em geral, são a análise textual e a representação de conteúdos por meio de diferentes técnicas, mas há várias abordagens e estudos, voltados para diferentes finalidades. O nosso objetivo é apresentar apenas algumas perspectivas e desafios no âmbito de trabalhos que exploram materiais produzidos em português, considerando somente o cenário do Direito Brasileiro. Afinal, o Direito, de país para país, tem especificidades linguísticas e culturais que repercutem muito sobre seus textos, discursos e tipo de vocabulário.

Por isso, iniciamos o capítulo apresentando alguns aspectos sócio-históricos do Direito Brasileiro, que acabam influenciando suas práticas de escrita e os seus conteúdos textuais. Em seguida, situamos exemplos de reconhecimento e exploração do vocabulário jurídico, dos seus modos de dizer e, especialmente, das suas terminologias. Vamos partir de dois diferentes cenários textuais: as leis e sentenças judiciais. A primeira parte de exemplos tem a ver com um trabalho que denominamos reconhecimento terminológico (RT). Esse trabalho, atualmente, é baseado em fontes escritas disponíveis em formato digital e se beneficia muito das técnicas da Linguística de *Corpus* (Sardinha, 2000) e do PLN. Depois dessa parte, mais dedicada ao vocabulário e terminologias, segue um exemplo de estudo em PLN, na área conhecida como **Análise de Sentimentos** (Capítulo 23).

O território de materiais para estudo e de enfoques, em Direito, é extremamente amplo, isso se ficarmos restritos aos trabalhos que lidam com os textos jurídicos da atualidade. Ainda assim, vale mencionar que uma série de estudos históricos sobre a linguagem jurídica brasileira, tratando de seus conceitos e até preconceitos, têm sido muito úteis para uma crítica social e política sobre o Direito. Para esses estudos históricos, os processos sobre crimes no período colonial e do império, reunidos em *corpora* que se exploram com apoio computacional, têm mostrado a importância de se fazer uma linha de tempo de ações e de entendimentos até os dias de hoje. No Brasil, temos já, por exemplo, diferentes pesquisas filológicas e linguísticas dedicadas a estudos de processos criminais dos séculos 17, 18 e 19.

Entretanto, para se trabalhar com textos antigos em português, há todo o processo de *normalizar* e padronizar a apresentação escrita das “palavras antigas”, para então podermos fazer o seu processamento. A normalização de textos é, assim, um desafio multidisciplinar de uma nova área de estudos denominada *Humanidades Digitais* e que inclui o PLN no tratamento de acervos antigos.

Conforme nos coloca o artigo de Cameron; Olival; Vieira (2023), os desafios são muitos.



Afinal, geralmente trabalha-se com textos em forma de arquivo provenientes de manuscritos que foram “decifrados” e transcritos. Isso significa enfrentar muitas questões associadas à variabilidade da escrita. Afinal, uma mesma palavra podia apresentar-se de vários modos, em um mesmo documento, escrito por uma mesma pessoa, como nos casos de ÁGUA/AGUA/AGOA ou UMA/HUA/HUMA. Há exemplos interessantes desses tipo de estudo histórico, no âmbito do Direito Penal e da Medicina Legal, com processos judiciais que envolveram crimes contra mulheres no Brasil do século 19. O artigo de Teixeira; Marengo; Finatto (2022) ilustra um exemplo de estudo bem interessante nesse tema da violência contra as mulheres.

Mas, voltando à atualidade dos textos e da linguagem do Direito, veremos, mais adiante, como exemplos, alguns textos jurídicos brasileiros, buscando ilustrar suas peculiaridades. Vamos destacar: **a)** o texto do Estatuto da Criança e do Adolescente (ECA), conforme apresentado na Lei 8.069-90, promulgada em 13 de julho de 1990 e atualizada em 2021; **b)** o texto da nossa Constituição do Brasil, de 1988 (CF88); e, **c)** um conjunto de Sentenças Judiciais dos chamados “tribunais de pequenas causas”, os Juizados Especiais Cíveis. Conforme pretendemos deixar claro, esses três tipos de fontes, em suas características linguísticas e textuais, podem estar associados a diferentes tarefas de PLN, desde a descrição do português até a pontos específicos de *Recuperação da Informação*, área conhecida como *Information Retrieval* (Capítulo 16).

Por isso, um outro exemplo que trazemos neste capítulo é o da análise de conteúdos em sentenças judiciais via *Análise de Sentimentos*. Este tipo de técnica pode ser muito útil para identificar, por exemplo, padrões de sentenças judiciais favoráveis ou desfavoráveis a um determinado assunto. A utilidade dessa técnica para os profissionais do Direito é grande, pois um profissional geralmente faz buscas para entender como um determinado tribunal já vem decidindo sobre um assunto específico. Nesse trabalho de pesquisa, são buscadas retornadas inúmeras sentenças e documentos. Vale destacar que existem tribunais no Brasil inteiro, que lidam diversos assuntos (trabalhistas, penais, civis, eleitorais, entre outros). Nesses órgãos são protocolados milhares de novos processos diariamente e neles existe uma base de milhões de processos já julgados, muitos já em formato digital. A criação de um método que possa filtrar, por exemplo, as causas que foram consideradas favoráveis, em um dado tema, tende a reduzir o trabalho de leitura individual de cada sentença, ajudando o profissional a buscar e encontrar a informação que precisa.

22.2 O Direito – uma moldura para a significação

Conforme já mostraram os estudos (Motta, 2021, 2022), o Direito se manifesta através da língua, pois são as palavras que emprega e os enunciados que produz que conferem e confirmam a sua existência peculiar (Maciel, 2001) como uma prática social e área de conhecimento. Assim, temos uma relação intensa entre o Direito e a língua em uso pelas pessoas que nele atuam. Isto é, pelo emprego de certas palavras¹, com um sentido particular e pela forma como suas proposições e teses são enunciadas vemos todo um cenário de valores. Isso é tão importante que temos uma área de estudos específica conhecida como *jurilinguística* (veja mais em Cornu (1990)).

Estudiosos dessa área da linguística (Montoro, 1998, p. 1998) explicam que a linguagem jurídica, sempre com destaque para escrita, compreende diversas “espécies” de práticas que

¹Veja mais sobre a delimitação de “palavras” e de unidades de processamento no Capítulo 4. Neste capítulo sobre Direito e PLN, vamos acentuar a noção de palavra como uma unidade da língua escrita, situada entre dois espaços em branco, ou entre espaço em branco e sinal de pontuação.



se subdividem, conforme uma dada finalidade e foco. Vejamos um detalhamento dessas espécies ou modos de se apresentar conforme seus propósitos (Petri, 2017, p. 47):

1. linguagem legislativa – a linguagem dos códigos - como o Código Penal ou Civil, das normas; sua finalidade: criar o Direito;
2. linguagem judiciária, forense ou processual - é a linguagem dos processos e sentenças; sua finalidade: aplicar o Direito;
3. linguagem convencional ou contratual – é a linguagem dos contratos, por meio dos quais se criam direitos e obrigações entre as partes;
4. linguagem doutrinária – é a linguagem dos mestres, dos doutrinadores, cuja finalidade é explicar os institutos jurídicos, é ensinar o Direito;
5. linguagem cartorária ou notarial – a linguagem jurídica que tem por finalidade registrar os atos de Direito.

Assim, embora se possa pensar numa linguagem jurídica em geral, quando lidamos com sentenças produzidas em processos judiciais, temos linguagem judiciária, forense ou processual. Quando lidamos com os textos de leis, decretos e portarias, temos a linguagem legislativa. Cada tipo de suporte e/ou instrumento jurídico tende a adotar usos diferenciados e um vocabulário diferenciado. E esses elementos podem ser importantes quando se trabalha com o processamento em larga ou pequena escala desses textos. Como há uma especificidade de discursos envolvida, considerar os seus elementos linguísticos e modos de dizer próprios poderá nos ajudar a desempenhar tarefas de um modo mais produtivo. Afinal, o “Direito é, por excelência, entre as que mais o sejam, a ciência da palavra. Mais precisamente: do uso dinâmico da palavra” (Xavier, 2002, p. 1).

Conforme já mencionamos no início deste capítulo, em cada país, a linguagem jurídica tende a realizar um uso particular da língua comum. Por isso, a linguagem do Direito de um país se diferencia da de um outro – como acontece com a linguagem jurídica dos diferentes países de Língua Portuguesa. Embora haja elos comuns, o Direito brasileiro é bastante distinto do de países como Angola ou Portugal. E, mesmo os textos jurídicos, em suas diversas formas (no Brasil conhecidos como petições, recursos, decisões judiciais etc.) podem adotar nomes e modelos de apresentação diversos. Conforme a cultura jurídica e as tradições de cada país, os produtores dos textos jurídicos serão também “autorias” diferentes, conforme o que é estabelecido no ordenamento legal de cada país.

O Direito no Brasil é regido pelo sistema da *civil law*, isso significa que uma a lei escrita tem preponderância sobre a jurisprudência – que são as decisões dos juízes – lembrando que os juízes são encarregados de verificar e direcionar a aplicação das leis. No Brasil, quem produz as leis são os membros do poder legislativo, eleitos, democraticamente, pelo povo. Os textos das leis são discutidos e votados, e então aprovados para entrarem em vigor. Os membros do poder executivo, também eleitos pelo povo, devem executar as leis aprovadas. Vejamos um resumo sobre como se organiza o Direito brasileiro, atualmente, em suas hierarquias:

- **Constituição Federal:** é a lei maior do país e define os direitos e deveres dos cidadãos, além de estabelecer a organização e o funcionamento dos poderes Executivo, Legislativo e Judiciário.
- **Legislação infraconstitucional:** são as leis, decretos e normas que regulamentam assuntos específicos, como por exemplo, o Código Civil, Código Penal, Código de Defesa do Consumidor, entre outros.



- **Poder Executivo:** é composto pelo Presidente da República, Vice-Presidente e ministros. É responsável pela administração do país e pela implementação das leis.
- **Poder Legislativo:** é formado pelo Congresso Nacional (Senado Federal e Câmara dos Deputados). É responsável por criar, modificar e aprovar as leis.
- **Poder Judiciário:** é composto pelos tribunais e juízes. É responsável por aplicar a lei em casos concretos, solucionar conflitos e garantir os direitos fundamentais dos cidadãos.

Apesar de o sistema jurídico brasileiro ser o *civil law*, há grande influência da jurisprudência nas decisões judiciais, principalmente quando agrupadas pelos tribunais e transformadas em *súmulas*. A súmula é um tipo de documento que consiste em um verbete que registra a interpretação pacífica ou majoritária adotada por um Tribunal a respeito de um tema específico. Portanto, quando textos legislativos e documentos processuais tornam-se objetos do PLN, com vistas a obter conhecimento para os profissionais do Direito, será preciso compreender esses elementos e valores diferenciados. Sem isso, há o risco de “misturar alhos com bugalhos”.

22.3 Entre as terminologias e as palavras no Direito do Brasil

Grosso modo, um RT equivale à identificação e à sistematização de denominações associadas a conceitos conforme utilizadas em um dado campo ou área do conhecimento. Geralmente, o RT envolve a produção de uma “lista” de nomes (termos) vinculados aos seus significados (conceitos). Além disso, junto de cada item dessa “lista”, tem-se um conjunto de informações que ajudam a contextualizar e a entender o seu uso ao longo de um conjunto de documentos escritos.

Assim, vamos pensar nesse processo ao longo de um conjunto de documentos jurídicos - em um dado tipo - tendo em mente a situação particular do uso de suas palavras. A Terminologia e os terminólogos dedicam-se a estudar – descrever e compreender - os diferentes fenômenos linguísticos da comunicação técnico-científica, o que se estende ao Direito, em seus variados cenários.

O que diferencia uma terminologia de uma palavra “comum” é, em primeiro plano, o seu ambiente comunicativo. E, repetindo a ideia de uma das maiores autoridades da nossa área da Terminologia (Cabré, 2005), podemos dizer: uma palavra não é um termo técnico-científico, ela está nessa condição em determinados contextos, que conferem a ela um significado “especial”. Esse significado ou modo de compreensão especial, chamaremos, grosso modo, de conceito.

Vejamos um exemplo, com a palavra/item CRIANÇA, muito corriqueira no nosso dia a dia. Como seu significado básico, geralmente, entendemos algo como “pessoa não adulta”. Mas, quando empregada e “significada” em um dado ambiente comunicativo de especialidade, como é o caso do Direito brasileiro, essa palavra “comum” assume contornos semânticos diferenciados.

No contexto do nosso Estatuto da Criança e do Adolescente, documento brasileiro conhecido como ECA, que corresponde à Lei 8.069-90, atualizada em 2021, que podemos enquadrar no domínio do Direito Civil do Brasil, temos o seguinte:

“Art. 2º Considera-se **criança**, para os efeitos desta Lei, a pessoa até doze anos de idade incompletos, e adolescente aquela entre doze e dezoito anos de idade” (BRASIL, 1990, grifo nosso).



Como se percebe, há um significado “especializado”, jurídico, uma delimitação em termos de anos de idade, que se soma ao nosso entendimento mais comum de **criança**. E você deve estar se perguntando: o que isso importará ou pode repercutir em um trabalho computacional sobre o tema das crianças em leis e documentos em português? A resposta é: importa muito! Se comparar com os que estabelece a OMS, Organização Mundial da Saúde, a faixa etária de uma pessoa considerada como **criança** é outra, pois compreende pessoas até 19 anos de idade. Isto é, os traços/valores de uso da palavra, que adquire estatuto terminológico, são variáveis. Além disso, temos uma conceituação jurídica específica/particular associada a um dado termo que, à primeira vista, não pareceria ser um termo.

No caso do segmento de lei acima, o ECA, podemos considerar que há uma definição específica para CRIANÇA, que se opõe à de ADOLESCENTE. Além disso, essa definição é circunscrita, isto é, ela vale apenas em um dado contexto ou “*frame* de significação”. Assim, teríamos um problema, para aquelas pessoas que se interessassem pelo Direito das Crianças, seja em sistemas jurídicos específicos, como o do Brasil, ou que busquem um mapeamento sobre esse tema no âmbito do Direito Internacional, não é mesmo?

Vamos supor uma aplicação de PLN que pudesse nos ajudar a dar conta de uma busca de informações sistematizada sobre esse tema, mas restrita ao cenário brasileiro. Como vimos, em Direito, temos uma definição que tende a ser circunscrita, isto é, ela vale apenas em um dado contexto, correspondendo a um valor que estabelece frente a todo um CONJUNTO DE OUTROS TERMOS E CONCEITOS com ela relacionados. Isso é o que chamamos de sistema conceitual, que tem a ver como uma rede de conceitos e terminologias que se entrelaçam. Como vimos, o ECA está subordinado à Constituição do Brasil, e ainda podemos ter, por exemplo, leis estaduais ou municipais – ou mesmo códigos e portarias – que “valem como leis locais” sobre o tratamento de crianças em estabelecimentos de Saúde em diferentes estados do Brasil. Além das normas, também podem haver interpretações jurídicas unânimes ou diversas sobre assuntos relacionados à criança e disponibilizadas em sentenças judiciais.

22.4 Um caso concreto: em pequeníssima escala

Para realizar um ensaio de um RT, podemos explorar um conjunto de textos que servem de referência ou espelhamento em uma dada área de conhecimento (veja um passo a passo detalhado com a *Constituição do Brasil* em Finatto; Esteves; Villar (2022)). Lidando com textos jurídicos, como vimos, será importante levar em conta suas naturezas e tipologias. Vamos supor que um RT associado, por hipótese ao tema “Direitos das Crianças no Brasil”. Esse RT poderia envolver identificar, em diferentes documentos relevantes previamente selecionados, os seguintes elementos:

- a) TERMOS e seus respectivos CONCEITOS
- b) TERMOS e seus respectivos FORMATOS LINGUÍSTICOS
- c) TERMOS, CONCEITOS e respectivos TERMOS E CONCEITOS RELACIONADOS.

Nos itens a) e b), acima, entra em jogo uma questão muito importante: a variação terminológica. Essa variação tem a ver com as diferentes formas das denominações, dentro de uma dada especialidade ou subárea, que um TERMO pode ter. Você poderá perguntar: vamos explorar esse tema no âmbito do Direito Civil até o Direito Criminal? Ou vamos ficar apenas em um dado recorte?

Para administrar a variabilidade de termos e conceitos, sem a ideia de condená-la, pois o enfoque linguístico e conceitual em um RT é sempre descritivo, temos, para nos socorrer,



os vocabulários controlados e/ou padronizados. Esses vocabulários mostram padrões de denominações que geralmente são colocados pela autoridade de órgãos profissionais associados a uma dada especialidade. Nesses vocabulários, encontramos as “terminologias padronizadas” e também as “normas técnicas” de uma área. Assim, uma forma de denominar um respectivo conceito/significado é estabelecida em um dado contexto, de modo a se garantir precisão e boa correlação com outros termos e conceitos relacionados. Isso será importante especialmente em situações de trocas de conhecimento e de trocas em geral.

Guardadas as devidas diferenças, é semelhante o caso, por exemplo, do conceito de CRIANÇA frente ao conceito de ADOLESCENTE no nosso Estatuto da Criança e do Adolescente, o ECA. Crianças não poderão ser confundidas, em um cenário legal e jurídico, com adolescentes ou pessoas adultas, salvo condições especiais definidas naquele texto, que funciona como uma moldura de significação para suas terminologias.

O mesmo vemos nos casos dos nomes “oficiais” para algumas doenças, que inclusive correspondem a um código numérico, conhecido como CID ou Classificação Internacional de Doenças. A ideia, nesse contexto de padronização das terminologias da área da Saúde, é evitar confusões e tentar garantir que todos possam ter um mesmo entendimento – ou conceito uniforme – de um dado TERMO + CONCEITO/DESCRIÇÃO DE SEU SIGNIFICADO. Abaixo, alguns exemplos dessa padronização da CID para o termo SARAMPO e seus tipos – uma doença, no Brasil, geralmente associada a **crianças**.

- CID 10 – B05 – Sarampo
- CID 10 – B05.0 – Sarampo complicado por encefalite
- CID 10 – B05.1 – Sarampo complicado por meningite

Dada a relevância e necessidade de tratar esse assunto, alguns tribunais como o Supremo Tribunal Federal e Superior Tribunal de Justiça criaram um site denominado “Tesouro” como forma de ferramenta para controle terminológico que tem por objetivo a padronização da informação. Nesta ferramenta, o tesouro, são apresentados os termos, conceitos, termos relacionados, mas também categorias, termos genéricos e termos específicos. A partir deste mapeamento, é possível orientar que os servidores públicos redijam os documentos judiciais com uma terminologia uniforme, para auxiliar na pesquisa e recuperação da informação posteriormente. Para saber mais sobre o tema dos *tesauros e sua interface com as terminologias*, vale consultar o trabalho de Vargas; Van der Lann (2011).

Dado o contexto que os tribunais pertencem, há uma variedade maior de termos relacionados em comparação à legislação e cada tribunal pode apresentar informações diversas nos *tesauros* para o mesmo termo. *Tesauros* são listas de assuntos, palavras-chave e de terminologias de uma dada área de conhecimento. Essas listagens dão suporte à indexação e catalogação de documentos em bibliotecas e em diferentes acervos, como bases de dados. Geralmente, quem produz esses tesouros são os bibliotecários, documentalistas e cientistas da informação que lidam com a catalogação de informações técnicas e científicas. Veja este exemplo, quando se busca pelo item CRIANÇA no *tesauro* do Supremo Tribunal Federal. Nessa busca, temos o seguinte resultado:

- **Termo Genérico:** Menor
- **Termos relacionados:** Adulto, Castigo Físico, Direito da Criança e do Adolescente, Educação, Educação Infantil, Estatuto da Criança e do Adolescente (ECA), Guarda de Menor, Investigação de Paternidade, Poder Familiar.
- **Categoria:** DCT Direito Constitucional, ECA Estatuto da Criança e do Adolescente.



O que se informa aqui não é um conceito para CRIANÇA, mas se aponta que ele tem um correspondente ou equivalente genérico nesse âmbito. Isto é, CRIANÇA = MENOR (genérico). Em seguida, nos termos relacionados, vemos assuntos em que se inclui esse item.

Feitas essas explicações sobre peculiaridades das terminologias, em suas diferentes circunstâncias e variabilidades de uso e de significações, um RT pode ser visto como um tipo de trabalho de mediação de comunicação, realizado por profissionais de uma área, terminólogos, linguistas, informatas, entre outros. Salienta-se, assim, a ideia de uma mediação terminológica (Conceição; Zanola, 2020). O RT pode ser um trabalho multidocumento e multitemático. Pode, ainda, apontar ligações entre documentos de diferentes naturezas, extrapolando-se o reconhecimento de um dado tópico para diferentes fronteiras. Um exemplo seriam os materiais sobre temas e políticas de Saúde Pública voltadas para crianças e os documentos jurídicos que estabelecem seus direitos. Outro exemplo de trabalho seria verificar como determinado tribunal interpreta e aplica a legislação sobre crianças, diante de problemas específicos.

22.5 Outros casos/exemplos: Direito Ambiental

Um RT legislativo também poderia servir de apoio para um recurso didático voltado para o cidadão comum, sem formação em Direito, ou mesmo para diferentes estudantes universitários interessados na legislação ambiental do Brasil. Nesse caso, vamos imaginar um conjunto composto, por exemplo, por 800 leis, as quais versam sobre diferentes aspectos ambientais. Vamos supor que estamos trabalhando em um RT para uso de jornalistas que lidam com temas ambientais. Como explorar essas 800 leis para chegar, por exemplo, a um conjunto de seus termos e conceitos conforme sejam mais comumente empregados nessas leis? Como apresentar a informação de forma a melhor atender o nosso suposto usuário jornalista, que, sem ter formação em Direito ou Biologia, precisaria ler e entender a legislação? Bastaria perguntar ao ChatGPT?

Naturalmente, hoje, dada a larga prática de digitalização desse tipo de documento e a garantia de seu acesso a todo o cidadão, parece ser fácil encontrar e percorrer uma base de dados com leis ambientais. O Senado Federal do Brasil, por exemplo, oferece todo um banco de leis, decretos e outros documentos afins para acesso público. Basta a pessoa acessar um site determinado e salvar os documentos no seu computador. Feito isso, “bastaria” a pessoa – o jornalista que imaginamos – ler, calmamente e com cuidado, todas as 800 leis do nosso caso imaginário e ir fazendo um registro, em um arquivo de texto, de suas terminologias e conceituações à medida que avance com a leitura. Outra opção seria o “nosso” jornalista consultar um dicionário especializado sobre esse tema, mas é nele, como um ponto final possível, entre outros, de um RT que queremos chegar com o que tratamos neste texto.

Como a legislação em alguns aspectos pode ser principiológica ou apenas fornecer diretrizes, ou ainda possuir conflitos de termos entre normas diversas, pode ser necessário associar um outro RT para identificar o entendimento prevalente. Neste caso, pode ser associado a um RT Judicial, como é o caso dos *tesauros* dos tribunais, mencionados anteriormente.

Naturalmente, além dessas fontes padronizadas, há dicionários e glossários descritivos sobre o tema do Direito Ambiental do Brasil. Um exemplo é o *Dicionário de Direito Ambiental* do Grupo Termisul da UFRGS, publicado em segunda edição em 2008. A produção desse dicionário demandou construir, desde 1994, toda uma base legislativa sobre



temas do meio ambiente associada à obra, a Base Legis². Essa base, que começa com o texto do Código de Águas do Brasil, de 1930³.

Mas, voltando aos tesouros, segundo o *tesauro* do Supremo Tribunal Federal, temos as seguintes informações:

- **Termo Genérico:** Direito (Ciência Jurídica)
- **Termo Relacionado:** Lei da Biossegurança
- **Categoria:** DAM Direito Ambiental

Como termo, há poucas informações sobre Direito Ambiental, mas ao acessar a categoria “DAM Direito Ambiental”, temos cerca de 200 termos relacionados. Vale destacar que os termos apresentados fazem parte do contexto ao qual o tribunal pertence e os processos judiciais que julga. Um RT legislativo pode ter mais termos que os apresentados no tesauro do Supremo Tribunal Federal, mas em contrapartida, o *tesauro* do tribunal, pode ter um nível de detalhamento maior.

Já para o Superior Tribunal de Justiça do Brasil, ao se consultar o termo/assunto **Direito Ambiental**, temos os seguintes resultados:

- **Termos relacionados:** Bioética, Meio Ambiente, Órgão público ambiental, Princípio da Precaução, Princípio da Prevenção, Princípio do In Dubio Pro Natura, Teoria do Risco Integral.

A vantagem do tesauro elaborado por alguns tribunais é que esse instrumento reduz esforços na aplicação do PLN ao Direito, uma vez que já associa os termos mais frequentes nas decisões judiciais e associa esses termos às decisões judiciais existentes. Dado que muitos dos tesouros não associam conceitos aos termos, ao trabalhar com RT de outras fontes, como o legislativo, podem ser obtidas estas informações.

22.6 Aplicação: Análise de Sentimentos em Direito: desafios e exemplos

A análise de sentimento em textos jurídicos envolve a aplicação de técnicas de PLN para determinar o tom emocional ou opinativo presente nos documentos legais.

Dentre diferentes possibilidades de aplicação da técnica de análise de sentimento em textos jurídicos, podemos trabalhar com sentenças judiciais. Assim, analisa-se o contexto das sentenças judiciais e identifica-se se o juiz foi favorável ou desfavorável ao pedido de cada parte. Para esse tipo de trabalho, alguns passos devem ser seguidos como: 1) Coleta de Dados; 2) Pré-Processamento do Texto; 3) Rotulação de Dados; 4) Escolha da técnica de análise de sentimento; 5) Execução da técnica e 6) Avaliação e Validação.

Na coleta de dados, é necessário escolher um repositório que possua os textos das decisões judiciais, sejam elas de forma resumida ou na íntegra. Dentre as opções públicas e gratuitas, estão os diários de justiça dos tribunais, uso de APIs (*Application Programming Interface*)

²A Base Legis Termisul-UFRGS é composta de textos da Legislação Ambiental do Brasil, Alemanha, Argentina, Estados Unidos, França, Paraguai e Uruguai. Também inclui códigos brasileiros, constituições dos países anteriormente mencionados e dos demais países de fala portuguesa (Angola, Cabo Verde, Guiné Bissau, Moçambique, Portugal, São Tomé e Príncipe e Timor Leste), Atos Internacionais relativos ao meio ambiente (Agenda 21, Convenção de Estocolmo, Declaração do Rio e Protocolo de Kyoto). Todos os textos possuem uma descrição e podem ser baixados em formato TXT.

³Você pode acessar em <http://www.ufrgs.br/termisul> na aba “Recursos”.



públicas como o DataJud do Conselho Nacional de Justiça e decisões disponibilizadas nos sistemas de busca dos tribunais. Para automatização desta coleta, é preciso o conhecimento de técnicas de *web crawling* e *web scraping* (Macohin; Carneiro, 2020). Essas técnicas consistem na automatização do download das páginas e arquivos que possuem decisões judiciais e posterior filtragem da informação que se deseja usar), respectivamente.

Um exemplo de informação que pode ser obtida de um tribunal pode ser verificado abaixo. Neste exemplo foram suprimidas algumas informações que pudessem identificar os envolvidos. Veja que lidamos com um tipo de texto que contém *uma parte* denominada *ementa* e outra parte que é o *acórdão*, com o resultado do processo.

Figura 22.1: Exemplo de resumo de acórdão judicial de um tribunal superior brasileiro.

PROCESSO			
RÉsp	/ SC		
RECURSO ESPECIAL			
2019/	-0		
RELATOR	ÓRGÃO JULGADOR	DATA DO JULGAMENTO	DATA DA PUBLICAÇÃO/FONTE
Ministro	T2 - SEGUNDA TURMA	/11/2022	DJe /12/2022
RELATOR PARA ACÓRDÃO			
Ministro			
EMENTA			
ADMINISTRATIVO. AÇÃO CIVIL PÚBLICA AJUIZADA PELA UNIÃO COM VISTA A OBTER O RESSARCIMENTO POR DANO PATRIMONIAL DECORRENTE DE EXPLORAÇÃO DE MINÉRIO (BASALTO) SEM AUTORIZAÇÃO. PRETENSÃO SUJEITA À INCIDÊNCIA DA PRESCRIÇÃO QUINQUENAL. RECURSO ESPECIAL A QUE SE NEGA PROVIMENTO.			
1. No caso dos autos, a União ajuizou ação civil pública com o objetivo de obter ressarcimento pela lavra ilegal de basalto.			
2. O Tribunal Regional Federal manteve a sentença de improcedência do pedido, pois, "Em se tratando de ação civil pública movida pelo Poder Público em face de particular (não abrangido pelo conceito de agente público), objetivando a reparação de dano decorrente da extração ilegal de recursos minerais, deve ser aplicado o prazo prescricional quinquenal delineado na Lei da Ação Popular (Lei n. 4.717/1965)".			
Inconformada, a União recorre defendendo o afastamento da prescrição.			
3. O entendimento adotado pelo Tribunal Regional Federal da 4ª Região não merece reparos, pois a imprescritibilidade da pretensão de ressarcimento ao erário se aplica somente em casos excepcionais, como é o do ato doloso de improbidade administrativa; e a incidência da prescrição, como regra, consagra o princípio da segurança jurídica (e até mesmo o da ampla defesa), não sendo cabível o sacrifício de direito fundamental do particular como medida de compensação da ineficiência da máquina pública.			
4. Recurso especial a que se nega provimento.			
ACÓRDÃO			
Vistos, relatados e discutidos esses autos em que são partes as acima indicadas, acordam os Ministros da SEGUNDA TURMA do Superior Tribunal de Justiça, na conformidade dos votos e das notas taquigráficas, o seguinte resultado de julgamento:			
"Prosseguindo-se no julgamento, após o voto-vista do Sr. Ministro Francisco Falcão, acompanhando o Sr. Ministro-Relator, dando provimento ao recurso especial, a Turma, por maioria, negou provimento ao recurso especial, nos termos do voto do Sr. Ministro Og Fernandes, vencidos os Srs. Ministros Herman Benjamin e Francisco Falcão. Lavrará o acórdão o Sr. Ministro Mauro Campbell Marques."			
Os Srs. Ministros Mauro Campbell Marques e Assusete Magalhães votaram com o Sr. Ministro Og Fernandes.			

A partir do download desta página, o objetivo é extrair a informação do *acórdão*, último parágrafo da imagem, onde consta se foi dado ou negado provimento ao pedido do autor. Neste caso, foi negado provimento ao autor do recurso, como se verifica através do uso das palavras “negou provimento”.

Em posse dos trechos das decisões judiciais que se deseja analisar, de forma automatizada, se o desfecho foi favorável ou desfavorável, pode-se iniciar a fase de pré-processamento do texto. A fase de pré-processamento pode contemplar diversas subtécnicas. Mas, para fins de exemplificação, vamos citar apenas a tokenização, a remoção de pontuações, conversão de todas as letras para minúsculas e remoção de *stop words*.

A tokenização consiste em dividir o texto em palavras ou unidades menores, chamadas de *tokens* (veja mais detalhes no Capítulo 4), que são conjuntos de caracteres separados por um espaço em branco. Um critério para separação dos *tokens* pode ser o espaço entre



as palavras. Já a remoção de pontuações visa eliminar pontuação e caracteres especiais que não são relevantes para a análise de sentimento.

Em seguida, a conversão para minúsculas consiste em transformar todas as palavras em minúsculas para garantir consistência nas comparações. Por fim, a remoção de *stop words* consiste na remoção de palavras que são comuns - as palavras gramaticais ou instrumentais - e não contribuem significativamente para uma análise de sentimento, como “a”, “o”, “em”, “por” etc. Vale destacar que a lista de *stop words* deve ser na mesma língua do texto analisado. Veja o Quadro 22.1 abaixo.

Quadro 22.1. Exemplo de pré-processamento de um trecho de decisão judicial.

<p>Texto Original: “Vistos, relatados e discutidos esses autos em que são partes as acima indicadas, acordam os Ministros da SEGUNDA TURMA do Superior Tribunal de Justiça, na conformidade dos votos e das notas taquigráficas, o seguinte resultado de julgamento: 'Prosseguindo-se no julgamento, após o voto-vista do Sr. Ministro Francisco Falcão, acompanhando o Sr. Ministro-Relator, dando provimento ao recurso especial, a Turma, por maioria, negou provimento ao recurso especial, nos termos do voto do Sr. Ministro Og Fernandes, vencidos os Srs. Ministros Herman Benjamin e Francisco Falcão. Lavrará o acórdão o Sr. Ministro Mauro Campbell Marques.' Os Srs. Ministros Mauro Campbell Marques e Assusete Magalhães votaram com o Sr. Ministro Og Fernandes.”</p> <p>Após Pré-processamento: “vistos relatados discutidos autos partes acima indicadas acordam ministros segunda turma superior tribunal justiça conformidade votos notas taquigráficas seguinte resultado julgamento prosseguindo julgamento após voto-vista sr ministro francisco falcão acompanhando sr ministro-relator dando provimento recurso especial turma maioria negou provimento recurso especial termos voto sr ministro og fernandes vencidos srs ministros herman benjamin francisco falcão lavrará acórdão sr ministro mauro campbell marques srs ministros mauro campbell marques assusete magalhães votaram sr ministro og fernandes”</p>

Ainda na fase de pré-processamento, é possível aperfeiçoar a tarefa e incluir novas *stop words*, com o objetivo de limpar mais ainda o texto e facilitar futuramente a identificação das palavras positivas ou negativas na sentença. No Quadro 22.1, verifica-se que as palavras “srs”, “sr”, “vistos”, não influenciam na interpretação da decisão judicial e podem ser removidas.

A próxima fase, rotulamento de dados, consiste em classificar manualmente algumas decisões como positivas ou negativas, para fins de validação futura se a classificação automatizada está desempenhando um bom resultado. A partir do Quadro 22.1, facilmente esta decisão seria classificada como “NEGATIVA”. Outras opções de rótulo seriam “POSITIVA” e “NEUTRA”. Os casos de neutro poderiam ser utilizados, por exemplo, quando o juiz decidiu parcialmente pelo provimento.

Já a fase da escolha da técnica de análise de sentimento, consiste em selecionar qual abordagem será utilizada, se baseada em regras ou baseada em aprendizado de máquina. Na abordagem baseada em regras, é criado um conjunto de regras e heurísticas que determinam o sentimento ou polaridade com base em palavras-chave, padrões gramaticais e outras características linguísticas. Por exemplo, certas palavras negativas podem indicar um sentimento negativo. Já, na abordagem baseada em aprendizado de máquina, é treinado um modelo de aprendizado de máquina usando-se os dados linguísticos rotulados. Algoritmos



como *Naïve Bayes*, *Support Vector Machines* (SVM) ou redes neurais podem ser usados para construir um modelo.

Para dar continuidade ao exemplo mencionado, utilizaremos a abordagem baseada em regras. Nesse caso, utilizamos um dicionário prévio com palavras positivas e negativas. Quando são usados dicionários, deve ser considerada a língua do texto. Como exemplo de dicionário de palavras positivas, negativas e neutras em português, temos o SentiLex-PT⁴ (Carvalho; Silva, 2017).

Na fase de execução da técnica e a partir do texto pré-processado anteriormente, cada palavra do texto é verificada se consta no dicionário como palavra positiva, negativa ou neutra. Segundo o SentiLex-PT, foi encontrado o seguinte resultado apresentado no Quadro 22.2.

⁴O SentiLex-PT está disponível em: <https://b2share.eudat.eu/records/93ab120efdaa4662baec6adee8e7585f>.



Quadro 22.2. Palavras positivas e negativas segundo o SentiLex-PT.

Dicionário:
 Positivas: [“acordam”, “conformidade”, “aprovar”]
 Negativas: [“negou”, “vencidos”]
 Sentimento:
 Positivas: 3 (acordam, conformidade, aprovar)
 Negativas: 2 (negou, vencidos)
 Sentimento Geral: 1 (POSITIVO) (Positivas - Negativas = 3 - 2)

O sentimento geral é calculado realizando uma subtração do número de palavras positivas com o número de palavras negativas. Um resultado com valor positivo indica um sentimento positivo, já um resultado com valor negativo indica um sentimento negativo e um valor próximo de zero indica um sentimento neutro. Este cálculo pode ser aperfeiçoado ao dividir o número encontrado pelo total de palavras (*tokens*) existentes no texto ((palavras positivas - palavras negativas) / total de palavras). Ou seja, se há 10 palavras no texto, 3 são positivas e 0 negativas, indica uma maior “probabilidade” que o texto realmente seja positivo. Por outro lado, se há 50 palavras no texto, apenas 1 negativa e nenhuma positiva, há probabilidade de ser um falso negativo. Essa divisão pode indicar que novos aperfeiçoamentos no dicionário podem ser necessários.

Neste exemplo, verifica-se que o resultado não reflete a resposta correta (NEGATIVO) e ajustes devem ser feitos. O dicionário SentiLex-PT pode ser adaptado para a linguagem jurídica, uma vez que “acordam” “conformidade” e “aprovar”, não indicam necessariamente que o juiz está dando provimento (julgando como positivo) a decisão judicial, logo, devem ser desassociados do sentimento “POSITIVO”. Outro ajuste que pode ser feito também é não associar a palavra “vencidos” ao sentimento “NEGATIVO”, uma vez que é comum, quando há divergência entre o grupo de juízes votantes, aparecer a palavra “vencidos”. Outro ajuste que também pode ser feito no dicionário é associar palavras frequentemente encontradas juntas e que reflitam a intenção da decisão judicial, por exemplo “negou provimento”, “negado provimento”, “não provido o recurso”, “ não dado provimento”, entre outros.

Feitos estes ajustes, teremos o seguinte resultado apresentado no Quadro 22.3.

Quadro 22.3. Resultado após os ajustes.

Dicionário:
 Positivas: [“ ”]
 Negativas: [“negou provimento”]
 Sentimento:
 Positivas: 0 ()
 Negativas: 1 (negou provimento)
 Sentimento Geral: -1 (NEGATIVO) (Positivas - Negativas = 0 - 1)

Lembrando de que essa abordagem é uma simplificação e pode não capturar todas as nuances de sentimento e/ou as polaridades em textos jurídicos complexos. Principalmente quando há uma variedade de pedidos sendo julgados com decisões diferentes para cada pedido. O contexto legal específico também pode influenciar a interpretação das palavras. Portanto, ajustes e validações são sempre necessários.

Por fim, com relação à fase de avaliação e validação, se foi utilizada a abordagem baseada em regras, como a acima exemplificada, é mais simples validar com a amostra rotulada



previamente e comparar a taxa de acertos e erros. Já no caso da abordagem baseada em aprendizado de máquina, é possível utilizar parâmetros estatísticos para demonstrar a precisão e desempenho do modelo.

Como mencionado anteriormente, é necessário fazer ajustes em cada fase da execução da análise de sentimento devido às peculiaridades discursivas dos textos jurídicos que nem sempre constam nos dicionários existentes. A partir das informações obtidas na última fase, de avaliação e validação, pode sugerir que novos aperfeiçoamentos sejam feitos nas fases anteriores, para que o algoritmo tenha um desempenho similar e até superior ao de uma atividade humana.

Dentre os desafios para aplicar este tipo de técnica em decisões judiciais, está principalmente na coleta dos dados. Os tribunais, no geral, não possuem repositórios com estas informações prontas, estruturadas e públicas e isto por si só, já dificulta iniciar qualquer trabalho de processamento de linguagem natural. Apesar do problema poder ser contornado com a criação de algoritmos de *web crawling* e *web scraping*, alguns tribunais fazem uso de *captchas* que impedem o acesso automatizado e massivo às informações. Apesar de iniciativas do Conselho Nacional de Justiça, como o DataJud, para centralizar e fornecer informações estruturadas por meio de uma API, ainda não há a íntegra das decisões disponibilizadas. Entretanto, como o DataJud continua em constante evolução, é possível que futuramente seja disponibilizado. A limpeza e seleção das informações contidas em uma página HTML ou arquivo PDF também é bastante custosa e somente a partir destes esforços torna possível dar seguimento à aplicação da técnica de análise de sentimento em Direito.

22.7 Considerações finais

Este capítulo tentou situar o “mundo textual” oferecido pelo Direito do Brasil e trazer alguns exemplos de trabalhos e estudos com a sua linguagem e as suas práticas de escrita. Comentamos um pouco dos estudos históricos com processos brasileiros antigos sobre crimes contra mulheres, mencionamos que a tarefa de RT é uma demanda no Direito Ambiental e trouxemos um breve exemplo de estudo de sentimentos ou de polaridades, em PLN, para ajudar a detectar os tipos de decisões que os juízes brasileiros tomam em determinados tipos de processos federais.

Buscamos salientar que é importante considerar as características e elementos de diferentes tipos de documentos, sejam leis, códigos, processos ou sentenças, produzidos por diferentes instâncias jurídicas. E, visto esse panorama, caso você possa se interessar especificamente por sentenças de tribunais administrativamente menores, como os Juizados Especiais Cíveis (JECs) - conhecidos como “tribunais de pequenas causas”, vale conhecer o trabalho de doutorado de Motta (2022). A autora estudou a complexidade das sentenças dos JECs, quanto ao vocabulário, terminologias e sintaxe, frente aos preceitos da legislação que estabelece que tais sentenças devem ser escritas em linguagem simples, que possibilitem fácil compreensão sobre o que se decide em uma causa. Afinal, o cidadão comum recorre aos JECs geralmente sem advogados, em meio a causas de valor limitado. Além de ampla análise, Motta (2022) oferece, no seu trabalho, acesso a todo um *corpus* de sentenças por ela reunido e analisado com a ferramenta NILC-METRIX⁵ (Leal et al., 2021). Também os *corpora* que ela usou como contraponto para ponderar a complexidade/facilidade de linguagem dessas sentenças estão disponíveis nos seus anexos.

⁵<http://fw.nilc.icmc.usp.br:23380/nilcmatrix>



O Direito é um mundo feito de palavras e modos de dizer, o que oferece um terreno fértil para os nossos trabalhos de análise linguístico-textual, em geral, e, em especial, para diferentes tarefas do PLN. Os resultados desses trabalhos beneficiam tanto os profissionais quanto o cidadão e a sociedade, que são os principais focos e beneficiários das ações do Direito.



Capítulo 23

PLN em Redes Sociais

Brenda S. Santana
Larissa A. de Freitas

23.1 Introdução

O Processamento de Linguagem Natural (PLN) desempenha um papel cada vez mais significativo no cenário das redes sociais. O volume de dados advindos de redes sociais a todo instante é imenso. Dentre os dados gerados, podemos citar os dados textuais, os quais variam desde conversas informais até discussões complexas. Nas redes sociais, as pessoas expressam ideias e opiniões de maneiras diversas. Isso inclui o uso de gírias, abreviações, emojis e outros elementos da linguagem cotidiana. Tratar esse tipo de dado não é uma tarefa trivial e é desafiador para os sistemas de PLN.

Os sistemas de PLN são ferramentas indispensáveis para compreender, analisar e extrair informações. O estudo dos estilos de linguagem utilizados nas redes sociais ajuda a melhorar a compreensão de textos informais. As redes sociais são fontes valiosas de informação e seus conteúdos podem ser utilizados como *corpora* para treinar e testar algoritmos de PLN, permitindo que pesquisadores e desenvolvedores trabalhem com exemplos reais e relevantes. Visto que o Brasil é um dos países com maior presença nas redes sociais, e o português é o idioma predominante nessas interações, tem-se aqui uma área muito fértil para o desenvolvimento de estudos de aplicações de abordagens e PLN.

