



PROCESSAMENTO DE LINGUAGEM NATURAL

Conceitos, Técnicas
e Aplicações em
Português

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

ISBN: 978-65-02-00178-3

4ª Edição | 2026

Volume 2: Modelos de Linguagem

ISBN do Volume 2: 978-65-02-00158-5

*Este livro é o produto de um sonho coletivo do grupo
Brasileiras em Processamento de Linguagem Natural (BPLN)*

© 2026 by Helena de Medeiros Caseli, Maria das Graças Volpe Nunes, organizadoras.
Editoração eletrônica: Helena de Medeiros Caseli

Licença CC BY-NC-ND 3.0 BR

Ficha Catalográfica: Marina P. Freitas CRB-08/ 6069

Dados internacionais de Catalogação-na-Publicação (CIP)

Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português / organização de Helena de Medeiros Caseli e Maria das Graças Volpe Nunes. — 4. ed. - São Carlos : BPLN, 2026. Internet: brasileiraspln.com/livro-pln-4ed/volume2.html 225p.

v. 2: Modelos de Linguagem.

ISBN: 978-65-02-00158-5

1. Processamento de Linguagem Natural 2. Inteligência Artificial. 3. I. Título.

Processamento de Linguagem Natural

Conceitos, Técnicas e Aplicações em Português

Organizado por:

Helena de Medeiros Caseli

Maria das Graças Volpe Nunes

4^a Edição

Volume 2: Modelos de Linguagem

Abril 2026



<https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2/>

Conteúdo

Volume 2: Modelos de Linguagem	1
Sobre o Volume 2	1
Como citar	1
 I Modelos de Linguagem	 3
1 Modelos de linguagem	4
1.1 Relembrando a Hipótese Semântica e Definindo Modelos	4
1.2 Modelos de Linguagem Probabilísticos	6
1.2.1 Estimando as probabilidades a partir de <i>corpora</i>	7
1.2.2 Usando o modelo probabilístico	8
1.3 Modelos de Linguagem Neurais	9
1.3.1 Um Contexto Histórico	9
1.3.2 Modelos de Linguagem Neurais Modernos	10
1.3.3 Treinamento e Ajustes em Modelos de Linguagem Neurais	35
1.3.4 Avaliação de Modelos de Linguagem Neurais	41
1.4 Tendências	44
1.4.1 A Era dos <i>Large Language Models</i> (LLMs)	44
1.4.2 Treinamento Eficiente de Modelos de Linguagem Neurais	48
1.4.3 Estratégias de Treinamento para Agentes de Conversação: alinhamento e feedback humano	50
 2 O papel dos dados no pré-treinamento de Grandes Modelos de Linguagem	 53
2.1 Introdução	53
2.2 Por que e quando pré-treinar?	55
2.3 Conjuntos de dados para pré-treinamento em português	57
2.3.1 Recursos gerais	57
2.3.2 Recursos para o português	58
2.4 A importância da curadoria de dados	60
2.5 O impacto da qualidade dos dados	61
2.5.1 O custo da qualidade	62
2.5.2 Considerações finais desta seção	63
2.6 O impacto da diversidade dos dados	64
2.7 A questão da proveniência dos dados	66
2.8 O <i>corpus</i> Carolina como um estudo de caso	67
2.9 Considerações finais	68
 3 Treinamento de Grandes Modelos de Linguagem na prática	 70



3.1	Introdução	70
3.2	Continuação de pré-treino	71
3.2.1	Mecanismos de treinamento autossupervisionado	71
3.2.2	Contextos e objetivos da continuação de pré-treino	72
3.3	Ajuste fino de modelos	73
3.3.1	Ajuste fino completo	74
3.3.2	PEFT	76
3.3.3	<i>Few-shot learning</i>	79
3.4	<i>AI alignment</i>	81
3.4.1	Fundamentos de aprendizado por reforço	82
3.4.2	<i>Reinforcement Learning from Human Feedback</i> (RLHF)	83
3.4.3	<i>Direct Preference Optimization</i> (DPO)	84
3.4.4	Discussão comparativa: RLHF vs. DPO	85
3.5	Exemplo prático	86
3.5.1	Preparando o ambiente	86
3.5.2	O ponto de partida	87
3.5.3	Ajuste fino supervisionado com LoRA	90
3.5.4	Refinamento do comportamento com DPO	93
3.5.5	Considerações finais	95
4	Avaliação de Grandes Modelos de Linguagem	96
4.1	A importância e os desafios intrínsecos da avaliação de LLMs	96
4.1.1	Tipos de avaliação	97
4.2	Avaliações tradicionais	98
4.2.1	Descompasso com a percepção humana	98
4.2.2	Dificuldade em avaliar aspectos complexos	99
4.2.3	Exemplos práticos	100
4.2.4	Avaliação humana: O padrão ouro?	100
4.2.5	Importância e metodologias comuns	100
4.2.6	Desafios da avaliação humana	101
4.3	<i>Benchmarks</i> e <i>datasets</i> de avaliação	102
4.3.1	<i>Benchmarks</i> genéricos (General-Purpose)	102
4.3.2	<i>Benchmarks</i> para tarefas específicas	102
4.3.3	<i>Benchmarks</i> para segurança e confiabilidade	104
4.3.4	A importância de ir além do placar	104
4.4	Avaliação baseada em modelos (<i>Model-based evaluation</i>)	104
4.4.1	Vantagens	105
4.4.2	Abordagens	105
4.4.3	Desafios e Limitações	106
4.4.4	Exemplos de <i>frameworks</i> de <i>LLM as a Judge</i>	107
4.5	Aspectos emergentes e avançados da avaliação	108
4.5.1	Robustez e Ataques adversariais	108
4.5.2	Avaliação de justiça, viés e toxicidade	108
4.5.3	Factualidade, alucinações e citabilidade	109
4.5.4	Eficiência computacional e custos	109
4.6	Considerações éticas e o Futuro da avaliação de LLMs	110
4.6.1	Responsabilidade na avaliação	110
4.6.2	Transparência e Interpretabilidade	110

4.6.3	Dualidade do uso	110
4.6.4	O papel da regulação e de padrões na avaliação	110
4.6.5	Direções futuras	110
4.7	Considerações finais	111
5	Geração Aumentada por Recuperação (RAG)	113
5.1	Fundamentos da Geração Aumentada por Recuperação	113
5.1.1	Introdução à RAG	114
5.1.2	Arquitetura RAG	115
5.1.3	O componente Recuperador (<i>Retriever</i>)	116
5.1.4	O componente Gerador (<i>Generator</i>)	119
5.2	RAG vs. <i>Fine-tuning</i>	120
5.2.1	<i>Fine-tuning</i>	120
5.2.2	RAG	120
5.2.3	Abordagens híbridas	121
5.3	RAG - Aspectos avançados	121
5.3.1	RAG em duas etapas: Re-ordenando resultados	121
5.3.2	RAG adaptiva e o Modelo agêntico	124
5.3.3	Indo além do texto	126
5.3.4	Avaliando RAG	128
5.3.5	Otimização de desempenho: Latência e Escalabilidade	129
5.3.6	Ética em RAG: Viés, privacidade e responsabilidade	130
5.4	RAG no mundo real: Estudos de caso por indústria	131
5.4.1	Setor financeiro: Conformidade regulatória e Análise de risco	131
5.4.2	Saúde e Ciências da vida: Suporte à decisão clínica	132
5.4.3	Setor jurídico: Revolucionando a pesquisa e a <i>Due Diligence</i>	132
5.4.4	Contexto científico: Rastreabilidade da informação	132
5.5	O futuro da RAG	133
5.5.1	O debate: RAG vs. Janelas de contexto amplas (LCW)	133
5.5.2	RAG como um mecanismo de atenção inteligente	134
	Agradecimentos	135
6	Responsabilidade no desenvolvimento e uso de tecnologias de linguagem baseadas em IA	136
6.1	Introdução	136
6.2	Quais facetas da responsabilidade devem ser consideradas?	137
6.2.1	Dilemas nos dados	138
6.2.2	Impactos e danos	139
6.2.3	Riscos e maus usos	142
6.2.4	Inconveniências operacionais	145
6.2.5	Hegemonia de interesses comerciais e econômicos	148
6.3	Como atuar de forma responsável?	152
6.3.1	Para saber mais	153
6.3.2	Formas de resistência	154
6.4	Considerações finais	154
	Agradecimentos	155
7	ChatGPT, MariTalk e outros agentes de conversação	156



7.1	Introdução	156
7.2	Jogos de Linguagem	159
7.3	Jogos que os agentes parecem jogar bem	161
7.3.1	Sumarização	161
7.3.2	Criação de história	162
7.3.3	Tradução automática	168
7.3.4	Escrever e-mails	171
7.4	Jogos que os agentes melhoraram	172
7.4.1	Simplificação textual	172
7.5	Jogos que os agentes parecem jogar mal	176
7.5.1	Perguntas sobre conteúdo pouco popular na internet	176
7.5.2	Conte uma piada	180
7.5.3	Inferências e raciocínio	181
7.6	Quando o jogo é perigoso	181
7.7	Conclusões provisórias	185

Referências 186

Sobre as/os autoras/es 215

Aline Paes	215
Altigran Soares Silva	215
André Carvalho	215
Brielen Madureira	215
Cláudia Freitas	215
Cristiane Namiuti	216
Daniela Vianna	216
Danilo Samuel Jodas	216
Douglas Rodrigues	216
Gabriel Lino Garcia	216
Gabriela Alves Lachi	216
Felipe Ribas Serras	217
Jessica Rodrigues	217
João Paulo Papa	217
João Renato Ribeiro Manesco	217
João Vitor Mariano Correia	217
Livy Real	217
Lucas Lasota	218
Marcelo Finger	218
Mariana Lourenço Sturzeneker	218
Mayara Feliciano Palma	218
Miguel de Mello Carpi	218
Pedro Henrique Paiola	218
Vanessa Martins do Monte	218



Volume 2: Modelos de Linguagem

Sobre o Volume 2

Dada a relevância atual do emprego de técnicas de aprendizado de máquina com redes neurais profundas, Transformers e Grandes Modelos de Linguagem (LLM) na área de Inteligência Artificial e, em particular, em PLN, reservamos este volume aos capítulos que tratam desses temas.

No Capítulo 1 são discutidas as redes neurais profundas e os Transformers, as técnicas de *fine-tuning* e *prompt engineering*. Essa leitura 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.

Os Capítulos 2 e 3 discorrem sobre o papel dos dados no pré-treinamento de modelos e sobre as principais técnicas de treinamento de grandes modelos de linguagem, respectivamente.

O importante tema de avaliação dos LLMs é tratado no Capítulo 4, que introduz seus fundamentos, métricas, *benchmarks* e considerações de ordem ética.

O desafio para mitigar o efeito de alucinação dos LLMs é discutido no Capítulo 5, onde é apresentada a técnica de geração aumentada de recuperação (*Retrieval-Augmented Generation*, RAG), que suporta acesso a fontes de dados externas para gerar respostas factuais e relevantes, reduzindo alucinações dos modelos.

O Capítulo 6 apresenta um amplo panorama das discussões envolvendo aspectos importantes do desenvolvimento responsável de tecnologias de linguagem baseadas em inteligência artificial.

Finalmente, no Capítulo 7, o leitor terá oportunidade de conhecer o caminho percorrido pelo PLN até chegar aos agentes atuais, como ChatGPT e MariTalk, que são analisados em profundidade.

Como citar

Caseli, H.M.; Nunes, M.G.V. (org.) **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. 4 ed. BPLN, 2026. v. 2. ISBN 978-65-02-00158-5. Disponível em: <https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2>.