Neste capítulo buscamos abordar algumas das principais áreas de aplicação de PLN em redes sociais, discutindo os desafios encontrados. Ainda, buscamos apresentar alguns dos recursos disponíveis para suporte no desenvolvimento de estudos voltados para as tarefas apresentadas, focando em dados em língua portuguesa. Para tanto, este capítulo se organiza da seguinte maneira: na Seção 23.2, apresentamos a definição de redes sociais e descrevemos sobre os conteúdos nela postados; na Seção 23.3, apresentamos as principais áreas de aplicação de PLN que utilizam essas redes sociais. E, na Seção 23.4, apresentamos as considerações finais.

23.2 Redes Sociais

Uma rede social é definida como um conjunto de dois elementos: atores e suas conexões (Wasserman; Faust, 1994). Nos últimos anos, as redes sociais (como: Facebook, Reddit, Youtube, Twitter/X, Whatsapp e Instagram) têm revolucionado a forma como indivíduos, grupos e comunidades interagem. Nelas, são compartilhados textos, fotos, vídeos e outros tipos de conteúdo. Assim, as redes sociais estabelecem um ambiente rico e dinâmico que oferece inúmeras oportunidades para o estudo e o aprimoramento de abordagens em PLN.



Segundo Recuero (2009), o estudo das redes sociais na Internet objetiva analisar como as estruturas sociais surgem, de que tipo elas são e como são compostas.

De acordo com Farzindar; Inkpen (2018), usar PLN em textos provindos de mídias tradicionais (como jornal, rádio e televisão) tem sido um tópico de pesquisa popular nos últimos 25 anos. Hoje, usar PLN em textos provindos de redes sociais é uma área de pesquisa que requer adaptações dos métodos tradicionais, já que os textos provindos de redes sociais têm várias peculiaridades, principalmente devido a sua natureza. Ainda, eles podem estar escritos em diferentes idiomas e pertencerem a diferentes fontes.

As redes sociais se popularizaram no Brasil em 2004, com a criação do Orkut¹. Desde lá, novas redes surgiram e com elas a percepção da necessidade e viabilidade de aplicação de abordagens em PLN para o estudo de conteúdos e comportamentos gerados nesse meio. Dentre as áreas de aplicação dessas abordagens, destacam-se a detecção de discurso de ódio e linguagem ofensiva, a detecção de ironia/sarcasmo/humor, a detecção de notícias falsas, a análise de sentimento, entre outras (Ferreira et al., 2017).

Na literatura, existe uma predominância do Twitter/X como fonte de dados, isso se deve, provavelmente, ao fato de ele oferecer uma API² que, de forma muito simples, consegue acessar mensagens publicadas e os dados associados a seus usuários (por exemplo, o número de seguidores deste). No caso do Facebook, é necessário criar um aplicativo e obter a autorização dos usuários para que seus dados possam ser acessados/capturados (Coello; Junqueira, 2019), o que pode ser visto como um limitante na extração de informações desta rede.

No ano de 2023, algumas mudanças ocorreram nas APIs do Twitter/X e do Reddit. No Twitter/X, os pesquisadores terão que se adaptar às restrições da versão gratuita ou assinar alguns dos planos pagos para manter suas atividades. Já, no Reddit, o uso da API³ passou a ser cobrado. Portanto, é de se esperar que mudanças aconteçam nas pesquisas que utilizam *corpora* advindos dessas redes sociais.

Como mencionado anteriormente, o conteúdo postado nas redes sociais pode variar muito de acordo com a plataforma, o público-alvo e a intenção por trás da postagem. Abaixo descrevemos brevemente as redes sociais mais utilizadas em trabalhos de PLN sobre redes sociais em língua portuguesa.

23.2.1 Facebook

O Facebook⁴ é atualmente a maior rede social do mundo, com 2.9 bilhões de usuários ativos em 2023. Ela permite que os usuários criem perfis pessoais, adicionem amigos, compartilhem textos, fotos, vídeos e atualizações de status. Os usuários podem interagir com as postagens de outros usuários através de curtidas, comentários e compartilhamentos. Além disso, essa rede social também permite a criação de páginas para empresas, tornando-se uma ferramenta importante para marketing e divulgação.

23.2.2 Reddit

O Reddit⁵ é uma plataforma online de compartilhamento de conteúdo e discussões, organizada em comunidades chamadas *subreddits*. Os usuários podem enviar postagens,

¹<http://orkut.com/>

²<https://developer.twitter.com/en/docs/twitter-api>

³<https://www.reddit.com/dev/api/>

⁴<https://www.facebook.com/>

⁵<https://www.reddit.com/>



comentar, votar em conteúdos e interagir uns com os outros. Essa plataforma abrange uma ampla variedade de tópicos e interesses, permitindo que os usuários encontrem comunidades específicas que correspondam aos seus interesses. É um espaço onde os usuários podem trocar informações, debater, compartilhar histórias, memes e muito mais.

23.2.3 Youtube

O Youtube⁶ é uma plataforma de compartilhamento de vídeos que permite que os usuários compartilhem e assistam vídeos de uma variedade de gêneros, incluindo filmes, programas de TV, vídeos musicais, documentários, entre outros. Também é uma ferramenta de marketing importante para muitas empresas e indivíduos, os quais usam a plataforma para compartilhar conteúdo promocional e aumentar a conscientização sobre seus produtos ou serviços.

23.2.4 Twitter/X

O Twitter/X⁷ é um serviço de microblogging que pode ser utilizado para transmitir pequenas atualizações de status (Russell, 2011). Nele podem ser analisados os vínculos entre amigos e seguidores, grafos sociais e descobertas de mais informações sobre os usuários, inspecionando as entidades presentes em seus tweets. Os tweets são mensagens curtas (contendo até 280 caracteres, incluindo texto, imagens, GIFs, vídeos e links para outros sites) e públicas postadas no Twitter/X. Eles têm um alcance imediato e podem se tornar viral rapidamente, dependendo do conteúdo e da quantidade de interação que recebem de outros usuários. Isso faz do Twitter/X uma plataforma poderosa para disseminar informações, ideias e tendências em tempo real.

23.2.5 Whatsapp

O Whatsapp⁸ permite que os usuários troquem mensagens privadas. Apesar de ser usado principalmente para conversas individuais, o WhatsApp possui recursos de grupos de conversação, onde podem participar até 256 usuários, e encaminhamento de mensagens (Cabral et al., 2021). Concebido como um aplicativo de mensagens instantâneas, o WhatsApp evoluiu para uma plataforma multifacetada, permitindo não apenas conversas privadas, mas também a formação de grupos e comunidades, compartilhamento de mídia, chamadas de voz e vídeo e até mesmo recursos empresariais.

23.2.6 Instagram

Instagram⁹ é uma rede social para compartilhamento de fotos e vídeos. Nela também é possível acompanhar (seguir) outras contas, curtir, comentar e compartilhar publicações. Todas as publicações realizadas no aplicativo são mostradas por meio do *feed* e o usuário pode visualizar as postagens das contas que ele segue. Ainda, esta rede social oferece diversas outras funcionalidades, como: *boomerang*, *live* e *stories*.

⁶<https://www.youtube.com/>

⁷<https://twitter.com/>. O Twitter mudou de nome em 2023 e agora se chama X. Por esse motivo, ao longo do texto, iremos usar o termo Twitter/X quando mencionarmos essa rede social.

⁸<https://www.whatsapp.com/>

⁹<https://www.instagram.com/>



23.3 Áreas de Aplicação

Abaixo são descritas quatro áreas de aplicações que surgiram com a finalidade de compreender, analisar e extrair informações de textos que são publicados diariamente nas redes sociais, são elas: detecção de discurso de ódio e linguagem ofensiva, análise de sentimento, detecção de notícias falsas e detecção de ironia/sarcasmo/humor.

23.3.1 Detecção de Discurso de Ódio e Linguagem Ofensiva

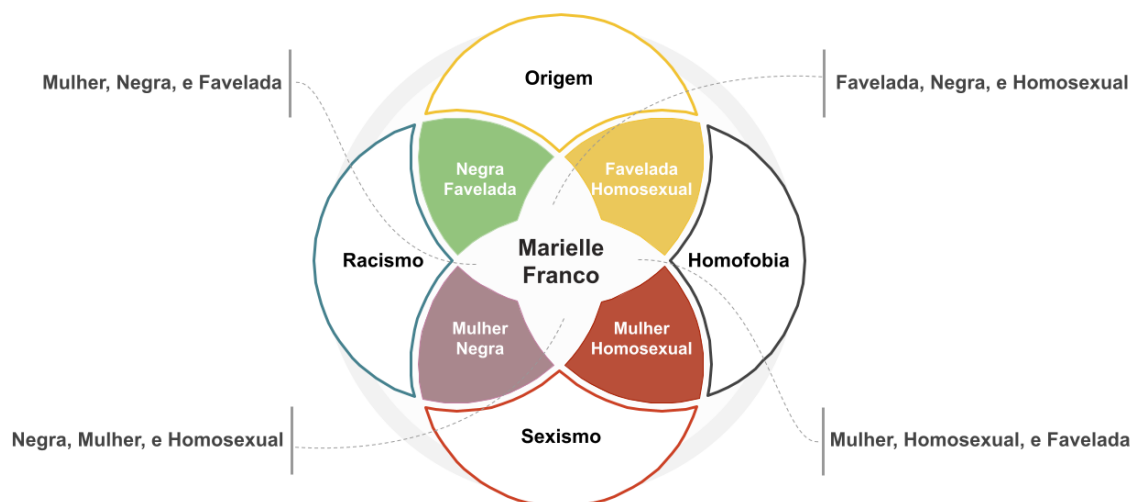
A partir de definições encontradas na literatura, termos diferentes, porém semelhantes, podem ser enquadrados como discursos simbolicamente prejudiciais (por exemplo, discurso perigoso, discurso tóxico, discurso de ódio, discurso intolerante e outros). Certos discursos possuem o potencial de causar danos significativos, inclusive críticos, e podem ser considerados tóxicos (Tirrell, 2018). Discursos tóxicos podem assumir diversas formas, podendo ser um discurso persistente ou momentâneo, afetar indivíduos ou a sociedade como um todo, causando danos temporários ou permanentes. O impacto de toxinas discursivas é de natureza social, afetando comunidades e prejudicando indivíduos pertencentes aos grupos-alvos. Essas toxinas podem incluir palavras ofensivas, insultos, discriminação, discurso de ódio, difamação, ameaças ou qualquer forma de linguagem que busque macular, menosprezar ou ferir a dignidade e a integridade de indivíduos pertencentes ao grupo-alvo. De acordo com Kumar et al. (2023), comentários tóxicos são a principal forma de ódio e assédio online.

Os grupos-alvos de discursos tóxicos podem variar dependendo do contexto e da natureza do discurso. Grupos frequentemente alvos de discursos tóxicos incluem minorias étnicas e raciais, comunidade LGBTQIA, mulheres, religiões minoritárias, portadores de deficiência, refugiados e imigrantes, e grupos políticos ou ideológicos. Contudo, qualquer grupo ou indivíduo pode ser alvo de discursos tóxicos, e a disseminação desse tipo de linguagem é prejudicial para a sociedade como um todo. Indivíduos pertencentes a diferentes grupos discriminados podem ainda ser alvos de discursos interseccionais que os atacam por múltiplas frentes. A Figura 23.1, adaptada de Santana (2023), ilustra um caso em que ataques interseccionais dirigidos a uma entidade foram disseminados na Internet. A imagem busca ilustrar o que aconteceu após o assassinato em 2018 da socióloga e política brasileira Marielle Franco, quando uma rede de ódio e desinformação gerou diversos comentários online atacando sua imagem por diversas características que ela possuía, e até outros traços que foram indevidamente atribuídos a ela. Teixeira; Zamora (2019) destacam que Marielle - mulher negra, assumidamente bissexual, favelada, defensora política dos direitos humanos - foi, sem dúvida, atravessada por todo tipo de opressão desencadeada pelo sistema machista, racista e classista. Os ataques registrados neste caso foram motivados pelo ódio.

A análise e a detecção de diferentes tipos de discursos tóxicos são um tópico de crescente interesse tanto para a área de PLN quanto demais áreas de interesse social como um todo. De acordo com Guimarães et al. (2020), quando focamos em comentários tóxicos, especialmente em notícias, o Facebook é a rede social que mais se destaca. O Reddit também introduz uma inclinação em relação à linguagem tóxica e ofensiva (Mohan et al., 2017). Por esse motivo, o conteúdo do Reddit tem sido usado para estudar microagressões (Breitfeller et al., 2019; Mollas et al., 2022) e depressão (Pirina; Çöltekin, 2018). De acordo com estudos realizados por Kumar et al. (2023), perfis que postam comentários tóxicos representam 3,1% de todas as contas que postam comentários no Reddit. Entretanto, ainda de acordo com os autores, apesar de seu percentual relativamente pequeno, tais contas



Figura 23.1: Exemplo de interseccionalidade entre diferentes formas de discurso de ódio observado no caso de Marielle Franco.



Fonte: Adaptada de (Santana, 2023)

desempenham um papel ativo e de alto impacto na plataforma.

Apesar dos diversos avanços pelos quais a área de PLN vem passando, a detecção de discursos tóxicos ainda é um desafio latente. O desenvolvimento de algoritmos de PLN e Aprendizado de Máquina (AM) para detectar esses tipos de conteúdo depende da disponibilidade de *corpora* anotados para treinamento. Conforme identificado por Trajano; Bordini; Vieira (2023) quase todos os sistemas de detecção de toxicidade usam modelos de aprendizado supervisionado que requerem uma grande quantidade de dados rotulados¹⁰. Entre estes *corpora*, podemos ressaltar recursos para a língua portuguesa como o ToLD-Br¹¹ desenvolvido por Leite et al. (2020).

O ToLD-Br (Leite et al., 2020) é um conjunto de dados capturado do Twitter/X entre julho e agosto de 2019 com a ferramenta GATE Cloud's Twitter Collector¹². Elaborado para estudos sobre classificação automática de comentários tóxicos, este conjunto de dados tem como objetivo equilibrar o viés de anotação. Para tanto, 42 anotadores foram selecionados, com base em suas informações demográficas. Este *corpus* apresenta um conjunto de 21000 tweets em português manualmente anotados por três diferentes anotadores em sete categorias: LGBTQ+fobia, obsceno, insulto, racismo, misoginia e/ou xenofobia.

O estudo de discursos tóxicos é de suma importância por várias razões, abordando questões sociais, éticas e técnicas. O volume de informações gerado a partir das redes sociais e plataformas online aumenta a exposição a discursos tóxicos, o que pode causar impactos negativos na saúde mental e emocional dos usuários. Compreender e identificar esses discursos é fundamental para criar um ambiente digital mais saudável e seguro para todos. Discursos tóxicos frequentemente incluem discursos de ódio e manifestação de linguagem imprópria, ou seja, atos que podem promover a violência, intolerância e discriminação contra grupos específicos. A análise desses discursos permite identificar

¹⁰Em <https://hatespeechdata.com> alguns *corpora* anotados sobre discurso de ódio, abuso online e linguagem ofensiva são catalogados e podem ser utilizados como base para o desenvolvimento de estudos que buscam detectá-los.

¹¹<https://github.com/JAugusto97/ToLD-Br>

¹²<https://cloud.gate.ac.uk>



padrões prejudiciais e trabalhar para mitigar seus efeitos negativos.

Adicionalmente, muitos discursos tóxicos envolvem a disseminação intencional de informações incorretas, desinformação e notícias falsas (Seção 23.3.3). Ao estudar esses discursos, podemos desenvolver técnicas para detecção precoce de conteúdo enganoso, ajudando a manter a qualidade da informação nas redes. Além disso, técnicas de PLN auxiliam na detecção de linguagem irônica, sarcástica e outros formatos frequentemente usados nas redes sociais para mascarar discursos de ódio (Seção 23.3.4). Essas abordagens avançadas permitem que plataformas de redes sociais aprimorem suas ferramentas de moderação, identificando automaticamente discursos de ódio e adotando medidas para removê-los ou sinalizá-los.

Muitas iniciativas têm sido empreendidas com o intuito de possibilitar a detecção automatizada de discursos de ódio nas diferentes plataformas. Conforme mencionado por Fortuna; Nunes (2018), esse crescente interesse não se restringe apenas à ampla cobertura midiática, mas também à crescente relevância política do tema. No entanto, os autores também destacam desafios latentes, como a falta de técnicas automáticas adequadas e a escassez de dados confiáveis sobre o discurso de ódio, que continuam motivando pesquisas nessa área. Analisando estatísticas brasileiras, Dadico (2020) explana que os dados indicam que o ódio sobrevitima pessoas de grupos identificados por critérios de raça, cor, etnia, sexo, orientação sexual, identidade de gênero, origem nacional e regional, sem-teto ou deficiência, entre outros atributos que os expõem a maior vulnerabilidade social. Apesar da normalização do ódio, esse discurso é parte de uma narrativa socio-histórica que traz em si os modos de pensar de uma cultura. É pela língua que nos mostramos como somos, e enquanto ela pode ser um instrumento de empoderamento, também pode gerar exclusão, opressão. O avanço de estudos de aplicação de abordagens de PLN para a detecção de tais conteúdos é essencial. Entretanto, avanços nesta área de estudos dependem fundamentalmente de conjuntos de dados anotados, ferramentas de análise de texto e modelos específicos disponibilizados para tal.

Para o português, Fortuna et al. (2019) criou um conjunto de dados para a classificação do discurso de ódio, o HLPDSD¹³. As instâncias deste conjunto foram coletadas através do uso da API do Twitter/X. Para isso, foram usadas palavras-chave e hashtags como #dyke ou #womensPlaceIsInTheKitchen coletadas entre janeiro e março de 2017 (majoritariamente). Este conjunto de dados contém conteúdo de 1156 usuários diferentes e abrange diferentes tipos de discriminação, com base sobre religião, gênero, orientação sexual, etnia, e migração. Nele foram feitas duas anotações: binária (“é discurso de ódio” ou “não é discurso de ódio”) e hierárquica (“racismo”, “sexismo”, ou “homofobia”). Na anotação binária, cada tweet foi anotado por três diferentes anotadores. Por fim, uma votação majoritária para determinar classificação final foi realizada nos 3059 tweets. Os autores realizaram experimentos utilizando uma LSTM combinada com *embeddings* pré-treinados para realizar uma classificação base a partir deste conjunto de dados e assim demonstrar seu potencial de uso. O resultado obtido foi a medida-F de 78%.

Outro conjunto de dados disponível na literatura que foi elaborado para estudos sobre a classificação do discurso de ódio é o HateBR¹⁴, elaborado por Vargas et al. (2022). O HateBR é composto por 7000 textos sobre o domínio político coletados através da API do Instagram. Neste conjunto de dados, constam postagens de seis contas pré-definidas (gênero - 4 mulheres e 2 homens, posição política - 3 liberais e 3 conservadores). Sua anotação foi feita de três maneiras: binária (“é ofensivo” ou “não é ofensivo”), granularidade (“levemente

¹³<https://b2share.eudat.eu/records/9005efe2d6be4293b63c3cffd4cf193e>

¹⁴<https://github.com/franciellevargas/HateBR>



ofensivo”, “moderadamente ofensivo” e “altamente ofensivo”) e grupos de discursos de ódio (“partidarismo”, “sexismo”, “intolerância religiosa”, “apologia pela ditadura”, “gordofobia”, “homofobia”, “racismo”, “anti-semitismo” e “xenofobia”).

Tal como diversas outras tarefas de aplicação de abordagens de PLN, apesar dos esforços recentes, a detecção de discurso de ódio em português fica muito atrás do inglês (Jahan; Oussalah, 2023). A detecção de discursos de ódio em língua portuguesa é, sem dúvida, uma área promissora de pesquisa no campo do PLN. Redes sociais são um terreno fértil para a disseminação de discursos de ódio em qualquer idioma. Dada a popularidade da língua portuguesa nas redes sociais, tem-se aqui uma área muito fértil para o desenvolvimento de estudos de aplicações de abordagens e PLN. A detecção de discursos de ódio em português envolve desafios únicos, como a diversidade linguística, o uso de gírias e expressões regionais, além das particularidades culturais. Isso torna a pesquisa nessa área empolgante e relevante, não apenas do ponto de vista técnico, mas também do ponto de vista social e ético.

Embora muito do que é visto em discursos tóxicos seja também discurso de ódio, cabe ressaltar que outras formas de toxicidade também são manifestas através de discursos. Há também o que chamamos de linguagem ofensiva. Diferentemente de discursos de ódio, os quais são voltados para indivíduos ou grupos específicos de pessoas com base em características identitárias, a linguagem ofensiva tem a intenção de magoar, insultar ou provocar os sentimentos das pessoas, sem necessariamente ter um objetivo discriminatório. É importante notar que a linha entre discurso de ódio e linguagem ofensiva nem sempre é clara, e o contexto em que o conteúdo é apresentado pode influenciar a percepção do quão prejudicial ele é. Isto é, todo discurso de ódio é uma linguagem ofensiva, mas nem toda linguagem que é ofensiva é também um discurso de ódio. Ambos podem ser prejudiciais e problemáticos em diferentes aspectos, e muitas vezes é necessário avaliar cuidadosamente o conteúdo para entender suas implicações e tomar medidas apropriadas para mitigar seus efeitos negativos. Tal qual os demais discursos considerados tóxicos, é importante também o desenvolvimento de meios de detecção de linguagem ofensiva.

Conjuntos de dados voltados para a detecção deste tipo de linguagem podem ser usados em um contexto que não é necessariamente de ódio. Para o português brasileiro, Trajano; Bordini; Vieira (2023) construíram um conjunto de dados voltados a detecção de linguagem ofensiva, nomeado OLID-Br¹⁵. Inspirado em outros *corpora* similares (do inglês, *Offensive Language Identification Datasets* ou OLID), construídos para outros idiomas, o OLID-Br reúne dados de diferentes fontes: Twitter/X, YouTube, e ainda de outros conjuntos de dados em português anotados com um esquema de anotação distinto do proposto. Os conjuntos de dados utilizados foram o OffComBR de Pelle; Moreira (2017), NCCVG¹⁶ de Nascimento et al. (2019), HLPDSD de Fortuna et al. (2019), e ToLD-Br de Leite et al. (2020). O conjunto de dados OLID-BR contém anotações para cinco tarefas, são elas: (1) classificação de comentário tóxico: classificação binária utilizada para identificar se um comentário é ou não tóxico; (2) detecção do tipo de toxicidade: classificação multi-rótulo que identifica os rótulos de toxicidade presentes em um comentário tóxico; (3) classificação de alvo de toxicidade: classificação binária que prevê se um comentário tóxico é direcionado ou não; (4) identificação do tipo de alvo de toxicidade: classificação multiclasse que identifica o tipo de alvo de um comentário direcionado; e (5) categorização de *spans*: tarefa voltada a detecção de *spans* (parte de um texto) em um comentário tóxico. O conjunto de dados contém 6.354 (extensível para 13.538) comentários rotulados usando um esquema de anotação de três camadas com granulação fina compatível com conjuntos de dados em outros idiomas, o

¹⁵<https://github.com/DougTrajano/olid-br>

¹⁶<https://github.com/LaCAfe/Dataset-Hatespeech>



que permite o treinamento de modelos multilíngues.

23.3.2 Análise de Sentimento

Com a proliferação das redes sociais e das plataformas de avaliação online (tais como: TripAdvisor¹⁷, Booking¹⁸ e Airbnb¹⁹), assim como em diversos sites de e-commerce, uma infinidade de textos opinativos são publicados diariamente. Estes textos têm grande potencial para apoiar os processos de tomada de decisão (Zhang et al., 2023). A Análise de Sentimento (AS) estuda as opiniões, sentimentos, avaliações, apreciações, atitudes e emoções em relação a entidades como produtos, serviços, organizações, indivíduos, problemas, eventos, tópicos e seus diferentes aspectos expressos em textos (Liu, 2012). Nesta área desenvolvem-se aplicações em diversos campos do conhecimento como: política, finanças e marketing²⁰.

Existem muitos nomes e tarefas ligeiramente diferentes, por exemplo, análise de sentimento no nível de aspecto, reconhecimento/classificação de emoções etc. A AS visa encontrar soluções computacionais para extrair e analisar as opiniões das pessoas sobre uma entidade e seus diferentes aspectos (Pereira, 2021). Como as opiniões podem ser categorizadas com polaridades (por exemplo, positivo e negativo), a AS pode ser considerada uma tarefa de classificação de texto (Zhang et al., 2023). E como se trata de uma tarefa de classificação de texto, de acordo com Tan; Lee; Lim (2023), três tipos de abordagens podem ser utilizadas, são elas: AM, Aprendizado Profundo (AP) e Aprendizado Conjunto (AC), usualmente referido como *ensemble learning*. Abordagens baseadas em AM, como classificador Ingênuo de Bayes (em inglês, *Naïve Bayes* ou NB) de Zhang (2004) e Máquina de Vetor de Suporte (em inglês, *Support Vector Machine* ou SVM) de Cortes; Vapnik (1995), usam modelos matemáticos para prever sentimentos. Já, as abordagens baseadas em AP, como Redes de Memória Longa de Curto Prazo (em inglês, *Long Short-Term Memory* ou LSTM) de Hochreiter; Schmidhuber (1997), utilizam Redes Neurais Artificiais para prever sentimentos. O AC combina vários classificadores para obter um melhor desempenho de AS.

No trabalho de Pereira (2021) é apresentada uma pesquisa de AS em língua portuguesa. Nele são apresentados os principais tipos de abordagens de AS, as quais podem ser baseadas em AM (classificação também proposta por Tan; Lee; Lim (2023)), em léxico de sentimento, em conceitos, e híbrida. Abordagens baseadas em AM utilizam algoritmos de AM tradicionais. Já, abordagens baseadas em léxico de sentimento obtêm o grau de polaridade de opinião ou emoção de um léxico de sentimento. As abordagens baseadas em conceito usam redes de conceito (por exemplo: ontologias) para realizar a análise semântica do texto. Por fim, as abordagens híbridas, combinam as abordagens mencionadas anteriormente.

Em geral, a AS tem sido investigada principalmente em três níveis de granularidade: documento, sentença ou aspecto (Liu, 2012). No nível de documento, um sentimento é atribuído ao documento como um todo, por exemplo: o texto “O café da manhã é **incrível**. O hotel é um **ótimo** lugar para relaxar e curtir cada momento.” possui polaridade positiva. No nível de sentença, um sentimento é atribuído a cada sentença do documento, por exemplo: a sentença “O café da manhã é **incrível**.” possui polaridade positiva. No nível de

¹⁷<https://www.tripadvisor.com.br/>

¹⁸<https://www.booking.com/>

¹⁹<https://www.airbnb.com.br/>

²⁰Veja exemplo na área do Direito no Capítulo 22.



aspecto, um sentimento é atribuído a cada aspecto de determinada entidade. É uma análise mais refinada, onde os aspectos podem ser atributos ou componentes de uma entidade, por exemplo: a sentença “O café da manhã é incrível.” possui o aspecto **café da manhã** que possui polaridade positiva.

Abordagens de AS são altamente dependentes do uso de ferramentas de PLN, pois precisam interpretar textos em linguagem natural. Logo, desenvolver soluções específicas para a língua portuguesa está diretamente condicionado ao desenvolvimento de recursos linguísticos para a língua. Segundo Lo et al. (2017), o português é uma das línguas com poucos recursos linguísticos disponíveis, apesar de estar entre as línguas mais utilizadas na Web.

Dentre os recursos para o português brasileiro, podemos citar os léxicos de sentimentos: OpLexicon²¹ (Souza et al., 2011), OpenWordNet-PT²² (De Paiva; Rademaker; Melo, 2012), SentiLex²³ (Silva; Carvalho; Sarmiento, 2012), Reli-Lex²⁴ (Freitas, 2013), Word NetAffect-BR²⁵ (Pasqualotti, 2015), Personalitatem Lexicon (Machado et al., 2015), AffectPT-BR²⁶ (Carvalho; Santos; Guedes, 2018), LexReli (Machado; Pardo; Ruiz, 2018) e Brazilian Portuguese LIWC Dictionary²⁷ (Balage Filho; Pardo; Aluísio, 2013).

- O OpLexicon (Souza et al., 2011) possui 30.322 palavras (23.433 adjetivos e 6.889 verbos) e foi construído com base em um *corpus* do português brasileiro (composto por 346 resenhas de filmes e 970 textos jornalísticos), no *thesaurus* denominado TEP²⁸ (do português, Thesaurus Eletrônico Básico para o Português do Brasil) de Dias-da-Silva; Morales (2003) e no léxico de sentimento de Hu; Liu (2004) traduzido para o português.
- A base de dados da OpenWordNet-PT (De Paiva; Rademaker; Melo, 2012) é o resultado da tradução da base de dados da WordNet de Princeton²⁹, portanto, contém uma base de dados com grande abrangência, possui 62034 sentidos de pares de palavras e 45421 palavras únicas.
- A versão 2 do SentiLex (Silva; Carvalho; Sarmiento, 2012) é composta por 82347 formas flexionadas, organizadas em adjetivos (16863), substantivos (1280), verbos (29504) e expressões idiomáticas (34700).
- O ReLi-Lex (Freitas, 2013) é derivado do *corpus* ReLi de Freitas et al. (2012), que é composto por resenhas de livros publicadas na internet e possui 1600 resenhas de treze livros (sete autores), este léxico contém 609 entradas.
- O WordNetAffect-BR (Pasqualotti, 2015) é um vocabulário de emoções que possui 289 palavras (adjetivos e substantivos).
- O Personalitatem Lexicon (Machado et al., 2015) contém lexemas de conotação afetiva baseada nos traços de personalidade e foi construído com base no *Linguistic Inquiry e Word Count* (LIWC) 2.015.
- O AffectPT-BR (Carvalho; Santos; Guedes, 2018) tem um total de 1.139 palavras atribuídas na categoria “afeto”, 479 em “posemo” e 661 em “negemo”.

²¹<https://www.inf.pucri.br/linatural/wordpress/recursos-e-ferramentas/oplexicon/>

²²<https://github.com/own-pt/openWordnet-PT>

²³<https://b2find.eudat.eu/dataset/b6bd16c2-a8ab-598f-be41-1e7aeecd60d3>

²⁴<https://www.linguateca.pt/Repositorio/ReLi/>

²⁵<https://www.inf.pucri.br/linatural/wordpress/recursos-e-ferramentas/wordnet affect br/>

²⁶<https://github.com/LaCAfe/AffectPT-br>

²⁷<http://143.107.183.175:21380/portlex/index.php/pt/projetos/liwc>

²⁸<http://www.nilc.icmc.usp.br/tep2/>

²⁹<https://wordnet.princeton.edu/>



- O LexReli (Machado; Pardo; Ruiz, 2018) é uma combinação de três léxicos, OpLexicon (Souza et al., 2011), SentiLex (Silva; Carvalho; Sarmiento, 2012) e Brazilian Portuguese LIWC Dictionary (Balage Filho; Pardo; Aluísio, 2013), especializado em identificar a polaridade de aspectos em textos opinativos sobre livros e contém 1.543 entradas.
- O Brazilian Portuguese LIWC Dictionary (Balage Filho; Pardo; Aluísio, 2013) é um léxico disponível para a língua portuguesa, construído a partir do LIWC de Pennebaker; Francis; Booth (2001), ou seja, foi resultado de tradução automática, utilizando diversos dicionários bilíngues português-inglês e possui 127.149 instâncias.

Além dos léxicos há também os *corpora* anotados para a tarefa de AS: ReLi (Freitas et al., 2012), comentários sobre hotéis publicados no TripAdvisor (Freitas, 2015), comentários sobre produtos publicados no Buscapé³⁰ (Avanço; Nunes, 2014), comentários sobre restaurantes (Farias et al., 2016), TweetSentBR³¹ (Brum; Nunes, 2018), UTLCorpus³² (Sousa; Brum; Nunes, 2019) e tweets sobre a pandemia de COVID-19 (Vargas; Santos; Rocha, 2020).

No ano de 2022, foi proposto um desafio sobre AS no nível de aspecto (em inglês, *Aspect-based Sentiment Analysis* ou ABSA) para língua portuguesa no IberLEF³³ denominado ABSAPT³⁴. A proposta do ABSAPT foi inspirada em competições propostas em outros idiomas, como SemEval (Pontiki et al., 2014, 2015, 2016) para o inglês e EVALITA (Mattei et al., 2020) para o italiano. Além disso, tinha como público-alvo acadêmicos, pesquisadores e profissionais de empresas privadas. Na competição participaram cinco equipes de diferentes universidades e institutos do Brasil. O *corpus* disponibilizado na competição foi desenvolvido por Freitas (2015) e Corrêa (2021). Os participantes usaram diferentes tipos de abordagens para resolver a tarefa de ABSA, a qual foi dividida em duas, identificação de aspectos e extração de polaridade destes aspectos. O time da UFSCAR (Assi et al., 2022) propôs uma solução baseada em regras e léxico de sentimento, os times do NILC (Machado; Pardo, 2022) e da UFPR (Heinrich; Marchi, 2022) propuseram soluções baseadas em AM, utilizando algoritmos de AM tradicionais como *Conditional Random Field* (CRF) e os times Deep Learning Brasil (Gomes et al., 2022), PiLN (Neto et al., 2022) e UFPR (Heinrich; Marchi, 2022) propuseram soluções baseada em AP, utilizando Transformers (Silva et al., 2022). Enfim, estratégias como estas, especialmente para línguas com poucos recursos como o português, são extremamente importantes.

23.3.3 Detecção de Notícias Falsas

Uma notícia falsa é uma mensagem transmitida conscientemente por um remetente para promover uma falsa crença ou conclusão por parte do destinatário (Fuller et al., 2006). Segundo Oliveira et al. (2020), a classificação de notícias falsas pode ser vista como uma execução de uma classificação binária entre falso ou verdadeiro. A principal diferença entre a definição dos problemas de classificação de notícias falsas é em função dos diferentes esquemas de anotação ou contextos de aplicação em diferentes conjuntos de dados. Em geral, os dados são coletados de declarações anotadas em sites de verificação de fatos, com os rótulos “verdadeiro” ou “falso”. No Brasil, algumas agências de checagem são: Agência

³⁰<https://github.com/avanco/LBC>

³¹<https://metatext.io/datasets/tweetsentbr>

³²<https://github.com/RogerFig/UTLCorpus>

³³<https://sites.google.com/view/iberlef2022/>

³⁴<https://sites.google.com/inf.ufpel.edu.br/absapt2022/>



Lupa³⁵, Aos Fatos³⁶, Fato ou Fake³⁷ e Comprova³⁸.

A detecção de notícias falsas, também conhecidas como *fake news*, em redes sociais é uma área de pesquisa crítica e desafiadora. A aplicação eficaz de técnicas de PLN nesse contexto é crucial para preservar a integridade da informação online e combater a desinformação que pode trazer consequências sérias tanto na política, quanto na economia, e ainda na sociedade como um todo. O PLN desempenha um papel fundamental no desenvolvimento de abordagens eficazes para lidar com esse problema. Apesar do problema de disseminação de notícias falsas estar presente em todas as redes sociais, algumas tendem a ter o compartilhamento deste tipo de conteúdo mais dissipado. De acordo com Cabral et al. (2021), o Whatsapp facilita a disseminação rápida de desinformação. No Brasil, cerca de 35% das notícias falsas são compartilhadas através do WhatsApp (Newman et al., 2020), e 40,7% destes mensagens são compartilhadas após serem desmentidas (Resende et al., 2019).

Na literatura, encontramos alguns *corpora*, descritos na língua portuguesa, anotados para a tarefa de detecção de notícias falsas em língua portuguesa, são eles: COVID-19³⁹, FakeTweetBr⁴⁰ de Cordeiro; Pinheiro (2019), Fake.br-Corpus⁴¹ de (Monteiro et al., 2018) e FakeWhatsApp⁴² de Cunha (2021). O COVID-19 contém notícias sobre a cura da COVID-19 postadas no Twitter/X. O FakeTweetBr é um *corpus* de notícias falsas também advindo do Twitter/X. O Fake.br-Corpus contém notícias classificadas em seis grandes categorias (política, TV e celebridades, sociedade e notícias diárias, ciência e tecnologia, economia, religião) extraídas do G1⁴³, Folha de São Paulo⁴⁴ e Estadão⁴⁵. O FakeWhatsApp possui mensagens anônimas do WhatsApp de grupos públicos do português brasileiro para detecção automática de desinformação textual e de usuários maliciosos.

Um dos trabalhos recentes que utiliza Aprendizado Profundo (AP) na detecção de notícias falsas é o trabalho de Narde (2021). Nele foram utilizados diferentes modelos (ELECTRA de Clark et al. (2020), RoBERTa de Liu et al. (2021), XLM-R de Conneau et al. (2020), Multilingual BERT de Devlin et al. (2019) e BERTimbau de Souza; Nogueira; Lotufo (2020)) para detectar notícias falsas em redes sociais. O modelo BERTimbau (Souza; Nogueira; Lotufo, 2020) com 6 épocas foi o que obteve acurácia e medida-F superior a todos outros os modelos utilizados nos experimentos, medida-F de 95%.

23.3.4 Detecção de Ironia/Sarcasmo/Humor

Apesar dos avanços na área de Análise de Sentimentos (Seção 23.3.2), ela ainda se depara com vários desafios. Entre eles, destaca-se o entendimento de figuras de linguagem (ironia/sarcasmo/humor). As figuras de linguagem são difundidas em quase qualquer gênero de texto e são especialmente comuns nos textos da Web e das redes sociais, em plataformas como o Twitter/X (Ghosh et al., 2015).

³⁵<https://piaui.folha.uol.com.br/lupa/>

³⁶<https://www.aosfatos.org/>

³⁷<https://g1.globo.com/fato-ou-fake/>

³⁸<https://projeto comprova.com.br/>

³⁹<https://wp.ufpel.edu.br/midiars/datasets/>

⁴⁰<https://github.com/prc992/FakeTweet.Br>

⁴¹<https://github.com/roneysco/Fake.br-Corpus>

⁴²<https://github.com/cabrau/FakeWhatsApp.Br>

⁴³<https://g1.globo.com/>

⁴⁴<http://m.folha.uol.com.br/>

⁴⁵<https://www.estadao.com.br/>



Existe uma linha tênue entre os conceitos de ironia, sarcasmo e humor. Reyes; Rosso; Buscaldi (2012) define a ironia como uma “ligeira fronteira no significado do sarcasmo e da sátira”. Gibbs; Colston (2001) afirmam que o sarcasmo, combinado com jocosidade, hipérbole, perguntas retóricas e eufemismo, são tipos de ironia. Tradicionalmente, a ironia é conhecida como o oposto do significado literal (Grice, 1975).

Dentre os trabalhos aplicados à tarefa de detecção de ironia para a língua portuguesa, podemos citar: Carvalho et al. (2009), Freitas et al. (2014) e Silva (2018). Nos trabalhos de Carvalho et al. (2009) e Freitas et al. (2014) são propostas pistas linguísticas para detectar ironia. Em Carvalho et al. (2009), os autores mostraram que é possível identificar opiniões irônicas em comentários, com precisão relativamente alta (de 45% a 85%), usando padrões linguísticos relativamente simples, tais como: emoticons, expressões onomatopeicas para risos, sinais de pontuação, aspas e interjeições positivas (“viva”, “parabéns”, “força” etc.). Ainda, em Freitas et al. (2014) padrões linguísticos foram aplicados no *corpus* sobre o assunto “fim do mundo” extraído do Twitter/X. Esses padrões estão relacionados ao português brasileiro, mas a metodologia pode ser facilmente transferida para análises em outras línguas. Isso foi feito no trabalho de Freitas; Santos; Deon (2020), no qual um subconjunto de padrões foi testado em *corpora* de diferentes idiomas (inglês, italiano e espanhol). Em Silva (2018), o autor descreve sobre o processo de geração de um *corpus* de ironia para a língua portuguesa, bem como, a criação de um modelo pré-treinado de uma Rede Neural Convolucional (em inglês, *Convolutional Neural Network* ou CNN) para detectar ironia. A CNN foi capaz de adaptar-se e detectar automaticamente as figuras de linguagem em questão. Tal abordagem mostrou-se satisfatória para detecção de ironia, obtendo medida-F de 89,78%.

No ano de 2021, foi proposto um desafio sobre detecção de ironia para língua portuguesa no IberLEF⁴⁶ denominado IDPT⁴⁷. A proposta do IDPT foi inspirada em competições proposta em outros idiomas, como SemEval (Hee; Lefever; Hoste, 2018) para o inglês, IronITA (Cignarella et al., 2018) para o italiano, IroSvA (Bueno et al., 2019) para o espanhol e IDAT (Ghanem et al., 2019) para o árabe. Participaram da tarefa seis equipes de universidades e de empresas de quatro diferentes países: Brasil, China, Portugal e Espanha. Os *corpora* disponibilizados na competição contêm textos (tweets e notícias) sobre diferentes temas. O conjunto de dados de treinamento foi desenvolvido por Freitas et al. (2014), Silva (2018) e Schubert; Freitas (2020). Os participantes usaram abordagens tradicionais de AM (como: SVM, NB e outros) e/ou AP (como: Transformers). Os times que atingiram os melhores resultados foram o BERT4EVER (Jiang et al., 2021) e PiLN (Anchiêta et al., 2021). BERT4EVER (Jiang et al., 2021) utilizou Transformers e obteve uma acurácia balanceada de 92% para conjunto de dados de notícias. Para o conjunto de dados composto por tweets, a equipe PiLN (Anchiêta et al., 2021) utilizou *superficial features* e SVM e obteve uma acurácia balanceada de 52%.

23.4 Considerações Finais

Falar sobre aplicações de PLN em redes sociais é de grande importância por diversas razões. As redes sociais desempenham um papel fundamental na comunicação e interação social na sociedade moderna. Compreender como o PLN é aplicado nessas plataformas é essencial para entender as dinâmicas sociais e o impacto da tecnologia na vida das

⁴⁶<https://sites.google.com/view/iberlef2021/>

⁴⁷<https://sites.google.com/inf.ufpel.edu.br/idpt2021/>



peçoas. Este capítulo forneceu uma visão geral sobre aplicações de abordagens de PLN em conteúdos de redes sociais. Demos ênfase ao desenvolvimento de pesquisas desenvolvidas com foco na língua portuguesa, dado o foco deste livro e de esta ser ainda uma língua com recursos escassos para algumas tarefas. Nessa primeira versão, deixamos de cobrir tópicos relevantes e atuais como reconhecimento/classificação de emoções, rastreo de transtorno mental, e detecção de postura. Reconhecendo a importância destes tópicos, pretendemos contemplá-los na versão seguinte deste livro.



Parte X

Sociedade



Capítulo 24

Questões éticas em IA e PLN

Maria das Graças Volpe Nunes
Tayane Arantes Soares
Mariza Ferro

Toda civilização, coletividade ou sociedade surge a partir do compartilhamento de necessidades, bens e valores comuns. A sobrevivência e a prosperidade de uma sociedade dependem de algum tipo de mediação das diferenças e da regulação do comportamento de seus integrantes. A mediação das diferenças e a regulação da conduta de indivíduos partem de pressupostos. A ética refere-se ao comportamento de indivíduos na tomada de decisões e na sua responsabilização por elas, frente aos valores compartilhados pela sociedade em que vivem.