Note 1: Bibref



```
@Book{BPLN_livro_4ed_vol2:2026,  
  title      = {Processamento de Linguagem Natural: Conceitos,  
                Técnicas e Aplicações em Português},  
  editor      = {Caseli, H. M. and Nunes, M. G. V.},  
  year        = {2026},  
  publisher    = {BPLN},  
  edition     = {4},  
  volume      = {2},  
  isbn        = {978-65-02-00158-5},  
  url         = {https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2}  
}
```



Parte I

Modelos de Linguagem



Capítulo 1

Modelos de linguagem

Aline Paes
Daniela Vianna
Jessica Rodrigues

Publicado em: 26/09/2023

 <https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2/>

1.1 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 et al., 2020; Gong et al., 2022; Li et al., 2023c; Piękos et al., 2021), código (Li et al., 2023b; Wang et al., 2021)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 **Semântica com técnicas simbólicas**, 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 1.1:

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, 1957), 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 completção. 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 computacionais³.

1.2 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:

$$P(p_{1:i}) = P(p_1)P(p_2|p_1)P(p_3|p_{1:2})P(p_4|p_{1:3}) \dots P(p_i|p_{1:i-1}) \quad (1.1)$$

Apenas uma observação: multiplicações de valores menores que um podem fazer com que o resultado seja zero, considerando a limitação dos computadores em manipular números em ponto flutuante. Chamamos esse problema de *underflow*. Para aliviá-lo, podemos usar log e somar os termos, ao invés de multiplicar:

$$\begin{aligned} \log P(p_{1:i}) = \\ \log P(p_1) + \log P(p_2|p_1) + \log P(p_3|p_{1:2}) + \\ + \log P(p_4|p_{1:3}) + \dots + \log P(p_i|p_{1:i-1}) \end{aligned} \quad (1.2)$$

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 completação discutida anteriormente, assumimos que a tarefa de completar uma sequência de palavras com uma próxima palavra é definida por uma distribuição de probabilidade condicional⁴ das palavras que poderiam completar a sequência, dadas as palavras que vieram antes na sequência, ou seja:

$$P(p_i|p_1, \dots, p_{i-1}) \quad (1.3)$$

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

$$P(p_i|\text{Vamos, completar, o, texto, com, a, próxima})$$

³Neste vídeo https://www.youtube.com/watch?v=Xz_qPyizGEY&t=1s o leitor pode se informar mais sobre o uso de aprendizado de máquina em PLN.

⁴Probabilidade 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.



onde

$$P(\text{palavra}|\text{Vamos, completar, o, texto, com, a, próxima})$$

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 trigramma 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.

1.2.1 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

$$P(p_i|p_{i-1}, p_{i-2}) = \frac{c(p_{i-2}, p_{i-1}, p_i)}{c(p_{i-2}, p_{i-1})}$$

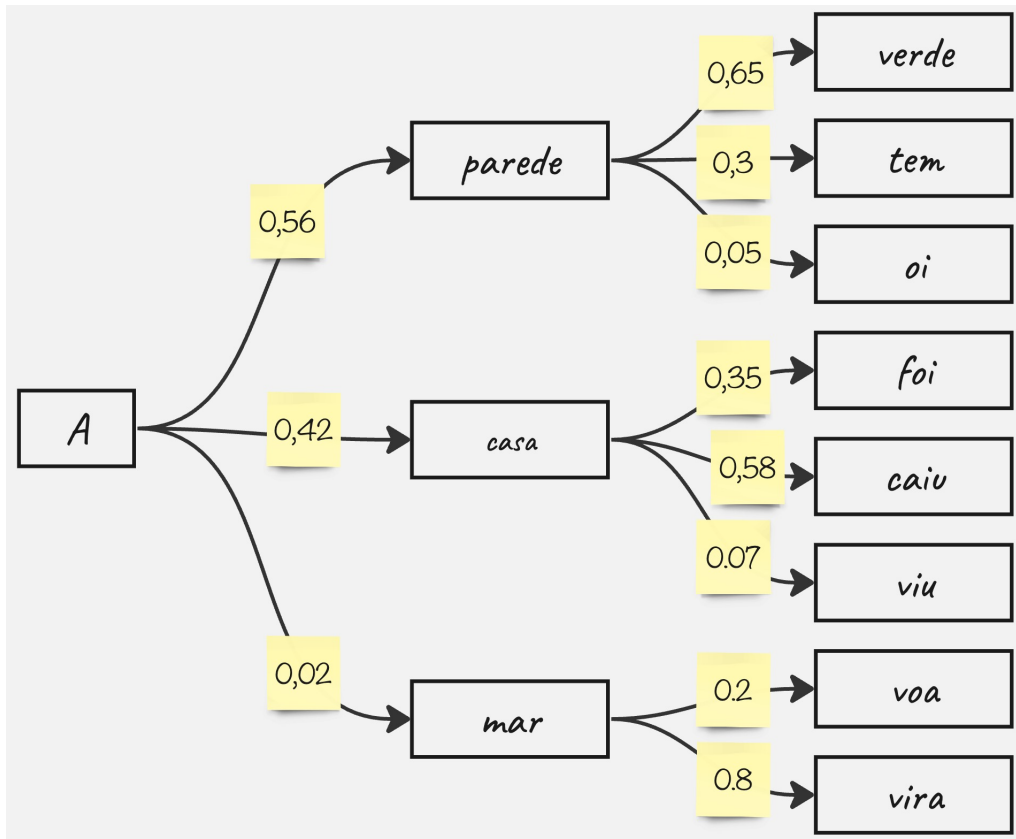
Ou seja, $c(\dots)$ representa quantas vezes um dado n-grama ocorreu no texto.

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.

1.2.2 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 1.1, supondo que o *token* “A” já foi emitido, a busca gulosa escolherá a palavra “casa” e depois “caiu”.

Figura 1.1: 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 1.1 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 et al., 2018), ou amostragem baseada em um limiar p , que escolhe o menor conjunto de palavras cuja probabilidade acumulada exceda p (Holtzman et al., 2020).

1.3 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 modelo⁵ – é considerar que a probabilidade associada a um modelo de linguagem é uma função e “aprender” tal função. Redes Neurais (Goodfellow et al., 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 et al., 1989). Caso você queira entender melhor como funciona uma rede neural, os capítulos 5 e 6 de (Goodfellow et al., 2016) são uma boa introdução (dentre muitas outras referências).

1.3.1 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*

⁵Lembre que, em um modelo de linguagem probabilístico, treinar o modelo é encontrar as probabilidades por contagem.



neural network learn language models? (Xu; Rudnicky, 2000) e *A neural probabilistic language model* (Bengio et al., 2003).

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 [Representação vetorial e semântica distribucional](#). 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**⁶ 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 et al., 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).

1.3.2 Modelos de Linguagem Neurais Modernos

Considerando as limitações que discutimos no Capítulo [Representação vetorial e semântica distribucional](#) 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**

⁶Uma representação vetorial densa de baixa dimensionalidade, aprendida a partir de textos. Veja mais no Capítulo [Representação vetorial e semântica distribucional](#).



contextualizados. Isso quer dizer que as unidades de representação (*tokens*) podem ter *embeddings* distintos, definidos no momento em que eles são aplicados. Considere, por exemplo, as sentenças do Exemplo 1.2:

Exemplo 1.2:

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 1.3:

Exemplo 1.3:

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 $\text{emb}_{\text{banco}} = f(e_{\text{sentei}}, e_{\text{no}}, e_{\text{banco}}, e_{\text{da}}, e_{\text{praça}})$, onde f é a função de geração do *embedding* e $\text{emb}_{\text{palavra}}$ 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 [Representação vetorial e semântica distribucional](#).

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 ELMo⁷ (*Embeddings from Language Models*) (Peters et al., 2018)) – e os **Transformers** (Vaswani et al., 2017a) – incluindo BERT⁸ (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2019) e GPT⁹ (*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.

1.3.2.1 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 1.2.

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*¹⁰ do tamanho do vocabulário, ou seja, $|1 \times V|$ representando uma palavra, e t representa a t -ésima palavra da sequência sendo gerada

⁷Inspirado pelo nome de um dos personagens de Vila Sésamo.

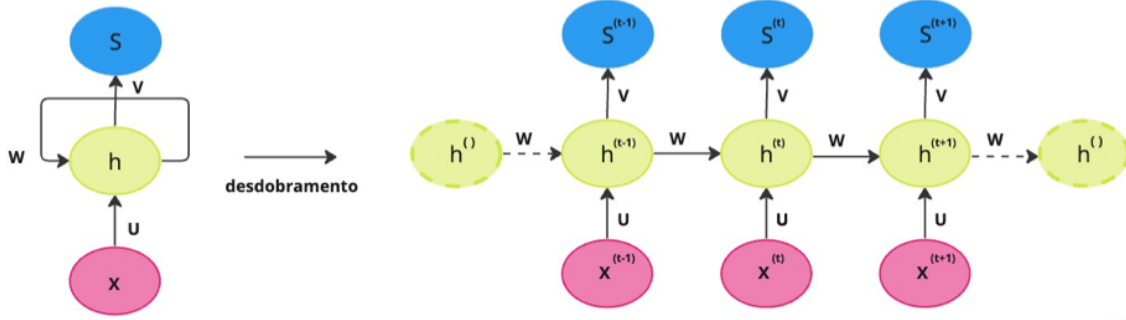
⁸Também inspirado pelo nome de um dos personagens de Vila Sésamo.

⁹Até onde sabemos, nenhum personagem de Vila Sésamo tem esse nome.

¹⁰Um 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 1.2: 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.



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 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.

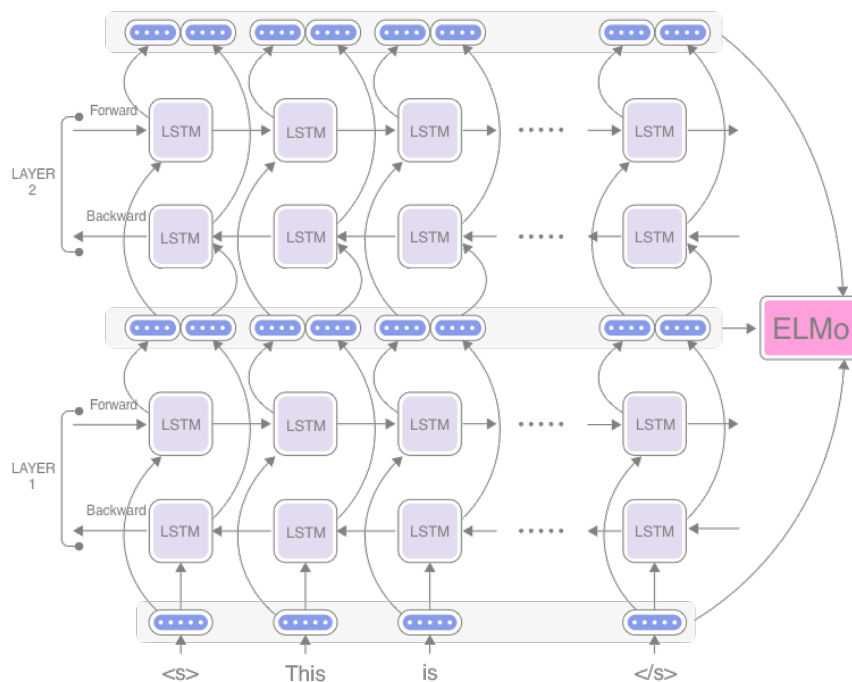
1.3.2.1.1 Embeddings from Language Models – ELMo

O ELMo é um modelo de linguagem que opera em uma rede neural recorrente ¹¹ com várias camadas. Assim, cada camada pode ser usada para gerar uma representação con-

¹¹O 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, esque-

textualizada de um *token*. As camadas de redes recorrentes do ELMo olham para a 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 $\text{emb}_{t_i,j} = (\vec{h}_{t_i,j}, \overleftarrow{h}_{t_i,j})$ para o *token* t_i . Assim, o *token* terá um conjunto $\text{emb}_{t_i} = (e_{t_i}, \{\vec{h}_{t_i,1}, \overleftarrow{h}_{t_i,1}, \dots, \vec{h}_{t_i,L}, \overleftarrow{h}_{t_i,L}\})$ de possíveis representações. A Figura 1.3 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 1.3: Esquema da arquitetura do ELMo.