A partir do momento em que os sistemas de inteligência artificial (IA) passam a fazer parte da sociedade, interagindo com humanos e mimetizando seus comportamentos, tomando decisões com certo grau de autonomia e eventualmente colocando pessoas ou sociedades em risco, problemas de natureza ética naturalmente emergem. Nesse contexto, tem havido uma preocupação crescente com as implicações éticas dos atuais sistemas inteligentes, e a sociedade acadêmica de IA tem se movimentado para alertar e promover mudanças para minimizá-los (“AI and Ethics”, 2023; Coeckelbergh, 2020).

24.1 Ética em IA

Atualmente (2023), a principal tecnologia de IA para dotar seus programas com inteligência caracteriza-se por fornecer, a algoritmos criados para aprender, grandes quantidades de dados sobre aquilo que deve ser aprendido, ou seja, sobre um conceito ou uma tarefa. E, conforme já discutido no Capítulo 14, se esses dados não forem coletados de maneira criteriosa, podem conter vieses que acabam por provocar comportamentos indesejáveis, incorretos ou injustos. A injustiça desses sistemas de IA ocorre muitas vezes por terem sido treinados com dados desbalanceados e sem curadoria, ou por terem aprendido correlações entre os dados que ou são irrelevantes para o conceito que se quer ensinar, ou que carregam algum viés indesejado.

Um exemplo concreto de vieses algorítmicos é relatado no documentário “*Coded Bias*” (Kantayya, 2020). A cientista da computação Joy Buolamwini, uma mulher negra, durante a sua pesquisa sobre softwares de visão computacional no MIT (*Massachusetts Institute of Technology*), não conseguia ter seu rosto reconhecido pelo software no qual estava trabalhando. Somente após colocar uma máscara facial branca que o sistema reconheceu a máscara como sendo um rosto. As pesquisadoras Joy Buolamwini e Timnit Gebru constataram que 79,6% dos dados de treinamento desse software eram compostos por pessoas de pele clara (Buolamwini; Gebru, 2018).



Esse fato evidencia o problema estrutural que permeia a criação de ferramentas de IA, tendo em vista a pouca ou nenhuma reflexão por parte de empresas e de pessoas que desenvolvem essas soluções sobre os impactos sociais que essas ferramentas podem causar, além de pouco envolvimento da sociedade no desenvolvimento dessas soluções tecnológicas (Hora, 2021). O’Neil (2021) relata vários outros exemplos de consequências negativas de se utilizar algoritmos de aprendizado de máquina (AM) em tomada de decisões.

Casos de discriminação de raça são frequentes. A ferramenta *Google Fotos* foi acusada de rotular a imagem de um casal negro como “gorilas”, e a do *Flickr* rotulou fotos de pessoas negras como “macaco” (Cruz, 2021). A pesquisa de Buolamwini; Gebru (2018) também revelou vieses de raça e de gênero em serviços de IA de empresas como Microsoft, IBM e Amazon. O Twitter, em 2020, foi denunciado por priorizar rostos de pessoas brancas na exibição de imagens publicadas pelos usuários (INFOBASE, 2021).

A IA também já foi acusada de impulsionar o ódio às minorias e influenciar os resultados de eleições (Cavaliere; Romeo, 2022), explorar fraquezas psicológicas e orientar decisões (Sartori; Theodorou, 2022), causando problemas como a intensa polarização social e ameaças aos princípios democráticos e aos direitos humanos (Artificial intelligence and human rights., 2021; Empoli, 2019).

Outra característica importante desses algoritmos que aprendem (ao menos na abordagem mais utilizada no ano de 2023, que são as redes neurais artificiais) é que aquilo que aprendem não tem sido recuperável de uma forma que seja compreensível para as pessoas/pesquisadores, ou seja, é impossível recuperar exatamente qual conhecimento foi apreendido pela máquina. Aferimos seu conhecimento apenas pelo seu comportamento numa determinada tarefa. Nesse sentido, dizemos que são obscuros, verdadeiras caixas-pretas, ou seja, não explicáveis. Essa característica indesejável deriva da tecnologia chamada *Deep Learning* (Goodfellow; Bengio; Courville, 2016), que nada mais é do que um modelo especial de redes neurais conhecidas há décadas. Performam bem, é verdade, mas não conseguimos entender como o fazem. Desse modo, a impossibilidade de se justificar, aliada à presença cada vez mais sentida desses sistemas no nosso cotidiano, tem gerado um sentimento de insegurança e desconfiança.

O ChatGPT, da OpenAI (OpenAI, 2022), de enorme repercussão no final de 2022, rapidamente teve sua reputação abalada devido à incapacidade de referenciar com exatidão a fonte de suas respostas (até porque, como foi mencionado anteriormente, é extremamente complicado recuperar com precisão o conhecimento apreendido pelo modelo a partir dos dados de treinamento (Heikkilä, 2021)). Consequentemente, torna-se desafiador determinar a fonte exata das respostas geradas, uma vez que o modelo atua essencialmente como um gerador de palavras prováveis com base em uma entrada inicial. Para algumas situações, em que essas respostas determinariam decisões ou teriam consequências importantes, a falta de confiança no sistema certamente gerou insegurança e afastou alguns usuários. Para saber mais sobre a tecnologia do ChatGPT, sugerimos a leitura de Capítulo 20.

A IA remete a problemas éticos na medida em que se constrói artefatos (sistemas, robôs) que interagem com humanos (de forma direta ou indireta, visível ou ubíqua). Com isso, a IA projeta a possibilidade de uma sociedade mista e comum, de pessoas e máquinas eventualmente autônomas e imprevisíveis interagindo, convivendo e compartilhando os mesmos ambientes. Considerando que sistemas inteligentes tendem a ultrapassar barreiras físicas, sociais, temporais e culturais (tendo em vista que são usados em vários lugares do mundo), devemos nos lembrar que as pessoas que desenvolvem tais sistemas sempre estarão inseridas num contexto cultural e moral específicos, o que pode entrar em conflito com o desenvolvimento ético desses sistemas de IA.



A fim de evitar problemas dessa natureza, em uma sociedade cada vez mais interativa com máquinas de IA, é fundamental investigar maneiras de construir esses artefatos de maneira responsável (Russel, 2019). Caso contrário, essas novas tecnologias continuarão a perpetuar pontos de vista hegemônicos, reforçando e codificando preconceitos e vieses humanos que ainda lutamos para combater (Bender et al., 2021). Nina da Hora, cientista da computação brasileira e pesquisadora na área de Pensamento Computacional, ressalta que, muitas vezes, a busca global pela ética em IA, por ser baseada na tentativa de manter as tecnologias que estão causando problemas, não aprofunda o entendimento e a investigação dos problemas enfrentados pelas pessoas afetadas por essas tecnologias. Segundo a pesquisadora, é necessário ir além dos aspectos técnicos ao buscar um desenvolvimento ético de sistemas de IA, e investigar também o impacto dessas novas tecnologias na vida das pessoas envolvidas (Hora, 2022).

Existem diversas recomendações e tentativas de regulação dos sistemas de IA ao redor do mundo, sendo a da União Europeia uma das pioneiras nesta normatização (Commission, 2021). No Brasil, encontra-se em tramitação na Câmara dos Deputados, desde setembro de 2021, o projeto de Lei 21/20, que estabelece fundamentos, princípios e diretrizes para o desenvolvimento e a aplicação da IA no país, propondo o Marco Legal do Desenvolvimento e Uso da IA¹. Além disso, até o momento existem pelo menos 36 documentos com princípios destinados a fornecer orientações normativas em relação aos sistemas baseados em IA em vários países, nos quais destacam-se os princípios promovidos pela OCDE (Organização para a Cooperação e Desenvolvimento Econômico) para classificação e avaliação de sistemas de IA, que fomenta a universalização de critérios para políticas de IA (OECD, 2022). Vale ainda destacar o documento da UNESCO, aprovado em novembro de 2021, reconhecendo os impactos positivos e negativos da IA nas sociedades e recomendando que os Estados-membros tomem providência quanto à violação de direitos (UNESCO, 2022). O objetivo é sempre recomendar princípios para que os sistemas de IA sejam confiáveis, desenvolvidos e utilizados para o bem da humanidade e do planeta e para preservar os valores por meio da proteção, promoção e respeito aos direitos humanos fundamentais, à liberdade e à igualdade.

A utilização de sistemas de IA pode afetar negativamente vários direitos fundamentais - estabelecidos pela Declaração Universal dos Direitos Humanos, adotada e proclamada pela Assembleia Geral das Nações Unidas (UNICEF, 1948) ou por instrumentos particulares de cada país, como a Constituição Brasileira ou a Carta dos Direitos Fundamentais da União Europeia. Os direitos fundamentais são inerentes a todos os seres humanos, independentemente da sua raça, sexo, nacionalidade, etnia, idioma, religião ou qualquer outra condição. Os direitos humanos incluem o direito à vida e à liberdade, liberdade de opinião e expressão, o direito ao trabalho e à educação, entre outros.

Entre os princípios mais comuns que norteiam as regulações e as recomendações para o desenvolvimento e o uso da IA ética e confiável estão a (1) justiça, diversidade e não discriminação, (2) transparência e explicabilidade, (3) robustez técnica e segurança, (4) privacidade e proteção de dados, (5) responsabilidade e prestação de contas².

1. Os princípios da justiça, diversidade e não discriminação estão intimamente ligados à promoção da justiça social e a salvaguardar a equidade e a não discriminação de qualquer tipo (gênero, raça, cor, nacionalidade, religião, língua, idade, opinião

¹<https://www.camara.leg.br/propostas-legislativas/2236340>

²Baseando-se em pontos comuns encontrados nos documentos supracitados da OCDE, UNESCO e regulamentação da UE.



política etc.), em conformidade com o direito internacional. O objetivo é garantir a distribuição igual e justa de benefícios e custos e garantir que indivíduos e grupos estejam livres de preconceitos, injustiças, discriminação e estigmatização. Ainda, minimizar e evitar reforçar ou perpetuar resultados discriminatórios ou tendenciosos (enviesados), ao longo do ciclo de vida dos sistemas de IA (Smith; Rustagi, 2020). Se preconceitos e injustiças não puderem ser evitados, os sistemas de IA podem aumentar a desigualdade social. Além disso, o uso de sistemas de IA nunca deve induzir as pessoas a serem enganadas ou prejudicar sua liberdade de escolha.

Dependendo da forma como é criada e utilizada, a IA tem potencial para criar e/ou reforçar vieses humanos. O viés pode entrar no desenvolvimento e uso de um sistema de IA, especialmente por meio do uso dos algoritmos de aprendizado de máquina durante a geração, a coleta, a rotulagem e o gerenciamento dos dados com os quais o algoritmo aprende; mas também pode ser introduzido durante o design e a avaliação dos algoritmos (Smith; Rustagi, 2020). Já existem muitos exemplos do uso de sistemas que utilizam AM, os quais, com base nos dados que recebem, têm apresentado resultados tendenciosos, imprecisos e injustos, os quais [representam riscos imensos para indivíduos e empresas.

São diversas as situações de discriminação de raça e de vieses e discriminação aos grupos minoritários ou culturas, principalmente com o uso de algoritmos de reconhecimento facial ou manipulação de imagens. Por exemplo, a dificuldade em reconhecer rostos de pessoas negras, como os exemplos mencionados no início deste capítulo, com a classificação automática de fotos de pessoas negras como “gorilas”; aplicativo que “desnudava” mulheres mostrando como as *deepfakes* prejudicam os mais vulneráveis (REVIEW, 2022); aplicativos que transformam fotos em caricaturas onde os avatares das mulheres, especialmente orientais, são “pornificadas”, enquanto o dos homens são astronautas, exploradores e inventores (Heikkilä, 2022). Além dos casos das mídias sociais, o Brasil vem sofrendo uma profusão de denúncias com o uso de sistemas de reconhecimento facial que levaram a abordagens policiais e até prisões. A Rede de Observatórios da Segurança monitorou, entre março e outubro de 2019, os casos de prisões e abordagens com o uso de reconhecimento facial em cinco estados brasileiros e revelou que 90,5% dos presos por monitoramento facial no Brasil são negros (Nunes, 2019).

Há duas formas para minimizar esses vieses: fazer com que os dados de treinamento reflitam fielmente o universo de situações a que o sistema final será exposto, e detectar, ainda em período de testes, os desvios prováveis e eliminá-los antes de colocar o sistema em uso. Considerando a natureza da tecnologia mais comum hoje em IA – o treinamento de redes neurais – as duas formas são de difícil execução. Quer seja porque nem sempre todos os dados são acessíveis, quer seja porque os testes são incapazes de prever todas as possibilidades, dada a complexidade da tarefa a ser realizada pelo sistema.

2. Os princípios da transparência e explicabilidade são fundamentais para desenvolver e manter a confiança dos usuários nos sistemas de IA. Isso significa que os processos precisam ser transparentes, ou seja, o objetivo dos sistemas de IA deve ser comunicado abertamente e o usuário deve saber que está em contato com um produto ou serviço fornecido diretamente ou com o auxílio de sistemas de IA. Além disso, as decisões tomadas pelo sistema – na medida do possível – devem ser explicáveis aos



afetados direta ou indiretamente. Nem sempre é possível explicar por que um modelo gerou uma determinada saída ou decisão (e qual combinação de fatores de entrada contribuiu para isso). Esses casos são chamados de algoritmos de “caixa-preta” como os já mencionados algoritmos de redes neurais e suas redes neurais profundas. A comunidade de IA já se mobiliza em direção a tornar seus sistemas mais compreensíveis para seus usuários. A IA explicável (*Explainable AI, XAI*) é uma recente área de pesquisa que tem como objetivo propor processos e métodos para tornar os recentes sistemas de aprendizado de máquina mais compreensíveis e confiáveis. Não é tarefa fácil, mas, juntamente com esforços para combinar aprendizado de máquina com métodos simbólicos de representação de conhecimento, é esperado que testemunharemos avanços nessa área.

3. Um componente crucial para alcançar uma IA confiável é a robustez técnica, que está intimamente ligada ao princípio da prevenção de danos. A robustez exige que os sistemas de IA se comportem conforme o planejado, sejam desenvolvidos com uma abordagem preventiva aos riscos, minimizando danos não intencionais e inesperados e evitando danos inaceitáveis. Além disso, vulnerabilidades a ataques (riscos de segurança) devem ser evitadas e eliminadas durante o ciclo de vida dos sistemas de IA para garantir proteção e segurança humana e ambiental.
4. A privacidade e a proteção de dados devem ser respeitadas, protegidas e promovidas ao longo do ciclo de vida dos sistemas de IA. É importante que os dados destinados aos sistemas de IA sejam coletados, utilizados, compartilhados, arquivados e apagados de modo compatível com os marcos jurídicos nacionais, regionais e internacionais relevantes. Por exemplo, na legislação brasileira, a LGPD (Lei Geral de Proteção aos Dados) estabelece diretrizes para o uso dos dados pessoais (LGPD, 2018). A lei similar europeia, GDPR, vai além e trata também do direito de que dados pessoais sejam apagados das bases de dados a qualquer momento, tratado como o direito ao esquecimento pelos modelos de IA.

O uso de dados pessoais ocorre em diferentes contextos. Exemplos incluem o uso de dados para a identificação de supostas emoções para análise do comportamento do usuário. No caso de um cliente, por exemplo, para criar publicidade direcionada durante as compras com foco nos produtos ou em sua disposição na loja virtual, sem a transparência desejada. A ViaQuatro, empresa que tem a concessão da linha 4-amarela do metrô de São Paulo, foi processada pelo Instituto Brasileiro de Defesa do Consumidor por usar câmeras que coletavam dados referentes às “emoções” dos passageiros, e que seriam usados pela companhia, sem o consentimento dos passageiros. “O sistema inteligente conseguiria identificar se o passageiro está feliz, insatisfeito, surpreso e neutro. Além disso, detectaria o gênero e a faixa etária das pessoas. Os dados capturados seriam usados para a empresa fazer a gestão de seu conteúdo institucional e até de anúncios publicitários” (Cruz, 2018). Mais um exemplo de uso sem transparência e desrespeito à privacidade e a proteção dos dados pessoais.

5. A responsabilização e a prestação de contas complementam os princípios acima e estão intimamente ligadas ao princípio da justiça ao tentar garantir mecanismos que determinem as responsabilidades éticas e jurídicas pelas decisões e ações de alguma forma baseadas em um sistema de IA e seus resultados, antes e depois do seu desenvolvimento, implantação e uso.



Apesar de ainda não haver uma regulamentação direcionada aos sistemas de IA no Brasil, no exemplo supracitado com a ViaQuatro, a responsabilização veio por meio da legislação vigente no Brasil. A captação de dados sensíveis, como biométricos, precisa de consentimento do usuário, de acordo com a LGPD e a finalidade da coleta também deve ter propósitos legítimos e comunicados aos titulares dos dados. Além disso, o direito à informação dos consumidores está consagrado como um princípio fundamental ao abrigo da lei do consumidor.

Porém, em casos não cobertos por outras leis, a responsabilização por danos causados por sistemas de IA ainda carece de lei própria. O problema é que o avanço tecnológico ocorre num ritmo muito mais rápido do que aquele da política e da justiça. Enquanto se discutem as leis para regular esta IA de hoje, ela continua avançando modificando nossa forma de interagir com ela e por meio dela, e antes que seja regulamentada por leis, ela já será outra.

A grande questão é se, um dia, um sistema de inteligência artificial estará programado para avaliar adequadamente as informações recebidas e as possíveis consequências que suas ações são capazes de causar ao ambiente e aos seres à sua volta. Será possível programá-lo para embasar suas decisões à luz de valores humanos a fim de exibir um comportamento ético?

Essa possibilidade esbarra em várias dificuldades, como a definição dos valores responsáveis por um comportamento ético, sua representação (seja de forma explícita ou por meio de exemplos), seu processamento por um algoritmo e sua incorporação por um sistema de inteligência artificial.

24.2 Ética em PLN

O PLN assume um papel importante no âmbito das questões éticas relacionadas aos sistemas de IA, sobretudo quando almejamos uma construção responsável, porque é ele que permite a interação entre humanos e máquinas de forma natural. Consequentemente, conhecimentos linguísticos também são relevantes para o desenvolvimento de sistemas inteligentes. Compreender a linguagem, suas variações, suas mudanças, seu papel na comunicação humana e na sociedade pode nos auxiliar na construção de tecnologias melhores e mais inclusivas (Bender, 2020).

Muitas línguas até então desfavorecidas de recursos tecnológicos, como o português, e também línguas minoritárias têm se beneficiado com sua inserção no mundo digital. No entanto, ainda temos um longo caminho a percorrer. Segundo Emily Bender, 90% das línguas do mundo e suas variedades usadas por mais de um bilhão de pessoas têm pouco ou nenhum suporte em termos de tecnologia linguística, reforçando a ideia de que essas novas tecnologias apresentam um potencial excludente (Bender, 2020).

Nesse sentido, ter em mente que vivemos em um mundo diverso linguisticamente é importante, principalmente para atuais e futuras pessoas desenvolvedoras e pesquisadoras em PLN. Não é razoável aceitar o inglês, idioma dos dados de treinamento da maioria dos modelos de língua, como representativo de toda variedade linguística e cultural existente. Ao trabalharmos com PLN, não podemos esquecer como linguagem e poder estão atrelados, que a linguagem cria o nosso mundo e molda nossa realidade (HALLIDAY; MATTHIESSEN, 1999). Portanto, o desafio de romper com esse monopólio linguístico e desenvolver modelos a partir de dados coletados de forma responsável, validados, balanceados e com menos



vieses em outras línguas, precisa ser aceito por mais pessoas, a fim de nos aproximarmos do ideal ético de construção dessas ferramentas.

É fato que o PLN tem se beneficiado muito com o uso de aprendizado de máquina. Muitas barreiras foram transpostas ao representar alguns fenômenos linguísticos por meio de exemplos. Para as tarefas clássicas de PLN – *taggers*, *parsers*, reconhecedores de entidades nomeadas, entre outras – os sistemas construídos por AM parecem cada vez melhores à luz de avaliações padronizadas. O problema que se coloca é o uso futuro desses sistemas, suas combinações e suas aplicações fora de qualquer controle (Chandran, 2023).

Quando presente em sistemas inteligentes, a língua tem papel fundamental. Muitos sistemas de IA são treinados com dados linguísticos (texto ou fala). Em sistemas de conversação, como os *chatbots*, a língua é fundamental. Nos sistemas de IA mais recentes, a competência linguística é adquirida por meio de treinamento com *corpora* muito grandes, gerando um modelo de língua, ou seja, um sistema capaz de prever qual(is) palavra(s) deve(m) seguir a última palavra vista. São os chamados LLM (*Large Language Models*), apresentados em Capítulo 15. O ChatGPT é um exemplo muito conhecido dessa tecnologia.

Se considerarmos que os *corpora* de treinamento de grandes modelos de língua tendem a ser compostos por uma quantidade massiva de dados linguísticos coletados na internet, e que o acesso à internet é desigual, os dados de treinamento têm grandes chances de não serem representativos e não levarem em conta a diversidade cultural e linguística existentes. Como discutido no Capítulo 23, manifestações carregadas de ofensas, preconceitos, discriminação, posturas antiéticas em geral são eventualmente reproduzidas nos textos gerados por esses modelos, manifestando um comportamento antitético do sistema (Perrigo, 2023).

Além disso, a popularização de modelos de língua, como o ChatGPT, tem suscitado questões importantes no âmbito da ética em PLN, especialmente no que diz respeito à propriedade intelectual e aos direitos autorais. Embora esses dados estejam disponíveis publicamente, uma preocupação fundamental está relacionada à forma como os dados são coletados para o treinamento desses modelos, considerando se as pessoas que produziram esses dados tinham ciência de que suas postagens textuais poderiam ser utilizadas como insumos para modelos de linguagem (Alisson, 2023). A pergunta que fica é: como podemos garantir que princípios éticos de transparência, proteção de dados e consentimento serão respeitados nesse processo de coleta?

24.3 Modelos de língua como fonte de conhecimento?

Modelos de língua nos surpreendem ao gerarem textos coerentes, muitas vezes, indistinguíveis de textos produzidos por seres humanos. No entanto, esses textos não passam de seqüências de palavras, de frases prováveis estatisticamente em um dado idioma que foram “cuspidas” pelo modelo a partir de alguma entrada textual. Os modelos em si não entendem os textos gerados. Os modelos apenas apreendem, a partir do *corpus* de treinamento, os padrões (combinações frequentes) linguísticos derivados dos dados, e reproduzem, como bons papagaios estocásticos, esses padrões em novas saídas (Bender, 2023).

Nesse sentido, se a língua é apreendida a partir de um *corpus*, nos modelos de línguas, as características desse *corpus* são determinantes para a qualidade linguística do que será gerado pelo sistema (Capítulo 14). Isso soa óbvio, mas, em se tratando de língua, há uma outra consequência. Em sistemas conversacionais, como os *chatbots*, a linguagem produzida por um sistema tem um efeito: ela estará atendendo a alguma expectativa do usuário, que pediu uma informação, ou uma sugestão, ou se queixou de algo, ou quer simplesmente



dialogar. Não basta, portanto, que a expressão linguística cumpra todos os requisitos de ortografia, gramática, coesão e coerência. É preciso atender a outros critérios. Não é rara a geração de uma expressão linguística correta e elegante, com um conteúdo ou uma informação incorreta ou enviesada pelos dados. Detectar essa imprecisão, no entanto, pode não ser tão fácil. Um interlocutor do *chat*, impressionado pela boa forma do texto, pode aceitar como verdade, sem questionar, o conteúdo expresso por ela. Acontece que um modelo de língua não é capaz de preencher os requisitos relativos à autenticidade e veracidade das expressões que gera. É o típico exemplo de uma ferramenta incrível de geração de língua sendo usada para um fim para o qual não foi projetada. Como são capazes de gerar uma infinidade de expressões linguísticas, tem-se a impressão de que, de fato, têm domínio em várias áreas de conhecimento e tarefas. A consequência é que suas “alucinações” podem ser confundidas com novas “verdades”, oferecendo um risco enorme à sociedade, na medida em que a crença nessas verdades pode levar a comportamentos imprevisíveis.

Tão logo foi disponibilizado o ChatGPT, em 2022, as consequências desse cenário têm sido discutidas por vários setores das sociedades em todo o mundo, incluindo o Brasil. Já se prevê mudanças no trabalho em toda sorte de setores que usam informações para tomada de decisão, bem como aqueles que têm a redação de textos como atividade relevante. Incluem-se, portanto, o jornalismo, a educação formal, a pesquisa, o direito, apenas para citar alguns. Percebe-se aí o perigo de se utilizar um sistema impróprio como se fosse um “gerador de conhecimento”.

Temos testemunhado que sociedades cada vez mais tecnológicas suscitam muitas questões de natureza ética. A velocidade com que os sistemas computacionais – em especial, os ditos inteligentes – evoluem tem nos mostrado que precisamos nos antecipar, de alguma forma, aos riscos que eles podem representar. Para alcançarmos um desenvolvimento e uma utilização ética e responsável de sistemas de IA, precisamos contar com esforços coletivos e transdisciplinares, além de um diálogo constante entre governo, empresas, especialistas e sociedade em geral. Nesse sentido, é fundamental promover debates mais amplos e plurais sobre os impactos dessas novas tecnologias, a fim de pensarmos de forma conjunta aplicações positivas dessas ferramentas em nossa sociedade.



Capítulo 25

E agora, PLN?

Maria das Graças Volpe Nunes

Neste último Capítulo elencamos alguns desafios e perspectivas para o PLN em língua portuguesa e finalizamos com uma discussão sobre os limites atuais do PLN.

25.1 Desafios e perspectivas para o PLN-Português

Por razões históricas e econômicas, os sistemas atuais de PLN “estado da arte” são muito mais comuns em inglês do que em qualquer outra língua. Enquanto que outras comunidades têm adaptado para suas línguas os sistemas originalmente criados para o inglês (por meio de novos treinamentos, mas com aproveitamento de parâmetros), comunidades linguísticas minoritárias e comunidades linguísticas de países menos desenvolvidos são invisibilizadas no mundo digital, com consequências negativas e diretas na sua economia e desenvolvimento.

Segundo o Instituto Camões, em 2022, a comunidade de falantes de português no mundo era estimada em cerca de 260 milhões de pessoas (3,7% da população mundial) sendo o quarto idioma mais usado, depois do mandarim, inglês e espanhol. Contudo, essa representatividade não é contemplada no estado da arte da ciência, que está majoritariamente nas mãos de instituições e organizações não falantes do português. Pesquisadores brasileiros e portugueses têm levantado a necessidade de unir forças para colocar o português no lugar de destaque que ele merece¹.

O processamento do português brasileiro tem avançado de maneira consistente desde meados da década de 1990, principalmente a partir do uso de AM e de abordagens *cross-language* e multilíngue, que facilitam a construção rápida de recursos e soluções, e permitem a geração de uma aplicação em uma língua a partir de uma aplicação em outra língua. Mas ainda é precária a união de esforços entre os países da Comunidade de Países de Língua Portuguesa (CPLP), que inclui Portugal, Angola, Moçambique, Cabo Verde, Guiné-Bissau, São Tomé e Príncipe, além do Brasil. Se as diferenças linguísticas entre os diferentes idiomas representam barreiras para a criação de sistemas comuns, não há dúvida de que a união de esforços trará benefícios para todos. Por ora, o esforço mais visível é aquele entre os mais fortes do grupo, Brasil e Portugal, que realizam um evento científico bianual comum, o PROPOR², e mantêm vínculos acadêmicos há várias décadas. Dois grandes repositórios de recursos e ferramentas linguístico-computacionais do Português, que pretendem abranger as diversas comunidades de língua portuguesa são a Linguateca³ e o Portulan Clarin⁴.

¹<https://www.publico.pt/2023/02/09/opiniao/opiniao/lingua-portuguesa-tecnologia-futuro-2038078>

² CE-PLN. PROPOR (*International Conference on Computational Processing of Portuguese Language*). Disponível em: <https://sites.google.com/view/ce-pln/eventos/propor>.

³ <https://www.linguateca.pt/>

⁴ <https://portulanclarin.net/>



Em países extensos como o Brasil, onde há uma grande variedade linguística, a exemplo das diferentes línguas indígenas faladas em território nacional⁵, das variações dialetais e sociais e dos sotaques regionais do português brasileiro, suas riquezas e diversidades linguísticas dificilmente são representadas nos *corpora*. Essa sub-representação nos dados de treinamento de modelos de aprendizado de máquina é um dos fatores que contribuem para aumentar a codificação de vieses por esses sistemas. Percebe-se, portanto, a importância de os dados linguísticos que alimentam tais sistemas serem coletados de forma responsável, buscando representar as variações linguísticas e idiomáticas das línguas faladas no país.

Um dos primeiros *corpora* em português brasileiro usado para treinar um modelo de língua é o BrWac (*Brazilian Portuguese Web as corpus*), composto por 3,53 milhões de documentos da web, totalizando 2,68 bilhões de *tokens*, com acesso público para pesquisadores⁶. Já o *corpus* Carolina, do Centro de IA, C4AI⁷, é, de acordo com os autores, “um corpus com um volume robusto de textos em Português Brasileiro contemporâneo (1970-2021), com informações de procedência e tipologia. O corpus está disponível em acesso aberto, para download gratuito, desde 8 de março de 2022. A versão atual, Ada 1.2 (8 de março de 2023), tem 823 milhões de *tokens*, mais de dois milhões de textos e mais de 11 GBs”⁸. Esse *corpus* é um importante passo para o treinamento de LLM do português brasileiro, e tem o mérito de incluir uma grande variedade de gêneros (jornalismo, literatura, poesia, judiciário, wikis, mídia social, legislativo, acadêmico etc.), mas ainda não contempla as diversidades regionais e culturais dessa língua, meta perseguida pelo C4AI com a construção do *corpus* de fala (transcrições) TaRSila⁹, previsto para contemplar os diferentes dialetos brasileiros. Todos esses *corpora* pretendem ser variados quanto a gênero textual e domínio.

No mesmo C4AI, o projeto PROINDL¹⁰ promete usar a IA em parceria com comunidades indígenas para o desenvolvimento de ferramentas que promovam a preservação, revitalização e disseminação de línguas indígenas do Brasil. Um dos objetivos é explorar as técnicas que utilizam poucos dados para criar tradutores automáticos tanto para texto como para fala, além de outras aplicações.

Mesmo com a limitação de variedade e tamanho de *corpora* em português para treinamento de LLMs, grandes modelos de língua para o português já são encontrados, quer sejam modelos com capacidade multilíngue (ex. os modelos PALM da Google), quer sejam treinados apenas em português (ex. BERTimbau(Souza; Nogueira; Lotufo, 2020), Sabiá (Pires et al., 2023), Albertina¹¹). Dessa forma, são claros os avanços em direção a produtos para a língua portuguesa. No entanto, o que pode parecer simples (*corpus* + redes neurais e Transformers + *fine-tuning* = LLM) pode ser, de fato, inviável. O custo de se produzir um LLM de qualidade é extremamente alto. Um ótimo LLM, como o LLaMA-65B, por exemplo, foi pré-treinado com 1.4 trilhão de palavras, em 40 mil GPU¹²-horas, consumindo energia equivalente ao consumo de cerca de 10 casas brasileiras em um ano¹³.

⁵ <https://www.gov.br/funai/pt-br/assuntos/noticias/2022-02/brasil-registra-274-linguas-indigenas-diferentes-faladas-por-305-etnias>

⁶ <https://www.inf.ufrgs.br/pln/wiki/index.php?title=BrWaC>

⁷ <https://c4ai.inova.usp.br/pt/sobre/>

⁸ <https://sites.usp.br/corpuscarolina/>

⁹ TaRSila. Disponível em: <https://sites.google.com/view/tarsila-c4ai>.

¹⁰ https://c4ai.inova.usp.br/pt/pesquisas/#PROINDL_port

¹¹ Família de modelos treinados para as variantes europeia e brasileira do português disponível em: <https://huggingface.co/PORTULAN/albertina-ptbr-nobrwaC>.

¹² Graphics Processing Unit – unidade de processamento gráfico.

¹³ https://www.youtube.com/watch?v=prJrQ8XL-AY&ab_channel=BrasileirasemPLN



De um lado, são necessárias muitas GPUs para treinar modelos competitivos: quanto maior o número de GPUs, mais parâmetros podem ser usados no modelo, aumentando sua eficácia numa tarefa. Atualmente, poucas instituições públicas ou privadas dispõem de infraestrutura para tal e, ainda assim, com número de GPUs bastante inferior (de 2 a 100) àquela disponível em nuvem (clusters de TPUs¹⁴) com preços de aluguel que podem chegar a um milhão de dólares. Pesquisadores costumam recorrer a recursos gratuitos e temporários oferecidos pelas gigantes internacionais (ex. Google Cloud). Essa dependência externa por recursos essenciais ao desenvolvimento tecnológico só pode ser minimizada por meio de ações e investimentos governamentais (p.ex. centralizados pelo CNPq) ou por iniciativas coletivas dos detentores de recursos no sentido de juntá-los para incrementar o poder computacional e compartilhá-lo com toda a comunidade. De outro lado, independentemente do fator financeiro, temos o custo energético, com efeito na emissão de carbono, que, como vimos, não é desprezível.

Essas questões nos fazem refletir sobre os próximos caminhos a seguir. Nem tudo se resolve com grandes modelos de língua, assim como há muitas aplicações interessantes que podem ser desenvolvidas ou com modelos mais modestos ou por meios distintos dos modelos de língua. Considerando tarefas e domínios de conhecimento particulares, é possível construir soluções a partir de modelos treinados apenas nesse domínio. De fato, os resultados tendem a ser melhores do que com o uso de modelos mais genéricos. Além disso, considerar uma tarefa mais específica pode levar a uma solução - qualquer que seja a abordagem - mais eficaz.

As limitações para a academia não impedem, no entanto, que o PLN seja cada vez mais usado por empresas e startups da área, cujo número vem crescendo muito em nosso país. Certamente isso é fruto da alta demanda por sistemas dessa natureza, mas também do investimento das universidades públicas na formação de recursos humanos nessa área. Estamos vivendo um momento de grande absorção dos profissionais de PLN pelo mercado. Mais um motivo para refletirmos sobre a formação desses profissionais frente aos grandes desafios que essa área (e a IA de modo geral) nos coloca.

Além de todas as questões levantadas anteriormente, vale ressaltar a relevância de se adequar os critérios de avaliação tradicionalmente usados para sistemas de IA e, em particular, de PLN, à nova realidade das aplicações oferecidas à sociedade. A cultura acadêmica sugere uma avaliação em cenários rigidamente controlados, usando apenas métricas objetivas (numéricas), visando quase que exclusivamente a comparação com outros sistemas. Assim é a ciência e assim ela evolui. No entanto, tendo em vista o alcance que as novas tecnologias têm na sociedade, é urgente que os métodos de avaliação considerem critérios de outras naturezas, critérios que ajudem a prever o comportamento do sistema em situações, de fato, reais, sabidamente complexas, onde a imprevisibilidade é um fator relevante.

25.2 Há limites para o PLN?

A língua é frequentemente citada como sinal de inteligência e, por isso, nos distinguiria de outros animais. É por essa razão, aliás, que o PLN sempre esteve ligado à área de Inteligência Artificial. Sistemas dotados de habilidades linguísticas estariam entre os (artificialmente) inteligentes. No entanto, inteligência é algo difícil de se definir. Apenas

¹⁴TPUs (Unidades de Processamento de Tensor) são aceleradores de treinamento e geração de modelos de machine learning.



parecer inteligente nos faz inteligentes? Essa questão sempre esteve presente na IA. Como definir um sistema inteligente? É necessário que ele raciocine como os humanos (seja bioinspirado), que tenha conhecimento explicitamente representado em seus algoritmos, ou basta que suas respostas sejam similares às de um humano nas mesmas situações? Não há acordo sobre isso, até porque sequer conseguimos concordar com os critérios de classificação de inteligência humana.

No caso do PLN, isso se traduz na seguinte questão: aos sistemas que mostram habilidade linguística pode-se atribuir inteligência? Ainda: eles de fato dominam o conhecimento total sobre a língua e todos os fenômenos que a língua em uso nos apresenta?

A língua tem sido objeto de estudo, análise e fascínio nas mais variadas áreas do conhecimento: filosofia, literatura, linguística, psicologia, psicanálise, ciências cognitivas, comunicação social, entre outras, e, recentemente, do PLN. Isso revela que a língua é um objeto de estudo bastante rico e complexo, e, portanto, não é possível abordá-lo segundo uma única disciplina.

O PLN tem sido apresentado como uma área comum a duas disciplinas, Computação e Linguística. No passado, isso parecia suficiente, pois apenas a porção formal, estrutural da língua era tratada computacionalmente¹⁵. Com o passar do tempo, a evolução das máquinas e as redes sociais, isso mudou. Essa língua em uso no cenário digital atual só pode ser tratada de forma transdisciplinar. Não é um caminho simples, nem cômodo, nem garantidor de que o PLN terá sucesso. Pelo contrário, não é improvável que, ao tratar a língua em toda sua complexidade, concluamos que há um limite para o PLN que independe de avanços tecnológicos.

Os capítulos anteriores evidenciam que PLN é uma área de grande potencial, porém repleta de desafios, sobre os quais é difícil fazer previsões. Várias tarefas de IA têm sido solucionadas pelas tecnologias atuais (Redes Neurais, Aprendizado de Máquina) que não são ideias novas; elas ficaram adormecidas até que o hardware das máquinas pudesse processá-las eficientemente. Em se tratando de PLN, no entanto, não é razoável prever que avanços de hardware, ou mesmo de métodos, garantam a solução completa para todos os sistemas que envolvem a língua. A demanda por sistemas que processam a língua não para de crescer. Vale notar que demandas e métodos são interdependentes: enquanto as demandas provocam novos métodos, estes últimos abrem caminho para novas demandas antes não possíveis.

Este livro também evidenciou que o desempenho linguístico dos sistemas atuais de PLN espelham aquilo que aprendem a partir dos dados de treinamento dos algoritmos de aprendizado: língua na norma culta, língua mal formada, discursos de ódio, misoginia ou racismo; o que quer que tenha sido oferecido ao algoritmo de aprendizado a título de exemplo eventualmente será reproduzido pelo sistema gerado. Como o conhecimento (a língua) adquirido nesses sistemas não é explicitamente representado (ele está imerso em valores probabilísticos ou parâmetros numéricos das redes neurais), não há um controle de quando e como ele será usado. Todos esses efeitos colaterais dessa tecnologia preocupam a sociedade e trazem para a comunidade de PLN desafios e responsabilidades não existentes antes. As trajetórias da IA e do PLN têm nos ensinado que o alcance de metas mais modestas e realistas, ao longo do tempo, tem nos levado a patamares cada vez mais surpreendentes.

Convidamos você a esperar para ver, ou fazer para acontecer.

¹⁵A rigor, somente a parte formal da língua é passível de processamento pela máquina. Toda tentativa de alcançar o extralinguístico trata-se apenas de uma aproximação.



Referências

- ABADJI, J. et al. **Towards a Cleaner Document-Oriented Multilingual Crawled Corpus**. Proceedings of the Thirteenth Language Resources and Evaluation Conference. **Anais...**Marseille, France: European Language Resources Association, jun. 2022. Disponível em: <<https://aclanthology.org/2022.lrec-1.463>>
- ABNEY, S. P. Parsing By Chunks. Em: BERWICK, R. C.; ABNEY, S. P.; TENNY, C. (Eds.). **Principle-Based Parsing: Computation and Psycholinguistics**. Dordrecht: Springer Netherlands, 1992. p. 257–278.
- ABREU, S. C. DE; VIEIRA, R. Relp: Portuguese open relation extraction. **KNOWLEDGE ORGANIZATION**, v. 44, n. 3, p. 163–177, 2017.
- AFANTENOS, S.; ASHER, N. **Counter-argumentation and discourse: A case study**. Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing. **Anais...**CEUR Workshop Proceedings, 2014.
- AFONSO, S. et al. **Floresta Sintá(c)tica: A treebank for Portuguese**. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02). **Anais...**Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA), maio 2002. Disponível em: <<http://www.lrec-conf.org/proceedings/lrec2002/pdf/1.pdf>>
- AGHAJANYAN, A.; GUPTA, S.; ZETTLEMOYER, L. **Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning**. (C. Zong et al., Eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. **Anais...**Association for Computational Linguistics, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.acl-long.568>>
- AGICHTEIN, E.; GRAVANO, L. **Snowball: Extracting relations from large plain-text collections**. Proceedings of the fifth ACM conference on Digital libraries. **Anais...**2000.
- AGIRRE, E. Cross-Lingual Word Embeddings. **Computational Linguistics**, v. 46, n. 1, p. 245–248, mar. 2020.
- AHN, L. VON; KEDIA, M.; BLUM, M. **Verbosity: A Game for Collecting Common-Sense Facts**. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. **Anais...**: CHI '06. New York, NY, USA: Association for Computing Machinery, 2006. Disponível em: <<https://doi.org/10.1145/1124772.1124784>>
- AI and Ethics**. Springer, 2023. Disponível em: <<https://link.springer.com/journal/43681/volumes-and-issues>>. Acesso em: 7 abr. 2023
- AJAY, H. B.; TILLET, P.; PAGE, E. B. **Analysis of essays by computer (AEC-II)**. Storrs, CT: University of Connecticut, 1973.
- AKÇAY, M. B.; OĞUZ, K. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. **Speech Communication**, v. 116, p. 56–76, 2020.
- ALAM, T.; KHAN, A.; ALAM, F. **Punctuation Restoration using Transformer**



- Models for High-and Low-Resource Languages.** Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020). *Anais...Online: Association for Computational Linguistics*, nov. 2020. Disponível em: <<https://aclanthology.org/2020.wnut-1.18>>
- ALCAIM, A.; SOLEWICZ, J. A.; MORAES, J. A. DE. Frequência de ocorrência dos fonemas e listas de frases foneticamente balanceadas no português falado no Rio de Janeiro. *Journal of Communication and Information Systems*, v. 7, n. 1, 1992.
- ALEIXO, P.; PARDO, T. A. S. **CSTTool: um parser multidocumento automático para o Português do Brasil.** IV Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence–WTDIA. *Anais...a2008*.
- ALEIXO, P.; PARDO, T. A. S. **CSTNews: um corpus de textos jornalísticos anotados segundo a teoria discursiva multidocumento CST (Cross-document Structure Theory.** [s.l.] Universidade de São Paulo (USP); São Carlos, SP, Brasil., b2008. Disponível em: <<http://repositorio.icmc.usp.br/handle/RIICMC/6761>>.
- ALENCAR, L. F. DE. Donatus: uma interface amigável para o estudo da sintaxe formal utilizando a biblioteca em Python do NLTK. *Alfa: Revista de Linguística (São José do Rio Preto)*, v. 56, n. 2, p. 523–555, jul. 2012.
- ALENCAR, L. F. DE; CUCONATO, B.; RADEMAKER, A. MorphoBr: an open source large-coverage full-form lexicon for morphological analysis of Portuguese. *Texto Livre*, v. 11, n. 3, p. 1–25, dez. 2018.
- ALENCAR, V.; ALCAIM, A. **LSF and LPC-derived features for large vocabulary distributed continuous speech recognition in Brazilian Portuguese.** 2008 42nd Asilomar Conference on Signals, Systems and Computers. *Anais...IEEE*, 2008.
- ALIKANIOTIS, D.; YANNAKOUDAKIS, H.; REI, M. **Automatic Text Scoring Using Neural Networks.** Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. *Anais...Association for Computational Linguistics*, 2016.
- ALISSON, S. **Their god is not our god.** Disponível em: <https://www.thecontinent.org/_files/ugd/287178_73f3d2af22614e678f277b631a62e491.pdf>. Acesso em: 11 jun. 2023.
- ALMEIDA, G. DE. **Translating the post-editor: an investigation of post-editing changes and correlations with professional experience across two Romance languages.** 2013. Disponível em: <<https://api.semanticscholar.org/CorpusID:60255248>>
- ALTUNYURT, L.; ORHAN, Z.; GÜNGÖR, T. **A Composite Approach for Part of Speech Tagging in Turkish.** 2006. Disponível em: <<https://api.semanticscholar.org/CorpusID:9439761>>
- ALUÍSIO, S. et al. **An Account of the Challenge of Tagging a Reference Corpus for Brazilian Portuguese.** (N. J. Mamede et al., Eds.) Computational Processing of the Portuguese Language. *Anais...Berlin, Heidelberg: Springer Berlin Heidelberg*, 2003.
- ALVARES, R. V.; GARCIA, A. C. B.; FERRAZ, I. **STEMBR: A stemming algorithm for the Brazilian Portuguese language.** Portuguese conference on artificial intelligence. *Anais...Springer*, 2005.
- AMARAL, D. O. F. DO. **O reconhecimento de entidades nomeadas por meio de conditional random fields para a língua portuguesa.** Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2013.
- AMARAL, D.; VIEIRA, R. Nerp-crf: uma ferramenta para o reconhecimento de entidades nomeadas por meio de conditional random fields. *Linguamática (Braga)*, 2014.
- AMORIM, E.; CANÇADO, M.; VELOSO, A. **Automated Essay Scoring in the Presence of Biased Ratings.** Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.