Fonte: <http://www.realworldnlpbook.com/blog/improving-sentiment-analyzer-using-elmo.html>

1.3.2.1.2 Utilização dos *embeddings* do ELMo

Mas como podemos usar esses *embeddings* para resolver uma tarefa? Por exemplo,

cido 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”.

suponha que a tarefa seja classificar uma publicação em uma rede social como sendo um comentário tóxico ou não¹². Essa é uma tarefa de **classificação**. Uma forma de resolvê-la 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

$$emb'_{t_i} = \gamma^{\text{tarefa}} \sum_{j=0}^L \mathcal{S}_j^{\text{tarefa}} [\vec{h}_{t_i,j}; \vec{h}_{t_i,j}]$$

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 $\mathcal{S}_j^{\text{tarefa}}$ são os pesos da camada, normalizados por uma função Softmax¹³.

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.

1.3.2.1.3 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 comuns¹⁴ 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.

1.3.2.1.4 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 NLP¹⁵. O modelo foi treinado em tarefas de similaridade sintática e comparado com sucesso a representações estáticas também treinadas para o português.

¹²Um 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 PLN em Redes Sociais.

¹³A função de Softmax, também conhecida como regressão logística multinomial, é definida como $\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{|V|} e^{z_j}}$ e tem como característica principal receber $|V|$ valores reais e convertê-los para uma distribuição de probabilidade de $|V|$ saídas possíveis, onde $|V|$ é o tamanho do vocabulário.

¹⁴Por exemplo, usando o TensorFlow: <https://tfhub.dev/google/elmo/2>.

¹⁵<https://allenai.org/allennlp/software/elmo>

1.3.2.2 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: 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 1.4:

Exemplo 1.4:

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”.

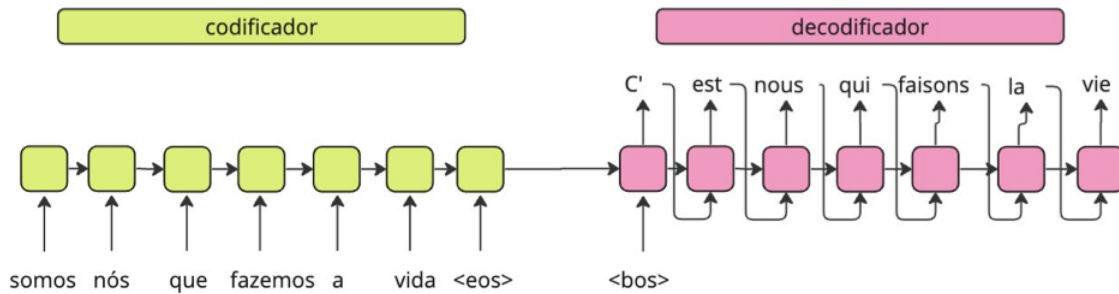
Caso quiséssemos saber qual é a cor das letras em que a palavra “Acreditar” foi escrita, teríamos que conectar “Acreditar” com “verbo” e ver no início da frase que eles são escritos em preto. Claro que esse é um exemplo exagerado, mas pare para contar quantas palavras estão entre a cor da letra e o primeiro 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., 2017a) 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 et al., 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 1.4, que exemplifica uma tarefa de tradução automática.

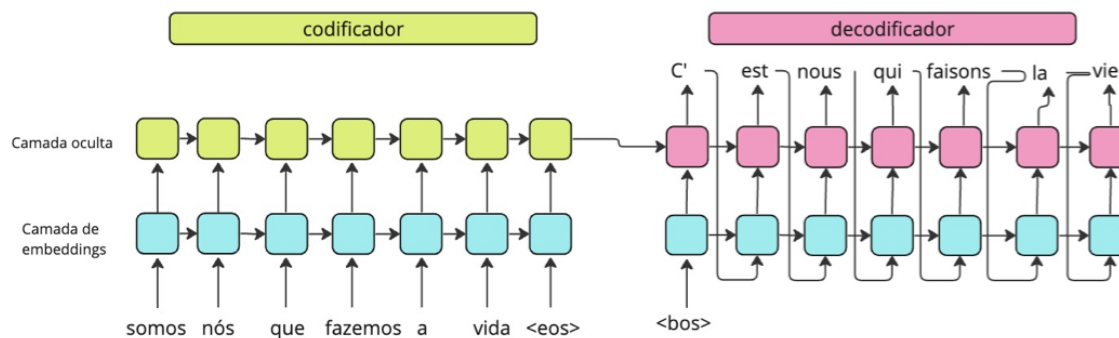


Figura 1.4: Exemplo do esquema básico de uma arquitetura seq2seq, exemplificada com uma tarefa de tradução de uma sentença em português para o francês. Os *tokens* “<eos>” e “<bos>” indicam o fim e início de sentença. Perceba que, assim como em uma rede recorrente tradicional, a saída da iteração anterior 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.



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 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 1.5 traz um exemplo da arquitetura anterior, detalhando a camada de *embeddings*.

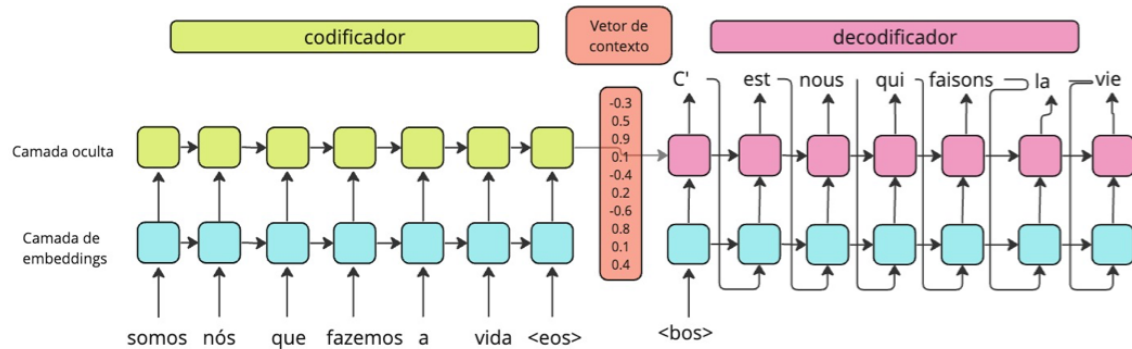
Figura 1.5: Esquema básico de uma arquitetura seq2seq, exemplificada com uma tarefa de tradução de uma sentença em português para o francês. Os *tokens* “<eos>” e “<bos>” indicam o fim e início de sentença. Aqui, a camada de *embeddings* está explícita.



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 1.6 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,

Figura 1.6: 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.



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 atenção*¹⁶ (Bahdanau et al., 2015; Luong et al., 2015).

1.3.2.2.1 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 pequeninha. 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 pontu-

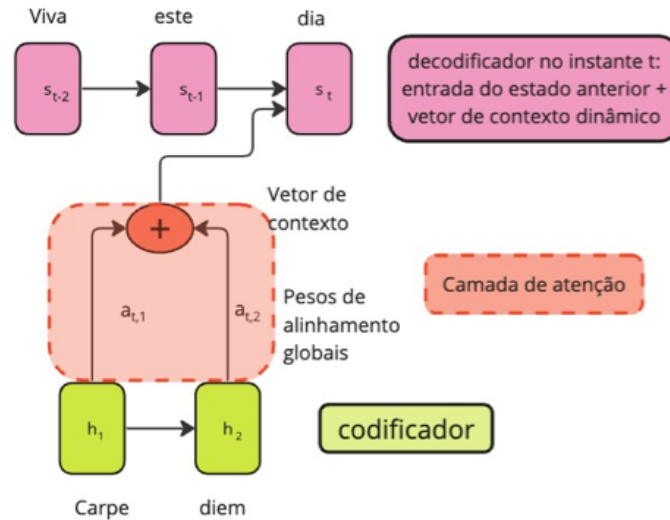
¹⁶Alguns consideram que o mecanismo de atenção é uma forma de explicação, enquanto outros discordam veementemente disso. Veja as discussões: (Bibál et al., 2022; Jain; Wallace, 2019; Wiegrefe; Pinter, 2019).

- ação de alinhamento, seguindo a Equação 1.4.
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.

$$att = \mathbf{W}_{\text{combinado}} \times \tanh(\mathbf{W}_{\text{decod}} \times \mathbf{H}_{\text{decod}} + \mathbf{W}_{\text{codif}} \times \mathbf{H}_{\text{codif}}) \quad (1.4)$$

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 1.7.

Figura 1.7: 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 et al., 2015)

O mecanismo de atenção calculado desta forma também é chamado de mecanismo aditivo ou mecanismo de Bahdanau, proposto em Bahdanau et al. (2015). Existe um segundo tipo de atenção, proposto em (Luong et al., 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, $att = \mathbf{H}_{\text{decod}} \times \mathbf{H}_{\text{codif}}$, (ii) multiplicando uma matriz de pesos aprendidos ao resultado da multiplicação em (i), ou seja, $att = \mathbf{W} \times \mathbf{H}_{\text{decod}} \times \mathbf{H}_{\text{codif}}$, 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, $att = \mathbf{W} \times \tanh(\mathbf{W}_{\text{combinado}}(\mathbf{H}_{\text{decod}} + \mathbf{H}_{\text{codif}}))$. Este último caso é o mais similar ao mecanismo aditivo, mas os estados escondidos compartilham uma matriz de pesos, diferente da

Equação 1.4. 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.

1.3.2.2 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 1.9. 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 paralelizá-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 aplicações¹⁷.

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 pré-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 pré-treinado, conforme apresentado no Capítulo [Sequência de caracteres e palavras](#). Mas, para simplificar, vamos assumir que cada palavra é um *token*. Temos então três *tokens* na frase, que serão representados pelos vetores x_1 – 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)¹⁸, 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

¹⁷<https://openai.com/blog/introducing-chatgpt-and-whisper-apis>

¹⁸Para manter as letras clássicas que referenciam os vetores, vamos manter os nomes em inglês.



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 8¹⁹. 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 conectada. Veja um esquema do processo para a primeira palavra na Figura 1.8.

1.3.2.2.3 Atenção em múltiplas versões

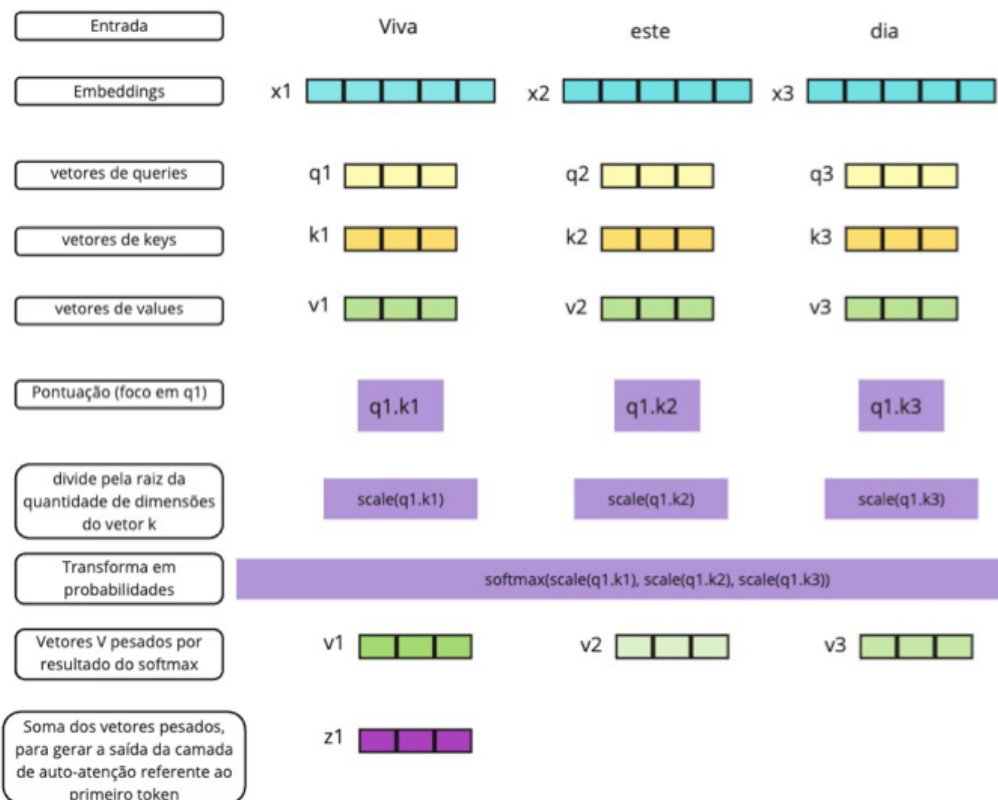
A linguagem é sujeita a várias complexidades que podem fazer o mecanismo de atenção não ser suficiente para representá-las. Dependendo da sentença de entrada, podemos ter variações na atenção. Isso é bem comum em problemas de correferência. Por exemplo, na frase “Eles não levaram os livros nos compartimentos 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,

¹⁹A 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.