- Anais...**Association for Computational Linguistics, 2018.
- AMORIM, E.; VELOSO, A. **A Multi-aspect Analysis of Automatic Essay Scoring for Brazilian Portuguese**. Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics. **Anais...**Valencia, Spain: Association for Computational Linguistics, abr. 2017.
- ANACLETO, J. et al. **Can Common Sense uncover cultural differences in computer applications?** (M. Bramer, Ed.)Artificial Intelligence in Theory and Practice. **Anais...**Boston, MA: Springer US, 2006.
- ANACLETO, J. C. et al. **A Common Sense-Based On-Line Assistant for Training Employees**. (C. Baranauskas et al., Eds.)Human-Computer Interaction – INTERACT 2007. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- ANANIADOU, S.; MCNAUGHT, J. **Text Mining for Biology And Biomedicine**. Norwood, MA, USA: Artech House, Inc., 2005.
- ANANTHAKRISHNAN, S.; NARAYANAN, S. S. Automatic Prosodic Event Detection Using Acoustic, Lexical, and Syntactic Evidence. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 16, n. 1, p. 216–228, 2008.
- ANCHIÊTA, R. T. et al. **PiLN IDPT 2021: Irony Detection in Portuguese Texts with Superficial Features and Embeddings**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021. **Anais...**2021.
- ANDERSEN, P. M. et al. **Automatic extraction of facts from press releases to generate news stories**. Third Conference on Applied Natural Language Processing. **Anais...**1992.
- ANTUNES, I. **Lutar com palavras: coesão e coerência**. [s.l.] Parábola, 2007.
- ANTUNES, I. **Textualidade: noções básicas e implicações pedagógicas**. [s.l.] Editora: Parábola Editorial, 2017.
- ARAUJO, P. H. L. DE et al. **LeNER-Br: A Dataset for Named Entity Recognition in Brazilian Legal Text**. Proceedings of the 13th International Conference. **Anais...**2018.
- ARDILA, R. et al. Common voice: A massively-multilingual speech corpus. **arXiv preprint arXiv:1912.06670**, 2019.
- Artificial intelligence and human rights**. 1. ed. [s.l.] Dykinson, S.L., 2021.
- ASAHARA, M.; MATSUMOTO, Y. **Japanese named entity extraction with redundant morphological analysis**. Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics. **Anais...**2003.
- ASHER, N. et al. **Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus**. 10th International Conference on Language Resources and Evaluation (LREC 2016). **Anais...**2016.
- ASHER, N.; LASCARIDES, A. **Logics of conversation**. [s.l.] Cambridge University Press, 2003.
- ASHER, N.; VIEU, L. Subordinating and coordinating discourse relations. **Lingua**, v. 115, n. 4, p. 591–610, 2005.
- ASSI, F. M. et al. **UFSCar’s Team at ABSAPT 2022: Using Syntax, Semantics and Context for Solving the Tasks**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022. **Anais...**2022.



- AUER, S. et al. **DBpedia: A Nucleus for a Web of Open Data**. (K. Aberer et al., Eds.)The Semantic Web. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- AVANÇO, L. V.; NUNES, M. DAS G. V. **Lexicon-Based Sentiment Analysis for Reviews of Products in Brazilian Portuguese**. Proceedings of the 2014 Brazilian Conference on Intelligent Systems. **Anais...**2014.
- AZIZ, W.; SPECIA, L. **Fully Automatic Compilation of a Portuguese-English Parallel Corpus for Statistical Machine Translation**. STIL 2011. **Anais...**Cuiabá, MT: 2011.
- BAADER, F. et al. **The Description Logic Handbook: Theory, Implementation and Applications**. Cambridge, Reino Unido: Cambridge University Press, 2003.
- BÄCKSTRÖM, T. et al. **Introduction to Speech Processing**. 2. ed. [s.l: s.n.].
- BADENE, S. et al. **Learning Multi-party Discourse Structure Using Weak Supervision**. 25th International conference on computational linguistics and intellectual technologies (Dialogue 2019). **Anais...**2019.
- BAEVSKI, A. et al. **wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.**, 2020. Disponível em: <<https://arxiv.org/abs/2006.11477>>
- BAEZA-YATES, R. A.; RIBEIRO-NETO, B. A. **Modern Information Retrieval-the concepts and technology behind search**. 2011.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação:- Conceitos e Tecnologia das Máquinas de Busca**. [s.l.] Bookman Editora, 2013.
- BAGGA, A.; BALDWIN, B. **Algorithms for Scoring Coreference Chains**. Proceedings of the first International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference. **Anais...**Granada, Spain: 1998.
- BAHDANAU, D.; CHO, K.; BENGIO, Y. **Neural Machine Translation by Jointly Learning to Align and Translate**. (Y. Bengio, Y. LeCun, Eds.)3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. **Anais...**San Diego, California.: 2015. Disponível em: <<http://arxiv.org/abs/1409.0473>>
- BAKER, C. F.; FILLMORE, C. J.; LOWE, J. B. **The Berkeley FrameNet Project**. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1. **Anais...**Montreal, Quebec, Canada: Association for Computational Linguistics, ago. 1998. Disponível em: <<https://aclanthology.org/P98-1013>>
- BAKER, C.; FELLBAUM, C.; PASSONNEAU, R. **Semantic Annotation of MASC**. Em: **Handbook of Linguistic Annotation**. [s.l.] Springer Netherlands, 2017. p. 699–717.
- BALAGE FILHO, P. P.; PARDO, T. A. S.; ALUÍSIO, S. M. **An Evaluation of the Brazilian Portuguese LIWC Dictionary for Sentiment Analysis**. Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology. **Anais...**2013.
- BANARESCU, L. et al. **Abstract Meaning Representation for Sembanking**. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. **Anais...**Sofia, Bulgaria: Association for Computational Linguistics, 2013. Disponível em: <<http://aclweb.org/anthology/W13-2322>>
- BANERJEE, S.; LAVIE, A. **METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments**. (J. Goldstein et al., Eds.)Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. **Anais...**Ann Arbor, Michigan: Association for Computational Linguistics, jun. 2005. Disponível em: <<https://aclanthology.org/W05-0909>>



- BANKO, M. et al. **Open Information Extraction from the Web**. Proceedings of the 20th International Joint Conference on Artificial Intelligence. **Anais...**: IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9909B5C03DA1A3CCFFF4263898B69100?doi=10.1.1.74.5174&rep=rep1&type=pdf>>
- BARBOSA, G. C. G.; GLAUBER, R.; CLARO, D. B. **Classificação de Relações Abertas Utilizando Features Independentes do Idioma**. Proceedings of the 4th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe). **Anais...SBC**, 2016.
- BARRAULT, L. et al. **Findings of the 2019 Conference on Machine Translation (WMT19)**. Proceedings of WMT. **Anais...**Florence, Italy: 2019.
- BARRAULT, L. et al. **Findings of the 2020 Conference on Machine Translation (WMT20)**. Proceedings of the Fifth Conference on Machine Translation. **Anais...**Online: Association for Computational Linguistics, nov. 2020. Disponível em: <<https://www.aclweb.org/anthology/2020.wmt-1.1>>
- BARREIRA, R.; PINHEIRO, V.; FURTADO, V. **FrameFOR – Uma Base de Conhecimento de Frames Semânticos para Perícias de Informática (FrameFOR - a Knowledge Base of Semantic Frames for Digital Forensics)**[In Portuguese]. Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology. **Anais...**Uberlândia, Brazil: Sociedade Brasileira de Computação, out. 2017. Disponível em: <<https://aclanthology.org/W17-6620>>
- BARROS, D. L. P. DE. Introdução à Linguística II: princípios de análise. Em: FIORIN, J. L. (Ed.). 5. ed. São Paulo: Contexto, 2021. p. 187–219.
- BASSO, R. M. **A Semântica das Relações Anafóricas entre Eventos**. tese de doutorado—[s.l.] Universidade Estadual de Campinas, SP, 2009.
- BATES, M. et al. Research in Knowledge Representation for Natural Language Understanding: Bolt, Beranek, and Newman. **SIGART Bull.**, n. 79, p. 30–31, jan. 1982.
- BATISTA, C.; DIAS, A. L.; NETO, N. Free resources for forced phonetic alignment in Brazilian Portuguese based on Kaldi toolkit. **EURASIP Journal on Advances in Signal Processing**, v. 2022, n. 1, p. 11, 19 fev. 2022.
- BECKMAN, M. E.; HIRSCHBERG, J.; SHATTUCK-HUFNAGEL, S. The original ToBI system and the evolution of the ToBI framework. Em: JUN, S.-A. (Ed.). **Prosodic typology: the phonology of intonation and phrasing**. Oxford: Oxford University Press, 2005. p. 9–54.
- BELTAGY, I.; PETERS, M. E.; COHAN, A. Longformer: The Long-Document Transformer. **CoRR**, v. abs/2004.05150, 2020.
- BENDER, E. M. **Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax**. Springer Nature Switzerland AG 2013: Springer Cham, 1959. p. XVII–166
- BENDER, E. M. **Linguistically Naïve != Language Independent: Why NLP Needs Linguistic Typology**. Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous? **Anais...**Athens, Greece: Association for Computational Linguistics, mar. 2009. Disponível em: <<https://www.aclweb.org/anthology/W09-0106>>
- BENDER, E. M. **The Power of Linguistics - Unpacking Natural Language Processing Ethics with Emily M. Bender**. [Podcast]. Disponível em: <<https://www.radicalai.org/e16-emily-bender>>. Acesso em: 7 abr. 2023.
- BENDER, E. M. et al. **On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?** . Proceedings of the 2021 ACM Conference on Fairness,



- Accountability, and Transparency. **Anais...** FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021. Disponível em: <<https://doi.org/10.1145/3442188.3445922>>
- BENDER, E. M. **You Are Not a Parrot And a chatbot is not a human. And a linguist named Emily M. Bender is very worried what will happen when we forget this.** Disponível em: <<https://nymag.com/intelligencer/article/ai-artificial-intelligence-chatbots-emily-m-bender.html>>. Acesso em: 9 abr. 2023.
- BENDER, E. M.; FRIEDMAN, B. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. **Transactions of the Association for Computational Linguistics**, v. 6, p. 587–604, 2018.
- BENGIO, Y. et al. A Neural Probabilistic Language Model. **J. Mach. Learn. Res.**, v. 3, n. null, p. 1137–1155, mar. 2003.
- BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, v. 35, n. 8, p. 1798–1828, 2013.
- BERTAGLIA, T. F. C.; NUNES, M. DAS G. V. **Exploring Word Embeddings for Unsupervised Textual User-Generated Content Normalization.** Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT). **Anais...** Osaka, Japan: The COLING 2016 Organizing Committee, dez. 2016. Disponível em: <<https://aclanthology.org/W16-3916>>
- BERTAGLIA, T. F. C.; NUNES, M. DAS G. V. **Normalização textual de conteúdo gerado por usuário.** mathesis—[s.l.] Universidade de São Paulo, 2017.
- BERTOLDI, A. **Os Limites da Criação Automática de Léxicos Computacionais Baseados em Frames: Um Estudo Contrastivo do Frame Criminal_process (The Limits of the Automatic Creation of Frame-based Computational Lexicons: a Contrastive Study of the Criminal_process Frame) [in Portuguese].** Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology. **Anais...**2011. Disponível em: <<https://aclanthology.org/W11-4510>>
- BERTSCH, A. et al. Unlimiformer: Long-Range Transformers with Unlimited Length Input. **CoRR**, v. abs/2305.01625, 2023.
- BERWICK, R. C.; CHOMSKY, N. **Por que apenas nós? Linguagem e evolução.** [s.l.] SciELO-Editora UNESP, 2017.
- BHARDWAJ, S.; AGGARWAL, S.; MAUSAM, M. **CaRB: A crowdsourced benchmark for open IE.** Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). **Anais...**2019.
- BIBAL, A. et al. **Is Attention Explanation? An Introduction to the Debate.** Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Dublin, Ireland: Association for Computational Linguistics, 2022. Disponível em: <<https://aclanthology.org/2022.acl-long.269>>
- BICK, E. **The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.** tese de doutorado—[s.l.] Aarhus University Press, Denmark; University of Aarhus, 2000.
- BICK, E. **A dependency-based approach to anaphora annotation.** Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language. **Anais...**Porto Alegre, Brazil: 2010.
- BICK, E. S. PFN-PT: A Framenet Annotator for Portuguese: Anotação semântica automática: um novo Framenet para o português. **Domínios de Linguagem**, v. 16(4)7,



- p. 1401–1435, 2009.
- BIDERMAN, M. T. C. **Teoria linguística: linguística quantitativa e computacional**. Rio de Janeiro: Martins Fontes, 1978.
- BIKEL, D. M.; SCHWARTZ, R.; WEISCHEDEL, R. M. An algorithm that learns what’s in a name. **Machine learning**, v. 34, p. 211–231, 1999.
- BIRD, S.; LOPER, E. **NLTK: The Natural Language Toolkit**. Proceedings of the ACL Interactive Poster and Demonstration Sessions. **Anais...**Barcelona, Spain: Association for Computational Linguistics, jul. 2004. Disponível em: <<https://aclanthology.org/P04-3031>>
- BIRON, T. et al. Automatic detection of prosodic boundaries in spontaneous speech. **PLoS ONE**, v. 16, n. 5, p. 1–21, maio 2021.
- BITTENCOURT JR., J. A. S. **Avaliação automática de redação em língua portuguesa empregando redes neurais profundas**. mathesis—[s.l.] Universidade Federal de Goiás, 2020.
- BLACKBURN, P.; BOS, J. **Representation and Inference for Natural Language: A First Course in Computational Semantics**. [s.l.] Center for the Study of Language; Information, 2005.
- BLEI, D. M.; MORENO, P. J. **Topic Segmentation with an Aspect Hidden Markov Model**. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. **Anais...**New York, NY, USA: Association for Computing Machinery, 2001.
- BLOM, J. D. **A dictionary of hallucinations**. [s.l.] Springer, 2010.
- BOBROW, D. G. et al. GUS, a frame-driven dialog system. **Artificial Intelligence**, v. 8, n. 2, p. 155–173, 1977.
- BOERSMA, P.; WEENINK, D. **Praat: doing phonetics by computer [Computer program]. Version 6.3.10.**, 2023. Disponível em: <<http://www.praat.org/>>
- BOJANOWSKI, P. et al. Enriching Word Vectors with Subword Information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017.
- BOJAR, O. et al. **Findings of the 2016 Conference on Machine Translation**. Proceedings of the First Conference on Machine Translation. **Anais...**Berlin, Germany: Association for Computational Linguistics, 2016.
- BOND, F.; FOSTER, R. **Linking and extending an open multilingual wordnet**. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Sofia, Bulgaria: Association for Computational Linguistics, ago. 2013. Disponível em: <<https://aclanthology.org/P13-1133>>
- BONIFACIO, L. H. et al. **mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset.**, 2021. Disponível em: <<https://arxiv.org/abs/2108.13897>>
- BOWMAN, S. R. et al. **A large annotated corpus for learning natural language inference**. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. **Anais...**Lisbon, Portugal: Association for Computational Linguistics, set. 2015. Disponível em: <<https://aclanthology.org/D15-1075>>
- BRANDES, N. et al. ProteinBERT: a universal deep-learning model of protein sequence and function. **Bioinform.**, v. 38, n. 8, p. 2102–2110, 2022.
- BRANDOM, R. B. **Articulating Reasons: An Introduction to Inferentialism**. Cambridge, Massachusetts, EUA: Harvard University Press, 2001.
- BRAUDE, D. A.; SHIMODAIRA, H.; YOUSSEF, A. B. **Template-warping based speech driven head motion synthesis**. Interspeech. **Anais...**2013.



- BRAUN, H. I. Understanding Scoring Reliability: Experiments in Calibrating Essay Readers. **Journal of Educational Statistics**, v. 13, n. 1, p. 1–18, 1988.
- BREEN, J. **JMdict: a Japanese-Multilingual Dictionary**. Proceedings of the Workshop on Multilingual Linguistic Resources. **Anais...Geneva**, Switzerland: COLING, 2004. Disponível em: <<https://aclanthology.org/W04-2209>>
- BREITFELLER, L. et al. **Finding Microaggressions in the Wild: A Case for Locating Elusive Phenomena in Social Media Posts**. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). **Anais...2019**.
- BREWSTER, C.; WILKS, Y. **Ontologies, taxonomies, thesauri: learning from texts**. (M. Deegan, Ed.) Proceedings of Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content Workshop. **Anais...2004**. Disponível em: <http://www.cbrewster.com/papers/KeyWord_FMO.pdf>
- BRIDGEMAN, B. Handbook of automated essay evaluation: Current applications and new directions. Em: SHERMIS, M. D.; BURSTEIN, J. (Eds.). [s.l.] Routledge/Taylor & Francis Group, 2013. p. 221–232.
- BRILL, E. **A Simple Rule-Based Part of Speech Tagger**. Proceedings of the Third Conference on Applied Natural Language Processing. **Anais...: ANLC '92.USA**: Association for Computational Linguistics, 1992. Disponível em: <<https://doi.org/10.3115/974499.974526>>
- BRIN, S. **Extracting patterns and relations from the world wide web**. International workshop on the world wide web and databases. **Anais...Springer**, 1998.
- BROWN, P. et al. **A statistical approach to language translation**. Proceedings of the 12th conference on Computational linguistics -. **Anais...Budapest**, Hungary: Association for Computational Linguistics, 1988. Disponível em: <<http://portal.acm.org/citation.cfm?doid=991635.991651>>. Acesso em: 10 jun. 2020
- BROWN, T. B. et al. **Language Models are Few-Shot Learners**. (H. Larochelle et al., Eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. **Anais...2020**. Disponível em: <<https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfb4967418bfb8ac142f64a-Abstract.html>>
- BRUM, H.; NUNES, M. DAS G. V. **Building a Sentiment Corpus of Tweets in Brazilian Portuguese**. (N. C. (Conference chair) et al., Eds.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). **Anais...Miyazaki**, Japan: European Language Resources Association (ELRA), mar. 2018.
- BUCKLEY, C.; VOORHEES, E. M. **Evaluating evaluation measure stability**. ACM SIGIR Forum. **Anais...ACM New York**, NY, USA, 2017.
- BUENO, R. O. et al. **Overview of the Task on Irony Detection in Spanish Variants**. Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing. **Anais...2019**.
- BUOLAMWINI, J.; GEBRU, T. **Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification**. (S. A. Friedler, C. Wilson, Eds.) Proceedings of the 1st Conference on Fairness, Accountability and Transparency. **Anais...: Proceedings of Machine Learning Research**. PMLR, 2018. Disponível em: <<https://proceedings.mlr.press/v81/buolamwini18a.html>>
- CABRAL, B.; SOUZA, M.; CLARO, D. B. **PortNOIE: A Neural Framework for Open Information Extraction for the Portuguese Language**. International Conference on Computational Processing of the Portuguese Language. **Anais...Springer**, 2022.



- CABRAL, L. et al. **FakeWhastApp.BR: NLP and Machine Learning Techniques for Misinformation Detection in Brazilian Portuguese WhatsApp Messages**. Proceedings of the International Conference on Enterprise Information Systems. **Anais...**2021.
- CABRÉ, M. T. **La terminología: representación y comunicación**. [s.l.] Editora: Documenta Universitaria, 1999.
- CABRÉ, M. T. A Terminologia, uma disciplina em evolução: passado, presente e alguns elementos de futuro. **Debate Terminológico**. ISSN: 1813-1867, n. 01, 2005.
- CAMERON, H.; OLIVAL, F.; VIEIRA, R. Planear a normalização automática: tipologia de variação gráfica do corpus das Memórias Paroquiais (1758). **LaborHistórico**, v. 9, n. 1, p. 52234, 2023.
- CANDIDO JUNIOR, A. et al. CORAA: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese. **CoRR**, v. abs/2110.15731, 2021.
- CANDIDO JUNIOR, A. et al. CORAA ASR: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese. **Language Resources & Evaluation**, 2022.
- CARDOSO, N. **Rembrandt - a named-entity recognition framework**. Proceedings of the Eighth International Conference on Language Resources and Evaluation. **Anais...**Istanbul, Turkey: 2012. Disponível em: <<http://www.lrec-conf.org/proceedings/lrec2012/summaries/409.html>>
- CARDOSO, P. C. F. et al. **CSTNews-a discourse-annotated corpus for single and multi-document summarization of news texts in Brazilian Portuguese**. Proceedings of the 3rd RST Brazilian Meeting. **Anais...**2011.
- CARDOSO, P. C. F. **Exploração de métodos de sumarização automática multidocumento com base em conhecimento semântico-discursivo**. tese de doutorado—[s.l.] Universidade de São Paulo, 2014.
- CARL, M.; WAY, A. (EDS.). **Recent Advances in Example-Based Machine Translation**. [s.l.] Springer Netherlands, 2003.
- CARLSON, L.; MARCU, D. Discourse tagging reference manual. **ISI Technical Report ISI-TR-545**, v. 54, n. 2001, p. 56, 2001.
- CARMO, D. et al. PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data. **CoRR**, v. abs/2008.09144, 2020.
- CARPINETO, C.; ROMANO, G. A survey of automatic query expansion in information retrieval. **Acm Computing Surveys (CSUR)**, v. 44, n. 1, p. 1–50, 2012.
- CARVALHO, F.; SANTOS, G. DOS; GUEDES, G. P. **AffectPT-br: an Affective Lexicon based on LIWC 2015**. Proceedings of the 37th International Conference of the Chilean Computer Science Society. **Anais...**2018.
- CARVALHO, P. et al. **Clues for Detecting Irony in User-Generated Contents: Oh...!! It's "so Easy";-)**. Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion. **Anais...**2009.
- CARVALHO, P.; SILVA, M. J. **SentiLex-PT 02**. <https://b2share.eudat.eu>, 2017. Disponível em: <<https://b2share.eudat.eu/records/93ab120efdaa4662baec6adee8e7585f>>
- CASANOVA, E. **Síntese de voz aplicada ao português brasileiro usando aprendizado profundo**. {B.S.} thesis—[s.l.] Universidade Tecnológica Federal do Paraná, 2019.
- CASANOVA, E. et al. TTS-Portuguese Corpus: a corpus for speech synthesis in Brazilian Portuguese. **Language Resources and Evaluation**, v. 56, n. 3, p. 1043–1055, 2022.



- CASANOVA, E.; SHULBY, C. D.; ALUÍSIO, S. M. Deep learning approaches for speech synthesis and speaker verification. **Acoustic communication: an interdisciplinary approach**, 2021.
- CASELI, H. DE M.; FREITAS, C.; VIOLA, R. Processamento de Linguagem Natural. Em: **Tópicos em Gerenciamento de Dados e Informações: Minicursos do SBB 2022**. [s.l.] Sociedade Brasileira de Computação, 2022. p. 1–28.
- CASELI, H. DE M.; INÁCIO, M. **NMT and PBSMT Error Analyses in English to Brazilian Portuguese Automatic Translations**. Proceedings of the Twelfth Language Resources and Evaluation Conference. **Anais...Marseille, France: European Language Resources Association, 2020**. Disponível em: <<https://aclanthology.org/2020.lrec-1.446>>
- CASTANO, A.; CASACUBERTA, F. **A connectionist approach to machine translation**. 5th European Conference on Speech Communication and Technology (Eurospeech 1997). **Anais...ISCA, set. 1997**. Disponível em: <<http://dx.doi.org/10.21437/eurospeech.1997-50>>
- CASTILHO, A. T. DE. O português culto falado no Brasil: história do Projeto NURC. Em: PRETI, D.; URBANO, H. (Eds.). **A linguagem falada culta na cidade de São Paulo**. São Paulo, SP: TAQ/Fapesp, 1990. v. 4 – Estudosp. 141–292.
- CASTILHO, A. T. DE. Gramática do Português Brasileiro: fundamentos, perspectivas. **Cadernos de Linguística**, v. 2, n. 1, p. e252, abr. a2021.
- CASTILHO, S. et al. **Does post-editing increase usability? A study with Brazilian Portuguese as Target Language**. Proceedings of the 17th annual conference of the European association for machine translation. **Anais...2014**.
- CASTILHO, S. et al. **A comparative quality evaluation of PBSMT and NMT using professional translators**. Proceedings of Machine Translation Summit XVI: Research Track. **Anais...a2017**.
- CASTILHO, S. et al. Is Neural Machine Translation the New State of the Art? **The Prague Bulletin of Mathematical Linguistics**, v. 108, n. 1, p. 109–120, jun. b2017.
- CASTILHO, S. et al. Approaches to Human and Machine Translation Quality Assessment. Em: **Translation Quality Assessment: From Principles to Practice**. Machine Translation: Technologies e Applications. [s.l.] Springer International Publishing, 2018. v. 1p. 9–38.
- CASTILHO, S. et al. Editors' foreword to the special issue on human factors in neural machine translation. **Machine Translation**, v. 33, n. 1–2, p. 1–7, maio 2019.
- CASTILHO, S. **On the Same Page? Comparing IAA in Sentence and Document Level Human MT Evaluation**. Proceedings of the Fifth Conference on Machine Translation. **Anais...Association for Computational Linguistics, nov. 2020**. Disponível em: <<https://www.aclweb.org/anthology/2020.wmt-1.137>>
- CASTILHO, S. **Towards Document-Level Human MT Evaluation: On the Issues of Annotator Agreement, Effort and Miscalculation**. Proceedings of the Workshop on Human Evaluation of NLP Systems. **Anais...Association for Computational Linguistics, abr. b2021**. Disponível em: <<https://www.aclweb.org/anthology/2021.humeval-1.4>>
- CASTILHO, S. et al. **DELA Corpus - A Document-Level Corpus Annotated with Context-Related Issues**. Proceedings of the Sixth Conference on Machine Translation. **Anais...Online: Association for Computational Linguistics, nov. 2021**. Disponível em: <<https://aclanthology.org/2021.wmt-1.63>>
- CASTILHO, S. **How Much Context Span is Enough? Examining Context-Related Issues for Document-level MT**. Proceedings of the Language Resources and Evaluation Conference. **Anais...Marseille, France: European Language Resources Association, 2022**.



- Disponível em: <<https://aclanthology.org/2022.lrec-1.323>>
- CASTILHO, S. et al. **Translation Systems Care for Context? What About a GPT Model?** Proceedings of the 24th Annual Conference of the European Association for Machine Translation. **Anais...**Tampere, Finland: EAMT, 2023. Disponível em: <<https://events.tuni.fi/uploads/2023/06/11678752-proceedings-eamt2023.pdf>>
- CASTILHO, S.; RESENDE, N. Post-Editese in Literary Translations. **Information**, v. 13, n. 2, p. 66, 2022.
- CASTILHO, S.; RESENDE, N.; MITKOV, R. **What Influences the Features of Post-editese? A Preliminary Study.** Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019). **Anais...**Varna, Bulgaria: Incoma Ltd., Shoumen, Bulgaria, set. 2019. Disponível em: <<https://aclanthology.org/W19-8703>>
- CASTRO, P. V. Q. DE; SILVA, N. F. F. DA; SOARES, A. DA S. **Portuguese Named Entity Recognition Using LSTM-CRF.** (A. Villavicencio et al., Eds.)Proceedings of the 13th International Conference on the Computational Processing of the Portuguese Language. **Anais...**2018.
- CAVALIERE, P.; ROMEO, G. From Poisons to Antidotes: Algorithms as Democracy Boosters. **European Journal of Risk Regulation**, v. 13, n. 3, p. 421–442, 2022.
- CHALMERS, D. J. Syntactic transformations on distributed representations. **Connectionist Natural Language Processing: Readings from Connection Science**, p. 46–55, 1992.
- CHANDRAN, R. **Indigenous groups in NZ, US fear colonisation as AI learns their languages.** Disponível em: <<https://www.context.news/ai/nz-us-indigenous-fear-colonisation-as-bots-learn-their-languages>>. Acesso em: 7 abr. 2023.
- CHANG, K.-W. et al. **Illinois-Coref: The UI system in the CoNLL-2012 shared task.** Joint Conference on EMNLP and CoNLL-Shared Task. **Anais...**Association for Computational Linguistics, 2012.
- CHARPENTIER, F.; STELLA, M. **Diphone synthesis using an overlap-add technique for speech waveforms concatenation.** ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing. **Anais...**IEEE, 1986.
- CHE, X. et al. **Punctuation Prediction for Unsegmented Transcript Based on Word Vector.** Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). **Anais...**Portorož, Slovenia: European Language Resources Association (ELRA), 2016. Disponível em: <<https://aclanthology.org/L16-1103>>
- CHEN, A.; CHEN, D. O. Simulation of a machine learning enabled learning health system for risk prediction using synthetic patient data. **Scientific Reports**, v. 12, n. 1, p. 17917, out. 2022.
- CHEN, K.; HASEGAWA-JOHNSON, M. A. **How prosody improves word recognition.** Speech Prosody 2004. **Anais...**2004.
- CHEN, L.-W.; RUDNICKY, A. **Exploring Wav2vec 2.0 Fine Tuning for Improved Speech Emotion Recognition.** ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). **Anais...**IEEE, 2023.
- CHEN, P. P. The Entity-Relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst.**, v. 1, n. 1, p. 9–36, 1976.
- CHILD, R. et al. Generating Long Sequences with Sparse Transformers. **CoRR**, v. abs/1904.10509, 2019.
- CHO, K. et al. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.** (A. Moschitti, B. Pang, W. Daelemans, Eds.)Proceedings of the 2014 Conference on Empirical Methods in Natural



- Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. **Anais...ACL**, 2014. Disponível em: <<https://doi.org/10.3115/v1/d14-1179>>
- CHOWDHERY, A. et al. PaLM: Scaling Language Modeling with Pathways. **CoRR**, v. abs/2204.02311, 2022.
- CHRISMAN, L. Learning recursive distributed representations for holistic computation. **Connection Science**, v. 3, n. 4, p. 345–366, 1991.
- CHRISTIANO, P. F. et al. **Deep Reinforcement Learning from Human Preferences**. (I. Guyon et al., Eds.)Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. **Anais...2017**. Disponível em: <<https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html>>
- CHUNG, Y.-A.; GLASS, J. **Speech2Vec: A Sequence-to-Sequence Framework for Learning Word Embeddings from Speech**. Proc. Interspeech 2018. **Anais...2018**.
- CIERI, C.; MILLER, D.; WALKER, K. **The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text**. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). **Anais...Lisbon, Portugal: European Language Resources Association (ELRA)**, 2004. Disponível em: <<http://www.lrec-conf.org/proceedings/lrec2004/pdf/767.pdf>>
- CIGNARELLA, A. T. et al. **Overview of the EVALITA 2018 Task on Irony Detection in Italian Tweets (IronITA)**. Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018). **Anais...2018**.
- CLARK, K. et al. **ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators**. 8th International Conference on Learning Representations, ICLR 2020. **Anais...Addis Ababa, Ethiopia: OpenReview.net**, abr. 2020. Disponível em: <<https://openreview.net/forum?id=r1xMH1BtvB>>
- CLIFTON, A. et al. **100,000 podcasts: A spoken English document corpus**. Proceedings of the 28th International Conference on Computational Linguistics. **Anais...2020**.
- COECKELBERGH, M. Artificial Intelligence, Responsibility Attribution, and a Relational Justification of Explainability. **Science and Engineering Ethics**, v. 26, p. 2051–2068, 2020.
- COELLO, J. M. A.; JUNQUEIRA, B. A. Automatic Analysis of Facebook Posts and Comments Written in Brazilian Portuguese. **International Journal for Innovation Education and Research**, 2019.
- COHEN, A. D. et al. LaMDA: Language Models for Dialog Applications. Em: **arXiv**. [s.l.: s.n.].
- COHEN, J. A Coefficient of Agreement for Nominal Scales. **Educational and Psychological Measurement**, v. 20, n. 1, p. 37–46, 1960.
- COLLOBERT, R.; WESTON, J. **A unified architecture for natural language processing: deep neural networks with multitask learning**. (W. W. Cohen, A. McCallum, S. T. Roweis, Eds.)Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008. **Anais...: ACM International Conference Proceeding Series.ACM**, 2008. Disponível em: <<https://doi.org/10.1145/1390156.1390177>>
- COLLOVINI, S. et al. **Summ-it: Um Corpus Anotado com Informações Discursivas**



- Visando a Sumarização Automática.** Proceedings of V Workshop em Tecnologia da Informação e da Linguagem Humana. **Anais...**Rio de Janeiro, Brasil: 2007.
- COLLOVINI, S. et al. **Extraction of Relation Descriptors for Portuguese Using Conditional Random Fields.** Proceedings of the 14th Ibero-American Conference on Advances in Artificial Intelligence. **Anais...**Santiago de Chile: 2014.
- COLLOVINI, S. et al. **IberLEF 2019 Portuguese Named Entity Recognition and Relation Extraction Tasks.** [s.l: s.n.].
- COMMISSION, E. **Proposal for a Regulation laying down harmonised rules on artificial intelligence.** Disponível em: < <https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence> >. Acesso em: 28 ago. 2023.
- CONCEIÇÃO, M. C.; ZANOLA, M. T. **Terminologia e mediação linguística: métodos, práticas e atividades.** Universidade do Algarve Editora, 2020.
- CONNEAU, A. et al. **Unsupervised cross-lingual representation learning at scale.** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. **Anais...**2020.
- CONNEAU, A.; LAMPLE, G. Cross-Lingual Language Model Pretraining. Em: **Proceedings of the 33rd International Conference on Neural Information Processing Systems.** Red Hook, NY, USA: Curran Associates Inc., 2019.
- CONSOLI, B. S. et al. **Embeddings for Named Entity Recognition in Geoscience Portuguese Literature.** Proceedings of The 12th Language Resources and Evaluation Conference. **Anais...**2020.
- CONSORTIUM, L. D. ACE (Automatic Content Extraction) English Annotation Guidelines for Events. **Version**, n. 5.4.3, 2005.
- COPESTAKE, A. et al. Minimal recursion semantics: An introduction. **Research on language and computation**, v. 3, p. 281–332, 2005.
- CORDEIRO, P. R.; PINHEIRO, V. **Um corpus de notícias falsas do twitter e verificação automática de rumores em lingua portuguesa.** Proceedings of the Symposium in Information and Human Language Technology. **Anais...**2019.
- COREIXAS, T. **Resolução De Correferência E Categorias De Entidades Nomeadas.** Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2010.
- CORMEN, T. et al. Introduction to Algorithms. Em: 2. ed. [s.l.] MIT Press; McGraw-Hill, 2001.
- CORNU, G. **Linguistique juridique.** [s.l: s.n.].
- CORRÊA, U. B. **Análise de sentimento baseada em aspectos usando aprendizado profundo: uma proposta aplicada à língua portuguesa.** tese de doutorado—[s.l.] Universidade Federal de Pelotas, 2021.
- CORTES, C.; VAPNIK, V. Support-Vector Networks. **Mach. Learn.**, v. 20, n. 3, p. 273–297, set. 1995.
- CORTIZ, D. et al. **A Weakly Supervised Dataset of Fine-Grained Emotions in Portuguese.** Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. **Anais...**Porto Alegre, RS, Brasil: SBC, 2021. Disponível em: <<https://sol.sbc.org.br/index.php/stil/article/view/17786>>
- COSTA, A. et al. A linguistically motivated taxonomy for Machine Translation error analysis. **Machine Translation**, v. 29, n. 2, p. 127–161, 2015.
- COUILLAULT, A. et al. **Evaluating corpora documentation with regards to the Ethics and Big Data Charter.** Proceedings of the Ninth International Conference on



Language Resources and Evaluation (LREC'14). **Anais...**Reykjavik, Iceland: European Language Resources Association (ELRA), 2014. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2014/pdf/424_Paper.pdf>

COWIE, J. R. **Automatic analysis of descriptive texts**. First Conference on Applied Natural Language Processing. **Anais...**1983.

COWIE, J.; LEHNERT, W. Information extraction. **Communications of the ACM**, v. 39, n. 1, p. 80–91, 1996.

CRISTEA, D.; IDE, N.; ROMARY, L. **Veins theory: A model of global discourse cohesion and coherence**. Coling-ACL Conference. **Anais...**1998.

CROFT, W. B.; METZLER, D.; STROHMAN, T. **Search engines: Information retrieval in practice**. [s.l.] Addison-Wesley, 2010. v. 520

CRUZ, B. S. **Concessionária do Metrô de SP é processada por ter câmeras que leem nossas emoções**. Disponível em: <<https://www.uol.com.br/tilt/noticias/redacao/2018/08/31/concessao-do-metro-de-sp-e-processada-por-ter-cameras-que-leem-emocoes.htm>>. Acesso em: 29 ago. 2023.

CRUZ, B. S. **Racismo Calculado**. Disponível em: <<https://www.uol.com.br/tilt/reportagens-especiais/como-os-algoritmos-espalham-racismo/#cover>>. Acesso em: 29 ago. 2023.

CUCCHIARELLI, A.; VELARDI, P. Unsupervised named entity recognition using syntactic and semantic contextual evidence. **Computational Linguistics**, v. 27, n. 1, p. 123–131, 2001.

CUI, H. et al. **Probabilistic query expansion using query logs**. Proceedings of the 11th international conference on World Wide Web. **Anais...**2002.

CUI, L.; WEI, F.; ZHOU, M. **Neural Open Information Extraction**. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). **Anais...**2018.

CULOTTA, A.; MCCALLUM, A.; BETZ, J. **Integrating probabilistic extraction models and data mining to discover relations and patterns in text**. Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. **Anais...**2006.

CUNHA, L. C. C. DA. **Um Corpus anotado de mensagens do WhatsApp em PT-BR para detecção automática de desinformação textual**. <https://github.com/cabrau/FakeWhatsApp.Br>, 2021.

DA SILVA JR., J. A. **Um avaliador automático de redações**. mathesis—[s.l.] Universidade Federal do Espírito Santo, 2021.

DADICO, C. M. **O Ódio Ancestral Como Elemento Constitutivo Do Estado Moderno e Seus Reflexos Na Compreensão dos Crimes De Ódio: Um Diálogo Entre o Direito Internacional e o Direito Brasileiro**. tese de doutorado—Porto Alegre, RS, Brazil: Programa de Pós-Graduação em Ciências Criminais da Escola de Direito da Pontifícia Universidade Católica do Rio Grande do Sul, 2020.

DAHL, V. Natural language processing and logic programming. **Journal of Logic Programming**, v. 19-20, n. 1, p. 681–714, 1994.

DAI, Z. et al. **Transformer-XL: Attentive Language Models beyond a Fixed-Length Context**. (A. Korhonen, D. R. Traum, L. Màrquez, Eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. **Anais...**Association for Computational Linguistics, 2019. Disponível em: <<https://doi.org/10.18653/v1/p19-1285>>

DALIANIS, H. Characteristics of Patient Records and Clinical Corpora. Em: **Clinical Text Mining: Secondary Use of Electronic Patient Records**. Cham: Springer



- International Publishing, 2018. p. 21–34.
- DARPA (ED.). **Proceedings of the 3rd Message Understanding Conference (MUC-3)**. San Diego, EUA: Morgan Kaufmann, 1991.
- DE PAIVA, V. et al. **An overview of Portuguese wordnets**. Proceedings of the 8th Global WordNet Conference (GWC). **Anais...**2016.
- DE PAIVA, V.; RADEMAKER, A.; MELO, G. DE. **OpenWordNet-PT: An Open Brazilian Wordnet for Reasoning**. Proceedings of COLING 2012: Demonstration Papers. **Anais...**2012.
- DE SOUSA, S. C.; AZIZ, W.; SPECIA, L. **Assessing the post-editing effort for automatic and semi-automatic translations of DVD subtitles**. Proceedings of the International Conference Recent Advances in Natural Language Processing 2011. **Anais...**2011.
- DEERWESTER, S. et al. Indexing by latent semantic analysis. **Journal of the American society for information science**, v. 41, n. 6, p. 391–407, 1990.
- DEJONG, G. Prediction and substantiation: A new approach to natural language processing. **Cognitive Science**, v. 3, n. 3, p. 251–273, 1979.
- DEL CORRO, L.; GEMULLA, R. **Clasie: clause-based open information extraction**. Proceedings of the 22nd international conference on World Wide Web. **Anais...**: WWW '13. New York, NY, USA: ACM; ACM, 2013. Disponível em: <<http://doi.acm.org/10.1145/2488388.2488420>>
- DEMNER-FUSHMAN, D.; CHAPMAN, W. W.; MCDONALD, C. J. What can natural language processing do for clinical decision support? **J Biomed Inform**, v. 42, n. 5, p. 760–772, ago. 2009.
- DEMPSEY, P. The teardown: Google Home personal assistant. **Engineering & Technology**, v. 12, n. 3, p. 80–81, 2017.
- DETTMERS, T. et al. QLoRA: Efficient Finetuning of Quantized LLMs. **arXiv preprint arXiv:2305.14314**, 2023.
- DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. (J. Burstein, C. Doran, T. Solorio, Eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019. **Anais...**Minneapolis, MN, USA: Association for Computational Linguistics, 2019. Disponível em: <<https://doi.org/10.18653/v1/n19-1423>>
- DHUMAL DESHMUKH, R.; KIWELEKAR, A. **Deep Learning Techniques for Part of Speech Tagging by Natural Language Processing**. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). **Anais...**mar. 2020.
- DI GANGI, M. A. et al. **MuST-C: a Multilingual Speech Translation Corpus**. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). **Anais...**Minneapolis, Minnesota: Association for Computational Linguistics, jun. 2019. Disponível em: <<https://aclanthology.org/N19-1202>>
- DIAS-DA-SILVA, B. C. **A face tecnológica dos estudos da linguagem: o processamento automático das línguas naturais**. 1996. 272f. tese de doutorado—[s.l.] Tese (Doutorado em Lingüística e Língua Portuguesa)—Faculdade de Ciências e ..., 1996.
- DIAS-DA-SILVA, B. C. **Wordnet.Br: An Exercise of Human Language Technology Research**. Proceedings of the Third International WordNet Conference. **Anais...**2005. Disponível em: <<http://semanticweb.kaist.ac.kr/conference/gwc/pdf2006/6.pdf>>



- DIAS-DA-SILVA, B. C.; MORALES, H. R. DE. A Construção de um Thesaurus Eletrônico para o Português. **Alfa**, 2003.
- DIAZ, F.; MITRA, B.; CRASWELL, N. **Query Expansion with Locally-Trained Word Embeddings**. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. **Anais...**2016.
- DODDINGTON, G. **Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics**. Proceedings of the Second International Conference on Human Language Technology Research. **Anais...**: HLT '02.San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- DODDINGTON, G. R. et al. **The automatic content extraction (ace) program-tasks, data, and evaluation**. Lrec. **Anais...**Lisbon, 2004.
- DOHERTY, S. et al. **Mapping the industry I: Findings on translation technologies and quality assessment**. QTLaunchPad – Mapping the Industry I: Findings on Translation Technologies and Quality Assessment. **Anais...**GALA, 2013. Disponível em: <http://doras.dcu.ie/19474/1/Version_Participants_Final.pdf>. Acesso em: 11 nov. 2015
- DOHERTY, S. et al. On Education and Training in Translation Quality Assessment. Em: MOORKENS, J. et al. (Eds.). **Translation Quality Assessment: From Principles to Practice**. Cham: Springer International Publishing, 2018. p. 95–106.
- DONG, Q. et al. A Survey for In-context Learning. **CoRR**, v. abs/2301.00234, 2023.
- DORR, B. et al. Machine translation evaluation and optimization. Em: **Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation**. [s.l.] Springer, 2011. p. 745–843.
- DU BOIS, J. W. et al. **Santa Barbara corpus of spoken American English. Parts 1–4**. Philadelphia: Linguistic Data Consortium, 2000–2005.
- DU BOIS, J. W. et al. **Discourse transcription**. Santa Barbara: Department of Linguistics, University of California, 1992. v. 4
- DURAN, M. S. et al. **The Dawn of the Portinari Multigenre Treebank: Introducing its Journalistic Portion**. Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. **Anais...**Porto Alegre, RS, Brasil: SBC, 2023.
- DURAN, M. S.; ALUÍSIO, S. M. **Propbank-Br: a Brazilian Treebank Annotated with Semantic Role Labels**. Proceedings of the 8th International Conference on Language Resources and Evaluation - LREC. **Anais...**2012.
- EBDEN, P.; SPROAT, R. The Kestrel TTS text normalization system. **Natural Language Engineering**, v. 21, p. 333–353, maio 2014.
- EIJCK, J. VAN; UNGER, C. **Computational Semantics with Functional Programming**. [s.l.] Cambridge University Press, 2010.
- EISENSTEIN, J. **Introduction to Natural Language Processing**. [s.l.] The MIT Press, 2019.
- EKMAN, P. An argument for basic emotions. **Cognition and Emotion**, v. 6, n. 3-4, p. 169–200, 1992.
- EL AYADI, M.; KAMEL, M. S.; KARRAY, F. Survey on speech emotion recognition: Features, classification schemes, and databases. **Pattern recognition**, v. 44, n. 3, p. 572–587, 2011.
- ELLIOT, N.; KLOBUCAR, A. Handbook of automated essay evaluation: Current applications and new directions. Em: SHERMIS, M. D.; BURSTEIN, J. (Eds.). [s.l.] Routledge/Taylor & Francis Group, 2013. p. 16–35.
- EMPOLI, G. DA. **Os engenheiros do caos: Como as fake news, as teorias da**



- conspiração e os algoritmos estão sendo utilizados para disseminar ódio, medo e influenciar eleições.** [s.l.] Vestígio Editora, 2019.
- ESTRELLA, P.; POPESCU-BELIS, A.; KING, M. The FEMTI guidelines for contextual MT evaluation: principles and resources. Em: WALTER DAELEMANS; VÉRONIQUE HOSTE (Eds.). **Evaluation of translation Technology.** Linguistica Antverpiensia new Series- themes em Translation Studies. [s.l.: s.n.].
- ETZIONI, O. et al. Unsupervised named-entity extraction from the web: An experimental study. **Artificial intelligence**, v. 165, n. 1, p. 91–134, 2005.
- Euromatrix. Survey of Machine Translation Evaluation.** [s.l.] Statistical; Hybrid Machine Translation Between All European Languages. Euromatrix, dez. 2007.
- FADER, A.; SODERLAND, S.; ETZIONI, O. **Identifying Relations for Open Information Extraction.** Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. **Anais...Edinburgh, Scotland, UK.:** Association for Computational Linguistics, jul. 2011. Disponível em: <<https://www.aclweb.org/anthology/D11-1142>>
- FAN, A.; LEWIS, M.; DAUPHIN, Y. **Hierarchical Neural Story Generation.** Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...Melbourne, Australia:** Association for Computational Linguistics, jul. 2018. Disponível em: <<https://aclanthology.org/P18-1082>>
- FARIAS, D. S. et al. **Opinion-Meter: A Framework for Aspect-Based Sentiment Analysis.** Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web. **Anais...2016.**
- FARZINDAR, A.; INKPEN, D. **Natural Language Processing for Social Media.** Second edition ed. [s.l.] Morgan; Claypool, 2018.
- FAYEK, H. M. **Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between.,** 2016. Disponível em: <<https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>>
- FEDERICO, M. et al. **Assessing the Impact of Translation Errors on Machine Translation Quality with Mixed-effects Models.** Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). **Anais...Doha, Qatar:** Association for Computational Linguistics, out. 2014. Disponível em: <<https://aclanthology.org/D14-1172>>
- FEIJÓ, D. DE V.; MOREIRA, V. P. Mono vs Multilingual Transformer-based Models: a Comparison across Several Language Tasks. **CoRR**, v. abs/2007.09757, 2020.
- FELLBAUM, C. (EDITOR). **WordNet: An electronic lexical database.** [s.l.] The MIT press, 1998.
- FELTRIM, V. D. et al. **A Construção de uma Ferramenta de Auxílio à Escrita de Resumos Acadêmicos em Português.** Anais do Encontro Nacional de Inteligência Artificial (ENIA'2003). **Anais...SBC, 2003.**
- FENNELLY, O. et al. Use of standardized terminologies in clinical practice: A scoping review. **Int J Med Inform**, v. 149, p. 104431, fev. 2021.
- FERNANDES, E. R.; SANTOS, C. N. DOS; MILIDIÚ, R. L. Latent trees for coreference resolution. **Computational Linguistics**, 2014.
- FERRADEIRA, J. E. DE S. **Resolução de anáfora pronominal.** mathesis—[s.l.] Universidade Nova de Lisboa; Dissertação de Mestrado, Universidade Nova de Lisboa, 1993.
- FERRÁNDEZ, Ó. et al. Tackling HAREM's portuguese named entity recognition task



- with spanish resources. **Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área. Linguateca** (http://www.linguateca.pt/aval_conjunta/LivroHAREM/Cap11-SantosCardoso2007-Ferrandezetal.pdf), 2007.
- FERREIRA, A. C. et al. Padrões linguísticos para detecção de ironia em múltiplos idiomas. **Revista Gestão & Tecnologia**, 2017.
- FERREIRA MELLO, R. et al. **Towards automated content analysis of rhetorical structure of written essays using sequential content-independent features in Portuguese**. (A. F. Wise, R. Martinez-Maldonado, I. Hilliger, Eds.) LAK22 Conference Proceedings. **Anais...United States of America: Association for Computing Machinery (ACM)**, 2022.
- FERREIRA, R. et al. Towards Automatic Content Analysis of Rhetorical Structure in Brazilian College Entrance Essays. Em: [s.l: s.n.]. p. 162–167.
- FILLMORE, C. J. et al. **Frame semantics and the nature of language**. Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech. **Anais...New York**, 1976.
- FINATTO, M. J. B.; ESTEVES, F. F.; VILLAR, G. S. Construindo uma terminologia de raiz: textos legislativos sob exploração terminológica. **Revista Platô**, v. 5, n. 9, 2022.
- FINE, K. Truthmaker semantics. **A Companion to the Philosophy of Language**, p. 556–577, 2017.
- FIRTH, J. R. The technique of semantics. **Transactions of the philological society**, v. 34, n. 1, p. 36–73, a1957.
- FIRTH, J. R. **A synopsis of linguistic theory 1930–1955**. [s.l.] Blackwell, 1957b. p. 1–32
- FLEISS, J. L. Measuring nominal scale agreement among many raters. **Psychological Bulletin**, v. 76, n. 5, p. 378–382, 1971.
- FLORES, F. N.; MOREIRA, V. P.; HEUSER, C. A. **Assessing the impact of stemming accuracy on information retrieval**. International Conference on Computational Processing of the Portuguese Language. **Anais...Springer**, 2010.
- FLORIAN, R. et al. **Named entity recognition through classifier combination**. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003. **Anais...2003**.
- FONSECA, E. B. **Resolução de correferências em língua portuguesa: pessoa, local e organização**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, 2014.
- FONSECA, E. B. et al. **Summ-it++: an enriched version of the summ-it corpus**. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). **Anais...2016**.
- FONSECA, E. B. **Resolução de correferência nominal usando semântica em língua portuguesa**. tese de doutorado—[s.l.] Pontifícia Universidade Católica do Rio Grande do Sul; Pontifícia Universidade Católica do Rio Grande do Sul, 2018.
- FONSECA, E. B.; VIEIRA, R.; VANIN, A. **Dealing With Imbalanced Datasets For Coreference Resolution**. Proceedings of The Twenty-Eighth International Flairs Conference. **Anais...2015**.
- FONSECA, E. B.; VIEIRA, R.; VANIN, A. **Adapting an Entity Centric Model for Portuguese Coreference Resolution**. Portorož, Slovenia, a2016.
- FONSECA, E. B.; VIEIRA, R.; VANIN, A. **CORP: Coreference Resolution for Portuguese.**, b2016.



- FONSECA, E. B.; VIEIRA, R.; VANIN, A. A. Coreference Resolution In Portuguese: Detecting Person, Location And Organization. **Journal of the Brazilian Computational Intelligence Society**, v. 12, n. 2, p. 86–97, 2014.
- FONSECA, E. R. et al. **Automatically Grading Brazilian Student Essays**. (A. Villavicencio et al., Eds.)Computational Processing of the Portuguese Language. **Anais...**Springer International Publishing, 2018.
- FONSECA, E. R.; ROSA, J. L. G. **Mac-Morpho Revisited: Towards Robust Part-of-Speech Tagging**. Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology. **Anais...**2013. Disponível em: <<https://aclanthology.org/W13-4811>>
- FONSECA, E. R.; ROSA, J. L.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in Portuguese. **Journal of the Brazilian Computer Society**, v. 21, n. 1, p. 32–38, fev. 2015.
- FONSECA, E.; VANIN, A.; VIEIRA, R. **Mention clustering to improve portuguese semantic coreference resolution**. International Conference on Applications of Natural Language to Information Systems. **Anais...**Springer, 2018.
- FONT LLITJÓS, A.; CARBONELL, J. G.; LAVIE, A. **A framework for interactive and automatic refinement of transfer-based machine translation**. Proceedings of the 10th EAMT Conference: Practical applications of machine translation. **Anais...**Budapest, Hungary: European Association for Machine Translation, 2005. Disponível em: <<https://aclanthology.org/2005.eamt-1.13>>
- FORCADA, M. L.; ÑECO, R. P. **Recursive hetero-associative memories for translation**. International Work-Conference on Artificial Neural Networks. **Anais...**Springer, 1997.
- FORTUNA, P. et al. **A Hierarchically-Labeled Portuguese Hate Speech Dataset**. Proceedings of the Third Workshop on Abusive Language Online. **Anais...**2019.
- FORTUNA, P.; NUNES, S. A survey on automatic detection of hate speech in text. **ACM Computing Surveys (CSUR)**, 2018.
- FREITAS, C. et al. **Relation detection between named entities: report of a shared task**. Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. **Anais...**Boulder, Colorado: 2009.
- FREITAS, C. et al. **Second HAREM: Advancing the State of the Art of Named Entity Recognition in Portuguese**. Proceedings of the International Conference on Language Resources and Evaluation. **Anais...**Valletta, Malta: 2010. Disponível em: <<http://www.lrec-conf.org/proceedings/lrec2010/summaries/412.html>>
- FREITAS, C. et al. **Vampiro que brilha... rá! Desafios na anotação de opinião em um corpus de resenhas de livros**. Proceedings of XI Encontro de Linguística de Corpus. **Anais...**2012.
- FREITAS, C. Sobre a construção de um léxico da afetividade para o processamento computacional do português. **Revista Brasileira de Linguística Aplicada**, 2013.
- FREITAS, C. et al. **Tagsets and Datasets: Some Experiments Based on Portuguese Language**. (A. Villavicencio et al., Eds.)Computational Processing of the Portuguese Language. **Anais...**Cham: Springer International Publishing, 2018.
- FREITAS, C. **Linguística Computacional**. [s.l.] Editora Parábola, 2022.
- FREITAS, C.; ROCHA, P.; BICK, E. **Floresta sintá (c) tica: bigger, thicker and easier**. International Conference on Computational Processing of the Portuguese Language. **Anais...**Springer, 2008.
- FREITAS, C.; SANTOS, D. **Gender Depiction in Portuguese: Distant reading**



- Brazilian and Portuguese literature.** 2nd Annual Conference of Computational Literary Studies. *Anais...*2023. Disponível em: <<https://www.linguateca.pt/Diana/download/FreitasSantos2023-2ndCCLS.pdf>>
- FREITAS, C.; SOUZA, E. Sujeito oculto às claras: uma abordagem descritivo-computacional / Omitted subjects revealed: a quantitative-descriptive approach. *REVISTA DE ESTUDOS DA LINGUAGEM*, v. 29, n. 2, p. 1033–1058, 2021.
- FREITAS, L. A. DE et al. **Pathways for irony detection in tweets.** Proceedings of the Symposium on Applied Computing (SAC). *Anais...*2014.
- FREITAS, L. A. DE. **Feature-level sentiment analysis applied to brazilian portuguese reviews.** tese de doutorado—[s.l.] Pontifícia Universidade Católica do Rio Grande do Sul, 2015.
- FREITAS, L. A. DE; SANTOS, L. DOS; DEON, D. Padrões linguísticos para detecção de ironia em múltiplos idiomas. *Revista Eletrônica de Iniciação Científica em Computação*, 2020.
- FULLER, C. et al. **An Analysis of Text-Based Deception Detection Tools.** Proceedings of the Twelfth Americas Conference on Information Systems. *Anais...*2006.
- FYFE, S. et al. Apophenia, theory of mind and schizotypy: perceiving meaning and intentionality in randomness. *Cortex*, v. 44, n. 10, p. 1316–1325, 2008.
- GAMALLO, P.; GARCIA, M. **Multilingual open information extraction.** (F. Pereira et al., Eds.)Portuguese Conference on Artificial Intelligence. *Anais...*Cham: Springer; Springer International Publishing, 2015. Disponível em: <https://doi.org/10.1007/978-3-319-23485-4_72>
- GAMALLO, P.; GARCIA, M.; FERNÁNDEZ-LANZA, S. **Dependency-based open information extraction.** Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP. *Anais...*: ROBUS-UNSUP '12.Stroudsburg, PA, USA: Association for Computational Linguistics; Association for Computational Linguistics, 2012. Disponível em: <<http://dl.acm.org/citation.cfm?id=2389961.2389963>>
- GAMON, M. et al. Handbook of automated essay evaluation: Current applications and new directions. Em: SHERMIS, M. D.; BURSTEIN, J. (Eds.). [s.l.] Routledge/Taylor & Francis Group, 2013. p. 251–266.
- GAO, T.; YAO, X.; CHEN, D. **SimCSE: Simple Contrastive Learning of Sentence Embeddings.** (M.-F. Moens et al., Eds.)Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. *Anais...*Association for Computational Linguistics, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.emnlp-main.552>>
- GARCIA, M.; GAMALLO, P. **An Entity-Centric Coreference Resolution System for Person Entities with Rich Linguistic Information.** Proceedings of 25th International Conference on Computational Linguistics. *Anais...*Dublin, Ireland: 2014. Disponível em: <<http://aclweb.org/anthology/C/C14/C14-1070.pdf>>
- GAUY, M. M.; FINGER, M. **Pretrained audio neural networks for Speech emotion recognition in Portuguese.** Proceedings of the Workshop on Automatic Speech Recognition for Spontaneous and Prepared Speech & Speech Emotion Recognition in Portuguese co-located with 15th edition of the International Conference on the Computational Processing of Portuguese (PROPOR 2022). *Anais...*2022.
- GEVA, M.; GUPTA, A.; BERANT, J. **Injecting Numerical Reasoning Skills into Language Models.** (D. Jurafsky et al., Eds.)Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. *Anais...*Association for Computational Linguistics, 2020. Disponível em: <<https://doi.org>>



g/10.18653/v1/2020.acl-main.89>

GHANEM, B. et al. **IDAT at FIRE2019: Overview of the Track on Irony Detection in Arabic Tweets**. Proceedings of the 11th Forum for Information Retrieval Evaluation. **Anais...2019**.

GHOSH, A. et al. **SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter**. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). **Anais...2015**.

GIBBS, R. W.; COLSTON, H. L. **The Risks and Rewards of Ironic Communication**. Say not to say: new perspectives on miscommunication. **Anais...2001**. Disponível em: <<https://api.semanticscholar.org/CorpusID:12510370>>

GLAUBER, R. et al. **Challenges of an Annotation Task for Open Information Extraction in Portuguese**. (A. Villavicencio et al., Eds.) Computational Processing of the Portuguese Language. **Anais...Cham: Springer International Publishing, 2018**.

GLAUBER, R.; CLARO, D. B. A systematic mapping study on open information extraction. **Expert Systems with Applications**, v. 112, p. 372–387, 2018.

GLAUBER, R.; CLARO, D. B.; OLIVEIRA, L. S. **Dependency Parser on Open Information Extraction for Portuguese Texts - DptOIE and DependentIE on IberLEF**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019) co-located with 35th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019). **Anais...<http://ceur-ws.org/Vol-2421/>: CEUR Workshop Proceedings, 2019**.

GLAUBER, R.; CLARO, D. B.; SENA, C. F. DE L. **Towards a Pragmatic Open Information Extraction for Portuguese Text - ICEIS17, InferPortOIE and PragmaticOIE on IberLEF**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019) co-located with 35th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019). **Anais...<http://ceur-ws.org/Vol-2421/>: CEUR Workshop Proceedings, 2019**.

GOLUB, G. H.; REINSCH, C. **Singular Value Decomposition and Least Squares Solutions**. [s.l.] Numer. Math 14, 1970. p. 403–420

GOMES, J. R. S. et al. **Deep Learning Brasil at ABSAPT 2022: Portuguese Transformer Ensemble Approaches**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022. **Anais...2022**.

GONÇALO OLIVEIRA, H. et al. **Avaliação à medida no Segundo HAREM**. (C. Mota, D. Santos, Eds.) Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. **Anais...Linguatca, 2008**.

GONÇALO OLIVEIRA, H. **Beyond the automatic construction of a lexical ontology for Portuguese: resources developed in the scope of Onto.PT**. Proceedings of Workshop on Tools and Resources for Automatically Processing Portuguese and Spanish. **Anais...: TorPorEsp.São Carlos, SP, Brasil: BDBComp, 2014**. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/torporesp/2014/004.pdf>>

GONÇALO OLIVEIRA, H.; GOMES, P. ECO and Onto-PT: a flexible approach for creating a Portuguese Wordnet automatically. **Language Resources and Evaluation**, v. 48, n. 2, p. 373–393, 2014.

GONÇALVES, M. et al. Avaliação de recursos computacionais para o português. **Linguamática**, v. 12, n. 2, p. 51–68, 2020.

GONÇALVES, S. C. L. Projeto ALIP (Amostra Linguística do Interior Paulista) e banco de dados Iboruna: 10 anos de contribuição com a descrição do português brasileiro. **Revista**



- Estudos Linguísticos**, v. 48, n. 1, p. 276–297, dez. 2019.
- GONÇALVES, T. et al. Clinical Screening Prediction in the Portuguese National Health Service: Data Analysis, Machine Learning Models, Explainability and Meta-Evaluation. **Future Internet**, v. 15, n. 1, p. 26, 2023.
- GONG, Z. et al. **Continual Pre-training of Language Models for Math Problem Understanding with Syntax-Aware Memory Network**. (S. Muresan, P. Nakov, A. Villavicencio, Eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. **Anais...** Association for Computational Linguistics, 2022. Disponível em: <<https://doi.org/10.18653/v1/2022.acl-long.408>>
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [s.l.] MIT Press, 2016. v. 1
- GRAHAM, Y. et al. **Is all that Glitters in Machine Translation Quality Estimation really Gold?** Proceedings of COLING 2016: Technical Papers. **Anais...** Osaka, Japan: The COLING 2016 Organizing Committee, dez. 2016. Disponível em: <<https://www.aclweb.org/anthology/C16-1294>>
- GRICE, H. P. Logic and Conversation. Em: **Syntax and Semantics: Vol. 3: Speech Acts**. [s.l.] Academic Press, 1975.
- GRIES, S. C. **Estatística com R para a Linguística**. [s.l.] FALE/ UFMG, 2019.
- GRIS, L. R. S. et al. **Bringing NURC/SP to digital life: the role of open-source automatic speech recognition models**. Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional. **Anais...** Porto Alegre, RS, Brasil: SBC, 2022. Disponível em: <<https://sol.sbc.org.br/index.php/eniac/article/view/22793>>
- GRIS, L. R. S. et al. **Evaluating OpenAI’s Whisper ASR for Punctuation Prediction and Topic Modeling of life histories of the Museum of the Person.**, 2023. Disponível em: <<https://arxiv.org/abs/2305.14580>>
- GRISHMAN, R.; SUNDHEIM, B. **Message Understanding Conference- 6: A Brief History**. COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics. **Anais...**1996. Disponível em: <<https://aclanthology.org/C96-1079>>
- GROSZ, B. J.; JOSHI, A. K.; WEINSTEIN, S. Centering: A framework for modelling the local coherence of discourse. **IRCS Technical Reports Series**, 1995.
- GROSZ, B. J.; SIDNER, C. L. Attention, intentions, and the structure of discourse. **Computational linguistics**, v. 12, n. 3, p. 175–204, 1986.
- GRUBER, A.; WEISS, Y.; ROSEN-ZVI, M. **Hidden Topic Markov Models**. Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics. **Anais...**: Proceedings of Machine Learning Research. San Juan, Puerto Rico: PMLR, mar. 2007.
- GRUBER, T. R. **Siri, A Virtual Personal Assistant-Bringing Intelligence to the Interface**. Semantic Technologies Conference. **Anais...**2009.
- GUARINO, N.; GUIZZARDI, G. **We need to Discuss the Relationship: Revisiting Relationships as Modeling Constructs**. Proceedings of the 27th International Conference on Advanced Information Systems Engineering (CAISE 2015). **Anais...**Springer-Verlag, 2015.
- GUIMARÃES, S. S. et al. **Characterizing Toxicity on Facebook Comments in Brazil**. Proceedings of the Brazilian Symposium on Multimedia and the Web. **Anais...**2020.
- GULATI, A. et al. Conformer: Convolution-augmented Transformer for Speech Recognition. **CoRR**, v. abs/2005.08100, 2020.
- GULDEN, C. et al. Extractive summarization of clinical trial descriptions. **International Journal of Medical Informatics**, v. 129, p. 114–121, 2019.



- GUMIEL, Y. B. et al. Temporal Relation Extraction in Clinical Texts: A Systematic Review. v. 54, n. 7, set. 2021.
- GURURANGAN, S. et al. **Annotation Artifacts in Natural Language Inference Data**. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). **Anais...**New Orleans, Louisiana: Association for Computational Linguistics, jun. 2018. Disponível em: <<https://aclanthology.org/N18-2017>>
- GURURANGAN, S. et al. **Don't Stop Pretraining: Adapt Language Models to Domains and Tasks**. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. **Anais...**Online: Association for Computational Linguistics, jul. 2020. Disponível em: <<https://aclanthology.org/2020.acl-main.740>>
- HABIBI, M. et al. Deep learning with word embeddings improves biomedical named entity recognition. **Bioinformatics**, v. 33, n. 14, p. i37–i48, 2017.
- HAENDCHEN FILHO, A. et al. **An approach to evaluate adherence to the theme and the argumentative structure of essays**. International Conference on Knowledge-Based Intelligent Information & Engineering Systems. **Anais...**2018.
- HAENDCHEN FILHO, A. et al. Imbalanced Learning Techniques for Improving the Performance of Statistical Models in Automated Essay Scoring. **Procedia Computer Science**, v. 159, p. 764–773, jan. 2019.
- HAKUTA, K. Handbook of Automated Essay Evaluation: Current Applications and New Directions. Em: SHERMIS, M. D.; BURSTEIN, J. (Eds.). [s.l.] Routledge/Taylor & Francis Group, 2013. p. 347–353.
- HALL, J. **A Probabilistic Part-of-Speech Tagger with Suffix Probabilities**. tese de doutorado—[s.l.: s.n.].
- HALLIDAY, M. A. K.; MATTHIESSEN, C. M. I. M. **Construing Experience Through Meaning: A Language Based Approach to Cognition**. [s.l.] Continuum, 1999.
- HAPKE, H.; HOWARD, C.; LANE, H. **Natural Language Processing in Action: Understanding, analyzing, and generating text with Python**. [s.l.] Manning, 2019.
- HARRIS, Z. S. Distributional Structure. **Word**, v. 10, n. 2-3, p. 146–162, 1954.
- HASEGAWA, T.; SEKINE, S.; GRISHMAN, R. **Discovering relations among named entities from large corpora**. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (acl-04). **Anais...**2004.
- HASSAN, H. et al. Achieving Human Parity on Automatic Chinese to English News Translation. **arXiv preprint 1803.05567**, 2018.
- HAVASI, C.; SPEER, R.; ALONSO, J. **ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge**. Recent Advances in Natural Language Processing. **Anais...**Borovets, Bulgaria: To appear, 2007.
- HE, K. et al. **Deep Residual Learning for Image Recognition**. 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. **Anais...**IEEE Computer Society, 2016. Disponível em: <<https://doi.org/10.1109/CVPR.2016.90>>
- HE, P. et al. **Deberta: decoding-Enhanced Bert with Disentangled Attention**. 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. **Anais...**OpenReview.net, 2021. Disponível em: <<https://openreview.net/forum?id=XPZiaotutsD>>
- HEARST, M. A. **Automatic acquisition of hyponyms from large text corpora**. Proceedings of the 14th conference on Computational linguistics-Volume 2. **Anais...**Association for Computational Linguistics, 1992.



- HEE, C. V.; LEFEVER, E.; HOSTE, V. **SemEval-2018 Task 3: Irony Detection in English Tweets**. Proceedings of the 12th International Workshop on Semantic Evaluation. **Anais...**2018.
- HEIKKILÄ, M. **Why you shouldn't trust AI search engines**. Disponível em: <<https://www.technologyreview.com/2023/02/14/1068498/why-you-shouldnt-trust-ai-search-engines/>>. Acesso em: 9 abr. 2023.
- HEIKKILÄ, M. **The viral AI avatar app Lensa undressed me—without my consent**. Disponível em: <<https://www.technologyreview.com/2022/12/12/1064751/the-viral-ai-avatar-app-lensa-undressed-me-without-my-consent/>>. Acesso em: 28 ago. 2023.
- HEIM, I. File Change Semantics and the Familiarity Theory of Definiteness. Em: **Formal Semantics**. [s.l.] Wiley-Blackwell, 2008. p. 223–248.
- HEINRICH, T.; MARCHI, F. **TeamUFPR at ABSAPT 2022: Aspect Extraction with CRF and BERT**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022. **Anais...**2022.
- HOCHREITER, S. Untersuchungen zu dynamischen neuronalen Netzen. **Diploma, Technische Universität München**, v. 91, n. 1, p. 31, 1991.
- HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, nov. 1997.
- HOFFMANN, J. et al. Training Compute-Optimal Large Language Models. **CoRR**, v. abs/2203.15556, 2022.
- HOFMANN, T. **Probabilistic Latent Semantic Indexing**. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99). **Anais...**New York, NY, USA: Association for Computing Machinery, 1999.
- HOLTZMAN, A. et al. **The Curious Case of Neural Text Degeneration**. ICLR. **Anais...**OpenReview.net, 2020. Disponível em: <<http://dblp.uni-trier.de/db/conf/iclr/iclr2020.html#HoltzmanBDFC20>>
- HORA, N. DA. **Coded Bias: linguagem acessível para entender vieses em algoritmos**. Disponível em: <<https://mittechreview.com.br/coded-bias-linguagem-acessivel-para-entender-vieses-em-algoritmos/>>. Acesso em: 7 abr. 2023.
- HORA, N. DA. **Ética em IA: a pergunta que não estamos fazendo**. Disponível em: <<https://mittechreview.com.br/etica-em-ia-a-pergunta-que-nao-estamos-fazendo/>>. Acesso em: 7 abr. 2023.
- HORNIK, K.; STINCHCOMBE, M. B.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989.
- HORMANN, T.; ZESCH, T. **Assigning Fine-grained PoS Tags based on High-precision Coarse-grained Tagging**. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. **Anais...**Osaka, Japan: The COLING 2016 Organizing Committee, dez. 2016. Disponível em: <<https://aclanthology.org/C16-1032>>
- HOU, Y.; MARKERT, K.; STRUBE, M. **A Rule-Based System for Unrestricted Bridging Resolution: Recognizing Bridging Anaphora and Finding Links to Antecedents**. Proceedings of the Conference on Empirical Methods in Natural Language Processing. **Anais...**Doha, Qatar: 2014. Disponível em: <<http://aclweb.org/anthology/D/D14/D14-1222.pdf>>
- HOULSBY, N. et al. **Parameter-Efficient Transfer Learning for NLP**. (K. Chaudhuri, R. Salakhutdinov, Eds.)Proceedings of the 36th International Conference on Machine



- Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. **Anais...** Proceedings of Machine Learning Research. PMLR, 2019. Disponível em: <<http://proceedings.mlr.press/v97/houlsby19a.html>>
- HOVY, E.; KING, M.; POPESCU-BELIS, A. **An introduction to MT evaluation**. Proceedings of Machine Translation Evaluation: Human Evaluators meet Automated Metrics. Workshop at the LREC 2002 Conference. Las Palmas, Spain. **Anais...**2002.
- HOWARD, J.; RUDER, S. **Universal Language Model Fine-tuning for Text Classification**. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Melbourne, Australia: Association for Computational Linguistics, jul. 2018. Disponível em: <[^5^](#)>
- HSU, W.-N. et al. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, v. 29, p. 3451–3460, 2021.
- HU, E. J. et al. **LoRA: Low-Rank Adaptation of Large Language Models**. The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. **Anais...**OpenReview.net, 2022. Disponível em: <<https://openreview.net/forum?id=nZeVKeeFYf9>>
- HU, M.; LIU, B. **Mining Opinion Features in Customer Reviews**. Proceedings of the 19th National Conference on Artificial Intelligence. **Anais...**2004.
- HUANG, J.-T.; HASEGAWA-JOHNSON, M.; SHIH, C. **Unsupervised prosodic break detection in Mandarin speech**. Proc. Speech Prosody 2008. **Anais...**2008.
- HUANG, X.; ACERO, A.; HON, H. W. **Spoken Language Processing: A Guide to Theory, Algorithm, and System Development**. [s.l.] Prentice Hall PTR, 2001.
- HUTCHINS, J. **Towards a definition of example-based machine translation.**, Proceedings of Second Workshop on Example-Based Machine Translation; **Anais...**2005.
- HUTCHINS, W. Machine Translation: A Concise History. **Journal of Translation Studies: Special Issue on The Teaching of Computer-aided Translation**, v. 13, p. 1–2, 2010.
- HUTCHINS, W. J. Machine translation over fifty years. **Histoire, Epistemologie, Langage**, v. XXII, n. 1, p. 7–31, 2001.
- IGNAT, O. et al. **A PhD Student’s Perspective on Research in NLP in the Era of Very Large Language Models.**, 2023. Disponível em: <<https://arxiv.org/abs/2305.12544>>
- ILARI, R.; GERALDI, J. W. **Semântica**. [s.l.] Ética, 1985.
- INÁCIO, M. L.; CASELI, H. DE M. **Word Embeddings at Post-Editing**. (P. Quaresma et al., Eds.)Computational Processing of the Portuguese Language. **Anais...**Cham: Springer International Publishing, 2020.
- INFOBASE. **Inteligência Artificial e a perpetuação do racismo**. Disponível em: <<https://infobase.com.br/inteligencia-artificial-e-a-perpetuacao-do-racismo/>>. Acesso em: 28 ago. 2023.
- ITO, K. **The LJ speech dataset**. <https://keithito.com/LJ-Speech-Dataset/>, 2017.
- IVGI, M.; SHAHAM, U.; BERANT, J. Efficient Long-Text Understanding with Short-Text Models. **Transactions of the Association for Computational Linguistics**, v. 11, p. 284–299, 2023.
- JACKSON, P.; MOULINIER, I. **Natural Language Processing for Online Applications – Text retrieval, extraction and categorization**. [s.l.] John Benjamins, 2002.
- JAHAN, M. S.; OUSSALAH, M. A systematic review of hate speech automatic detection



- using natural language processing. **Neurocomputing**, 2023.
- JAIN, S.; WALLACE, B. C. **Attention is not Explanation**. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). **Anais...**Minneapolis, Minnesota: Association for Computational Linguistics, 2019. Disponível em: <<https://aclanthology.org/N19-1357>>
- JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of IR techniques. **ACM Transactions on Information Systems (TOIS)**, v. 20, n. 4, p. 422–446, 2002.
- JEON, J. H.; LIU, Y. **Semi-supervised Learning for Automatic Prosodic Event Detection Using Co-training Algorithm**. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. **Anais...**Suntec, Singapore: Association for Computational Linguistics, ago. 2009. Disponível em: <<https://aclanthology.org/P09-1061>>
- JI, Z. et al. Survey of Hallucination in Natural Language Generation. **ACM Comput. Surv.**, v. 55, n. 12, mar. 2023.
- JIANG, S. et al. Multi-Ontology Refined Embeddings (MORE): A hybrid multi-ontology and corpus-based semantic representation model for biomedical concepts. **Journal of Biomedical Informatics**, v. 111, p. 103581, 2020.
- JIANG, S. et al. **Irony Detection in the Portuguese Language using BERT**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021. **Anais...**2021.
- JIN, X. et al. **Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora**. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. **Anais...**Seattle, United States: Association for Computational Linguistics, jul. 2022. Disponível em: <<https://aclanthology.org/2022.naacl-main.351>>
- JOHNSON, K. **Acoustic and Auditory Phonetics**. [s.l.] Wiley, 2011.
- JONES, K. H. et al. Toward the Development of Data Governance Standards for Using Clinical Free-Text Data in Health Research: Position Paper. **J Med Internet Res**, v. 22, n. 6, p. e16760, jun. 2020.
- JOOS, M. Description of language design. **Journal of Acoustical Society of America - JASA**, v. 22, p. 701–708, 1950.
- JOSHI, M. et al. **BERT for Coreference Resolution: Baselines and Analysis**. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). **Anais...**Hong Kong, China: Association for Computational Linguistics, nov. 2019. Disponível em: <<https://aclanthology.org/D19-1588>>
- JOSHI, M. et al. SpanBERT: Improving Pre-training by Representing and Predicting Spans. **Transactions of the Association for Computational Linguistics**, v. 8, p. 64–77, 2020.
- JOYCE, J. M. Kullback-Leibler Divergence. Em: LOVRIC, M. (Ed.). **International Encyclopedia of Statistical Science**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 720–722.
- JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech**



- Recognition**. 3rd. ed. USA: Prentice Hall PTR, 2023.
- KAMBHATLA, N. **Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction**. Proceedings of the ACL interactive poster and demonstration sessions. *Anais...*2004.
- KANTAYYA, S. **Coded Bias**. Disponível em: < <https://www.codedbias.com>>. Acesso em: 7 abr. 2023.
- KE, Z. et al. **Continual Pre-training of Language Models.**, 2023. Disponível em: <<https://arxiv.org/abs/2302.03241>>
- KENEDY, E.; OTHERO, G. DE Á. **Para conhecer sintaxe**. São Paulo: Contexto, 2018.
- KHAYRALLAH, H.; KOEHN, P. **On the Impact of Various Types of Noise on Neural Machine Translation**. Proceedings of the 2nd Workshop on Neural Machine Translation and Generation. *Anais...*Melbourne, Australia: Association for Computational Linguistics, jul. 2018. Disponível em: <<https://aclanthology.org/W18-2709>>
- KIANPOUR, M.; WEN, S.-F. **Timing Attacks on Machine Learning: State of the Art**. Intelligent Systems Conference. *Anais...*Springer, 2020.
- KILGARRIFF, A. I Don't Believe in Word Senses. **Computers and the Humanities**, 1997.
- KILGARRIFF, A. **Thesauruses for Natural Language Processing**. Proceedings of Natural Language Processing and Knowledge Engineering. *Anais...*2003. Disponível em: <<https://www.kilgarriff.co.uk/Publications/2003-K-Beijing-thes4NLP.pdf>>
- KIM, J. et al. Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search. **arXiv preprint arXiv:2005.11129**, 2020.
- KIM, J.; KONG, J.; SON, J. **Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech**. International Conference on Machine Learning. *Anais...*PMLR, 2021.
- KIPPER, K.; DANG, H. T.; PALMER, M. **Class-Based Construction of a Verb Lexicon**. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. *Anais...*AAAI Press, 2000.
- KIRSTAIN, Y.; RAM, O.; LEVY, O. **Coreference Resolution without Span Representations**. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). *Anais...*2021.
- KLATT, D. H. Software for a cascade/parallel formant synthesizer. **the Journal of the Acoustical Society of America**, v. 67, n. 3, p. 971–995, 1980.
- KLIE, J.-C. et al. **The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation**. Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. *Anais...*Santa Fe, USA: Association for Computational Linguistics, 2018. Disponível em: <<http://tubiblio.ulb.tu-darmstadt.de/106270/>>
- KNUTH, D. E. **Fundamental Algorithms. The Art of Computer Programming**. 3. ed. [s.l.] Addison-Wesley, 1997. v. 1
- KOCH, I. G. V. **O texto e a construção do sentido**. 7. ed. Campinas, SP: Contexto, 2003.
- KOCH, I. G. V.; TRAVAGLIA, L. **Texto e coerência**. 13. ed. [s.l.] Cortez, 2012.
- KOEHN, P. et al. **Moses: Open Source Toolkit for Statistical Machine Translation**. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. *Anais...*Prague, Czech



- Republic: Association for Computational Linguistics, jun. 2007. Disponível em: <<https://aclanthology.org/P07-2045>>
- KOEHN, P. **Statistical Machine Translation**. [s.l.] Cambridge University Press, 2009.
- KOEHN, P. **Neural Machine Translation**. [s.l.] Cambridge University Press, 2020.
- KOEHN, P.; OCH, F. J.; MARCU, D. **Statistical phrase-based translation**. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03. **Anais...**Association for Computational Linguistics, 2003. Disponível em: <<http://dx.doi.org/10.3115/1073445.1073462>>
- KOIZUMI, Y. et al. Miipher: A Robust Speech Restoration Model Integrating Self-Supervised Speech and Text Representations. **arXiv preprint arXiv:2303.01664**, b2023.
- KOIZUMI, Y. et al. LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus. **arXiv preprint arXiv:2305.18802**, a2023.
- KOJIMA, T. et al. **Large Language Models are Zero-Shot Reasoners**. NeurIPS. **Anais...**2022. Disponível em: <http://papers.nips.cc/paper/_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html>
- KOLECK, T. A. et al. Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review. **J Am Med Inform Assoc**, v. 26, n. 4, p. 364–379, abr. 2019.
- KONSTANTINOVA, N. **Review of relation extraction methods: What is new out there?** Analysis of Images, Social Networks and Texts: Third International Conference, AIST 2014, Yekaterinburg, Russia, April 10-12, 2014, Revised Selected Papers 3. **Anais...**Springer, 2014.
- KRINGS, H. P. **Repairing Texts: Empirical Investigations of Machine Translation Post-editing Processes**. [s.l.] Kent State University Press, 2001.
- KRIPPENDORFF, K. Estimating the Reliability, Systematic Error and Random Error of Interval Data. **Educational and Psychological Measurement**, v. 30, n. 1, p. 61–70, 1970.
- KRUSE, J. S.; BARBOSA, P. A. Alinha-PB: a phonetic aligner for Brazilian Portuguese. **Journal of Communication and Information Systems**, v. 36, n. 1, p. 192–199, dez. 2021.
- KUDO, T. **Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates**. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Melbourne, Australia: Association for Computational Linguistics, jul. 2018. Disponível em: <<https://aclanthology.org/P18-1007>>
- KUMAR, D. et al. **Understanding the Behaviors of Toxic Accounts on Reddit**. Proceedings of the ACM Web Conference 2023. **Anais...**2023.
- KUMAWAT, D.; JAIN, V. POS Tagging Approaches: A Comparison. **International Journal of Computer Applications**, v. 118, n. 6, p. 32–38, maio 2015.
- KUO, Y. et al. **Community-Based Game Design: Experiments on Social Games for Commonsense Data Collection**. Proceedings of the ACM SIGKDD Workshop on Human Computation. **Anais...**: HCOMP '09. New York, NY, USA: Association for Computing Machinery, 2009. Disponível em: <<https://doi.org/10.1145/1600150.1600154>>
- KUZI, S.; SHTOK, A.; KURLAND, O. **Query expansion using word embeddings**. Proceedings of the 25th ACM international conference on information and knowledge management. **Anais...**2016.