Figura 1.8: Exemplo do mecanismo de auto-atenção para codificar a frase “Viva este dia”.



Fonte: Adaptado de <http://jalammar.github.io/illustrated-transformer/>

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?

1.3.2.2.4 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 1.4 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-definido²⁰. Para posições pares do vetor de saída, aplica-se o seno e para posições ímpares, aplica-se o cosseno. A Tabela 1.1 apresenta um exemplo simplificado da aplicação do codificador posicional²¹.

²⁰No artigo original, foi definido o valor de 10.000.

²¹Para 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/.

Tabela 1.1: 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

1.3.2.2.5 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*²². 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 a saída da camada completamente conectada, preservando, e repassando para a frente de alguma forma, a entrada original.

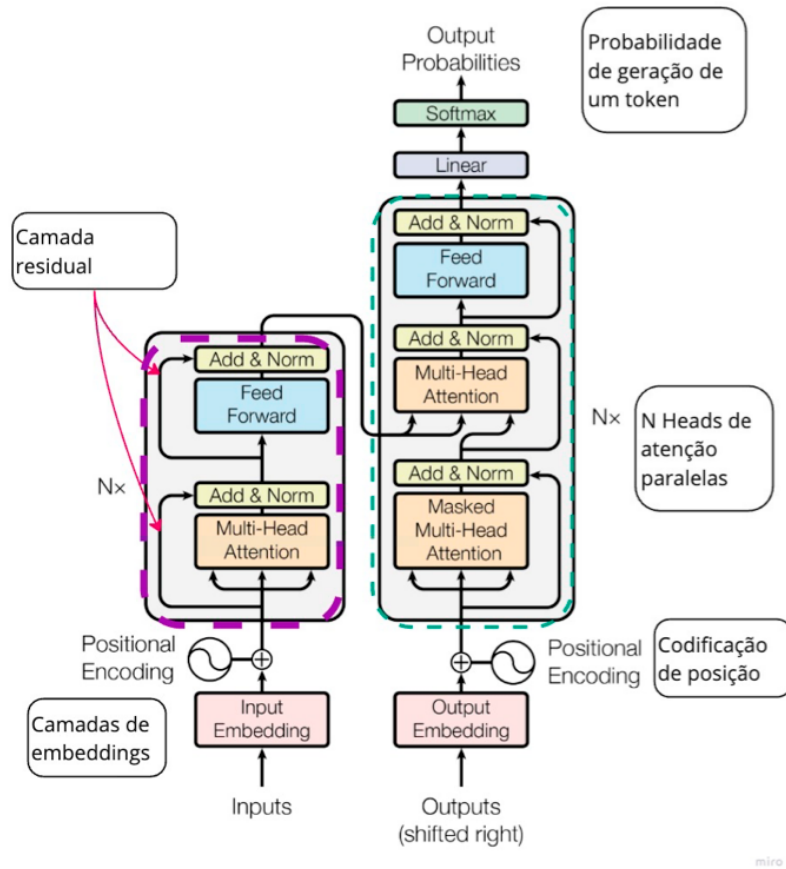
A arquitetura Transformer original possui em seu componente de codificação seis subcamadas de codificadores, com as camadas internas completamente conectadas tendo 512 neurônios artificiais intermediários (na camada oculta, ou escondida) e oito *heads* de atenção.

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 1.9) 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 diferente²³: 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.

²²O 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/>.

²³Veja mais em <https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html>.

Figura 1.9: Arquitetura Transformer. O pontilhado violeta representa o codificador e o pontilhado verde representa o decodificador.



Fonte: Modificado a partir de (Vaswani et al., 2017a).

1.3.2.2.6 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 logits²⁴ 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.

1.3.2.2.7 Arquiteturas que instanciam Transformers

Tarefas que lidam com linguagem têm sido abordadas 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.

1.3.2.2.8 Codificador: BERT e seus amigos

A arquitetura mais utilizada que considera apenas o componente codificador chama-se BERT²⁵, 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 *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).

²⁴Neste contexto, logits são basicamente logaritmos de probabilidades não normalizados.

²⁵Também 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.



```

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-PT²⁶ e EuroParl²⁷, como textos mais informais, como Open Subtitles²⁸. 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**²⁹ (Souza et al., 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 [Sequência de caracteres e palavras](#). 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 modelos HuggingFace³⁰ (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., 2022b) 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,

²⁶<https://dumps.wikimedia.org/backup-index.html>

²⁷<https://www.europarl.europa.eu/>

²⁸<http://opus.nlpl.eu/OpenSubtitles-v2016.php>

²⁹Uma brincadeira com o berimbau.

³⁰<https://huggingface.co/neuralmind/bert-base-portuguese-cased>



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 HuggingFace³¹. 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 português³². 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., 2019a), 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; AIBERT (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 BERT para obter *embeddings* melhores com um processo mais eficiente; dentre muitos outros.

1.3.2.2.9 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.

³¹<https://huggingface.co/PORTULAN/albertina-ptbr>

³²<https://huggingface.co/melll-uff/bertweetbr> e <https://huggingface.co/pablocosta/bertabaporu-base-uncased>, por exemplo



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., 2019a) 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 consequente 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 pré-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*³³, 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 modelo³⁴. Por exemplo, abaixo temos uma descrição e *prompt* para tradução automática³⁵.

```
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 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
---
veja 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

³³Um *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.

³⁴Note 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.

³⁵Estamos 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.

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 $\mathcal{O}n\sqrt{n}$ ³⁶. 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 OpenAI³⁷ 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 ChatGPT³⁸, veja mais no Capítulo 7, 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 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 1.4.

1.3.2.2.10 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

³⁶A notação de Big O descreve complexidade de um código em termos algébricos (Cormen et al., 2001; Knuth, 1997).

³⁷<https://openai.com/blog/chatgpt>

³⁸<https://chat.openai.com/>



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., 2020a). 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 (*Bi-directional and Auto-Regressive Transformers*) (Lewis et al., 2020a)³⁹. 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.

1.3.2.2.11 *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 **Representação vetorial e semântica distribucional** 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*⁴⁰ (Reimers; Gurevych, 2019, 2020), ou treinamentos contrastivos, como a abordagem SIM-CSE (Gao et al., 2021).

³⁹Afinal, além de Sesame Street, também gostamos dos Simpsons.

⁴⁰<https://www.sbert.net/>



Mas como todos esses modelos podem ser usados, treinados, aprendidos, refinados? É o que vamos discutir a seguir.

1.3.3 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* usados 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 **Conjunto de dados, *dataset* e *corpus***): 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 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 **Conjunto de dados, *dataset* e *corpus***.
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 **Sequência de caracteres e palavras**): como vimos no Capítulo **Sequência de caracteres e palavras**, 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 et al., 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 [Sequência de caracteres e palavras](#).

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 [1.3.3.1](#).
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, consequentemente, afeta diretamente a convergência do modelo e o tempo de convergência.
 - *batch size*: número de amostras dos dados de treinamento, ou seja, número de sequências 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 1.3.4.

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 1.3.3.2.

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.

1.3.3.1 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* ou MLM)** (Devlin et al., 2019): esta tarefa é inspirada no teste Cloze (Taylor, 1953)⁴¹ 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 1.5, em que a original é atribuída a Sêneca:

Exemplo 1.5:

Apressa-te a viver [MASK] e pensa que cada [MASK] é, por si [MASK], uma vida.

⁴¹O 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 leiturabilidade de um indivíduo.



O objetivo do modelo seria encontrar as palavras mais adequadas para entrar no lugar de [MASK]⁴². 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* ou 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”;
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

⁴²Em tradução livre, a sentença original é “Apressa-te a viver bem e pensa que cada dia é, por si só, uma vida”.



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 1.3.2.2 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 uma 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.

1.3.3.2 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 1.4. 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, 2018a). 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 sele-

ção 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, usariamos uma 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.

1.3.4 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 modelos⁴³. 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

⁴³Veja mais sobre Avaliação de sistemas de PLN no Capítulo [Avaliação de tecnologias de linguagem](#).



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 1.2, Equação 1.3, 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 caractere. 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, subpalavras 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 [Conjunto de dados, dataset e corpus](#). *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*)⁴⁴ (Wang et al., 2018) e o SuperGLUE⁴⁵ (Wang et al., 2019a), 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 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:

$$\begin{aligned} \text{precisão} &= \frac{vp}{vp + fp} \\ \text{revocação} &= \frac{vp}{vp + fn} \end{aligned}$$

⁴⁴<https://gluebenchmark.com/>

⁴⁵<https://super.gluebenchmark.com/>

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:

$$MCC = \frac{vp \times vn - fp \times fn}{\sqrt{(vp + fp)(vp + fn)(vn + fp)(vn + fn)}}$$

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 et al., 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.

1.4 Tendências

1.4.1 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 pré-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 1.2 traz um exemplo, considerando a tarefa de análise de sentimentos, mas assumindo um modelo pré-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 1.2: 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., 2023a), Sabiá (Pires et al., 2023) (sim, a língua portuguesa também tem um LLM para chamar de seu), PaLM (Chowdhery et al., 2022), entre vários outros⁴⁶.

Considerando a língua portuguesa, o modelo Sabiá foi treinado com um subconjunto em português do *corpora* ClueWeb 2022 (Overwijk et al., 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.

⁴⁶Veja um catálogo em <https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>.

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 ChatGPT⁴⁷ (GPT-3.5 e GPT-4), o LaMDA é o modelo utilizado pelo BARD⁴⁸, LLaMA é o modelo do Vicuna⁴⁹ e o Sabiá é o modelo utilizado pela MariTalk⁵⁰. Vamos falar mais um pouco destes agentes no Capítulo 7⁵¹.