- KYLE, K. K. J. F. S.; JOSE, K. A. C. Y. B.; SOTELO, S. M. **Char2wav: End-to-end speech synthesis**. International Conference on Learning Representations, workshop. **Anais...**2017.
- LAN, Z. et al. **ALBERT: A Lite BERT for Self-supervised Learning of Language Representations**. 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. **Anais...**OpenReview.net, 2020. Disponível em: <<https://openreview.net/forum?id=H1eA7AEtvS>>
- LÄUBLI, S. et al. A set of recommendations for assessing human-machine parity in language translation. **Journal of Artificial Intelligence Research**, v. 67, p. 653–672, 2020.
- LÄUBLI, S.; SENNRICH, R.; VOLK, M. **Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation**. Proceedings of EMNLP. **Anais...**Brussels, Belgium: 2018.
- LEACOCK, C. et al. **Automated Grammatical Error Detection for Language Learners**. [s.l.] Morgan; Claypool Publishers, 2010.
- LEAL, S. E. et al. NILC-Matrix: assessing the complexity of written and spoken language in Brazilian Portuguese. **CoRR**, v. abs/2201.03445, 2021.
- LÉCHELLE, W.; GOTTI, F.; LANGLAIS, P. WiRe57: A Fine-Grained Benchmark for Open Information Extraction. **arXiv preprint arXiv:1809.08962**, 2018.
- LEE, H. et al. **Stanford’s multi-pass sieve coreference resolution system at the CoNLL-2011 shared task**. Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. **Anais...**2011.
- LEE, H. et al. Deterministic coreference resolution based on entity-centric, precision-ranked rules. **Computational Linguistics**, v. 39, n. 4, p. 885–916, 2013.
- LEE, J. et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, v. 36, n. 4, p. 1234–1240, set. 2019.
- LEE, K. et al. End-to-end neural coreference resolution. **arXiv preprint arXiv:1707.07045**, 2017.
- LEE, S. et al. A Survey on Evaluation Metrics for Machine Translation. **Mathematics**, v. 11, n. 4, 2023.
- LEHNERT, W.; SUNDHEIM, B. A performance evaluation of text-analysis technologies. **AI magazine**, v. 12, n. 3, p. 81–81, 1991.
- LEITE, H. et al. **WRITEME: uma Ferramenta de Auxílio à Escrita de READMEs Baseada em Dados Abertos**. Anais do XVII Congresso Latino-Americano de Software Livre e Tecnologias Abertas. **Anais...**Porto Alegre, RS, Brasil: SBC, 2020.
- LEITNER, E.; REHM, G.; SCHNEIDER, J. M. **Fine-Grained Named Entity Recognition in Legal Documents**. (M. Acosta et al., Eds.)Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference. **Anais...**2019.
- LENAT, D. B.; GUHA, R. V. **Building large knowledge-based systems: representation and inference in the Cyc project**. [s.l.] Addison-Wesley, 1989.
- LESK, M. **The seven ages of information retrieval**., 1995. Disponível em: <<https://archive.ifla.org/VI/5/op/udtop5/udt-op5.pdf>>
- LESTER, B.; AL-RFOU, R.; CONSTANT, N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. (M.-F. Moens et al., Eds.)Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. **Anais...**Association for Computational Linguistics, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.emnlp-main.243>>



- LEWIS, M. et al. **BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.** (D. Jurafsky et al., Eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. **Anais...** Association for Computational Linguistics, a2020. Disponível em: <<https://doi.org/10.18653/v1/2020.acl-main.703>>
- LEWIS, P. S. H. et al. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.** (H. Larochelle et al., Eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. **Anais...** b2020. Disponível em: <<https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>>
- LGPD. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 9 abr. 2023.
- LI, J. et al. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. **arXiv preprint arXiv:2004.05080**, 2020.
- LI, P. et al. Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models. **arXiv preprint arXiv:2304.03271**, a2023.
- LI, Q.; JI, H. **Incremental joint extraction of entity mentions and relations.** Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...** 2014.
- LI, R. et al. StarCoder: may the source be with you! **CoRR**, v. abs/2305.06161, b2023.
- LI, W. W. et al. **BERT Is Not The Count: Learning to Match Mathematical Statements with Proofs.** (A. Vlachos, I. Augenstein, Eds.) Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023. **Anais...** Association for Computational Linguistics, c2023. Disponível em: <<https://aclanthology.org/2023.eacl-main.260>>
- LI, X. L.; LIANG, P. **Prefix-Tuning: Optimizing Continuous Prompts for Generation.** (C. Zong et al., Eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. **Anais...** Association for Computational Linguistics, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.acl-long.353>>
- LIANG, X. et al. **Contrastive Demonstration Tuning for Pre-trained Language Models.** (Y. Goldberg, Z. Kozareva, Y. Zhang, Eds.) Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. **Anais...** Association for Computational Linguistics, 2022. Disponível em: <<https://aclanthology.org/2022.findings-emnlp.56>>
- LIKERT, R. **A Technique for the Measurement of Attitudes.** [s.l.] Archives of Psychology, 1932.
- LIMA, T. B. DE et al. Avaliação Automática de Redação: Uma revisão sistemática. **Revista Brasileira de Informática na Educação**, v. 31, p. 205--221, maio 2023.
- LIN, C.-H. et al. **Rich prosodic information exploration on spontaneous Mandarin speech.** 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP). **Anais...** Tianjin: 2016.
- LIN, C.-H. et al. Hierarchical prosody modeling for Mandarin spontaneous speech. **The Journal of the Acoustical Society of America**, v. 145, n. 4, p. 2576–2596, 2019.
- LIN, C.-Y. **ROUGE: A Package for Automatic Evaluation of Summaries.** Text



- Summarization Branches Out. **Anais...**Barcelona, Spain: Association for Computational Linguistics, jul. 2004. Disponível em: <<https://aclanthology.org/W04-1013>>
- LIN, J.; NOGUEIRA, R.; YATES, A. Pretrained Transformers for Text Ranking: BERT and Beyond. **arXiv preprint arXiv:2010.06467**, 2020.
- LIU, B. Sentiment Analysis and Opinion Mining. **Synthesis Lectures on Human Language Technologies**, 2012.
- LIU, H.; SINGH, P. **Commonsense Reasoning in and Over Natural Language**. (M. Gh. Negoita, R. J. Howlett, L. C. Jain, Eds.)Knowledge-Based Intelligent Information and Engineering Systems. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
- LIU, T.; YAO, J.-G.; LIN, C.-Y. **Towards improving neural named entity recognition with gazetteers**. Proceedings of the 57th annual meeting of the association for computational linguistics. **Anais...**2019.
- LIU, Y. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **CoRR**, v. abs/1907.11692, 2019.
- LIU, Y. et al. Multilingual Denoising Pre-training for Neural Machine Translation. **Trans. Assoc. Comput. Linguistics**, v. 8, p. 726–742, 2020.
- LIU, Z. et al. De-identification of clinical notes via recurrent neural network and conditional random field. **J Biomed Inform**, v. 75S, p. S34–S42, jun. 2017.
- LIU, Z. et al. **A Robustly Optimized BERT Pre-Training Approach with Post-Training**. Chinese Computational Linguistics: 20th China National Conference, CCL 2021, Hohhot, China, August 13–15, 2021, Proceedings. **Anais...**Berlin, Heidelberg: Springer-Verlag, 2021. Disponível em: <https://doi.org/10.1007/978-3-030-84186-7_31>
- LO, C. **YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources**. Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1. **Anais...**2019. Disponível em: <<https://doi.org/10.18653/v1/w19-5358>>
- LO, C.; WU, D. **MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles**. The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA. **Anais...**2011. Disponível em: <<https://aclanthology.org/P11-1023/>>
- LO, S. L. et al. Multilingual Sentiment Analysis: From Formal to Informal and Scarce Resource Languages. **Artificial Intelligence Review**, 2017.
- LOMMEL, A.; MELBY, A. **Tutorial: MQM-DQF: A Good Marriage (Translation Quality for the 21st Century)**. Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track). **Anais...**Boston, MA: Association for Machine Translation in the Americas, mar. 2018. Disponível em: <<https://aclanthology.org/W18-1925>>
- LOPE, J.; GRAÑA, M. An ongoing review of speech emotion recognition. **Neurocomputing**, 2023.
- LOPES, L. et al. **PortiLexicon-UD: a Portuguese Lexical Resource according to Universal Dependencies Model**. Proceedings of the Language Resources and Evaluation Conference. **Anais...**Marseille, France: European Language Resources Association, jun. 2022. Disponível em: <<https://aclanthology.org/2022.lrec-1.715>>
- LOPES, L. et al. **Disambiguation of Universal Dependencies Part-of-Speech Tags of Closed Class Words in Portuguese**. (A. Britto, K. V. Delgado, Eds.)Proceedings of the 12th Brazilian Conference on Intelligent Systems (BRACIS). **Anais...**2023.



- LOUIS, A.; HIGGINS, D. **Off-topic essay detection using short prompt texts**. Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications. *Anais...Los Angeles, California: Association for Computational Linguistics*, jun. 2010.
- LOVINS, J. B. Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, v. 11, n. 1-2, p. 22–31, 1968.
- LUCY, L.; BAMMAN, D. **Gender and Representation Bias in GPT-3 Generated Stories**. Proceedings of the Third Workshop on Narrative Understanding. *Anais...Virtual: Association for Computational Linguistics*, jun. 2021. Disponível em: <<https://aclanthology.org/2021.nuse-1.5>>
- LUDUSAN, B.; SYNNAEVE, G.; DUPOUX, E. **Prosodic boundary information helps unsupervised word segmentation**. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. *Anais...2015*.
- LUO, X. **On Coreference Resolution Performance Metrics**. Proceedings of the Conference on Empirical Methods in Natural Language Processing. *Anais...Vancouver, Canada: 2005*.
- LUONG, T.; PHAM, H.; MANNING, C. D. **Effective Approaches to Attention-based Neural Machine Translation**. (L. Màrquez et al., Eds.) Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. *Anais...The Association for Computational Linguistics, 2015*. Disponível em: <<https://doi.org/10.18653/v1/d15-1166>>
- LYONS, J. **Semantics: Volume 2**. [s.l.] Cambridge university press, 1977. v. 2
- MA, Q. et al. **Blend: a Novel Combined MT Metric Based on Direct Assessment - CASICT-DCU submission to WMT17 Metrics Task**. Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017. *Anais...2017*. Disponível em: <<https://doi.org/10.18653/v1/w17-4768>>
- MACDONALD, C.; TONELLOTTI, N. **Declarative Experimentation in Information Retrieval using PyTerrier**. Proceedings of ICTIR 2020. *Anais...2020*.
- MACHADO, A. A. A. et al. **Personalitatem Lexicon: um léxico em português brasileiro para mineração de traços de personalidade em textos**. Proceedings of the Brazilian Symposium on Computers in Education. *Anais...2015*.
- MACHADO, M. T.; PARDO, T. A. S. **NILC at ABSAPT 2022: Aspect Extraction for Portuguese**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022. *Anais...2022*.
- MACHADO, M. T.; PARDO, T. A. S.; RUIZ, E. E. S. **Creating a portuguese context sensitive lexicon for sentiment analysis**. Proceedings of the 13th international conference on computational processing of the Portuguese Language (PROPOR). *Anais...2018*.
- MACIEL, A. M. B. **Para o reconhecimento da especificidade do termo jurídico**. mathesis—[s.l.] Universidade Federal do Rio Grande do Sul, RS, 2001.
- MACOHIN, A.; CARNEIRO, J. V. V. Web Crawling e Web Scraping em sites de tribunais: publicidade processual e proteção de dados pessoais nas experiências europeia e brasileira. Em: WACHOWICZ, M. (Ed.). **Proteção de Dados Pessoais em Perspectiva: LGPD e RGPD na Ótica do Direito Comparado**. Curitiba: Gedai, UFPR, 2020.
- MALENCHINI, F. M. et al. **Um Benchmark para Sistemas de Extração de Informação Aberta em Português**. Proceedings of the Symposium in Information and



- Human Language Technology (STIL 2019). **Anais...**Salvador, Bahia: SBC, out. 2019.
- MANN, W. C.; THOMPSON, S. A. Rhetorical structure theory: Toward a functional theory of text organization. **Text-interdisciplinary Journal for the Study of Discourse**, v. 8, n. 3, p. 243–281, 1988.
- MANNING, C. D.; SCHÜTZE, H.; RAGHAVAN, P. **Introduction to information retrieval**. [s.l.] Cambridge University Press Cambridge, 2008.
- MARCACINI, R. M.; CANDIDO JUNIOR, A.; CASANOVA, E. **Overview of the Automatic Speech Recognition for Spontaneous and Prepared Speech & Speech Emotion Recognition in Portuguese (SE&R) Shared-tasks at PROPOR 2022**. Proceedings of the Workshop on Automatic Speech Recognition for Spontaneous and Prepared Speech & Speech Emotion Recognition in Portuguese co-located with 15th edition of the International Conference on the Computational Processing of Portuguese (PROPOR 2022). **Anais...**2022.
- MARCU, D. **From local to global coherence: A bottom-up approach to text planning**. AAAI/IAAI. **Anais...**Citeseer, 1997.
- MARCU, D.; CARLSON, L.; WATANABE, M. **The automatic translation of discourse structures**. 1st Meeting of the North American Chapter of the Association for Computational Linguistics. **Anais...**2000.
- MARCUSCHI, L. A. **Produção textual, análise de gêneros e compreensão**. [s.l.] Parábola Ed., 2008.
- MARIE, B.; FUJITA, A.; RUBINO, R. Scientific Credibility of Machine Translation Research: A Meta-Evaluation of 769 Papers. **arXiv:2106.15195 [cs]**, jun. 2021.
- MARINHO, J. et al. **Automated Essay Scoring: An approach based on ENEM competencies**. Anais do XIX Encontro Nacional de Inteligência Artificial e Computacional. **Anais...**SBC, 2022.
- MARINHO, J.; ANCHIÊTA, R.; MOURA, R. Essay-BR: a Brazilian Corpus to Automatic Essay Scoring Task. **Journal of Information and Data Management**, v. 13, n. 1, p. 65–76, 2022.
- MARKOV, A. A. The theory of algorithms. **Trudy Matematicheskogo Instituta Imeni VA Steklova**, v. 42, p. 3–375, 1954.
- MARNEFFE, M.-C. DE et al. Universal Dependencies. **Computational Linguistics**, v. 47, n. 2, p. 255–308, jun. 2021.
- MARTINS, D. B. DE J. **Pós-edição automática de textos traduzidos automaticamente de inglês para português do Brasil**. Mestrado—São Carlos: Universidade Federal de São Carlos, 2014.
- MARTINS, D. B. DE J.; CASELI, H. DE M. Automatic machine translation error identification. **Machine Translation**, v. 29, n. 1, p. 1–24, 2015.
- MARTINS, H. Sobre a estabilidade do significado em Wittgenstein. **Veredas**, v. 4, n. 2, p. 19–42, 2000.
- MARTINS, H. Três Caminhos na Filosofia da Linguagem. Em: **Introdução à Linguística. Volume III**. [s.l.] Editora Cortez, 2004.
- MARTINS, R. T. et al. **An interlingua aiming at communication on the Web: How language-independent can it be?** NAACL-ANLP 2000 Workshop: Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP. **Anais...**2000. Disponível em: <<https://aclanthology.org/W00-0204>>
- MARTINS, R.; NUNES, M. DAS G. V.; HASEGAWA, R. **Curupira: A Functional Parser for Brazilian Portuguese**. (N. J. Mamede et al., Eds.)Computational Processing of the Portuguese Language. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.



- MARTSCHAT, S.; STRUBE, M. Latent Structures for Coreference Resolution. **Transactions of the Association for Computational Linguistics**, v. 3, p. 405–418, 2015.
- MATTEI, L. D. et al. ATE ABSITA@ EVALITA2020: Overview of the Aspect Term Extraction and Aspect-based Sentiment Analysis Task. **Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)**, 2020.
- MATTHEWS, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. **Biochimica et Biophysica Acta (BBA) - Protein Structure**, v. 405, n. 2, p. 442–451, 1975.
- MATTHIESSEN, M. C. M. I. Applying systemic functional linguistics in healthcare contexts. **Text and Talk**, v. 33, n. 4-5, p. 437–447, 19 ago. 2013.
- MATTHIESSEN, M. C. M. I.; TERUYA, K.; WU, C. Multilingual studies as a multi-dimensional space of interconnected language studies. Em: **Meaning in context : strategies for implementing intelligent applications of language studies**. [s.l.] Continuum, 2008. p. 146–221.
- MAYFIELD, E.; BLACK, A. W. **Should You Fine-Tune BERT for Automated Essay Scoring?** Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. **Anais...Association for Computational Linguistics**, jul. 2020.
- MAZIERO, E. G. et al. **A base de dados lexical e a interface web do TeP 2.0: thesaurus eletrônico para o Português do Brasil**. Proceedings of the XIV Brazilian Symposium on Multimedia and the Web. **Anais...Salvador, Brazil: 2008**.
- MAZIERO, E. G. **Análise retórica com base em grande quantidade de dados**. tese de doutorado—[s.l.] Universidade de São Paulo, 2016.
- MAZIERO, E. G.; HIRST, G.; PARDO, T. A. S. **Adaptation of discourse parsing models for the Portuguese language**. 2015 Brazilian Conference on Intelligent Systems (BRACIS). **Anais...IEEE**, 2015.
- MAZIERO, E. G.; JORGE, M. L. DEL R. C.; PARDO, T. A. S. Identifying Multidocument Relations. **NLPCS**, v. 7, p. 60–69, 2010.
- MAZIERO, E. G.; PARDO, T. A. S. Automatic Identification of Multi-document Relations. **Proceedings of the PROPOR 2012 PhD and MSc/MA Dissertation Contest**, p. 1–8, 2012.
- MAZUMDER, M. et al. **Multilingual spoken words corpus**. Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). **Anais...2021**.
- MCCALLUM, A.; LI, W. **Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons**. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. **Anais...2003**.
- MCCANN, B. et al. **Learned in Translation: Contextualized Word Vectors**. Proceedings of the 31st International Conference on Neural Information Processing Systems. **Anais...: NIPS'17**. Red Hook, NY, USA: Curran Associates Inc., 2017.
- MCCRAE, J. P. et al. **English WordNet 2019 – An Open-Source WordNet for English**. Proceedings of the 10th Global Wordnet Conference. **Anais...Wroclaw, Poland: Global Wordnet Association**, jul. 2019. Disponível em: <<https://aclanthology.org/2019.gwc-1.31>>
- MCDONALD, R. et al. **Universal Dependency Annotation for Multilingual Parsing**.



- Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). **Anais...**Sofia, Bulgaria: Association for Computational Linguistics, ago. 2013. Disponível em: <<https://aclanthology.org/P13-2017>>
- MELLO, H.; RASO, T.; ALMEIDA FERRARI, L. DE. **C-ORAL–Brasil II: Corpus de referência do português brasileiro falado informal.**, no prelo no prelo.
- MELO, G. DE; WEIKUM, G. **Towards a universal wordnet by learning from combined evidence.** Proceedings of the 18th ACM conference on Information and knowledge management. **Anais...**2009.
- MENDES, R. B.; OUSHIRO, L. **Mapping Paulistano Portuguese: the SP2010 Project.** Proceedings of the VIIth GSCP International Conference: Speech and Corpora. **Anais...**Firenze, Italy: Fizenze University Press, 2012.
- MEYER, C. F. et al. The world wide web as linguistic corpus. Em: **Corpus Analysis.** [s.l.] Brill Rodopi, 2003. p. 241–254.
- MIKKULAINEN, R.; DYER, M. G. Natural Language Processing With Modular Pdp Networks and Distributed Lexicon. **Cognitive Science**, v. 15, n. 3, p. 343–399, 1991.
- MIKOLOV, T. et al. **Efficient Estimation of Word Representations in Vector Space.**, a2013. Disponível em: <<https://arxiv.org/abs/1301.3781>>
- MIKOLOV, T. et al. **Distributed Representations of Words and Phrases and their Compositionality.** (C. J. C. Burges et al., Eds.)Advances in Neural Information Processing Systems. **Anais...**Curran Associates, Inc., b2013. Disponível em: <<https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>>
- MINSKY, M. A framework for representing knowledge. **The psychology of computer vision**, 1975.
- MITKOV, R. **The Oxford handbook of Computational Linguistics.** [s.l.] Oxford University Press, 2003.
- MITKOV, R. 21 Discourse Processing. **The handbook of computational linguistics and natural language processing**, p. 599, 2010.
- MIWA, M.; BANSAL, M. **End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures.** Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Association for Computational Linguistics, 2016.
- MOHAN, S. et al. **The Impact of Toxic Language on the Health of Reddit Communities.** Proceedings of the Canadian Conference on AI. **Anais...**2017.
- MOLLAS, I. et al. ETHOS: a multi-label hate speech detection dataset. **Complex & Intelligent Systems**, 2022.
- MONTEIRO, R. A. et al. **Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results.** Proceedings of the 13th international conference on computational processing of the Portuguese Language. **Anais...**2018.
- MONTORO, A. F. **Curso de Teoria Geral do Direito - Aula 2: A linguagem do direito: semântica, sintática e pragmática.** Disponível em: <<http://www.dialdata.com.br/ilam/aula2>>.
- MOORE, R. K. Spoken language processing: Piecing together the puzzle. **Speech Communication**, v. 49, n. 5, p. 418–435, 2007.
- MOORKENS, J. et al. Correlations of perceived post-editing effort with measurements of actual effort. **Machine Translation**, v. 29, n. 3/4, p. 267–284, 2015.
- MOORKENS, J. Under pressure: translation in times of austerity. **Perspectives**, v. 25, n. 3, p. 464–477, fev. 2017.
- MOTA, C. R3M, uma participação minimalista no Segundo HAREM. **quot; In Cristina**



- Mota; Diana Santos (ed) **Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM** Linguateca 2008, 2008.
- MOTA, C.; SANTOS, D. (EDS.). **Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM**. [s.l.] Linguateca, 2008.
- MOTA, C.; SANTOS, D.; RANCHHOD, E. Avaliação de reconhecimento de entidades mencionadas: princípio de HAREM. **Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa**, p. 161–175, 2007.
- MOTTA, E. Sentenças Judiciais e Acessibilidade Textual e Terminológica. **Domínios de Lingu@gem**, v. 15, n. 3, p. 761–813, 2021.
- MOTTA, E. **SENTENÇAS JUDICIAIS E LINGUAGEM SIMPLES: um encontro possível e necessário**. mathesis—[s.l.] Universidade Federal do Rio Grande do Sul, RS, 2022.
- MULLER, P. et al. **Manuel d’annotation en relations de discours du projet annodis.**, 2012.
- MUNIZ, M. C. M. **A construção de recursos linguístico-computacionais para o português do Brasil: o projeto Unitex-PB**. mathesis—[s.l.] Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo - ICMC/USP, 2004.
- NADEAU, D. **Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision**. tese de doutorado—[s.l.] University of Ottawa, 2007.
- NAGAO, M. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. Em: NIRENBURG, S.; SOMERS, H. L.; WILKS, Y. A. (Eds.). **Readings in Machine Translation**. [s.l.] The MIT Press, 1984.
- NAMIUTI, C. O Corpus Anotado do Português Histórico: um avanço para as pesquisas em Linguística Histórica do Português. **Revista Virtual de Estudos da Linguagem**, v. 2, p. 1–9, ago. 2004.
- NARDE, W. **Análise de notícias falsas em rede social: uma abordagem utilizando transferência de aprendizagem e Transformers**. https://www.monografias.ufop.br/bitstream/35400000/3122/6/MONOGRAFIA_AnáliseNotíciasFalsas.pdf, 2021.
- NASAR, Z.; JAFFRY, S. W.; MALIK, M. K. Named entity recognition and relation extraction: State-of-the-art. **ACM Computing Surveys (CSUR)**, v. 54, n. 1, p. 1–39, 2021.
- NASCIMENTO, G. et al. **Hate speech detection using brazilian imageboards**. Proceedings of the 25th Brazillian Symposium on Multimedia and the Web. **Anais...2019**.
- NATH, N.; LEE, S.-H.; LEE, I. NEAR: Named Entity and Attribute Recognition of Clinical Concepts. **J. of Biomedical Informatics**, v. 130, n. C, jun. 2022.
- NAVIGLI, R.; PONZETTO, S. P. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. **Artificial intelligence**, v. 193, p. 217–250, 2012.
- NECO, R. P.; FORCADA, M. L. **Asynchronous translations with recurrent neural nets**. Proceedings of International Conference on Neural Networks (ICNN’97). **Anais...1997**.
- NETO, F. A. R. et al. **Team PiLN at ABSAPT 2022: Lexical and BERT Strategies for Aspect-Based Sentiment Analysis in Portuguese**. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022. **Anais...2022**.
- NEVES, M. H. DE M. **Texto e gramática**. [s.l.] Contexto, 2013.



- NEWMAN, N. et al. **Reuters institute digital news report 2020**. [s.l.] Report of the Reuters Institute for the Study of Journalism, 2020.
- NG, V.; CARDIE, C. **Improving machine learning approaches to coreference resolution**. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. **Anais...Association for Computational Linguistics**, 2002.
- NGUYEN, D. B.; THEOBALD, M.; WEIKUM, G. J-NERD: joint named entity recognition and disambiguation with rich linguistic features. **Transactions of the Association for Computational Linguistics**, v. 4, p. 215–229, 2016.
- NIJKAMP, E. et al. ProGen2: Exploring the Boundaries of Protein Language Models. **CoRR**, v. abs/2206.13517, 2022.
- NIVRE, J. et al. **The CoNLL 2007 Shared Task on Dependency Parsing**. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). **Anais...Prague, Czech Republic: Association for Computational Linguistics**, jun. 2007. Disponível em: <<https://aclanthology.org/D07-1096>>
- NOGUEIRA, R. et al. Document expansion by query prediction. **arXiv preprint arXiv:1904.08375**, 2019.
- NOORALAHZADEH, F.; ØVRELID, L. **Syntactic Dependency Representations in Neural Relation Classification**. Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP. **Anais...Melbourne, Australia: Association for Computational Linguistics**, jul. 2018. Disponível em: <<https://aclanthology.org/W18-2907>>
- NOZAKI, J. et al. **End-to-end Speech-to-Punctuated-Text Recognition**. Proc. Interspeech 2022. **Anais...2022**.
- NUNES, M. DAS G. V. et al. O uso de interlíngua para comunicação via Internet: a decodificação UNL-português. **Revista Tecnologia da Informação**, v. 3, n. 1, p. 49–55, 2003.
- NUNES, P. **LEVANTAMENTO REVELA QUE 90,5% DOS PRESOS POR MONITORAMENTO FACIAL NO BRASIL SÃO NEGROS**. Disponível em: <<https://www.intercept.com.br/2019/11/21/presos-monitoramento-facial-brasil-negros/>>. Acesso em: 28 ago. 2023.
- O'BRIEN, S. Towards predicting post-editing productivity. **Machine translation**, v. 25, p. 197–215, 2011.
- O'BRIEN, S. et al. **Dynamic Quality Evaluation Framework**. [s.l.] TAUS Labs Report. The Translation Automation User Society-TAUS, 2011.
- O'NEIL, C. **Algoritmos de Destruição em Massa**. [s.l.] Editora Rua do Sabão, 2021.
- OCH, F. J.; NEY, H. The Alignment Template Approach to Statistical Machine Translation. **Computational Linguistics**, v. 30, n. 4, p. 417–449, dez. 2004.
- OECD. **The OECD Framework for the Classification of AI systems**. Disponível em: <<https://wp.oecd.ai/app/uploads/2022/02/Classification-2-pager-1.pdf>>. Acesso em: 28 ago. 2023.
- OLIVEIRA, F. S. et al. **CML-TTS: A Multilingual Dataset for Speech Synthesis in Low-Resource Languages**. International Conference on Text, Speech, and Dialogue. **Anais...Springer**, 2023.
- OLIVEIRA JR., M. NURC Digital: um protocolo para a digitalização, anotação, arquivamento e disseminação do material do Projeto da Norma Urbana Linguística Culta (NURC). **CHIMERA: Revista de Corpus de Lenguas Romances y Estudios Lingüísticos**, v. 3, n. 2, p. 149–174, set. 2016.



- OLIVEIRA, L. E. S. et al. SemClinBr - a multi-institutional and multi-specialty semantically annotated corpus for Portuguese clinical NLP tasks. **Journal of Biomedical Semantics**, v. 13, n. 1, a2022.
- OLIVEIRA, L. F. A. DE et al. **Challenges In Annotating A Treebank Of Clinical Narratives In Brazilian Portuguese**. Computational Processing of the Portuguese Language: 15th International Conference, PROPOR 2022, Fortaleza, Brazil, March 21–23, 2022, Proceedings. **Anais...Berlin, Heidelberg: Springer-Verlag, b2022**. Disponível em: <https://doi.org/10.1007/978-3-030-98305-5_9>
- OLIVEIRA, L.; CLARO, D.; SOUZA, M. DptOIE: a Portuguese open information extraction based on dependency analysis. **Artificial Intelligence Review**, v. 56, p. 1–32, dez. 2022.
- OLIVEIRA, M. R. DE. Manual de Linguística. Em: MARTELOTTA, M. E. (Ed.). São Paulo: Contexto, 2008. p. 193–204.
- OLIVEIRA, N. et al. Processamento de Linguagem Natural para Identificação de Notícias Falsas em Redes Sociais: Ferramentas, Tendências e Desafios. Em: [s.l.] SBC, 2020.
- OPENAI. **ChatGPT: OpenA’s conversational AI model**. Disponível em: <<https://openai.com/blog/chatgpt/>>. Acesso em: 7 abr. 2023.
- ORENGO, V. M.; BURIOL, L. S.; COELHO, A. R. **A study on the use of stemming for monolingual ad-hoc Portuguese information retrieval**. Workshop of the Cross-Language Evaluation Forum for European Languages. **Anais...Springer, 2006**.
- ORENGO, V. M.; HUYCK, C. **A Stemming Algorithm for the Portuguese Language**. Proceedings Eighth Symposium on String Processing and Information Retrieval. **Anais...IEEE Computer Society, 2001**.
- OSBORNE, T.; GERDES, K. The status of function words in dependency grammar: A critique of Universal Dependencies (UD). **Glossa: a journal of general linguistics (2016-2021)**, jan. 2019.
- OSGOOD, C. E.; SUCCI, G. J.; TENENBAUM, P. H. **The Measurement of meaning**. Urbana: University of Illinois Press, 1957.
- OSTENDORF, M.; PRICE, P.; SHATTUCK-HUFNAGEL, S. **The Boston University Radio news corpus**, 1995. Disponível em: <<https://doi.org/10.35111/Z7XK-Z229>>
- OUYANG, L. et al. **Training language models to follow instructions with human feedback**. NeurIPS. **Anais...2022**. Disponível em: <http://papers.nips.cc/paper/_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html>
- OVCHINNIKOVA, E. **Integration of World Knowledge for Natural Language Understanding**. [s.l.] Atlantis Press, 2012.
- OVERWIJK, A.; XIONG, C.; CALLAN, J. **ClueWeb22: 10 Billion Web Documents with Rich Information**. (E. Amigó et al., Eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. **Anais...ACM, 2022**. Disponível em: <<https://doi.org/10.1145/3477495.3536321>>
- ÖZSEVEN, T. Investigation of the effect of spectrogram images and different texture analysis methods on speech emotion recognition. **Applied Acoustics**, v. 142, p. 70–77, 2018.
- PAGE, E. B.; PETERSEN, N. S. The Computer Moves into Essay Grading: Updating the Ancient Test. **Phi Delta Kappan**, v. 76, p. 561–565, mar. 1995.
- PAIŞ, V.; TUFIŞ, D. Capitalization and punctuation restoration: a survey. **Artificial Intelligence Review**, v. 55, p. 1681–1722, 2022.
- PALMER, M.; GILDEA, D.; KINGSBURY, P. **The Proposition Bank: An Annotated**



- Corpus of Semantic Roles.** Computational Linguistics, 31: 1. **Anais...**The MIT Press Journals, 2005.
- PAPINENI, K. et al. **BLEU: A Method for Automatic Evaluation of Machine Translation.** Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. **Anais...** ACL '02.USA: Association for Computational Linguistics, 2002. Disponível em: <<https://doi.org/10.3115/1073083.1073135>>
- PARDO, T. et al. **Porttinari - a Large Multi-genre Treebank for Brazilian Portuguese.** Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. **Anais...**Porto Alegre, RS, Brasil: SBC, 2021. Disponível em: <<https://sol.sbc.org.br/index.php/stil/article/view/17778>>
- PARDO, T. A. S. **Métodos para análise discursiva automática.** tese de doutorado—[s.l.] Universidade de São Paulo, 2005.
- PARK, D. S. et al. **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.** Interspeech 2019. **Anais...**ISCA, set. 2019. Disponível em: <<https://doi.org/10.21437%2Finterspeech.2019-2680>>
- PAROUBEK, P.; CHAUDIRON, S.; HIRSCHMAN, L. **Principles of Evaluation in Natural Language Processing.** Traitement Automatique des Langues, Volume 48, Numéro 1 : Principes de l'évaluation en Traitement Automatique des Langues [Principles of Evaluation in Natural Language Processing]. **Anais...**France: ATALA (Association pour le Traitement Automatique des Langues), 2007. Disponível em: <<https://aclanthology.org/2007.tal-1.1>>
- PASQUALOTTI, P. R. **WordNet Affect BR – uma base de expressões de emoção em Português.** [s.l.] Novas Edições Acadêmicas, 2015.
- PELLE, R. P. DE; MOREIRA, V. **Offensive Comments in the Brazilian Web: a dataset and baseline results.** Anais do VI Brazilian Workshop on Social Network Analysis and Mining. **Anais...**2017.
- PENNEBAKER, J. W.; FRANCIS, M. E.; BOOTH, R. J. **Linguistic Inquiry and Word Count.** [s.l.] Lawrence Erlbaum Associates, 2001.
- PENNINGTON, J.; SOCHER, R.; MANNING, C. **GloVe: Global Vectors for Word Representation.** Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). **Anais...**Doha, Qatar: Association for Computational Linguistics, out. 2014. Disponível em: <<https://aclanthology.org/D14-1162>>
- PEREIRA, D. A. A Survey of Sentiment Analysis in the Portuguese Language. **Artificial Intelligence Review**, 2021.
- PEREIRA, V.; PINHEIRO, V. **Report - um sistema de extração de informações aberta para língua portuguesa.** Anais do X Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. **Anais...**SBC, 2015.
- PERRIGO, B. Disponível em: <<https://time.com/6247678/openai-chatgpt-kenya-workers/>>. Acesso em: 9 abr. 2023.
- PERSING, I.; NG, V. **Modeling Prompt Adherence in Student Essays.** Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. **Anais...**Baltimore, Maryland: Association for Computational Linguistics, jun. 2014.
- PETERS, M. E. et al. **Deep Contextualized Word Representations.** (M. A. Walker, H. Ji, A. Stent, Eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers). **Anais...**Association for Computational Linguistics, 2018. Disponível em: <<https://doi.org/10.18653/v1/n18-1202>>



- PETRI, M. J. C. **Manual de Linguagem Jurídica**. 3rd. ed. São Paulo: Saraiva, 2017.
- PIEKOS, P.; MALINOWSKI, M.; MICHALEWSKI, H. **Measuring and Improving BERT's Mathematical Abilities by Predicting the Order of Reasoning**. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). **Anais...Online**: Association for Computational Linguistics, ago. 2021. Disponível em: <<https://aclanthology.org/2021.acl-short.49>>
- PING, W. et al. Deep voice 3: 2000-speaker neural text-to-speech. **arXiv preprint arXiv:1710.07654**, 2017.
- PINHEIRO, V. et al. **InferenceNet.Br: Expression of Inferentialist Semantic Content of the Portuguese Language**. (T. A. S. Pardo et al., Eds.) Computational Processing of the Portuguese Language. **Anais...Berlin, Heidelberg**: Springer Berlin Heidelberg, 2010.
- PIRES, R. et al. **Sabiá: Portuguese Large Language Models**. Anais da XII Brazilian Conference on Intelligent Systems - BRACIS 2023. **Anais...2023**. Disponível em: <<https://arxiv.org/abs/2304.07880>>
- PIRINA, I.; ÇÖLTEKIN, ÇAĞRI. **Identifying Depression on Reddit: The Effect of Training Data**. Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task. **Anais...2018**.
- POESIO, M.; STUCKARDT, R.; VERSLEY, Y. **Anaphora Resolution: Algorithms, Resources, and Applications**. 1. ed. [s.l.] Springer, 2016.
- PONTIKI, M. et al. **SemEval-2014 Task 4: Aspect Based Sentiment Analysis**. Proceedings of the 8th International Workshop on Semantic Evaluation. **Anais...2014**.
- PONTIKI, M. et al. **SemEval-2015 Task 12: Aspect Based Sentiment Analysis**. Proceedings of the 9th International Workshop on Semantic Evaluation. **Anais...2015**.
- PONTIKI, M. et al. **SemEval-2016 Task 5: Aspect Based Sentiment Analysis**. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). **Anais...2016**.
- POPOVIC, M.; BURCHARDT, A. **From Human to Automatic Error Classification for Machine Translation Output**. Proceedings of the 15th Conference of the European Association for Machine Translation. **Anais...Leuven, Belgium**: 2011. Disponível em: <<https://aclanthology.org/2011.eamt-1.36.pdf>>
- POPOVIĆ, M. **chrF: character n-gram F-score for automatic MT evaluation**. Proceedings of the Tenth Workshop on Statistical Machine Translation. **Anais...Lisbon, Portugal**: Association for Computational Linguistics, set. 2015. Disponível em: <<https://aclanthology.org/W15-3049>>
- PORTER, M. F. An algorithm for suffix stripping. **Program**, v. 14, n. 3, p. 130–137, 1980.
- POSNER, J.; RUSSELL, J. A.; PETERSON, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. **Development and psychopathology**, v. 17, n. 3, p. 715–734, 2005.
- PRADHAN, S. et al. **CoNLL-2011 shared task: Modeling unrestricted coreference in ontonotes**. Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task. **Anais...Portland, Oregon**: Association for Computational Linguistics, 2011.
- PRADHAN, S. et al. **CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes**. Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Conference on Natural Language Learning - Shared



- Task. **Anais...**Jeju Island, Korea: 2012.
- PRADHAN, S. et al. **Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation**. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. **Anais...**Baltimore, MD, USA: 2014. Disponível em: <<http://aclweb.org/anthology/P/P14/P14-2006.pdf>>
- PRATAP, V. et al. **Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters.**, a2020. Disponível em: <<https://arxiv.org/abs/2007.03001>>
- PRATAP, V. et al. MLS: A Large-Scale Multilingual Dataset for Speech Research. **Proc. Interspeech 2020**, p. 2757–2761, b2020.
- PROVILKOV, I.; EMELIANENKO, D.; VOITA, E. **BPE-Dropout: Simple and Effective Subword Regularization**. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. **Anais...**Online: Association for Computational Linguistics, jul. 2020.
- PURINGTON, A. et al. ” **Alexa is my new BFF**” **Social Roles, User Satisfaction, and Personification of the Amazon Echo**. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems. **Anais...**2017.
- QIU, Q. et al. BiLSTM-CRF for geological named entity recognition from the geoscience literature. **Earth Science Informatics**, v. 12, n. 4, p. 565–579, 2019.
- QUINTANILHA, I. M.; NETTO, S. L.; BISCAINHO, L. W. P. An open-source end-to-end ASR system for Brazilian Portuguese using DNNs built from newly assembled corpora. **Journal of Communication and Information Systems**, v. 35, n. 1, p. 230–242, 2020.
- RABINER, L. R.; JUANG, B. H. **Fundamentals of Speech Recognition**. [s.l.] Pearson Education, 1993.
- RADEMAKER, A. et al. **Universal Dependencies for Portuguese**. Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017). **Anais...**Pisa, Italy: Linköping University Electronic Press, set. 2017. Disponível em: <<https://aclanthology.org/W17-6523>>
- RADEV, D. R. **A common theory of information fusion from multiple text sources step one: cross-document structure**. 1st SIGdial workshop on Discourse and Dialogue. **Anais...**2000.
- RADFORD, A. et al. **Language Models are Unsupervised Multitask Learners**. 2019.
- RADFORD, A. et al. Robust speech recognition via large-scale weak supervision. **arXiv preprint arXiv:2212.04356**, 2022.
- RADFORD, A.; NARASIMHAN, K. **Improving Language Understanding by Generative Pre-Training**. 2018.
- RAE, J. W. et al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. **CoRR**, v. abs/2112.11446, 2021.
- RAFFEL, C. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **J. Mach. Learn. Res.**, v. 21, p. 140:1–140:67, 2020.
- RAHMAN, A.; NG, V. **Coreference Resolution with World Knowledge**. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. **Anais...**Portland, Oregon, USA: a2011. Disponível em: <<http://www.aclweb.org/anthology/P11-1082>>
- RAHMAN, A.; NG, V. Narrowing the modeling gap: a cluster-ranking approach to coreference resolution. **Journal of Artificial Intelligence Research**, p. 469–521, b2011.
- RAMISCH, R. **Caracterização de desvios sintáticos em redações de estudantes do ensino médio: subsídios para o processamento automático das línguas naturais**.



- mathesis—[s.l.] Universidade Federal de São Carlos, 2020.
- RANCHHOD, E.; MOTA, C.; BAPTISTA, J. **A Computational Lexicon of Portuguese for Automatic Text Parsing**. SIGLEX99: Standardizing Lexical Resources. *Anais...1999*. Disponível em: <<https://aclanthology.org/W99-0511>>
- RAO, K. S.; KOOLAGUDI, S. G.; VEMPADA, R. R. Emotion recognition from speech using global and local prosodic features. *International journal of speech technology*, v. 16, p. 143–160, 2013.
- RASO, T.; MELLO, H. **C-ORAL–BRASIL I: corpus de referência do português brasileiro falado informal**. Belo Horizonte: Editora UFMG, 2012a.
- RASO, T.; MELLO, H. C-ORAL–BRASIL I: corpus de referência do português brasileiro falado informal. A general presentation. *Speech and Corpora*, p. 16, b2012.
- RASO, T.; TEIXEIRA, B.; BARBOSA, P. Modelling automatic detection of prosodic boundaries for Brazilian Portuguese spontaneous speech. *Journal of Speech Sciences*, v. 9, p. 105–128, set. 2020.
- RAU, L. F. **Extracting company names from text**. Proceedings the Seventh IEEE Conference on Artificial Intelligence Application. *Anais...IEEE Computer Society*, 1991.
- READ, J. et al. **Sentence Boundary Detection: A Long Solved Problem?** Proceedings of COLING 2012: Posters. *Anais...Mumbai, India: The COLING 2012 Organizing Committee*, dez. 2012. Disponível em: <<https://aclanthology.org/C12-2096>>
- REAL, L.; FONSECA, E.; GONÇALO OLIVEIRA, H. **The ASSIN 2 Shared Task: A Quick Overview**. Computational Processing of the Portuguese Language: 14th International Conference, PROPOR 2020, Evora, Portugal, March 2–4, 2020, Proceedings. *Anais...Berlin, Heidelberg: Springer-Verlag*, 2020. Disponível em: <https://doi.org/10.1007/978-3-030-41505-1_39>
- RECASENS, M.; HOVY, E. H. BLANC: Implementing the Rand index for coreference evaluation. *Natural Language Engineering*, v. 17, n. 4, p. 485–510, 2011.
- RECUERO, R. **Redes Sociais na Internet**. [s.l.] Ciber Cultura, 2009.
- REI, R. et al. **COMET: A Neural Framework for MT Evaluation**. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). *Anais...Online: Association for Computational Linguistics*, nov. 2020. Disponível em: <<https://aclanthology.org/2020.emnlp-main.213>>
- REIMERS, N.; GUREVYCH, I. **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. *Anais...Association for Computational Linguistics*, nov. 2019. Disponível em: <<https://arxiv.org/abs/1908.10084>>
- REIMERS, N.; GUREVYCH, I. **Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation**. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. *Anais...Association for Computational Linguistics*, nov. 2020. Disponível em: <<https://arxiv.org/abs/2004.09813>>
- RESENDE, G. et al. **(Mis)Information Dissemination in WhatsApp: Gathering, Analyzing and Countermeasures**. Proceedings of the World Wide Web Conference. *Anais...2019*.
- REVIEW, M. T. **Um aplicativo de Inteligência Artificial que “desnudava” mulheres mostra como as deepfakes prejudicam os mais vulneráveis**. Disponível em: <<https://mittechreview.com.br/um-aplicativo-de-inteligencia-artificial-que-desnudava-mulheres-mostra-como-as-deepfakes-prejudicam-os-mais-vulneraveis/>>. Acesso em: 28 ago. 2023.
- REYES, A.; ROSSO, P.; BUSCALDI, D. From Humor Recognition to Irony Detection:



- The Figurative Language of Social Media. **Data & Knowledge Engineering**, 2012.
- RIJSBERGEN, C. JOOST. VAN. **Information Retrieval**. [s.l.] Butterworths, 1979.
- RILOFF, E. et al. **Automatically constructing a dictionary for information extraction tasks**. AAAI. **Anais...Citeseer**, 1993.
- RILOFF, E.; JONES, R.; et al. **Learning dictionaries for information extraction by multi-level bootstrapping**. AAAI/IAAI. **Anais...1999**.
- RIZZOLATTI, G.; ARBIB, M. A. Language within our grasp. **Trends in Neurosciences**, v. 21, n. 5, p. 188–194, 1998.
- RO, Y.; LEE, Y.; KANG, P. **Multi-2OIE: Multilingual Open Information Extraction Based on Multi-Head Attention with BERT**. Findings of the Association for Computational Linguistics: EMNLP 2020. **Anais...Online: Association for Computational Linguistics**, nov. 2020. Disponível em: <<https://aclanthology.org/2020.findings-emnlp.99>>
- ROARK, B.; CHARNAK, E. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. **arXiv preprint cs/0008026**, 2000.
- ROBERTSON, S. E.; SPÄRCK JONES, K. Relevance weighting of search terms. **Journal of the American Society for Information science**, v. 27, n. 3, p. 129–146, 1976.
- ROCCHIO-JR, J. J. Relevance feedback in information retrieval. **The SMART retrieval system: experiments in automatic document processing**, 1971.
- ROCHA, M. A corpus-based study of anaphora in English and Portuguese, Corpus-based and Computational Approaches to Discourse Anaphora. Em: [s.l.] John Benjamins Publishing Company, 2000. p. 81–94.
- RODRIGUES, J. et al. Advancing Neural Encoding of Portuguese with Transformer Albertina PT-. **CoRR**, v. abs/2305.06721, 2023.
- RODRIGUES, R. C. et al. **Portuguese Language Models and Word Embeddings: Evaluating on Semantic Similarity Tasks**. (P. Quaresma et al., Eds.)Computational Processing of the Portuguese Language. **Anais...Springer Nature Switzerland AG: Springer International Publishing**, 2020.
- ROMERA-PAREDES, B.; TORR, P. H. S. **An embarrassingly simple approach to zero-shot learning**. (F. R. Bach, D. M. Blei, Eds.)Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. **Anais...: JMLR Workshop e Conference Proceedings.JMLR.org**, 2015. Disponível em: <<http://proceedings.mlr.press/v37/romera-paredes15.html>>
- RONCARATI, C. **As cadeias do texto: construindo sentidos**. [s.l.] Parábola, 2010.
- ROTH, D.; YIH, W. Global inference for entity and relation identification via a linear programming formulation. **Introduction to statistical relational learning**, p. 553–580, 2007.
- RUPPENHOFER, J. et al. **FrameNet II: Extended theory and practice**. [s.l.: s.n.].
- RUSSEL, S. **Human Compatible Artificial Intelligence and the Problem of Control**. [s.l.] Penguin Books, 2019.
- RUSSELL, M. A. **Mineração de Dados da Web Social**. Primeira edição ed. São Paulo: O'Reilly Novatec, 2011.
- SAEKI, T. et al. **Virtuoso: Massive Multilingual Speech-Text Joint Semi-Supervised Learning for Text-To-Speech.**, 2023. Disponível em: <<https://arxiv.org/abs/2210.15447>>
- SAG, I. A. et al. **Multiword Expressions: A Pain in the Neck for NLP**. Conference on Intelligent Text Processing and Computational Linguistics. **Anais...2002**. Disponível em: <<https://api.semanticscholar.org/CorpusID:1826481>>



- SAGER, N. Natural language information formatting: the automatic conversion of texts to a structured data base. Em: **Advances in computers**. [s.l.] Elsevier, 1978. v. 17p. 89–162.
- SAGER, N.; FRIEDMAN, C.; LYMAN, M. S. **Medical language processing: computer management of narrative data**. [s.l.] Addison-Wesley Longman Publishing Co., Inc., 1987.
- SAI, A. B.; MOHANKUMAR, A. K.; KHAPRA, M. M. A Survey of Evaluation Metrics Used for NLG Systems. **ACM Comput. Surv.**, v. 55, n. 2, p. 26:1–26:39, 2023.
- SALESKY, E. et al. The multilingual tedx corpus for speech recognition and translation. **arXiv preprint arXiv:2102.01757**, 2021.
- SALOMÃO, M. M. M. FrameNet Brasil: A work in progress. **Calidoscópico**, v. 7, p. 171–182, 2009.
- SALTON, G.; ALLAN, J. Text retrieval using the vector processing model. dez. 1994.
- SALTON, G.; MCGILL, M. J. **Introduction to Modern Information Retrieval**. [s.l.] McGraw-Hill, 1983.
- SANDERSON, M. et al. Test collection based evaluation of information retrieval systems. **Foundations and Trends® in Information Retrieval**, v. 4, n. 4, p. 247–375, 2010.
- SANG, E. T. K.; DE MEULDER, F. **Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition**. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. **Anais...2003**.
- SANH, V. et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. **CoRR**, v. abs/1910.01108, 2019.
- SANTANA, B. P. **Morfologia ornamental: as vogais temáticas do português brasileiro o Unitex-PB**. mathesis—Curitiba, PR: Universidade Federal do Paraná, Setor de Ciências Humanas, Programa de Pós-Graduação em Letras, 2019.
- SANTANA, B. S. **A computational-linguistic-based approach to support the analysis of the discursive configuration of violence on social media**. tese de doutorado—[s.l.] Universidade Federal do Rio Grande do Sul, 2023.
- SANTOS, A. A. et al. O teste de Cloze na avaliação da compreensão em leitura. **Psicologia: reflexão e crítica**, v. 15, p. 549–560, 2002.
- SANTOS, C. N. DOS; GUIMARÃES, V. **Boosting Named Entity Recognition with Neural Character Embeddings**. (X. Duan et al., Eds.) Proceedings of the 5th Named Entity Workshop. **Anais...Association for Computational Linguistics**, 2015.
- SANTOS, D. Avaliação conjunta. Em: SANTOS, D. (Ed.). **Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa**. Lisboa, Portugal: IST Press, 2007. p. 1–12.
- SANTOS, D.; CARDOSO, N. **Breve introdução ao HAREM**. (D. Santos, N. Cardoso, Eds.) Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área. **Anais...Linguatca**, a2007. Disponível em: <<http://www.linguatca.pt/LivroHAREM/>>
- SANTOS, D.; CARDOSO, N. **A golden resource for named entity recognition in portuguese**. Proceeding of the 7th International conference on the computational processing of portuguese. **Anais...b2007**.
- SANTOS, D.; CARDOSO, N.; SECO, N. **Avaliação no HAREM: Métodos e medidas**. (D. Santos, N. Cardoso, Eds.) Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área. **Anais...Linguatca**, 2007.
- SANTOS, D.; ROCHA, P. **The key to the first CLEF with Portuguese: Topics**,



- questions and answers in CHAVE. Workshop of the Cross-Language Evaluation Forum for European Languages. *Anais...*2004.
- SANTOS, H. D. P. D.; ULBRICH, A. H. D. P. S.; VIEIRA, R. **Evaluation of a Prescription Outlier Detection System in Hospital's Pharmacy Services**. 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). *Anais...IEEE*, 2021.
- SANTOS, J. et al. **Assessing the Impact of Contextual Embeddings for Portuguese Named Entity Recognition**. Proceedings of the 8th Brazilian Conference on Intelligent Systems. *Anais...*2019.
- SANTOS, J. et al. **De-identification of clinical notes using contextualized language models and a token classifier**. Brazilian Conference on Intelligent Systems. *Anais...*Springer, 2021.
- SANTOS, J.; SANTOS, H. D. P. DOS; VIEIRA, R. **Fall Detection in Clinical Notes using Language Models and Token Classifier**. (A. G. S. de Herrera et al., Eds.)Proceedings of the 33rd IEEE International Symposium on Computer-Based Medical Systems. *Anais...*2020.
- SANTOS, V. G. et al. **CORAA NURC-SP Minimal Corpus: a manually annotated corpus of Brazilian Portuguese spontaneous speech**. Proc. IberSPEECH 2022. *Anais...*2022.
- SARAH HICKEY. **Nimdzi 100 - Language Services Industry Market Report 2020.pdf**. [s.l: s.n.].
- SARDINHA, T. B. Lingüística de Corpus: histórico e problemática. **DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada**, v. 16, n. 2, p. 323–367, 2000.
- SARMENTO, C. DA S. **Da Abordagem do Léxico em Livros Didáticos de Língua Portuguesa: os Anos Finais do Ensino Fundamental**. mathesis—Brasília: UnB, 2019.
- SARMENTO, L.; PINTO, A. S.; CABRAL, L. **REPENTINO – a wide-scope gazetteer for entity recognition in portuguese**. Proceedings of International Workshop on Computational Processing of the Portuguese Language. *Anais...*Springer, 2006.
- SARTORI, L.; THEODOROU, A. A Sociotechnical Perspective for the Future of AI: Narratives, Inequalities, and Human Control. **Ethics and Inf. Technol.**, v. 24, n. 1, mar. 2022.
- SCAO, T. L. et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. **CoRR**, v. abs/2211.05100, 2022.
- SCARTON, C. E.; ALUISIO, S. M. **Towards a cross-linguistic VerbNet-style lexicon for Brazilian portuguese**. Workshop on Creating Cross-language Resources for Disconnected Languages and Styles - CREDISLAS. *Anais...ELRA*, 2012.
- SCARTON, C. E.; ALUÍSIO, S. M. Análise da Inteligibilidade de textos via ferramentas de Processamento de Língua Natural: adaptando as métricas do Coh-Metrix para o Português. **Linguamática**, v. 2, n. 1, p. 45–61, abr. 2010.
- SCHANK, R. C. et al. **MARGIE: Memory Analysis Response Generation, and Inference on English**. IJCAI. *Anais...*1973.
- SCHICK, T.; SCHÜTZE, H. **Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference**. (P. Merlo, J. Tiedemann, R. Tsarfaty, Eds.)Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021. *Anais...*Association for Computational Linguistics, 2021. Disponível em:



- <<https://doi.org/10.18653/v1/2021.eacl-main.20>>
- SCHMID, H. **Part-of-Speech Tagging with Neural Networks.**, 1994. Disponível em: <<https://arxiv.org/abs/cmp-lg/9410018>>
- SCHMIDHUBER, J.; HEIL, S. Sequential neural text compression. **IEEE Transactions on Neural Networks**, v. 7, n. 1, p. 142–146, 1996.
- SCHMITZ, M. et al. **Open language learning for information extraction.** Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. **Anais...**: EMNLP-CoNLL '12. Stroudsburg, PA, USA: Association for Computational Linguistics; Association for Computational Linguistics, 2012. Disponível em: <<http://dl.acm.org/citation.cfm?id=2390948.2391009>>
- SCHUBERT, G.; FREITAS, L. A. DE. **A Construção de um Corpus para Detecção de Ironia e Sarcasmo em Português.** Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional. **Anais...**2020.
- SCHUSTER, M.; NAKAJIMA, K. **Japanese and Korean voice search.** 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). **Anais...**2012.
- SEARA, I. **Estudo Estatístico dos Fonemas do Português Brasileiro Falado na Capital de Santa Catarina para Elaboração de Frases Foneticamente Balanceadas.** tese de doutorado—[s.l.] Dissertação de Mestrado, Universidade Federal de Santa Catarina ..., 1994.
- SEKINE, S. **Description of the Japanese NE system used for MET-2.** Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998. **Anais...**1998.
- SELLAM, T.; DAS, D.; PARIKH, A. P. **BLEURT: Learning Robust Metrics for Text Generation.** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. **Anais...**2020. Disponível em: <<https://doi.org/10.18653/v1/2020.acl-main.704>>
- SELLARS, W. Inference and Meaning. **Mind**, v. 62, n. 247, p. 313–338, 1953.
- SENA, C. F. L.; CLARO, D. B. InferPortOIE: A Portuguese Open Information Extraction system with inferences. **Natural Language Engineering**, v. 25, n. 2, p. 287–306, 2019.
- SENA, C. F. L.; CLARO, D. B. PragmaticOIE: a pragmatic open information extraction for Portuguese language. **Knowl. Inf. Syst.**, v. 62, n. 9, p. 3811–3836, 2020.
- SENA, C. F. L.; GLAUBER, R.; CLARO, D. B. **Inference Approach to Enhance a Portuguese Open Information Extraction.** Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 3: ICEIS. **Anais...**INSTICC; SciTePress, 2017.
- SENNRICH, R.; HADDOW, B.; BIRCH, A. **Improving Neural Machine Translation Models with Monolingual Data.** Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). **Anais...**a2016. Disponível em: <<https://arxiv.org/abs/1511.06709>>
- SENNRICH, R.; HADDOW, B.; BIRCH, A. **Neural Machine Translation of Rare Words with Subword Units.** Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Berlin, Germany: Association for Computational Linguistics, ago. b2016. Disponível em: <<https://aclanthology.org/P16-1162>>
- SENO, E. R. M. **RHeSumaRST: um sumariador automático de estruturas RST.** mathesis—[s.l.] Universidade Federal de São Carlos, 2005.



- SERRA, C. R. **Realização e percepção de fronteiras prosódicas no português do Brasil: fala espontânea e leitura**. tese de doutorado—Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2009.
- SHANNON, C. E. Prediction and entropy of printed English. **Bell System Technical Journal**, v. 30, n. 1, p. 50–64, 1951.
- SHAOWEI, Z. et al. Survey of Supervised Joint Entity Relation Extraction Methods. **Journal of Frontiers of Computer Science & Technology**, v. 16, n. 4, 2022.
- SHAPIRO, S. C. SNePS: A Logic for Natural Language Understanding and Commonsense Reasoning. Em: **Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language**. Cambridge, MA, USA: MIT Press, 2000. p. 175–195.
- SHEIKHALISHAHI, S. et al. Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review. **JMIR Med Inform**, v. 7, n. 2, p. e12239, abr. 2019.
- SHEN, J. et al. **Natural tts synthesis by conditioning wavenet on mel spectrogram predictions**. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). **Anais...IEEE**, 2018.
- SHERMIS, M. D.; BURSTEIN, J. **Handbook of Automated Essay Evaluation: Current Applications and New Directions**. [s.l.] Routledge/Taylor & Francis Group, 2013.
- SHI, Z.; LIPANI, A. **Don't Stop Pretraining? Make Prompt-based Fine-tuning Powerful Learner.**, 2023. Disponível em: <<https://arxiv.org/abs/2305.01711>>
- SHICKEL, B. et al. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. **IEEE J Biomed Health Inform**, v. 22, n. 5, p. 1589–1604, out. 2017.
- SHIMANAKA, H.; KAJIWARA, T.; KOMACHI, M. Machine Translation Evaluation with BERT Regressor. **arXiv**, v. abs/1907.12679, 2019.
- SHTERIONOV, D. et al. Human versus Automatic Quality Evaluation of NMT and PBSMT. **Machine Translation**, v. 32, n. 3, p. 217–235, 2018.
- SIDDHI, D.; VERGHESE, J. M.; BHAVIK, D. Survey on various methods of text to speech synthesis. **International Journal of Computer Applications**, v. 165, n. 6, 2017.
- SIDNER, C. **A progress report on the discourse and reference components of PAL**. [s.l.] Massachusetts Institute of Tech Cambridge Artificial Intelligence LAB, 1978.
- SILVA, A. P. DA et al. Risco de queda relacionado a medicamentos em hospitais: abordagem de aprendizado de máquina. **Acta Paulista de Enfermagem**, v. 36, 2023.
- SILVA, E.; PARDO, T.; ROMAN, N. **Etiquetagem morfossintática multigênero para o português do Brasil segundo o modelo Universal Dependencieš**. Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. **Anais...Porto Alegre, RS, Brasil: SBC**, 2023. Disponível em: <<https://sol.sbc.org.br/index.php/stil/article/view/25438>>
- SILVA, F. L. V. DA et al. ABSAPT 2022 at IberLEF: Overview of the Task on Aspect-Based Sentiment Analysis in Portuguese. **Procesamiento del Lenguaje Natural**, 2022.
- SILVA, F. R. A. DA. **Deteção de Ironia e Sarcasmo em Língua Portuguesa: uma abordagem utilizando Deep Learning**. <https://github.com/fabio-ricardo/deteccao-ironia>, 2018.
- SILVA, I. A. L. DA et al. Translation, post-editing and directionality. **Translation in transition: Between cognition, computing and technology**, p. 107–134, 2017.
- SILVA, J. F. DA. **Resolução de correferência em múltiplos documentos utilizando**



- aprendizado não supervisionado.** Dissertação de Mestrado, Universidade de São Paulo, 2011.
- SILVA, M. J.; CARVALHO, P.; SARMENTO, L. **Building a Sentiment Lexicon for Social Judgement Mining.** Proceedings of the 10th International Conference on Computational Processing of the Portuguese Language. **Anais...**2012.
- SIMÕES, A.; GUINOVART, X. G. **Bootstrapping a Portuguese WordNet from Galician, Spanish and English Wordnets.** IberSPEECH Conference. **Anais...**2014. Disponível em: <<https://api.semanticscholar.org/CorpusID:10377782>>
- SINGH, P. et al. **Open Mind Common Sense: Knowledge Acquisition from the General Public.** (R. Meersman, Z. Tari, Eds.) On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE. **Anais...**Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.
- SINGH, Y. B.; GOEL, S. A systematic literature review of speech emotion recognition approaches. **Neurocomputing**, 2022.
- Slator 2019 Language Industry Market Report. p. 33, 2019.
- SMIRNOVA, A.; CUDRÉ-MAUROUX, P. Relation extraction using distant supervision: A survey. **ACM Computing Surveys (CSUR)**, v. 51, n. 5, p. 1–35, 2018.
- SMITH, G.; RUSTAGI, I. **Mitigating Bias in Artificial Intelligence: An Equity Fluent Leadership Playbook.** [s.l.] Berkeley Haas Center for Equity, Gender; Leadership, 2020.
- SMITH, K. S. **On Integrating Discourse in Machine Translation.** Proceedings of the Third Workshop on Discourse in Machine Translation. **Anais...**2017.
- SNOVER, M. G. et al. **A Study of Translation Edit Rate with Targeted Human Annotation.** Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006. **Anais...**2006. Disponível em: <<https://aclanthology.org/2006.amta-papers.25/>>
- SOCHER, R. et al. **Semantic compositionality through recursive matrix-vector spaces.** Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. **Anais...**2012.
- SODERLAND, S. et al. **CRYSTAL inducing a conceptual dictionary.** Proceedings of the 14th international joint conference on Artificial intelligence-Volume 2. **Anais...**1995.
- SOLORIO, T. MALINCHE: A NER system for Portuguese that reuses knowledge from Spanish. **Reconhecimento de entidades mencionadas em português: Documentação e atas do HAREM, a primeira avaliação conjunta na área, Capítulo**, v. 10, p. 123–136, 2007.
- SOON, W. M.; NG, H. T.; LIM, C. Y. A Machine Learning Approach to Coreference Resolution of Noun Phrases. **Computational Linguistics**, v. 27, n. 4, p. 521–544, 2001.
- SOUSA, A. et al. **Cross-Lingual Annotation Projection for Argument Mining in Portuguese.** (G. Marreiros et al., Eds.) Progress in Artificial Intelligence. **Anais...**Springer International Publishing, 2021.
- SOUSA, R. F. DE; BRUM, H. B.; NUNES, M. DAS G. V. **A bunch of helpfulness and sentiment corpora in brazilian portuguese.** Proceedings of Symposium in Information and Human Language Technology. **Anais...**2019.
- SOUZA, E. DE. **Construção e avaliação de um treebank padrão ouro.** Mestrado—[s.l.] PUC-Rio, 2023.
- SOUZA, E. DE; FREITAS, C. **Explorando variações no tagset e na anotação Universal Dependencies (UD) para Português: Possibilidades e resultados com base no treebank PetroGold.** Anais do XIV Simpósio Brasileiro de Tecnologia da



- Informação e da Linguagem Humana. **Anais...** Association for Computational Linguistics, 2023.
- SOUZA, E. N. P. DE; CLARO, D. B.; GLAUBER, R. A Similarity Grammatical Structures Based Method for Improving Open Information Systems. **j-jucs**, v. 24, n. 1, p. 43–69, 28 jan. 2018.
- SOUZA, E. N. P.; CLARO, D. B. Extração de Relações utilizando Features Diferenciadas para Português. **Linguamática**, v. 6, n. 2, p. 57–65, 2014.
- SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. **BERTimbau: pretrained BERT models for Brazilian Portuguese**. (R. Cerri, R. C. Prati, Eds.) Proceedings of the 2020 Brazilian Conference on Intelligent Systems. **Anais...** Springer International Publishing, 2020.
- SOUZA, J. W. DA C. **Descrição linguística da complementaridade para a sumarização automática multidocumento**. mathesis—[s.l.] Universidade Federal de São Carlos, 2015.
- SOUZA, J. W. DA C. **Aprofundamento da caracterização linguístico-computacional da complementaridade em um corpus jornalístico multidocumento**. tese de doutorado—[s.l.] Universidade Federal de São Carlos, 2019.
- SOUZA, M. et al. **Construction of a Portuguese Opinion Lexicon from multiple resources**. Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology. **Anais...**2011.
- SPÄRCK JONES, K. Report on the need for and provision of an 'ideal' information retrieval test collection. **Computer Laboratory**, 1975.
- SPÄRCK JONES, K.; WALKER, S.; ROBERTSON, S. E. A probabilistic model of information retrieval: development and comparative experiments. **Information processing & management**, v. 36, n. 6, p. 809–840, 2000.
- SPEER, R.; CHIN, J.; HAVASI, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. **CoRR**, v. abs/1612.03975, 2016.
- STAB, C. et al. **Argumentation Mining in Persuasive Essays and Scientific Articles from the Discourse Structure Perspective**. ArgNLP. **Anais...**2014.
- STANOJEVIC, M.; SIMA'AN, K. **BEER: BEtter Evaluation as Ranking**. Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA. **Anais...**2014. Disponível em: <<https://doi.org/10.3115/v1/w14-3354>>
- STANOVSKY, G. et al. **Supervised open information extraction**. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). **Anais...**2018.
- STEVENS, S. S. A Scale for the Measurement of the Psychological Magnitude Pitch. **Acoustical Society of America Journal**, v. 8, n. 3, p. 185, jan. 1937.
- STIENNON, N. et al. **Learning to summarize with human feedback**. (H. Larochelle et al., Eds.) Advances in Neural Information Processing Systems. **Anais...**Curran Associates, Inc., 2020. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf>
- SU, K.-Y.; WU, M.-W.; CHANG, J.-S. **A new quantitative quality measure for machine translation systems**. Proceedings of the 14th conference on Computational linguistics -. **Anais...**Association for Computational Linguistics, 1992. Disponível em: <<http://dx.doi.org/10.3115/992133.992137>>
- SUCHANEK, F. M.; KASNECI, G.; WEIKUM, G. **Yago: a core of semantic knowledge**. Proceedings of the 16th international conference on World Wide Web. **Anais...**2007.
- SUNKARA, M. et al. **Multimodal Semi-Supervised Learning Framework for**



- Punctuation Prediction in Conversational Speech.** Proc. Interspeech 2020. **Anais...2020.**
- SUNKARA, M. et al. Neural Inverse Text Normalization. **CoRR**, v. abs/2102.06380, 2021.
- SUTSKEVER, I.; VINYALS, O.; LE, Q. V. **Sequence to Sequence Learning with Neural Networks.** (Z. Ghahramani et al., Eds.)Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. **Anais...2014.** Disponível em: <<https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html>>
- TABOADA, M.; MANN, W. C. Rhetorical structure theory: Looking back and moving ahead. **Discourse studies**, v. 8, n. 3, p. 423–459, 2006.
- TACHIBANA, H.; UENOYAMA, K.; AIHARA, S. Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention. **arXiv preprint arXiv:1710.08969**, 2017.
- TAKAMATSU, S.; SATO, I.; NAKAGAWA, H. **Reducing wrong labels in distant supervision for relation extraction.** Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...2012.**
- TAN, K. L.; LEE, C. P.; LIM, K. M. A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. **Applied Sciences**, 2023.
- TAN, X. et al. A survey on neural speech synthesis. **arXiv preprint arXiv:2106.15561**, 2021.
- TANAKA, E. et al. Cem Mil Podcasts: A Spoken Portuguese Document Corpus. **arXiv preprint arXiv:2209.11871**, 2022.
- TANG, Y. et al. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning. **CoRR**, v. abs/2008.00401, 2020.
- TAUS. **TAUS - The Translation Industry in 2022 Report.**, 2020. Disponível em: <<https://info.taus.net/translation-industry-2022-report-download>>. Acesso em: 19 ago. 2020
- TAYLOR, R. et al. Galactica: A Large Language Model for Science. **CoRR**, v. abs/2211.09085, 2022.
- TAYLOR, W. L. “Cloze procedure”: A new tool for measuring readability. **Journalism quarterly**, v. 30, n. 4, p. 415–433, 1953.
- TEIXEIRA, B. H. F. **Detecção automática de fronteiras prosódicas na fala espontânea.** tese de doutorado—Belo Horizonte: Universidade Federal de Minas Gerais, 2022.
- TEIXEIRA, B. H. F.; MITTMAN, M. M. Acoustic Models for the Automatic Identification of Prosodic Boundaries in Spontaneous Speech. **Revista de Estudos da Linguagem**, v. 26, n. 4, p. 1455–1488, 2018.
- TEIXEIRA, B.; BARBOSA, P.; RASO, T. **Automatic Detection of Prosodic Boundaries in Brazilian Portuguese Spontaneous Speech.** (A. Villavicencio et al., Eds.)Computational Processing of the Portuguese Language. **Anais...Cham: Springer International Publishing**, 2018.
- TEIXEIRA, J. P. et al. **Phonetic Events from the Labeling the European Portuguese DataBase for Speech Synthesis, FEUP/IPBDB.** Seventh European Conference on Speech Communication and Technology. **Anais...2001.**
- TEIXEIRA, J. P.; FREITAS, D.; FUJISAKI, H. **Prediction of Fujisaki model’s phrase commands.** Eighth European Conference on Speech Communication and Technology.



Anais...2003.

TEIXEIRA, S. C. S. B.; MARENGO, S. M. D. A.; FINATTO, M. J. B. Construindo fichas terminológicas para estudos sócio-históricos. **Revista Diálogos**, v. 10, n. 3, p. 261–279, 2022.

TEIXEIRA, S. H.; ZAMORA, M. H. Pensando a interseccionalidade a partir da vida e morte de Marielle Franco. **Dignidade Re-Vista**, 2019.

TESNIÈRE, L. **Éléments de Syntaxe Structurale**. Paris: Klincksieck, 1959.

THOPPILAN, R. et al. LaMDA: Language Models for Dialog Applications. **CoRR**, v. abs/2201.08239, 2022.

TIRRELL, L. Toxic Speech: Inoculations and Antidotes. **The Southern Journal of Philosophy**, 2018.

TOKUDA, K. et al. **Speech parameter generation algorithms for HMM-based speech synthesis**. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100). **Anais...IEEE**, 2000.

TOLLES, J.; MEURER, W. J. Logistic Regression: Relating Patient Characteristics to Outcomes. **JAMA**, v. 316, n. 5, p. 533–534, 2016.

TORAL, A. et al. **Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation**. Proceedings of WMT. **Anais...Brussels**, Belgium: 2018.

TORRENT, T. T. et al. **Copa 2014 FrameNet Brasil: a frame-based trilingual electronic dictionary for the Football World Cup**. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations. **Anais...Dublin**, Ireland: Dublin City University; Association for Computational Linguistics, ago. 2014. Disponível em: <<https://aclanthology.org/C14-2003>>

TORRENT, T. T.; ELLSWORTH, M. Behind the Labels: Criteria for Defining Analytical Categories in FrameNet Brasil. **Veredas-Revista de Estudos Linguísticos**, v. 17, n. 1, p. 44–66, 2013.

TOUVRON, H. et al. LLaMA: Open and Efficient Foundation Language Models. **CoRR**, v. abs/2302.13971, 2023.

TRAJANO, D.; BORDINI, R. H.; VIEIRA, R. OLID-BR: offensive language identification dataset for Brazilian Portuguese. **Language Resources and Evaluation**, 2023.

TURCHIOE, M. R. et al. Systematic review of current natural language processing methods and applications in cardiology. **Heart**, v. 108, n. 12, p. 909–916, 2022.

UCHIDA, H.; ZHU, M.; DELLA SENTA, T. A gift for a millennium. **IAS/UNU, Tokyo**, 1999.

UNESCO, D. G. **Recomendação sobre a Ética da Inteligência Artificial**. Disponível em: <https://unesdoc.unesco.org/ark:/48223/pf0000381137_por>. Acesso em: 28 ago. 2023.

UNICEF. **Declaração Universal dos Direitos Humanos**. Disponível em: <<https://www.unicef.org/brazil/declaracao-universal-dos-direitos-humanos>>. Acesso em: 28 ago. 2023.

USZKOREIT, H.; LOMMEL, A. **Multidimensional Quality Metrics: A New Unified Paradigm for Human and Machine Translation Quality Assessment**. [s.l: s.n.].

UZÊDA, V. R.; PARDO, T. A. S.; NUNES, M. G. V. A comprehensive comparative evaluation of RST-based summarization methods. **ACM Transactions on Speech and Language Processing (TSLP)**, v. 6, n. 4, p. 1–20, 2010.

VALLE, R. et al. Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis. **arXiv preprint arXiv:2005.05957**, 2020.



- VARGAS, D. F.; VAN DER LANN, R. H. A contribuição da terminologia na construção de linguagens documentárias como os tesouros. **Biblos**, v. 25, n. 1, p. 21–34, 2011.
- VARGAS, F. et al. **HateBR: A Large Expert Annotated Corpus of Brazilian Instagram Comments for Offensive Language and Hate Speech Detection**. Proceedings of the Thirteenth Language Resources and Evaluation Conference. **Anais...2022**.
- VARGAS, F. A.; PARDO, T. A. S. **Aspect clustering methods for sentiment analysis**. Proceedings of International conference on computational processing of the Portuguese language. **Anais...Springer**, 2018.
- VARGAS, F. A.; SANTOS, R. S. S. D.; ROCHA, P. R. **Identifying Fine-Grained Opinion and Classifying Polarity on Coronavirus Pandemic**. Proceedings of the Brazilian Conference on Intelligent Systems. **Anais...2020**.
- VASWANI, A. et al. **Attention is All you Need**. (I. Guyon et al., Eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. **Anais...2017**. Disponível em: <<https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>>
- VEAUX, C. et al. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. **University of Edinburgh. The Centre for Speech Technology Research (CSTR)**, 2017.
- VIEIRA, F. E.; FARACO, C. A. **Texto e discurso. Escrever na universidade**. [s.l.] Parábola, 2019.
- VIEIRA, R. et al. Coreference and anaphoric relations of demonstrative noun phrases in multilingual corpus. **Anaphora Processing: linguistic, cognitive and computational modeling**, p. 385–403, 2005.
- VIEIRA, R.; GONÇALVES, P. N.; SOUZA, J. G. C. DE. Processamento computacional de anáfora e correferência. **Revista de Estudos da Linguagem**, v. 16, n. 1, 2012.
- VILAIN, M. et al. **A model-theoretic coreference scoring scheme**. Proceedings of the 6th Conference on Message understanding. **Anais...Columbia, Maryland: 1995**.
- VILAR, D. et al. **Error Analysis of Statistical Machine Translation Output**. Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06). **Anais...Genoa, Italy: European Language Resources Association (ELRA)**, 2006. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2006/pdf/413_pdf.pdf>
- VRANDEČIĆ, D.; KRÖTZSCH, M. Wikidata: a free collaborative knowledgebase. **Communications of the ACM**, v. 57, n. 10, p. 78–85, 2014.
- WAGNER FILHO, J. A. et al. **The brWaC Corpus: A New Open Resource for Brazilian Portuguese**. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). **Anais...Miyazaki, Japan: European Language Resources Association (ELRA)**, 2018. Disponível em: <<https://aclanthology.org/L18-1686>>
- WAGNER, J. et al. Dawn of the transformer era in speech emotion recognition: closing the valence gap. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2023.
- WALLIS, S. Completing Parsed Corpora. Em: ABEILLÉ, A. (Ed.). **Trebanks: Building and Using Parsed Corpora**. Dordrecht: Springer Netherlands, 2003. p. 61–71.
- WANG, A. et al. **GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding**. Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. **Anais...Brussels**,



- Belgium: Association for Computational Linguistics, nov. 2018. Disponível em: <https://aclanthology.org/W18-5446/>
- WANG, A. et al. **SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems**. Advances in Neural Information Processing Systems. **Anais...**2019.
- WANG, B.; KOMATSUZAKI, A. **GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model**. <https://github.com/kingoflolz/mesh-transformer-jax>, 2021.
- WANG, C. et al. Covost: A diverse multilingual speech-to-text translation corpus. **arXiv preprint arXiv:2002.01320**, 2020.
- WANG, C. et al. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. **arXiv preprint arXiv:2101.00390**, a2021.
- WANG, C. et al. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. **arXiv preprint arXiv:2301.02111**, 2023.
- WANG, C.; WU, A.; PINO, J. Covost 2 and massively multilingual speech-to-text translation. **arXiv preprint arXiv:2007.10310**, 2020.
- WANG, L. et al. **Relation classification via multi-level attention cnns**. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**2016.
- WANG, W. Y.; GEORGILA, K. **Automatic detection of unnatural word-level segments in unit-selection speech synthesis**. 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. **Anais...IEEE**, 2011.
- WANG, Y. et al. Tacotron: A fully end-to-end text-to-speech synthesis model. **arXiv preprint arXiv:1703.10135**, 2017.
- WANG, Y. et al. **CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation**. (M.-F. Moens et al., Eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. **Anais...**Association for Computational Linguistics, b2021. Disponível em: <<https://doi.org/10.18653/v1/2021.emnlp-main.685>>
- WANI, T. M. et al. A comprehensive review of speech emotion recognition systems. **IEEE Access**, v. 9, p. 47795–47814, 2021.
- WASSERMAN, S.; FAUST, K. **Social network analysis: Methods and applications**. [s.l.] Cambridge university press, 1994.
- WAY, A. Quality Expectations of Machine Translation. Em: MOORKENS, J. et al. (Eds.). **Translation Quality Assessment: From Principles to Practice**. Cham: Springer International Publishing, 2018. p. 159–178.
- WAY, A.; FORCADA, M. L. Editors' foreword to the invited issue on SMT and NMT. **Machine Translation**, v. 32, n. 3, p. 191–194, set. 2018.
- WEI, J. et al. Emergent Abilities of Large Language Models. **Trans. Mach. Learn. Res.**, v. 2022, a2022.
- WEI, J. et al. **Chain-of-Thought Prompting Elicits Reasoning in Large Language Models**. NeurIPS. **Anais...**b2022. Disponível em: <http://papers.nips.cc/paper/_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html>
- WERBOS, P. J. Backpropagation through time: what it does and how to do it. **Proc. IEEE**, v. 78, n. 10, p. 1550–1560, 1990.
- WIEGREFFE, S.; PINTER, Y. **Attention is not not Explanation**. (K. Inui et al., Eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language



- Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). **Anais...**Hong Kong, China: Association for Computational Linguistics, nov. 2019. Disponível em: <<https://aclanthology.org/D19-1002>>
- WIGHTMAN, C. W.; OSTENDORF, M. Automatic recognition of prosodic phrases. [Proceedings] **ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing**, v. 1, p. 321–324, 1991.
- WILLIAMS, I. et al. **Contextual speech recognition in end-to-end neural network systems using beam search**. 2018. Disponível em: <https://www.isca-speech.org/archive/Interspeech_2018/pdfs/2416.pdf>
- WOLF, T. et al. **Transformers: State-of-the-Art Natural Language Processing**. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. **Anais...**Online: Association for Computational Linguistics, out. 2020. Disponível em: <<https://www.aclweb.org/anthology/2020.emnlp-demos.6>>
- WOLINSKI, F.; VICHOT, F.; DILLET, B. **Automatic processing proper names in texts**. Proc. Conference on European Chapter of the Association for Computational Linguistics. **Anais...**EACL, 1995.
- WU, H. et al. SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. **J Am Med Inform Assoc**, v. 25, n. 5, p. 530–537, 2018.
- WU, Y. et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. **arXiv preprint arXiv:1609.08144**, 2016.
- WU, Y. et al. **Memorizing Transformers**. The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. **Anais...**OpenReview.net, 2022. Disponível em: <<https://openreview.net/forum?id=TrjbxzRcnf->>
- XAVIER, C. C.; LIMA, V. L. S. DE; SOUZA, M. Open information extraction based on lexical semantics. **Journal of the Brazilian Computer Society**, v. 21, n. 1, p. 1–14, 2015.
- XAVIER, R. C. **Português no Direito: Linguagem Forense**. Rio de Janeiro: Forense, 2002. p. 1
- XIE, S. M. et al. **An Explanation of In-context Learning as Implicit Bayesian Inference**. The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. **Anais...**OpenReview.net, 2022. Disponível em: <<https://openreview.net/forum?id=RdJVFCHjUMI>>
- XIONG, R. et al. **On Layer Normalization in the Transformer Architecture**. Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. **Anais...**: Proceedings of Machine Learning Research.PMLR, 2020. Disponível em: <<http://proceedings.mlr.press/v119/xiong20b.html>>
- XU, W.; RUDNICKY, A. **Can artificial neural networks learn language models?** Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000). **Anais...**2000.
- XU, Y. et al. **Hard Sample Aware Prompt-Tuning**. (A. Rogers, J. L. Boyd-Graber, N. Okazaki, Eds.)Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023. **Anais...**Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.acl-long.690>>
- XUE, L. et al. **mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer**. (K. Toutanova et al., Eds.)Proceedings of the 2021 Conference



- of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021. **Anais...**Association for Computational Linguistics, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.naacl-main.41>>
- YAMAGUCHI, A. et al. **Frustratingly Simple Pretraining Alternatives to Masked Language Modeling**. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. **Anais...**Online; Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. 2021. Disponível em: <<https://aclanthology.org/2021.emnlp-main.249>>
- YAN, M. Y.; GUSTAD, L. T.; NYTRØ, Ø. Sepsis prediction, early detection, and identification using clinical text for machine learning: a systematic review. **J Am Med Inform Assoc**, v. 29, n. 3, p. 559–575, jan. 2022.
- YANG, H. et al. Clinical Trial Classification of SNS24 Calls with Neural Networks. **Future Internet**, v. 14, n. 5, p. 130, 2022.
- YANG, J.-H. et al. **Enriching Mandarin speech recognition by incorporating a hierarchical prosody model**. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). **Anais...**2011. Disponível em: <<https://doi.org/10.1109/ICASSP.2011.5947492>>
- YANG, M. et al. **Learning ASR pathways: A sparse multilingual ASR model.**, 2023. Disponível em: <<https://arxiv.org/abs/2209.05735>>
- YANG, P.; FANG, H.; LIN, J. **Anserini: Enabling the use of lucene for information retrieval research**. Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. **Anais...**2017.
- YANG, X. et al. **An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming**. Proceeding of Association for Computational Linguistics. **Anais...**2008.
- YANG, Z. et al. **XLNet: Generalized Autoregressive Pretraining for Language Understanding**. (H. M. Wallach et al., Eds.)Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. **Anais...**2019. Disponível em: <<https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html>>
- YI, J.; TAO, J. Self-attention Based Model for Punctuation Prediction Using Word and Speech Embeddings. **ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, p. 7270–7274, 2019.
- YU, X.; LAM, W. **Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach**. Coling 2010: Posters. **Anais...**2010.
- YUAN, W.; NEUBIG, G.; LIU, P. **BARTScore: Evaluating Generated Text as Text Generation**. Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual. **Anais...**2021. Disponível em: <<https://proceedings.neurips.cc/paper/2021/hash/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Abstract.html>>
- YUAN, Y. et al. **A relation-specific attention network for joint entity and relation extraction**. International joint conference on artificial intelligence. **Anais...**International Joint Conference on Artificial Intelligence, 2021.
- ZE, H.; SENIOR, A.; SCHUSTER, M. **Statistical parametric speech synthesis using deep neural networks**. 2013 iee international conference on acoustics, speech and signal



- processing. **Anais...IEEE**, 2013.
- ZELASKO, P. et al. **Punctuation Prediction Model for Conversational Speech**. (B. Yegnanarayana, Ed.) Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018. **Anais...ISCA**, 2018. Disponível em: <<https://doi.org/10.21437/Interspeech.2018-1096>>
- ZELENKO, D.; AONE, C.; RICHARDELLA, A. Kernel methods for relation extraction. **Journal of machine learning research**, v. 3, n. Feb, p. 1083–1106, 2003.
- ZEMAN, D. **Reusable Tagset Conversion Using Tagset Drivers**. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08). **Anais...Marrakech, Morocco: European Language Resources Association (ELRA)**, 2008. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf>
- ZEMAN, D.; RESNIK, P. **Cross-Language Parser Adaptation between Related Languages**. Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages. **Anais...2008**. Disponível em: <<https://aclanthology.org/I08-3008>>
- ZEN, H. et al. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. **Proc. Interspeech 2019**, p. 1526–1530, 2019.
- ZENG, D. et al. **Relation classification via convolutional deep neural network**. Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers. **Anais...2014**.
- ZEWDU, A.; YITAGESU, B. Part of speech tagging: a systematic review of deep learning and machine learning approaches. **Journal of Big Data**, v. 9, jan. 2022.
- ZHANG, A. et al. **Dive into Deep Learning**. [s.l.] Cambridge University Press, 2023.
- ZHANG, H. **The Optimality of Naive Bayes**. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference. **Anais...2004**.
- ZHANG, S.; DUH, K.; VAN DURME, B. **Mt/ie: Cross-lingual open information extraction with neural sequence-to-sequence models**. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. **Anais...2017**.
- ZHANG, T. et al. **BERTScore: Evaluating Text Generation with BERT**. 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. **Anais...OpenReview.net**, 2020. Disponível em: <<https://openreview.net/forum?id=SkeHuCVFDr>>
- ZHAO, J. et al. **Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods**. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). **Anais...New Orleans, Louisiana: Association for Computational Linguistics**, jun. 2018. Disponível em: <<https://aclanthology.org/N18-2003>>
- ZHAO, S.; GRISHMAN, R. **Extracting relations with integrated information using kernel methods**. Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05). **Anais...2005**.
- ZHAO, W. X. et al. A Survey of Large Language Models. **CoRR**, v. abs/2303.18223, 2023.
- ZHOU, C. et al. LIMA: Less Is More for Alignment. **CoRR**, v. abs/2305.11206, 2023.
- ZIEGLER, D. M. et al. Fine-Tuning Language Models from Human Preferences. **CoRR**, v. abs/1909.08593, 2019.
- ZIN, K. K. **Hidden Markov model with rule based approach for part of speech tagging of Myanmar language**. International Conference on Intelligent Cloud Computing. **Anais...2009**. Disponível em: <<https://api.semanticscholar.org/CorpusID:>



63473605>

ZOBEL, J. **How reliable are the results of large-scale information retrieval experiments?** Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. **Anais...**1998.



<https://brasileiraspln.com/livro-pln/1a-edicao/>



Apêndice 1 (Capítulo 2)

*Camila de Araújo Azevedo
Heliana Ribeiro de Mello
Priscila Osório Côrtes*

Alguns pressupostos:¹⁶ estatística¹⁷, probabilidade, teoria da informação

Aleatoriedade e incerteza desempenham um papel importante em muitas disciplinas científicas. A maioria dos problemas de processamento de linguagem falada pode ser caracterizada em um contexto probabilístico. A teoria da probabilidade e estatística fornecem a linguagem matemática para descrever e analisar tais sistemas.