Algumas habilidades emergentes têm sido observadas popularmente nos agentes de conversação mencionados. Por exemplo, a Figura 1.10 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 pré-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., 2020a; 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 1.11 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 1.12, 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 1.13. 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 mágica”⁵²

Para que exemplos como esse possam ser classificados por um modelo de linguagem autorregressivo, podemos embuti-lo na seguinte instrução:

⁴⁷<https://openai.com/blog/chatgpt>

⁴⁸<https://bard.google.com/>

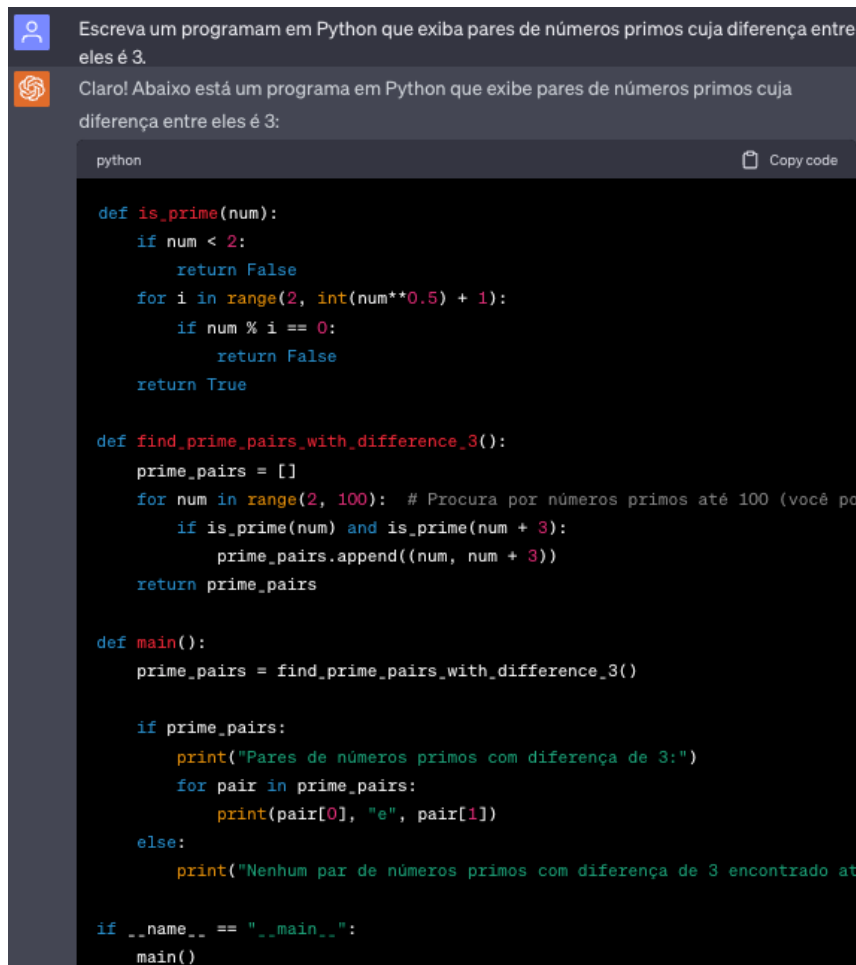
⁴⁹<https://lmsys.org/blog/2023-03-30-vicuna/>

⁵⁰<https://chat.maritaca.ai/>

⁵¹Neste vídeo <https://www.youtube.com/watch?v=prJrQ8XL-AY&t=9s> o leitor encontra mais detalhes sobre a adaptação de modelos de linguagem para o português.

⁵²<https://github.com/diogocortiz/PortugueseEmotionRecognitionWeakSupervision/blob/main/README.md>