Os critérios e métodos usados para estimar as probabilidades desconhecidas e densidades de probabilidade formam a base para a teoria da estimativa. A teoria da estimativa forma as bases para a aprendizagem de parâmetros no reconhecimento de padrões.

O teste de significância também é importante em estatística, que lida com a confiança da inferência estatística, como saber se a estimativa de algum parâmetro pode ser aceita com confiança. No reconhecimento de padrões, o teste de significância é extremamente importante para determinar se a diferença observada entre dois classificadores diferentes é real.

A teoria da informação foi originalmente desenvolvida para sistemas de comunicação eficientes e confiáveis. Ela evoluiu para uma teoria matemática preocupada com a essência do processo de comunicação. Ela fornece uma estrutura para o estudo de questões fundamentais, como a eficiência da representação da informação e as limitações na transmissão confiável de informação através de um canal de comunicação. Muitos desses problemas são fundamentais para o processamento de fala.

Abordamos brevemente nesta seção algumas dessas questões, a fim de fornecer um panorama de conhecimentos fundamentais para o profissional que trabalha com o processamento da fala.

Probabilidade

A ideia de incerteza e probabilidade remonta a cerca de 3500 a.C., quando jogos de azar com objetos ósseos foram desenvolvidos no Egito. Dados cúbicos com marcações virtualmente idênticas aos dados modernos foram encontrados em túmulos egípcios datados de aproximadamente 2000 a.C. O jogo de dados desempenhou um papel importante no desenvolvimento inicial da teoria da probabilidade. A teoria matemática moderna da probabilidade acredita-se ter sido iniciada pelos matemáticos franceses Blaise Pascal (1623-1662) e Pierre Fermat (1601-1665) quando eles trabalharam em certos problemas de jogo

¹⁶Esta seção é amplamente baseada nas discussões em (Gries, 2019; Jurafsky; Martin, 2023)

¹⁷Recomendamos o ambiente de programação R para a computação estatística. Para acessá-lo, cf. An Introduction to R (r-project.org) (<https://cran.r-project.org/doc/manuals/R-intro.html>)



envolvendo dados. O matemático inglês Thomas Bayes (1702-1761) foi o primeiro a usar a probabilidade indutivamente e estabeleceu uma base matemática para a inferência de probabilidade, levando ao que é agora conhecido como teorema de Bayes. A teoria da probabilidade tem se desenvolvido constantemente desde então e tem sido amplamente aplicada em diversos campos de estudo.

A teoria da probabilidade lida com as médias de fenômenos em massa que ocorrem sequencial ou simultaneamente. Frequentemente usamos expressões probabilísticas em nosso dia a dia, como quando dizemos: “É muito provável que o Dow (índice Dow Jones Industrial) atinja 12.000 pontos no próximo mês”, ou “A chance de chuvas dispersas em Seattle neste fim de semana é alta”. Cada uma dessas expressões se baseia no conceito de probabilidade, ou seja, a probabilidade de que algum evento específico ocorra.

A probabilidade pode ser usada para representar o grau de confiança no resultado de algumas ações (observações) que não são definitivas. Na teoria da probabilidade, o termo espaço amostral, S , é usado para se referir à coleção (conjunto) de todos os possíveis resultados. Um evento refere-se a um subconjunto do espaço amostral ou uma coleção de resultados. A probabilidade do evento A , denotada como $P(A)$, pode ser interpretada como a frequência relativa com que o evento A ocorreria se o processo fosse repetido um grande número de vezes sob condições semelhantes. Com base nessa interpretação, $P(A)$ pode ser calculado simplesmente contando o número total, S_N , de todas as observações e o número de observações A_N cujo resultado pertence ao evento A . Ou seja,

$$P(A) = \frac{N_A}{N_S}$$

$P(A)$ está limitado entre zero e um, ou seja, $0 \leq P(A) \leq 1$ para todos os A .

O limite inferior da probabilidade $P(A)$ é zero quando o conjunto de eventos A é um conjunto vazio. Por outro lado, o limite superior da probabilidade $P(A)$ é um quando o conjunto de eventos A acontece de ser S .

Variáveis aleatórias

Os elementos em um espaço amostral podem ser numerados e referidos pelos números atribuídos. Uma variável X que especifica a quantidade numérica em um espaço amostral é chamada de variável aleatória. Portanto, uma variável aleatória X é uma função que mapeia cada resultado possível s no espaço amostral S para números reais $X(s)$. Como cada evento é um subconjunto do espaço amostral, um evento é representado como um conjunto de $\{s\}$ que satisfaz $\{s \mid X(s) = x\}$. Usamos letras maiúsculas para denotar variáveis aleatórias e letras minúsculas para denotar valores fixos da variável aleatória. Assim, a probabilidade de $X = x$ é denotada como:

$$P(X = x) = P(s \mid X(s) = x)$$

Uma variável aleatória X é uma variável aleatória discreta, ou X tem uma distribuição discreta, se X pode assumir apenas um número finito n de valores diferentes x_1, x_2, \dots, x_n , ou no máximo, uma sequência infinita de valores diferentes x_1, x_2, \dots . Se a variável aleatória X é uma variável aleatória discreta, a função de probabilidade (f.p.) ou função de massa de probabilidade (f.m.p.) de X é definida como a função p tal que, para qualquer número real x :

$$P_X(x) = P(X = x)$$



Média e Variância

A média é uma medida de tendência central que representa o valor médio de um conjunto de dados. É calculada somando todos os valores do conjunto e dividindo pelo número total de valores. A média é útil para determinar um valor representativo do conjunto de dados e fornecer uma estimativa de seu centro.

A variância, por outro lado, é uma medida de dispersão que quantifica a variabilidade ou a dispersão dos valores em relação à média. Ela indica o quão afastados os valores individuais estão da média. A variância é calculada encontrando a diferença entre cada valor e a média, elevando ao quadrado essas diferenças, somando-as e dividindo pelo número total de valores. A variância fornece uma ideia da dispersão dos dados e é utilizada para avaliar a variabilidade de um conjunto de dados.

A Lei dos Grandes Números

A Lei dos Grandes Números é um conceito fundamental na estatística que descreve o comportamento dos resultados médios de uma sequência de experimentos aleatórios. Ela estabelece que, à medida que o número de observações aumenta, a média dessas observações se aproxima do valor esperado teórico ou verdadeiro.

Em termos mais simples, a Lei dos Grandes Números afirma que, quando repetimos um experimento aleatório um grande número de vezes, a média dos resultados observados se aproximará cada vez mais da média esperada ou valor esperado. Isso implica que, quanto mais dados são coletados, mais confiáveis e representativos se tornam os resultados.

Essa lei é de fundamental importância na estatística, pois permite que façamos inferências e tomemos decisões com base em amostras representativas dos dados. Ela estabelece uma relação entre o tamanho da amostra e a precisão das estimativas estatísticas, proporcionando uma base sólida para a análise e interpretação de dados em diversas áreas, inclusive em PLN.

Covariância e Correlação

A covariância e a correlação são medidas estatísticas que descrevem o relacionamento entre duas variáveis.

A covariância mede a relação linear entre duas variáveis aleatórias. Ela indica como as duas variáveis variam juntas. Se a covariância for positiva, isso significa que as variáveis tendem a aumentar ou diminuir juntas. Por outro lado, se a covariância for negativa, indica que as variáveis têm uma relação inversa, ou seja, uma tende a aumentar quando a outra diminui. Uma covariância de zero indica que não há uma relação linear aparente entre as variáveis.

No entanto, a covariância por si só não fornece uma medida padronizada para a força e a direção da relação entre as variáveis. É aí que a correlação entra em cena. A correlação é uma medida padronizada que varia entre -1 e 1, que indica a força e a direção da relação linear entre as variáveis.

Uma correlação de 1 indica uma relação linear positiva perfeita, onde as variáveis aumentam ou diminuem juntas na mesma proporção. Uma correlação de -1 indica uma relação linear negativa perfeita, onde as variáveis têm uma relação inversa perfeita. Uma correlação de 0 indica que não há uma relação linear aparente entre as variáveis.

A correlação é uma medida mais útil que a covariância pois é independente da escala das variáveis e fornece uma medida padronizada para a força da relação linear entre elas.



Ela é amplamente utilizada na análise de dados, no planejamento de experimentos e na modelagem estatística para entender e quantificar o relacionamento entre variáveis.

Vetores aleatórios e distribuições multivariadas

Um vetor aleatório é uma coleção de variáveis aleatórias que são agrupadas como um único objeto. Cada elemento do vetor aleatório representa uma variável aleatória diferente, e o vetor como um todo pode ser usado para descrever simultaneamente o comportamento dessas variáveis aleatórias.

Uma distribuição multivariada é uma distribuição de probabilidade que descreve conjuntamente as probabilidades de ocorrência de múltiplas variáveis aleatórias. Ela fornece informações sobre as relações e dependências entre as variáveis aleatórias em um conjunto de dados.

Ao lidar com vetores aleatórios, é comum usar distribuições multivariadas para modelar a relação conjunta entre as variáveis. Uma distribuição multivariada especifica a forma como as variáveis aleatórias se relacionam entre si e como a probabilidade é distribuída em seu espaço de valores conjunto.

Essas distribuições são amplamente utilizadas em áreas como estatística, ciência de dados e análise de dados para analisar e entender as relações entre múltiplas variáveis aleatórias. Elas permitem realizar análises conjuntas, como calcular a probabilidade conjunta de eventos, estimar parâmetros e fazer previsões baseadas nas relações entre as variáveis.

Algumas distribuições úteis

Existem várias distribuições relevantes para aplicações de probabilidades e estatística em sistemas de língua falada. A escolha da distribuição adequada depende do contexto e dos dados específicos sendo analisados. Aqui estão alguns exemplos:

- **Distribuição Binomial:** É usada para modelar o número de sucessos em um número fixo de tentativas independentes, onde cada tentativa tem apenas dois resultados possíveis (sucesso ou fracasso). Por exemplo, pode ser usada para modelar a probabilidade de acerto em um teste de múltipla escolha, onde cada pergunta tem apenas duas opções de resposta.
- **Distribuição de Poisson:** É usada para modelar a ocorrência de eventos raros em um intervalo de tempo ou espaço. Por exemplo, pode ser usada para modelar a taxa de ocorrência de palavras específicas em um discurso ou texto.
- **Distribuição Gaussiana (Normal):** É uma das distribuições mais comuns e amplamente utilizadas. Ela descreve dados simétricos ao redor de uma média, com a maioria dos valores concentrados perto da média e uma cauda que se estende para ambos os lados. É frequentemente usada para modelar características acústicas de fala, como duração de fonemas ou intensidade de som.
- **Distribuição de Bernoulli:** É uma distribuição específica da distribuição binomial, usada quando há apenas duas possibilidades de resultado (sucesso ou fracasso) em um único evento. Pode ser usada para modelar a probabilidade de uma determinada palavra aparecer em uma frase ou ocorrência de um determinado evento em um diálogo.
- **Distribuição de Dirichlet:** É uma distribuição multivariada usada para modelar a distribuição de probabilidades em um espaço de múltiplas categorias. É



frequentemente usada em processamento de linguagem natural para modelar a distribuição de palavras em um *corpus* de texto.

Teoria da Estimação e Teste de Significância

A Teoria da Estimação é um ramo da estatística que lida com métodos e técnicas para estimar parâmetros desconhecidos com base em dados amostrais. Ela envolve a utilização de informações amostrais para fazer inferências sobre características de uma população maior. A teoria da estimação fornece procedimentos e medidas para determinar o valor estimado de um parâmetro desconhecido, bem como sua precisão e confiabilidade.

No contexto de sistemas de língua falada, a teoria da estimação pode ser aplicada de várias maneiras. Por exemplo, pode ser usada para estimar a taxa de erro de reconhecimento de fala em um sistema de reconhecimento automático de fala. Com base em uma amostra de dados de entrada e saída do sistema, é possível estimar a taxa de erro geral do sistema e a variação dessa estimativa.

Existem vários tipos de estimação utilizados na teoria da estimação. Aqui estão alguns exemplos:

- Estimação de ponto: Nesse tipo de estimação, um único valor é estimado para o parâmetro desconhecido. Por exemplo, se quisermos estimar a média de altura de uma população com base em uma amostra, podemos usar a média amostral como uma estimativa pontual desse parâmetro.
- Estimação por intervalo: Ao contrário da estimação de ponto, a estimação por intervalo fornece um intervalo de valores dentro do qual o parâmetro desconhecido provavelmente está contido. Por exemplo, podemos usar uma estimativa intervalar para estimar a proporção de falantes nativos de uma língua específica em uma determinada região, fornecendo um intervalo de confiança em torno dessa estimativa.
- Estimação por máxima verossimilhança: Nesse método, o objetivo é encontrar o valor do parâmetro que maximiza a probabilidade de obter os dados observados. A estimativa de máxima verossimilhança é frequentemente usada para estimar os parâmetros de distribuições de probabilidade em sistemas de língua falada, como a probabilidade de ocorrência de palavras em um *corpus* de texto.
- Estimação por mínimos quadrados: Esse método é amplamente utilizado na regressão linear, onde a linha de melhor ajuste é encontrada minimizando a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela linha. Por exemplo, na modelagem da fala, a estimação por mínimos quadrados pode ser usada para encontrar os coeficientes que melhor ajustam um modelo de predição acústica aos dados de fala.

O teste de significância, por sua vez, é uma técnica estatística utilizada para avaliar a força da evidência contra uma hipótese nula. Ele permite determinar se os resultados observados são estatisticamente significativos ou se podem ser atribuídos ao acaso. O teste de significância envolve a comparação dos dados observados com uma distribuição de probabilidade teórica esperada sob a hipótese nula.

Em sistemas de língua falada, o teste de significância pode ser usado para determinar se existe uma diferença significativa na taxa de reconhecimento entre dois algoritmos de reconhecimento de fala. Realizando testes estatísticos apropriados, é possível avaliar se a diferença observada na taxa de reconhecimento é estatisticamente significativa ou se pode ser atribuída ao acaso.



Existem diferentes tipos de testes de significância estatística que podem ser aplicados em diversos cenários. Aqui estão alguns exemplos:

- Teste t de Student: É usado para testar se há diferença significativa entre as médias de duas amostras independentes. Por exemplo, pode ser aplicado para determinar se a pontuação média de fluência em uma língua é significativamente diferente entre dois grupos de alunos.
- Teste qui-quadrado: É utilizado para testar a independência entre duas variáveis categóricas. Por exemplo, pode ser usado para verificar se existe uma associação significativa entre o gênero dos falantes e o uso de determinados fonemas em um idioma.
- Teste de ANOVA: É aplicado para testar se há diferenças significativas entre as médias de três ou mais grupos independentes. Por exemplo, pode ser usado para comparar as médias de desempenho em fala entre diferentes grupos de falantes nativos e não nativos.
- Teste de correlação: É utilizado para avaliar se existe uma relação significativa entre duas variáveis contínuas. Por exemplo, pode ser aplicado para verificar se há uma correlação significativa entre a frequência fundamental da voz e a altura percebida da fala.
- Teste de qui-quadrado de ajuste: É usado para verificar se uma distribuição observada se ajusta a uma distribuição teórica específica. Por exemplo, pode ser aplicado para verificar se a distribuição de frequência de palavras em um *corpus* de texto se ajusta à distribuição de Zipf.

Essas técnicas estatísticas, teoria da estimação e teste de significância, são essenciais para analisar dados de sistemas de língua falada, avaliar o desempenho de algoritmos e fazer inferências estatísticas relevantes. Elas permitem tomar decisões informadas com base em evidências estatísticas sólidas.

Teoria da Informação, Entropia e Informação Mútua

A Teoria da Informação é um campo da matemática e da ciência da computação que se preocupa em quantificar e estudar a transmissão de informações. Foi desenvolvida por Claude Shannon na década de 1940 e tem uma ampla gama de aplicações, incluindo o processamento da língua natural falada.

A entropia é um conceito fundamental na Teoria da Informação. Ela mede a quantidade média de informação contida em uma fonte de dados. Quanto maior a entropia, maior a incerteza e, portanto, maior a quantidade de informação necessária para descrever ou transmitir os dados. A entropia é calculada com base na distribuição de probabilidade dos eventos. Em processamento da língua natural falada, a entropia pode ser usada para medir a previsibilidade das palavras em um *corpus* de fala, ou seja, quanto elas são esperadas ou inesperadas com base nas frequências de ocorrência.

A informação mútua é uma medida de dependência entre duas variáveis aleatórias. Ela quantifica a quantidade de informação que uma variável fornece sobre a outra. A informação mútua é calculada com base nas distribuições de probabilidade conjuntas das variáveis. Em termos simples, a informação mútua mede quanto a informação de uma variável reduz a incerteza sobre a outra variável. Em processamento da língua natural falada, a informação mútua pode ser usada para medir a associação entre diferentes palavras em um *corpus* de fala ou a dependência entre características acústicas e fonéticas.



A relação entre esses três conceitos é que a entropia é a medida fundamental da quantidade de informação contida em uma fonte de dados, enquanto a informação mútua mede a dependência ou associação entre duas fontes de informação. A entropia pode ser usada para calcular a informação mútua entre duas variáveis aleatórias, fornecendo uma medida da quantidade de informação compartilhada por essas variáveis.

Em relação às aplicações em processamento da língua natural falada, esses conceitos têm várias utilidades:

- Modelagem de linguagem: A entropia pode ser usada para medir a complexidade ou a incerteza da sequência de palavras em um *corpus* de fala, ajudando a desenvolver modelos de linguagem mais eficientes e precisos.
- Codificação de voz: A Teoria da Informação fornece princípios para a compressão e transmissão eficiente de sinais de voz, reduzindo a taxa de bits necessária para transmitir a informação de forma confiável.
- Detecção de padrões: A informação mútua pode ser aplicada para identificar padrões relevantes em dados de fala, como a associação entre determinados fonemas ou palavras em um contexto específico.
- Reconhecimento de fala: A informação mútua pode ajudar a melhorar a precisão dos sistemas de reconhecimento de fala, fornecendo informações sobre a dependência entre as características acústicas e os fonemas ou palavras correspondentes.



Sobre as/os autoras/es

Helena de Medeiros Caseli

Helena Caseli é formada em Ciência da Computação pela Universidade Federal de Uberlândia (UFU), com pós-graduação em inteligência artificial e PLN no ICMC/USP. É docente na Universidade Federal de São Carlos (UFSCar) desde 2008, onde coordena o LALIC (Laboratório de Linguística e Inteligência Computacional). Suas áreas de interesse incluem aprendizado de máquina, tradução automática, aprendizado multimodal e PLN aplicado a redes sociais em domínios como e-commerce e política. Foi a idealizadora e é uma das organizadoras deste livro, também colaborou com os Capítulos 1, 4 e 18.

Email: helenacaseli@ufscar.br

Maria das Graças Volpe Nunes

Maria das Graças Volpe Nunes é formada em Ciência da Computação pela UFSCar, mestre pela USP e doutora pela PUC-Rio. Foi docente no ICMC-USP, São Carlos, de 1981 a 2013, onde agora é professora sênior. Atualmente é membro do C4AI, da USP, IBM e FAPESP. É uma das fundadoras do NILC, onde atuou em várias áreas do PLN para português: revisão ortográfica e gramatical, tradução automática, sumarização, análise de sentimentos, entre outras. É uma das organizadoras deste livro e colaborou com os Capítulos 1, 24 e 25.

Email: gracan@icmc.usp.br

Adriana Pagano

Adriana Silvina Pagano é formada em Letras pela Universidad Nacional de La Plata. Fez seu mestrado no Programa de Inglês e Literatura Correspondente da UFSC e seu doutorado no Programa de Estudos Linguísticos da UFMG. É professora titular da Faculdade de Letras da UFMG desde 1995. Suas áreas de interesse incluem tradução, produção textual multilíngue e modelagem sistêmico-funcional da linguagem. Neste livro, colaborou com os Capítulos 1, 6, 7 e 21.

Email: apagano@ufmg.br

Aline Aver Vanin

Aline Aver Vanin é licenciada em Letras pela UCS, com mestrado e doutorado em Linguística pela PUCRS. Atuou em estágio pós-doutoral no Laboratório de PLN do Programa de Pós-Graduação em Computação da PUCRS. É docente do Departamento de Educação e Humanidades da UFCSPA desde 2014 e colaboradora do Programa de Pós-Graduação em Letras da UNISC desde 2019. Tem interesse na interface discurso e PLN. Neste livro, colaborou com o Capítulo 12.

Email: alinevanin@ufcspa.edu.br



Aline Macohin

Aline Macohin é formada em Direito e Computação, com Mestrado em Computação Aplicada pela UTFPR e Doutorado em Direito pela UFPR. É advogada, analista de sistemas no poder público há 13 anos, membro da Comissão de Direito Digital e Proteção de dados da OAB/PR e professora de pós-graduação na PUCPR. Autora do site <https://www.iaresponsavel.com.br>. Suas áreas de interesse incluem aprendizado de máquina, PLN, governança de inteligência artificial e proteção de dados. Neste livro, colaborou com o Capítulo 22.

Email: aline.macohin@gmail.com

Aline Paes

Aline Paes é formada em Ciência da Computação pela UERJ, com mestrado e doutorado em inteligência artificial pela COPPE/Sistemas/ UFRJ e estágio sanduíche de doutorado no Imperial College London. É docente na UFF desde 2013. Suas áreas de interesse incluem aprendizado de máquina relacional, aprendizado de representações para linguagem, IA para impacto social positivo e IA explicável. Neste livro, colaborou principalmente com os Capítulos 15 e 20 e está tentando contribuir com o Capítulo 23.

Email: alinepaes@ic.uff.br

Aline Villavicencio

Aline Villavicencio é formada em Ciência da Computação pela PUC-RS e doutora pela Universidade de Cambridge (Inglaterra). É Professora Titular da Universidade de Sheffield (Inglaterra) e a partir de 2024 da Universidade de Exeter (Inglaterra) onde assume a direção do Institute of Data Science and Artificial Intelligence. Sua pesquisa é em PLN inspirado em modelos cognitivo computacionais de processamento de linguagem e com foco em processamento de expressões multipalavras. Neste livro, colaborou com o Capítulo 5.

Email: a.villavicencio@sheffield.ac.uk

Amanda Rassi

Amanda Pontes Rassi é bacharel e mestre em Linguística pela UFG e Doutora na mesma área pela UFSCar. Tem experiência com correção automática de redações, tendo atuado na Redação Nota 1000 e atualmente na Somos Educação. Suas áreas de interesse incluem sintaxe, semântica, lexicologia, lexicografia e PLN. Neste livro, colaborou com os Capítulos 4, 6 e 19.

Email: amanda.rassi@somoseduacao.com.br

Ana Clara Souza Pagano

Ana Clara Souza Pagano é formada em Linguística Teórica e Descritiva pela UFMG. Fez seu TCC em Estilometria computacional. Suas áreas de interesse incluem: aprendizado de máquina, IA, tradução e PLN. Neste livro, colaborou com os Capítulos 6 e 7.

Email: anapagano.ufmg@gmail.com



Arnaldo Candido Junior

Arnaldo Candido Junior possui graduação em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho (2005), mestrado em Ciências da Computação e Matemática Computacional pelo Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) (2008) e doutorado em Ciências da Computação e Matemática Computacional também pelo ICMC/USP. É professor da Universidade Estadual Paulista. Atua nas áreas de linguagens de programação e inteligência artificial (PLN e processamento da fala). Neste livro, colaborou com o Capítulo 3.

Email: arnaldo.candido@unesp.br

Brenda Salenave Santana

Brenda Salenave Santana é formada em Ciência da Computação pela UFSM, com mestrado e doutorado em Computação pela UFRGS. É docente na Unipampa desde 2023. Suas áreas de interesse incluem aprendizado de máquina e detecção de discursos de ódio. Neste livro, colaborou com o Capítulo 23.

Email: brendasantana@unipampa.edu.br

Camila Azevedo

Camila Azevedo foi aluna da primeira turma do Bacharelado em Linguística da UFSCar, atua há mais de 10 anos com síntese de voz e atualmente trabalha no time de processamento de fala do SiDi. Neste livro, colaborou com o Capítulo 2.

Email: ca.araujo.azevedo@gmail.com

Carlos Ramisch

Carlos Ramisch é formado em ciência da computação pela UFRGS e concluiu o doutorado em 2012, em co-tutela entre a UFRGS e a Université de Grenoble (França). Desde 2013, ele é professor na Universidade de Aix-Marseille e pesquisador afiliado ao Laboratoire d'Informatique et Systèmes, na França. Sua pesquisa é focada na área de semântica computacional, em especial no processamento de expressões multipalavras. Neste livro, colaborou com o Capítulo 5.

Email: carlos.ramisch@lis-lab.fr

Cláudia Freitas

Cláudia Freitas é formada em Letras pela PUC-Rio, com pós-graduação em Estudos da Linguagem pela mesma instituição. Foi docente da PUC-Rio de 2002 a 2023, e desde 2023 é pesquisadora do ICMC/USP, vinculada ao C4AI. Suas áreas de interesse incluem construção de *datatsets/corpora* anotados para o aprendizado de máquina, extração de informação (em suas várias formas), e representações linguísticas para o aprendizado de máquina. Neste livro, colaborou com os Capítulos 8, 14 e 20.

Email: claudiafreitas@usp.br



Claudia Moro

Claudia Moro é engenheira de computação pela PUCPR, com mestrado (UNICAMP) e doutorado (USP) em engenharia elétrica/biomédica. Docente da PUCPR desde 1995, vinculada ao Programa de Pós-graduação em Tecnologia em Saúde (desde 2003), atuando nas áreas de PLN em textos clínicos, inteligência artificial em saúde e, interoperabilidade entre sistemas de informações em saúde. Neste livro, colaborou com o Capítulo 21.

Email: claudia.moro@gmail.com

Daniela Barreiro Claro

Daniela Barreiro Claro é professora associada da Universidade Federal da Bahia (UFBA), fez seu Mestrado em Ciências da Computação pela UFSC e o seu Doutorado na Université d'Angers na França. Em 2009, ela fundou o Grupo de Pesquisa FORMAS na área de Semântica e PLN. Suas principais áreas de pesquisa incluem extração de informação, aprendizado de máquina e aprendizado multimodal. Neste livro, colaborou com os Capítulos 10 e 17.

Email: dclaro@ufba.br

Daniela Vianna

Daniela Vianna tem Ph.D. em Ciência da Computação pela Rutgers University, USA, e mestrado e bacharelado em Ciência da Computação pela Universidade Federal Fluminense, Brasil. Em 2022, completou um pós-doutorado na Universidade Federal do Amazonas, Brasil, em uma parceria com a Jusbrasil. Seus interesses estão nas áreas de aprendizado de máquina e PLN. Neste livro, colaborou com o Capítulo 15.

Email: dvianna@gmail.com

Edresson Casanova

Edresson Casanova é bacharel em Ciência da Computação pela UTFPR, com pós-graduação em inteligência artificial e PLN com foco em processamento de fala no ICMC/USP. Atualmente, é Deep Learning engineer na coqui.ai. Suas áreas de interesse incluem síntese de fala, clonagem de voz, verificação de locutores e reconhecimento automático de fala. Neste livro, colaborou com o Capítulo 3.

Email: edresson1@gmail.com

Elisa Terumi Rubel Schneider

Elisa Terumi é formada em Informática pela Universidade Tecnológica Federal do Paraná (UTFPR), com especialização em Desenvolvimento Web (PUCPR), mestrado em Bioinformática (UFPR) e doutorado em Informática atuando na linha de pesquisa de Inteligência Artificial (PUCPR). Tem interesse em PLN aplicado na área de saúde. Neste livro, colaborou com os Capítulos 7 e 21.

Email: lisa.terumi@gmail.com

Eloize Seno

Eloize Seno é Bacharel em Análise e Desenvolvimento de Sistemas pela UNILINS, com mestrado em inteligência artificial e PLN pelo Programa de Pós-graduação em Ciência da



Computação da UFSCar e doutorado em inteligência artificial e PLN pelo ICMC/USP. É docente no Instituto Federal de São Paulo desde 2011. Suas áreas de interesse incluem sumarização automática, tradução automática, análise semântica e mineração de opinião. Neste livro, colaborou com os Capítulos 9 e 10.

Email: eloizeseno@gmail.com

Evandro Fonseca

Evandro Fonseca possui graduação em Ciência da Computação (2011) pela Universidade Católica de Pelotas (UCPEL), mestrado (2014) e doutorado (2018) em Ciência da Computação, com foco em PLN, pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Atualmente atua como Team Lead | AI/NLP Specialist na Blip. Neste livro, colaborou com o Capítulo 12.

Email: metalmorphy@gmail.com

Flaviane R. Fernandes Svartman

Flaviane Svartman possui graduação em Linguística pela Unicamp, doutorado em Linguística pela mesma instituição, com período de estágio de doutorado na Faculdade de Letras da Universidade de Lisboa e pós-doutorado também pela Unicamp. Atua como docente na graduação, na área de Filologia e Língua Portuguesa na Faculdade de Filosofia, Letras e Ciências Humanas da USP desde 2010. Suas investigações concernem ao estudo da fonologia e da fonética da língua portuguesa, com especial interesse na prosódia, na interface sintaxe-fonologia, na comparação entre variedades africanas, brasileiras e europeias do português e no PLN. Neste livro, colaborou com o Capítulo 3.

Email: flaviane@gmail.com

Heliana Mello

Heliana Mello é formada em Letras pela UFMG, com doutorado em Linguística pela CUNY. É docente na UFMG desde 1998. Suas áreas de interesse incluem métodos empíricos e quantitativos para a análise da linguagem, compilação e estudo de *corpora* de fala. Neste livro colaborou com o Capítulo 2.

Email: hmello@ufmg.br

Jackson Wilke da Cruz Souza

Jackson Souza é bacharel (2012), mestre (2015) e doutor (2019) em Linguística pela Universidade Federal de São Carlos. Atualmente, é professor da Universidade Federal da Bahia (UFBA), atuando junto ao Departamento de Ciência, Tecnologia e Inovação e ao Programa de Pós-Graduação em Língua e Cultura da UFBA. Tem interesse nas áreas de análise de discurso, sumarização automática e análise e descrição linguística. Neste livro, colaborou com o Capítulo 11.

Email: jackcruzsouza@gmail.com

Jessica Rodrigues

Jessica Rodrigues tem graduação e mestrado em Ciência da Computação, sendo o mestrado focado em *machine learning* e *natural language processing*, pela Universidade Federal de São Carlos. A experiência acadêmica continua com o PhD em *social data science* pela



University of Oxford, com foco em *AI safety, fairness and biases in language models*. Jessica também é Lead Data Scientist e é especialista em AI aplicada. Neste livro, colaborou com os Capítulos 10 e 15.

Email: jessica.rodriques@oii.ox.ac.uk

Joaquim Santos

Joaquim Santos fez licenciatura em Matemática na URCA, é mestre em Ciência da Computação pela PUCRS e atualmente é estudante de doutorado no programa de Computação Aplicada da UNISINOS. Tem trabalhado intensamente com PLN em problemas de extração de informação, modelos de linguagem e bases de conhecimento. Neste livro, colaborou com o Capítulo 17.

Email: netojoaquim@edu.unisinos.br

Laila Mota

Laila Mota é formada em Ciência e Tecnologia pela UFBA, com pós-graduação em Tecnologia da Informação pela UFBA. Suas áreas de interesse incluem aprendizado de máquina, linguística computacional e modelos de linguagem. Neste livro, colaborou com o Capítulo 10.

Email: lailapmota@gmail.com

Larissa Freitas

Larissa Freitas é doutora em Ciência da Computação pela PUCRS (2015). Atualmente é professora da Universidade Federal de Pelotas (UFPel). Suas áreas de interesse incluem análise de sentimento, ontologias, *chatbots*, detecção de sarcasmo/ironia, detecção de discurso de ódio e detecção de notícias falsas. Neste livro, colaborou com o Capítulo 23.

Email: larissa@inf.ufpel.edu.br

Leidiana Iza Andrade Freitas

Leidiana Iza Andrade Freitas é formada em Letras Português/Francês pela UFC, tecnóloga em Análise e Desenvolvimento de Sistemas pela Unifametro. Suas áreas de interesse além de PLN são engenharia de software, inteligência artificial, tradução automática. Neste livro, colaborou com o Capítulo 13.

Email: izafreit@gmail.com

Lilian Mie Mukai Cintho

Lilian Mie Mukai Cintho é formada em Enfermagem e Obstetrícia pela UFSCAR, com pós-graduação em Tecnologia em Saúde na PUCPR. É docente colaboradora na UEPG. Suas áreas de interesse incluem, gerenciamento de programas de promoção à saúde, enfermagem médico cirúrgica e informática em saúde. Neste livro, colaborou com o Capítulo 21.

Email: miemukai@hotmail.com

Lucelene Lopes

Lucelene Lopes é doutora em Ciência da Computação pela PUCRS (2012). Consultora Empresarial na área de PLN desde 2016, e desde 2021 é pesquisadora no ICMC/USP,



vinculada ao C4AI. Suas áreas de interesses incluem PLN, aprendizado de máquina, *parsing* e modelos de linguagem. Neste livro, colaborou com o Capítulo 4.

Email: lucelene@gmail.com

Maria José Bocorny Finatto

Maria José Bocorny Finatto é linguista especializada em Estudos do Léxico e Terminologia. Investiga a história das terminologias médicas e modos de aplicar técnicas da Linguagem Simples para maior acessibilidade das informações escritas para diferentes perfis de pessoas no Brasil. Análise textual com recursos PLN é seu foco. Orientadora de mestrado e doutorado junto à UFRGS. Coordena o grupo de pesquisa do DgP/CNPq “Acessibilidade Textual e Terminológica”. Neste livro, colaborou com os Capítulos 4 e 22.

Email: maria.finatto@gmail.com

Mariza Ferro

Mariza Ferro é formada em Ciência da Computação pela UNIOESTE, com mestrado em Inteligência Artificial no ICMC/USP e doutorado em Modelagem Computacional pelo LNCC. É docente do Instituto de Computação da Universidade Federal Fluminense (UFF) e coordenadora do Núcleo de Referência em IA Ética e Confiável desde 2020. Suas áreas de interesse incluem IA ética, IA verde e sustentável, IA para o estar social e aprendizado de máquina para predição de eventos extremos. Neste livro, colaborou com o Capítulo 24.

Email: mariza@ic.uff.br

Marli Quadros Leite

Marli Quadros Leite é Professora Titular do Departamento de Letras Clássicas e Vernáculas (FFLCH | USP) e atua como pesquisadora nas áreas de análise do Discurso Oral e na de História das Ideias Linguísticas. Atualmente é Pró-Reitora de Cultura e Extensão Universitária da USP. Neste livro, colaborou com o Capítulo 3.

Email: mqleite@usp.br

Marlo Souza

Marlo Souza é formado em Ciência da Computação pela Unifacs e em Matemática pela UFBA, com pós-graduação em PLN e Lógica pela PUC-RS e UFRGS, respectivamente. Atualmente, é professor adjunto na UFBA e co-líder do grupo de pesquisa FORMAS. Tem interesse em pesquisas nas áreas de semântica computacional, extração de informação e modelos de representação de conhecimento em PLN. Neste livro, colaborou com o Capítulo 17.

Email: msouza1@ufba.br

Paula Christina Figueira Cardoso

Paula Figueira Cardoso tem graduação em Informática pelo ILES de Santarém, mestrado em Engenharia Elétrica pela UFPA e doutorado pelo Instituto de Ciências Matemáticas e de Computação - USP/São Carlos. É docente na Universidade Federal de Lavras-MG desde 2015. Suas áreas de interesse incluem geração de textos, sumarização automática, tecnologias educacionais e pesquisas aplicadas na área de interface homem-máquina. Neste



livro, colaborou com o Capítulo 11.

Email: paulastm@gmail.com

Priscilla de Abreu Lopes

Priscilla de Abreu Lopes é formada em Ciência da Computação pela UFSCar, com pós-graduação em inteligência artificial no DC/UFSCar. Suas áreas de interesse incluem aprendizado de máquina e PLN. Neste livro, colaborou com o Capítulo 19.

Email: alopes.priscilla@gmail.com

Priscila Osório Côrtes

Priscila Osório Côrtes é formada em Letras Alemão/Português pela UFMG e mestre em linguística pela Albert Ludwigs-Universität Freiburg, Alemanha. Trabalha há 7 anos na área de PLN, e há 5 com processamento de fala na empresa SiDi Campinas. Neste livro, colaborou com o Capítulo 2.

Email: pritico@gmail.com

Renata Ramisch

Renata Ramisch é formada em Letras pela UFRGS e mestre em Linguística pela UFSCar. É linguista na Redação Nota 1000, que faz parte do grupo Somos Educação. Suas áreas de interesse incluem correção automática de redações e tratamento computacional de expressões multipalavras. Neste livro, colaborou com o Capítulo 5 (a ser lançado na segunda edição).

Email: renata.ramisch@gmail.com

Renata Vieira

Renata Vieira é PhD em Informática e Ciências Cognitivas pela Universidade de Edimburgo. É pesquisadora da área de PLN atuando em diversas áreas de aplicação. É investigadora principal convidada do Centro de História, Culturas e Sociedades da Universidade de Évora e conselheira científica do Instituto de Inteligência Artificial na Saúde. Neste livro, colaborou com o Capítulo 12.

Email: renata.vieira@gmail.com

Ricardo Marcacini

Ricardo Marcacini é formado em Informática pela Universidade de São Paulo (USP), com pós-graduação em ciência de computação e matemática computacional no ICMC/USP. Atualmente é docente do Departamento de Ciências de Computação (SCC) do ICMC/USP. Realiza pesquisa na área de aprendizado de máquina, com foco em mineração de textos, análise de eventos e análise de sentimentos. Neste livro, colaborou com o Capítulo 3.

Email: ricardo.marcacini@icmc.usp.br

Roana Rodrigues

Roana Rodrigues é formada em Letras, mestra e doutora em Linguística pela Universidade Federal de São Carlos (UFSCar). É docente do Departamento de Letras Estrangeiras da Universidade Federal de Sergipe (UFS). Suas áreas de interesse incluem sintaxe, semântica,



estudos do léxico e PLN. Neste livro, colaborou com o Capítulo 11.

Email: roana@academico.ufs.br

Sandra Maria Aluísio

Sandra Maria Aluísio é formada em Ciência da Computação pela UFSCar, com Doutorado em IA e PLN, pela USP. Foi docente de 1988 a 2018, no ICMC/USP, onde hoje atua no Programa Professor Sênior. As áreas de interesse incluem linguística de *corpus*, simplificação textual, recursos semânticos, detecção automática da estrutura do discurso científico e ferramentas de suporte à escrita, análise automática de distúrbios de linguagem, e processamento de fala. Neste livro, colaborou com o Capítulo 3.

Email: sandra@icmc.usp.br

Sheila Castilho

Sheila Castilho é licenciada em Letras pela UNIOESTE. Possui mestrado em PLN pela Universidade do Algarve (PT) e pela University of Wolverhampton (UK). Completou seu doutorado pela Dublin City University em 2016. Atualmente, ela é Assistant Professor na Dublin City University. Seus interesses incluem tradução automática, avaliação automática e humana, tecnologias da tradução, e PLN para tradução. Neste livro, colaborou com o Capítulo 18.

Email: sheila.castilho@dcu.ie

Solange Rezende

Solange Rezende é docente e pesquisadora no Laboratório de Inteligência Computacional (LABIC) do Departamento de Ciências de Computação do ICMC/USP desde 1991. Tem experiência na área de IA, atuando principalmente nos temas relacionados com mineração de dados e textos, análise de sentimentos e sistemas de recomendação. Possui graduação em licenciatura em Ciências Habilitação em Matemática pela UFU (1986), mestrado em Ciências de Computação e Matemática Computacional pela USP (1990) e doutorado em Engenharia Mecânica – São Carlos pela USP (1993). Realizou pós-doutorado na Universidade de Minnesota, USA (1995-1996). Neste livro, colaborou com o Capítulo 3.

Email: solange@icmc.usp.br

Tayane Soares

Tayane Soares é Cientista de Dados na empresa Blip, formada em Letras pela UFMG e atualmente estudante de mestrado em Linguística Aplicada pelo PosLin-UFMG. Seu trabalho de pesquisa tem como foco a avaliação de viés de gênero em modelos de tradução automática, especificamente no par linguístico inglês-português. Seu interesse de pesquisa abrange a interseção entre linguagem e tecnologia, explorando a troca de conhecimentos entre computação e linguística para viabilizar a criação de aplicações de PLN, especialmente no que diz respeito a aprendizado de máquina, tradução automática e questões éticas em IA e PLN. Neste livro, colaborou com o Capítulo 24.

Email: arantesoares@gmail.com



Valéria de Paiva

Valeria de Paiva é formada em Matemática pela PUC-Rio, com pós-graduação em Álgebra na PUC e e mestrado/doutorado na Universidade de Cambridge, UK. É pesquisadora principal (Principal Researcher) no Topos Institute desde 2021. Suas áreas de interesse incluem semântica de linguagem natural, inferência em linguagem natural, recursos léxico-semânticos e lógica, especialmente lógica categórica. Gosta de sistemas híbridos de PLN, open software, dependências universais e da OpenWordNet-PT. Fundadora de “Women in Logic” e “Lógicas Brasileiras”. Neste livro, colaborou com o Capítulo 9.

Email: valeria@topos.institute

Vinícius G. Santos

Vinícius G. Santos é graduado, mestre e doutor em Letras pela Universidade de São Paulo. Atualmente realiza pesquisa de pós-doutorado vinculada ao C4AI-USP. Atua na área de descrição e análise linguística de variedades africanas e brasileiras da língua portuguesa, com especial interesse em fonologia, prosódia, entoação e elaboração de recursos para PLN. Neste livro, colaborou com o Capítulo 3.

Email: vinicius.santos@alumni.usp.br

Viviane P. Moreira

Viviane P. Moreira é doutora em computação na Middlesex University (2004), mestre em ciência da computação na UFRGS (1999) e bacharel em análise de sistemas na UCPel (1996). É professora na UFRGS e orientadora do Programa de Pós-Graduação em Computação. Sua pesquisa concentra-se em recuperação de informação, PLN e mineração de dados. Neste livro, colaborou com o Capítulo 16.

Email: viviane@inf.ufrgs.br

Vlória Pinheiro

Vlória Pinheiro é formada em Ciência da Computação pela UFC, com doutorado em Ciências da Computação na mesma universidade. É docente do Programa de Pós-Graduação em Informática Aplicada, na UNIFOR, desde 2011. Suas áreas de interesse incluem recursos semânticos, *parsers* semânticos, extração de informação, sumarização de textos, *chatbots*, aplicações de inteligência artificial e ciência de dados. Neste livro, colaborou com o Capítulos 9 e 13.

Email: vladiacelia@gmail.com

Yohan Bonescki Gumiel

Yohan Bonescki Gumiel é formado em Engenharia de Controle e Automação, com uma carreira dedicada à tecnologia e inovação. Ele fez doutorado na área de PLN voltada a textos clínicos, um campo de estudo que demonstra sua paixão por unir tecnologia e saúde. Atualmente, Yohan ocupa o cargo de cientista de dados na A3data e é pós-doutorando na UFMG, onde continua a aprofundar sua pesquisa e conhecimento em áreas críticas. Suas áreas de interesse incluem aprendizado de máquina, PLN e aplicações voltadas à saúde. Neste livro, colaborou com o Capítulo 21.

Email: yohan.gumiel@gmail.com