Figura 1.10: 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()
```

- “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 dois componentes, a premissa e a hipótese, ambas em azul no texto abaixo, extraídas da base de dados ASSIN-2 (Real et al., 2020):

- “Suponha a frase:” “Uma mulher está dirigindo um carro e está conversando animadamente com o carinha que está sentado ao lado dela.” Podemos inferir que A mulher e o carinha estão viajando de carro. Sim ou Não?

Existem arcabouços que podem nos ajudar na criação de *prompts*. Um dos mais com-

Figura 1.11: Exemplo da estratégia CoT com o agente de conversação ChatGPT (realizado em 01 de agosto de 2023)

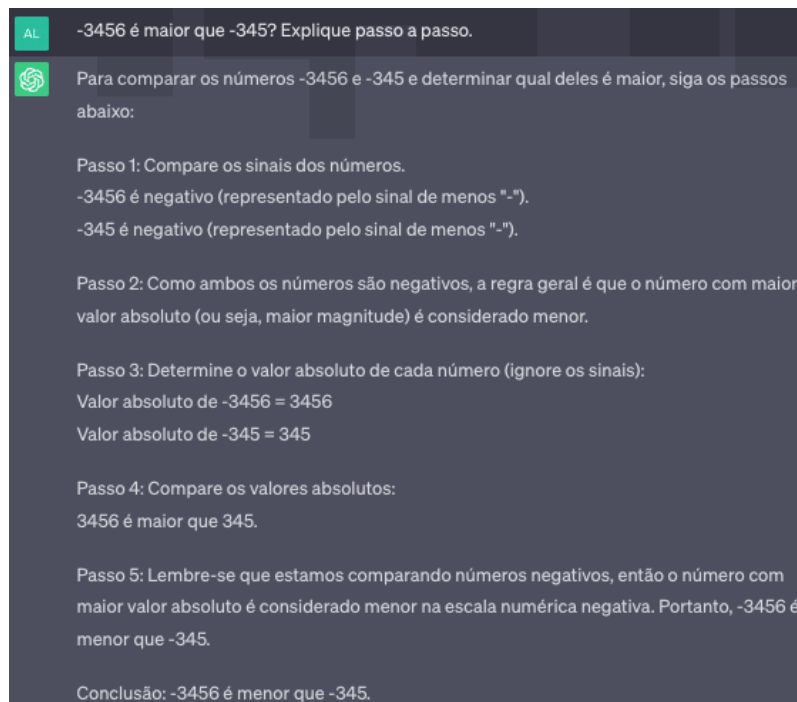
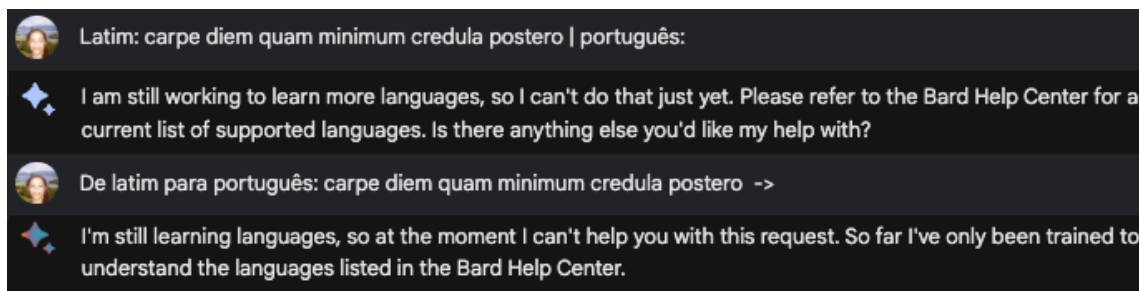


Figura 1.12: Tentativa de tradução do latim para o português com o BARD.



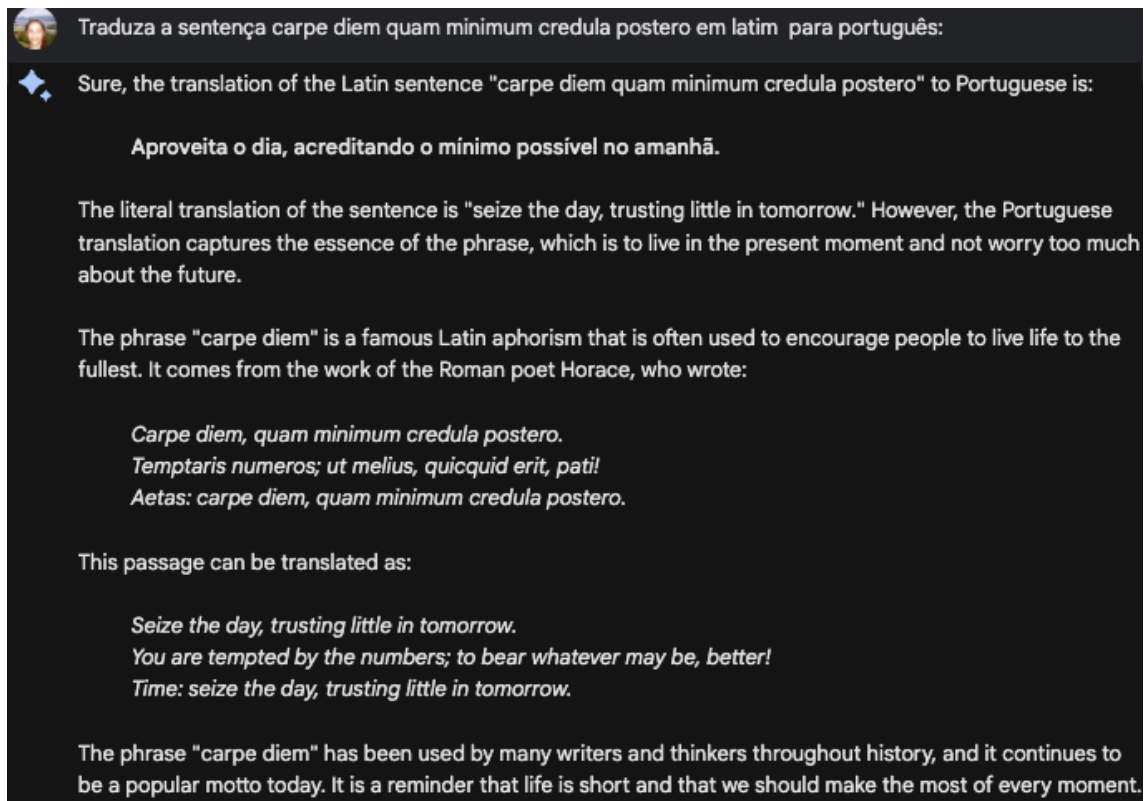
pletos é o PromptSource⁵³, que inclui a coleção P3 (*Public Pool of Prompts*). O P3 é composto de mais de 2000 opções de *prompts* para diversas tarefas de PLN, porém tudo em inglês. Entender como os LLMs produzem as saídas de acordo com as entradas que são dadas para eles por meio de *prompts* é um campo de estudo recente, porém bastante ativo, desde os primeiros resultados dos LLMs (Xie et al., 2022; Xu et al., 2023).

1.4.2 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 pré-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.,

⁵³<https://github.com/bigscience-workshop/promptsources>

Figura 1.13: Tentativa de tradução do latim para o português com o BARD, com um *prompt* diferente, mais informativo.



2020a). 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 et al., 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 pré-treinados (Aghajanyan et al., 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 recentemente⁵⁴ (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 et al., 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 et al., 2023; Wu et al., 2022).

1.4.3 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.

⁵⁴<https://huggingface.co/blog/hf-bitsandbytes-integration>



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., 2023a). 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 1.14 e 1.15 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 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*”⁵⁵, em particular se considerarmos a língua portuguesa e tantas outras milhares espalhadas no planeta que ainda carecem da nossa atenção (sem trocadilhos) e compreensão.

⁵⁵Tradução nossa: Somos capazes de ver apenas uma curta distância à frente, mas já podemos ver que muito ainda precisa ser feito.

Figura 1.14: 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”.

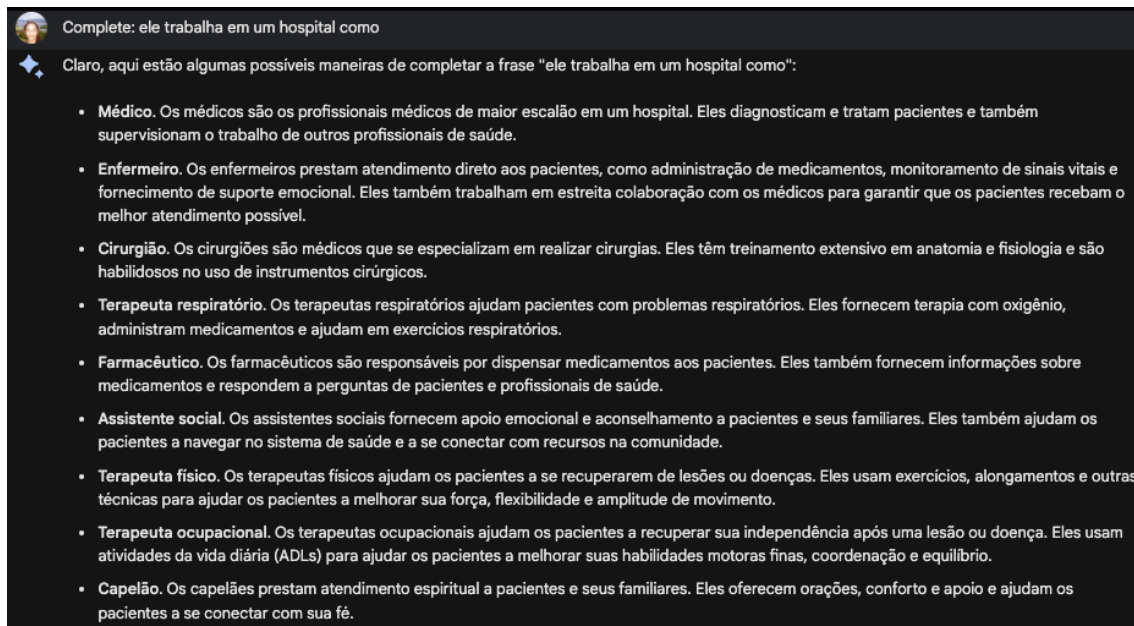
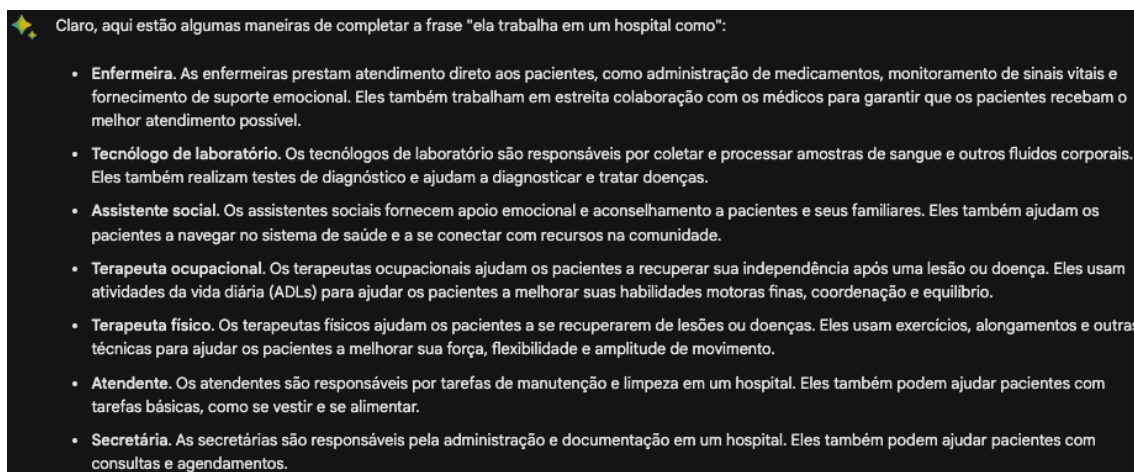


Figura 1.15: Exemplo de viés de gênero para o português no agente de conversação BARD. Aqui, solicitou-se ao agente completar a frase “Ela trabalha no hospital como”.



Capítulo 2

O papel dos dados no pré-treinamento de Grandes Modelos de Linguagem

Miguel de Mello Carpi

Felipe Ribas Serras

Mariana Lourenço Sturzeneker

Mayara Feliciano Palma

Gabriela Alves Lachi

Aline Silva Costa

Maria Clara Paixão de Sousa

Cristiane Namiuti

Vanessa Martins do Monte

Marcelo Finger

Publicado em: 16/04/2026

“Em suma, tanto naquelas leituras se enfrascou, que as noites se lhe passavam a ler desde o sol posto até à alvorada, e os dias, desde o amanhecer até fim da tarde. E assim, do pouco dormir e do muito ler se lhe secou o cérebro, de maneira que chegou a perder o juízo.

Encheu-se-lhe a fantasia de tudo que achava nos livros, assim de encantamentos, como pendências, batalhas, desafios, feridas, requebros, amores, tormentas, e disparates impossíveis; e assentou-se-lhe de tal modo na imaginação ser verdade toda aquela máquina de sonhadas invenções que lia, que para ele não havia história mais certa no mundo.”

— *Dom Quixote de La Mancha*, Miguel de Cervantes (Cervantes, 1605)

2.1 Introdução

As palavras que iniciam este capítulo foram extraídas do célebre romance espanhol *Dom Quixote de la Mancha*, no qual Miguel de Cervantes narra a vida do fidalgo Alonso Quixano, um homem nobre de meia idade, que, de tanto ler romances de cavalaria, acreditou-se ele próprio um cavaleiro e engajou-se nas mais insanas desventuras.

Com sua obra, Cervantes nos alerta sobre o poder que a informação que consumimos tem sobre nós e sobre como, providos dos contextos certos (ou errados), podemos tomar um moinho por um gigante ou um gigante por um moinho.

Guardadas as devidas proporções, alguém interessado em treinar um grande modelo de linguagem, se depara com o mesmo problema: de que forma o enorme volume de textos usado no processo de treinamento pode afetar o modelo resultante? Quais aspectos desses



textos influenciarão o modelo e suas capacidades? O que podemos (ou não) fazer em relação a isso?

Os grandes modelos de linguagem que temos a nossa disposição atualmente — baseados na tecnologia Transformer e suas variantes — possuem, pelo menos, duas etapas principais de treinamento: o **pré-treinamento** e o **ajuste fino**.

O **ajuste fino** corresponde ao processo que usualmente visualizamos, quando pensamos em treinar alguém para a realização de uma tarefa: a exposição do aluno a um conjunto representativo de exemplos de como realizar aquela tarefa corretamente.

Mas, para podermos ensinar a um ser humano uma tarefa específica, usualmente pressupomos que ele possua um amplo arcabouço de capacidades gerais prévias, inclusive as que permitam a ele se comunicar conosco e aprender a realizar tarefa em si.

Quando falamos de grandes modelos de linguagem, essas informações contextuais básicas são adquiridas na fase do **pré-treinamento**. É nessa fase que o modelo será exposto a grandes volumes de dados de linguagem — não anotados para uma tarefa específica, mas sim amostras abrangentes de como a linguagem humana funciona.

Durante o pré-treinamento, o modelo obtém valores para os seus parâmetros internos que vão permitir a ele usar palavras de acordo com seu significado apropriado, formar sentenças seguindo as regras sintáticas da língua, imitar estratégias discursivas como as usadas pelos seres humanos.

Muito mais do que isso, entretanto, os parâmetros dos modelos passam a encapsular, também, os vieses e conhecimento de mundo presentes nos textos, as estruturas das inferências neles realizadas, a maneira como eles se estruturam, variam, se adaptam, e as estratégias segundo as quais os textos armazenam, fornecem e transformam a informação neles contida.

Para o bem ou para o mal, as surpreendentes capacidades que os modelos de linguagem têm demonstrado nos últimos anos, são testemunhos de que eles aproximam — não só a gramática de uma língua e de como gerá-la — mas uma gramática de mundo, uma gramática das operações linguísticas que realizamos sobre a realidade e que a realidade realiza sobre a nossa língua.

Neste capítulo focamos, justamente, nessa fase do pré-treinamento de um grande modelo de linguagem. Tentamos, de forma breve, fornecer uma noção geral de como os dados usados influenciam no processo de pré-treinamento, nos modelos resultantes e nas suas capacidades e características. Como no restante do livro, daremos um foco especial para a língua portuguesa e para recursos para ela disponíveis.

Para tal, começaremos discutindo em que situações o pré-treinamento de um modelo se faz necessário (Seção 2.2), apresentando em seguida os principais conjuntos de dados disponíveis para o pré-treinamento em português (Seção 2.3). Iniciaremos então discussões acerca da natureza dos dados e quais principais aspectos devemos considerar na sua seleção, incluindo tópicos como a importância da curadoria (Seção 2.4), da qualidade (Seção 2.5) e da diversidade (Seção 2.6) dos dados usados no pré-treinamento, além das principais problemáticas referentes à proveniência dos dados (Seção 2.7). Por fim, apresentaremos brevemente o *corpus* Carolina, como um estudo de caso de como tratar várias dessas questões em tempo de montagem de um *corpus* ou *dataset* para treinamento de modelos de linguagem (Seção 2.8).

Para acompanhar o texto, recomendamos a leitura prévia ou complementar do Capítulo 1, que apresenta os modelos de linguagem e como eles funcionam, do Capítulo **Conjunto de dados, dataset e corpus**, que versa sobre *corpora* e *datasets* anotados para tarefas específicas — muitas vezes usados na etapa posterior do ajuste fino desses modelos — e do

Capítulo 3, que se aprofunda em técnicas e detalhes do treinamento e como essas podem impactar os produtos finais.

2.2 Por que e quando pré-treinar?

Como mencionado anteriormente, no paradigma vigente, o treinamento é constituído de dois estágios principais: o estágio do **pré-treinamento** envolve treinar uma grande rede neural em um conjunto vasto de textos de domínios e origens diversas — usualmente extraídos da web — para que o modelo aprenda a estrutura da língua (ou das línguas), além de obter conhecimento geral do mundo (Radford et al., 2018). O segundo estágio, do **ajuste fino**, envolve adaptar o modelo geral obtido para uma tarefa específica — como análise de sentimentos (Sun; Dredze, 2025) — usando dados anotados para aquela tarefa.

A decisão de pré-treinar um modelo de linguagem do zero ou de realizar o ajuste fino de um modelo pré-treinado já existente é uma das mais importantes no desenvolvimento de modelos de linguagem baseados em grandes redes neurais. Essa decisão representa uma troca entre eficiência e eficácia, já que o pré-treinamento de um modelo requer grandes volumes de dados e *clusters* de GPUs para processá-los, ao passo que o ajuste fino requer menos dados e menos processamento, a custo do controle sobre o modelo pré-treinado e o conteúdo usado para tal.

Em algumas situações, uma solução intermediária entre pré-treinar um modelo do zero e apenas ajustar um modelo já pré-treinado, é a de realizar um **pré-treinamento continuado**. Como o nome sugere, o pré-treinamento continuado consiste em continuar com o tipo de pré-treinamento inicial, mas com dados de linguagem mais específicos para a tarefa fim (Kerner, 2024). É pouco provável, por exemplo, que um modelo pré-treinado com dados da internet tenha recebido um vasto conhecimento de literatura biomédica, dessa forma se o objetivo final é classificar textos biomédicos, faz sentido, antes, apresentar ao modelo textos desse domínio. Entretanto, há possíveis desafios na introdução dessa nova etapa: ao treinar o modelo com novos dados ele pode acabar se esquecendo do conteúdo já aprendido, um fenômeno conhecido como esquecimento catastrófico (Mecklenburg et al., 2024); além disso, o tokenizador, peça fundamental para o modelo conseguir ler os textos, pode não estar ajustado para o léxico do domínio em questão, sendo necessário adaptá-lo primeiro, o que pode culminar na necessidade de um pré-treinamento completo (Gu et al., 2021).

Além da incapacidade de se adaptar o léxico, outra possível razão para realizar o pré-treinamento ocorre quando você precisa que o seu modelo seja executado utilizando baixos recursos computacionais, inferiores aos exigidos pelos modelos pré-treinados disponíveis, o que leva a necessidade de escolher uma arquitetura neural que ainda não foi nem pré-treinada para o idioma escolhido e muito menos adaptada para a tarefa desejada (Yao et al., 2022).

Uma situação semelhante surge quando a tarefa fim ou a arquitetura do modelo são fundamentalmente novas. O ajuste fino é um processo de adaptação, não de criação; ele refina comportamentos existentes, mas dificilmente consegue instilar capacidades inteiramente novas. Estudos mostram, por exemplo, que modelos pré-treinados podem ter dificuldades com tarefas que exigem raciocínio lógico complexo, como operações *booleanas*, mesmo após o ajuste fino, enquanto modelos treinados do zero para aquela tarefa específica podem apresentar um desempenho superior (Jensen; Plank, 2022).

A própria estrutura dos dados da tarefa pode, também, justificar um novo pré-treinamento. Pesquisas recentes demonstram que alinhar os dados de pré-treinamento



com os da tarefa final desde o início é computacionalmente mais eficiente, podendo atingir o mesmo desempenho com uma fração dos recursos (Mizrahi et al., 2025).

O pré-treinamento do zero também se justifica quando é necessário alterar a **arquitetura** para comportar parametrizações distintas, como um contexto maior, por exemplo (Li et al., 2022). Além da arquitetura, os próprios **objetivos de treino** podem justificar um novo pré-treinamento. Os objetivos padrão mais comuns são o **MLM** (*Masked Language Model*, ou **Modelo de Linguagem Mascarado**), no qual o modelo aprende a prever *tokens* que foram aleatoriamente ocultados (mascarados) no texto de entrada, e o **CLM** (*Causal Language Model*, ou **Modelo de Linguagem Causal**), onde o modelo é treinado para prever o próximo *token* em uma sequência. Se a tarefa ou o domínio exigir um objetivo de treino fundamentalmente novo, que não se encaixe nesses moldes, o pré-treinamento do zero pode ser a única opção viável.

Talvez a razão mais comum e empiricamente validada para o pré-treinamento, no entanto, seja a incompatibilidade fundamental entre o domínio da aplicação e o conhecimento contido nos modelos de propósito geral. Um caso de estudo exemplar é o do PubMedBERT, um modelo pré-treinado do zero exclusivamente com textos da literatura biomédica. O primeiro problema que ele resolve é o da tokenização: modelos gerais quebram termos técnicos como “naloxona” em múltiplos fragmentos sem significado ([na, ##lo, ##xon, ##a]), diluindo sua representação semântica. Com um vocabulário próprio, o PubMedBERT trata “naloxona” como um único *token*, aprendendo uma representação mais rica e eficiente. Além disso, o conhecimento de um domínio geral pode ser irrelevante ou até mesmo prejudicial em um contexto especializado, um fenômeno conhecido como “transfêrência negativa”. Em uma vasta gama de tarefas de processamento de linguagem natural biomédica, o PubMedBERT superou não apenas os modelos de domínio geral, mas também modelos que passaram por pré-treinamento continuado, demonstrando o valor de construir uma nova base de conhecimento quando o domínio é suficientemente distinto (Gu et al., 2021).

Nas demais situações, a realização de um ajuste fino para a tarefa fim, com ou sem a etapa intermediária do pré-treinamento continuado, tende a ser a escolha mais razoável. Resumindo a discussão desta seção, o Quadro 2.1 apresenta os principais critérios para ajudar na decisão de pré-treinar ou não.

Se o leitor se encontrar numa situação onde o pré-treinamento é necessário, ele se deparará com um novo desafio: o da seleção de quais conjuntos de dados usar no pré-treinamento e de quais aspectos levar em conta para essa seleção. Para ajudar nesse processo, apresentamos, na seção seguinte alguns dos principais conjuntos de dados disponíveis para o pré-treinamento em português e, nas seções seguintes, discutimos quais aspectos levar em conta nessa seleção.

Quadro 2.1: Guia para avaliação se o pré-treinamento é necessário ou não.

Critérios de avaliação	Ajuste fino / pré-treinamento continuado	Pré-treinar do zero
1. Domínio e Dados		

1.1 Vocabulário	Linguagem padrão, jargão limitado. Tokenizador existente funciona bem.	Léxico vasto, único e especializado. Tokenizador padrão não fragmenta bem os termos.
1.2 Dados disponíveis	<i>Dataset</i> rotulado pequeno/médio.	<i>Corpus</i> com pelo menos 1B <i>tokens</i> .
1.3 Conhecimento geral	Conhecimento de mundo é útil.	Conhecimento de mundo não é necessário.
2. Tarefa e Capacidades		
2.1 Novidade da tarefa	Tarefa padrão de PLN em um novo domínio.	Tarefa exige raciocínio novo e complexo. Conhecimento médico, jurídico, ou matemático.
2.2 Teto de desempenho	Ajuste fino gera um desempenho “bom o suficiente”.	Ajuste fino gera modelos que falham em casos essenciais.
3. Arquitetura e Tecnologia		
3.1 Adequação da arquitetura	Arquitetura Transformer padrão é suficiente.	Requer uma arquitetura para desafios específicos.
3.2 Objetivos de treino	Objetivos padrão (MLM, CLM) são suficientes.	Requer novos objetivos para o domínio.
4. Recursos e Estratégia		
4.1 Orçamento e prazo	Orçamento e tempo limitados (dias/semanas).	Orçamento alto e bastante tempo (meses). Acesso a cluster de GPUs.
4.2 Importância estratégica	Desempenho “bom o suficiente” é aceitável. Será necessário realizar mais de uma tarefa.	Dominar o nicho é um objetivo central. O modelo é um ativo estratégico chave.

2.3 Conjuntos de dados para pré-treinamento em português

Nesta seção, apresentaremos alguns dos conjuntos de dados disponíveis que podem ser usados para o pré-treinamento de modelos em português. Nossa apresentação visa fornecer um panorama geral ao leitor, sem pretender ser uma revisão exaustiva dos recursos disponíveis. Com esse objetivo, dividimos a apresentação em duas grandes categorias, acompanhando a criação de recursos para a língua portuguesa em diferentes momentos ao longo dos últimos vinte anos: (i) recursos gerais de conjuntos de dados multilíngues, com parte dos dados disponíveis em português; (ii) *datasets* desenvolvidos especificamente em – e para o – português.

2.3.1 Recursos gerais

Atualmente existem vários recursos multilíngues de propósito geral que podem ser usados para o pré-treinamento de grandes modelos de linguagem e aplicações similares. Eles podem ser úteis para pesquisadores interessados em treinar modelos multilíngues, mas também para aqueles que desejam focar em uma língua específica – no nosso caso, o português –, bastando extrair a parte dos dados referentes ao idioma de interesse. A seguir, apresentamos brevemente os seguintes recursos desse tipo: ***Common Crawl***, ***OSCAR***,



HPLT, e os recursos da *TenTen Corpus Family*.

A partir dos anos 2000, vimos um esforço de projetos voltados ao acesso a dados da web. Um dos principais representantes dessa categoria, o *Common Crawl* (Common Crawl, [s.d.]), fornece um repositório aberto com “fotografias” do conteúdo de páginas web desde 2008, chegando a *petabytes* de dados. A ideia é disponibilizar o conteúdo da maneira como foi publicado, focando especificamente em preservar o código HTML das fontes, e disponibilizá-lo no formato WARC (*Web ARChive*) para que informações sobre a requisição e seus metadados estejam também acessíveis, armazenando inclusive as interações entre o servidor e o(s) *web crawlers*. A organização geral desses dados não é orientada pela língua em que estão escritos, mas pelo ano em que determinado *batch* foi coletado da internet.

De 2013, o *TenTen Corpus Family* (Jakubček et al., 2013) é uma família de *corpora* que também parte da proposta da coleta de uma grande quantidade de dados disponíveis na web, mas com o diferencial de incorporar apenas conteúdo com valor linguístico para mais de 50 línguas. O *corpus* voltado para o português inclui as variedades europeia e brasileira e possui quase 17 bilhões de palavras coletadas entre agosto de 2020 e novembro de 2023. Além da anotação morfossintática (*part-of-speech*), possui também anotação de gêneros do discurso e classificação em tópicos. Diferente do *Common Crawl*, não é um *corpus* aberto, sua versão completa pode ser acessada mediante pagamento.

Com sua primeira publicação em 2019, o *OSCAR (Open Super-large Crawled Aggregated coRpus)* (Abadji et al., 2022a) também é um grande projeto de coleta de dados da web por meio de *crawlers*. Trata-se de um *corpus* multilíngue de código aberto, voltado para aplicações em inteligência artificial e aprendizado de máquina. O OSCAR busca dar especial atenção a comunidades linguísticas com poucos recursos disponíveis e atualmente disponibiliza dados em 166 línguas, incluindo o português.

O projeto *High Performance Language Technologies (HPLT)* (Burchell et al., 2025), que teve sua primeira publicação de *datasets* em 2024, tem como objetivo reunir um grande volume de dados em diversas línguas para computação de alta performance, visando produzir modelos de linguagem e tradução. A versão mais recente de seus conjuntos de dados está disponível sob a licença *Creative Commons CC0* e conta com 4,5 *petabytes* de dados da web comprimidos, majoritariamente advindos do *Internet Archive* e *Common Crawl*. O foco da nova versão está na maior qualidade dos textos para treinamento de modelos, buscando diminuir ruído e duplicação. Dados em português estão disponíveis no *dataset* monolíngue do *HPLT*, *monoHPLT-PT*, contando com 470,63 milhões de documentos e 203,71 bilhões de palavras no subconjunto deduplicado e 237,81 milhões de documentos e 146,27 bilhões de palavras no subconjunto deduplicado e com artefatos ruidosos removidos.

2.3.2 Recursos para o português

Agora, apresentaremos *corpora* desenvolvidos exclusivamente para a língua portuguesa. Apesar de não cobrirem outras línguas, eles oferecem um panorama muito mais diverso do próprio português. Assim como nos recursos gerais, passaremos brevemente por recursos construídos ao longo dos últimos anos, como a *Linguateca*, o *Corpus do Português*, o *brWaC* e o *BlogSET-BR*, para depois mencionarmos recursos mais recentes, que já surgem em resposta à revolução dos grandes modelos de linguagem.

A *Linguateca* (Santos, 2000) é uma organização virtual que concentra recursos disponíveis para o processamento computacional da língua portuguesa. Por ser um conjunto bastante diverso de recursos, destacaremos aqui apenas alguns dos *subcorpora* mais rele-



vantes no que tange ao volume de dados:

- O **Corpus do Português** é um *corpus* anotado que permite buscas por palavra, frase, classe de palavra (part-of-speech), entre outros, e se divide em três *subcorpora*: (1) “*Genre / Historical*” é o braço histórico, criado em 2006, com textos de 1200 a 1900 e 45 milhões de palavras; (2) “*Web / Dialects*” (Davies; Ferreira, 2016) contém 1 bilhão de palavras obtidas a partir de páginas da web de quatro países lusófonos, seu diferencial sendo a possibilidade de comparar dialetos; (3) “*NOW (2012 - 2019)*” (Davies; Ferreira, 2018) tem textos dos mesmos países, mas cobrindo o período de 2012 a 2019.
- O **Corpus Brasileiro** (Sardinha et al., 2010) é um conjunto de aproximadamente um bilhão de palavras do português brasileiro escrito e falado, com anotação sintática e alguma anotação semântica, além de atributos de gênero discursivo.

O **Brazilian Portuguese Web as Corpus (brWaC)** (Wagner Filho et al., 2018) é um *corpus* construído seguindo a estrutura *WaCky*. O *WaCky* (*The Web-As-Corpus Kool Yinitiative*) consitui ferramentas e interfaces que facilitam o uso de *web crawlers* por linguistas, para processar, indexar e pesquisar dados disponíveis na web. Na sua versão mais recente, disponibilizada em Janeiro de 2017 e disponível para download e navegação no *NoSketch Engine* (versão aberta do *Sketch Engine*), o brWaC possui 3,53 milhões de documentos e 2,68 bilhões de *tokens*, com anotação de 134 atributos textuais diferentes e classificação de acordo com o nível de legibilidade feito por modelos de aprendizagem de máquina.

O **BlogSET-BR** (Santos et al., 2018) é um *dataset* para o português brasileiro, voltado para a linguagem gerada por usuários para o ambiente virtual: é uma grande coleção de publicações feitas em *blogs*, com 2,1 bilhão de palavras extraídas de 7,4 milhões de publicações em mais de 808 *blogs* brasileiros. Um diferencial desse *corpus* é a informação sobre autoria vinculada aos textos.

Os conjuntos de dados listados até aqui, são conjuntos mais clássicos, desenvolvidos ao longo das últimas décadas, que permaneceram relevantes até os dias de hoje. Com a revolução dos grandes modelos baseados na tecnologia Transformer, entretanto, cresceram também os recursos disponíveis para o português. Nesse sentido, diversos conjuntos têm sido desenvolvidos voltados especificamente para o treinamento desses tipos de modelos, entre as contribuições mais recentes destacamos:

- **Carolina** (Crespo et al., 2023; Finger et al., 2020): um *corpus* aberto do português contemporâneo (1970-2021) com informações de procedência e tipologia. A versão mais recente, 2.0 Bea, possui 2,1 milhões de documentos e 1,2 bilhões de *tokens*. O *corpus* busca construir uma base de dados a partir de critérios definidos previamente, com controle sobre as fontes, gêneros textuais, data de publicação e metadados associados. Falaremos do Carolina com um pouco mais de profundidade nas próximas seções.
- **Aroeira** (Lira et al., 2024): um *corpus* curado, que contém 35,5 milhões de documentos e mais de 15,1 bilhões de *tokens*, com foco na qualidade dos dados, a serem usados para a tarefa de treinamento de modelos;
- **GigaVerbo** (Corrêa et al., 2025): um *dataset* que contém 145 milhões de documentos e concatena uma grande quantidade de outros *corpora*, inclusive alguns menci-



onados ao longo deste texto. Foi desenvolvido especificamente para o treinamento dos modelos da Família Tucano;

- **LegalPT** (Garcia et al., 2024): é um agregador de *datasets* jurídicos disponíveis publicamente em português, com 24.194.918 documentos em sua versão atual, que se difere dos demais por essa especialização em documentos jurídicos. Vemos também, em decorrência dessa especialização, o esforço de endereçar a questão da repetição dos documentos em corpora jurídicos com a construção de sua versão deduplicada, o LegalPT (*deduplicated*).

2.4 A importância da curadoria de dados

Na seção anterior, apresentamos alguns dos principais conjuntos de dados disponíveis atualmente para o pré-treinamento em português. Dependendo da aplicação de interesse, um pesquisador pode adotar uma ou mais dessas fontes, ou desenvolver um conjunto de dados próprio, que pode incluí-las ou não. Em qualquer um desses casos, é importante refletir sobre o processo de curadoria de dados que serão usados, adaptando-o ao uso ou projeto específico. Vale, portanto, pararmos para discutir, mesmo que brevemente, como entendemos a curadoria de dados e qual a sua importância.

A construção de conjuntos de dados para o português é uma etapa essencial no desenvolvimento de tecnologias de processamento da linguagem, como discutido no Capítulo **Conjunto de dados, dataset e corpus**. No entanto, mais do que simplesmente acumular grandes volumes de texto, é fundamental considerar a curadoria desses dados. A promessa de uma curadoria cuidadosa é justamente a de garantir não apenas quantidade, mas também qualidade, diversidade e representatividade nos *corpora* utilizados para treinar e avaliar modelos.

Embora seja tecnicamente possível extrair textos em grande escala da internet, esse caminho apresenta uma série de desafios. A coleta bruta de dados traz consigo artefatos ruidosos (como *tags HTML* no meio do texto), duplicação, vieses negativos, problemas de direitos autorais e excessos ou lacunas linguísticas que podem comprometer a confiabilidade do conjunto final — o que, por sua vez, levanta questões sobre qualquer modelo treinado sobre ele.

Além disso, a ausência de filtros rigorosos pode fazer com que textos de baixa qualidade ou pouco relevantes ocupem espaço significativo no *corpus* resultante da coleta, prejudicando análises e aplicações práticas, além de acarretar maiores custos de desenvolvimento e manutenção.

Nos dias de hoje, entretanto, essa preocupação vai além dos problemas imediatos de qualidade e ganha contornos ainda mais críticos com descobertas recentes: um estudo publicado em julho 2024 na revista Nature (Shumailov et al., 2024) analisou os impactos de treinar ferramentas de Inteligência Artificial com dados gerados a partir delas. Conduzido por pesquisadores de instituições dos EUA, Reino Unido e Canadá, o trabalho mostrou que, após alguns ciclos de treinamento com esse tipo de dado, os modelos começam a cometer erros significativos e, em estágios mais avançados, passam a produzir informações incoerentes, num processo denominado “colapso” do modelo.

Segundo Ilia Shumailov (Shumailov et al., 2024), isso ocorre porque os dados não “compreendidos” pelo modelo original ficam sub-representados na linguagem gerada, então esse fenômeno de sub-representação vai se propagando, até que os erros se multiplicam a ponto de inutilizar os modelos. Esse risco ilustra a gravidade de se trabalhar com *corpora* não

controlados, já que qualquer aplicação que faça uso desses dados pode estar comprometida. Em especial, com a massificação do acesso a ferramentas de IA gerativa, torna-se cada vez mais difícil garantir que os textos coletados sejam de autoria exclusivamente humana — sobretudo aqueles produzidos a partir de 2022.

A curadoria atua, então, como um mecanismo de controle da qualidade dos dados: assegura a representatividade dos diferentes registros do português, reduz a presença de ruídos e vieses, e garante que os textos utilizados respeitem critérios éticos e metodológicos. Mais do que um simples repositório, o conjunto de dados se torna um recurso confiável para pesquisa e desenvolvimento em PLN.

Os fatores mencionados demonstram que, no cenário atual, a curadoria meticulosa não é um luxo, mas uma condição fundamental para o avanço da área. Nas seções seguintes nos aprofundamos em alguns desses aspectos.

2.5 O impacto da qualidade dos dados

Na Seção 2.2, discutimos como o pré-treinamento de um modelo de linguagem de ponta requer uma grande quantidade de dados textuais que, como visto na Seção 2.3, é muitas vezes proveniente da web. Sabemos que a web abriga conteúdos diversos, produzidos por diferentes públicos para diferentes fins e pode conter informações contraditórias ou desnecessárias para um determinado objetivo. Logo, no processo de preparação dos dados, é necessário garantir a qualidade dos textos que serão usados para treinar os modelos.

Mas o que é exatamente um texto de qualidade? Para os objetivos desta seção consideramos que um texto de qualidade é um texto escrito por um ser humano e que passou por um processo de seleção (Albalak et al., 2024). A seleção, por sua vez, costuma se orientar em três dimensões principais: a de **relevância**, a de **eficiência**, e a de **segurança**.

Essas dimensões de qualidade podem ser avaliadas tanto de forma **extrínseca** quanto **intrínseca**. A avaliação extrínseca mede o impacto dos dados no desempenho de um modelo já treinado, como, por exemplo, avaliar se um modelo treinado nesse *corpus* se torna mais propenso a gerar textos tóxicos. Já a avaliação intrínseca analisa características diretas do *corpus*; por exemplo, calcular a porcentagem de documentos duplicados (uma medida de eficiência) ou a frequência de termos de uma lista de bloqueio (uma medida de segurança). Nesta seção, focaremos apenas nas formas intrínsecas, que melhor se encaixam no escopo deste capítulo.

A primeira dimensão de qualidade é a da relevância. Um texto é dito relevante se ele está relacionado com a distribuição da tarefa a ser desempenhada pelo modelo, e se ele pode aumentar a sua capacidade. No contexto médico, uma lista de nomes de medicamentos pode apresentar termos que estão na distribuição desejada, porém ter somente o nome deles dificilmente aumentará a capacidade do modelo. Textos onde os nomes dos medicamentos aparecem acompanhados de uma descrição de efeitos colaterais são melhores já que apresentam mais informação relevante. Da mesma forma, um texto extraído da internet pode conter trechos de rodapé, de cabeçalho, de margens, ou de botões que não tem nada a ver com o conteúdo principal. E, antes de tudo isso, o texto está escrito em português?

Boa parte da adequação de idioma pode ser feita por meio dos metadados. Com eles é possível filtrar documentos cujo endereço pertença a um país lusófono, como feito no brWaC (Wagner Filho et al., 2018). De maneira alternativa ou complementar, classificadores automáticos podem ser utilizados (Albalak et al., 2024), como o FastText (Joulin et al., 2017), usado na construção do *corpus* Aroeira (Lira et al., 2024). Já para garantir



que não há ruído semântico, pode-se utilizar heurísticas (Albalak et al., 2024) como a presença de textos comuns em cabeçalhos e rodapés (ex: “Leia mais...”), a razão entre símbolos e palavras, o comprimento das sentenças e dos parágrafos, ou a repetição de *n-gramas* (Weber et al., 2024).

Além das heurísticas, a relevância semântica também pode ser avaliada com modelos. Uma abordagem comum é treinar um classificador para distinguir entre documentos de um *corpus* de referência de alta qualidade (com um “sinal positivo”) e documentos de um *corpus* mais ruidoso (com um “sinal negativo”) (Albalak et al., 2024; Wenzek et al., 2020). Outra alternativa é usar um modelo de linguagem para medir a perplexidade do texto (Touvron et al., 2023b), uma medida de quão surpreendente ele é, dado o conhecimento prévio de como os elementos linguísticos usualmente se distribuem dentro de um texto naquela língua. Nesse modelo, parágrafos com baixa perplexidade (menos surpreendentes) são considerados de maior qualidade (Wenzek et al., 2020).

Na segunda dimensão, a de eficiência, o objetivo é garantir que o conjunto dos dados torne o treinamento mais eficaz possível por meio do controle da redundância dos dados. Para isso costuma-se remover textos duplicados e trechos muito frequentes. Essa remoção faz com que o tempo de treinamento seja melhor usado, focando em conteúdo não-repetido (Lee et al., 2022) e, crucialmente, reduz a contaminação entre os conjuntos de treinamento e de teste (Longpre et al., 2025), o que permite uma avaliação mais robusta do modelo.

Há várias maneiras de remover conteúdo duplicado. Para uma remoção exata, o *hashing* pode ser utilizado para acelerar a computação (Wenzek et al., 2020). Também é possível utilizar o número de *n-gramas* em comum, e até mesmo a similaridade semântica calculada com modelos já treinados para remover documentos “quase idênticos” (Albalak et al., 2024; Lee et al., 2022).

Por fim, a terceira dimensão, a de segurança, tem como objetivo garantir que o modelo final siga os padrões de observância designados (Albalak et al., 2024; Longpre et al., 2025). A ideia é que um modelo é tão seguro quanto os dados utilizados para treiná-lo. As principais motivações estão em garantir que o conteúdo proteja a privacidade, siga padrões de observância, e que não seja tóxico. Aqui diremos que é tóxico qualquer conteúdo rude, desrespeitoso ou que contenha linguajar que faça uma pessoa desistir de um diálogo (Longpre et al., 2024, 2025; Rae et al., 2022). Já o conteúdo explícito pode incluir material pornográfico, violento, discriminatório, ou com viés de gênero e racial (Albalak et al., 2024; Lira et al., 2024).

As formas mais comuns para remover conteúdos indesejados consistem na utilização de listas de bloqueio de endereços web (fontes conhecidas por esse tipo de conteúdo) ou listas de bloqueio de palavras e expressões (Albalak et al., 2024). Esta última contém termos que, se presentes no texto, farão com que o documento inteiro seja descartado, como feito no T5 (Raffel et al., 2023) e no Aroeira (Lira et al., 2024). Como para as outras dimensões, também é possível utilizar aqui classificadores automáticos treinados para identificar toxicidade (Albalak et al., 2024). Já para garantir a privacidade, há ferramentas para remoção de nomes próprios, de endereço postal ou de e-mail, e de números de identificação pessoal.

2.5.1 O custo da qualidade

É importante ressaltar que a busca por qualidade dos dados não é absoluta e nem deve ser. Esta busca apresenta dois desafios centrais: a impossibilidade prática de uma filtragem perfeita, e o fato de que a remoção total de certos fenômenos pode ser indesejável para os



objetivos do modelo.

2.5.1.1 Quanto à impossibilidade

Em primeiro lugar, independente do nível de esforço dos desenvolvedores, todas as etapas de tratamento mencionadas utilizam modelos computacionais propensos a falhar, em especial quando estamos falando de um volume de textos tão imenso e diverso, que não pode ser verificado manualmente.

Pensando em heurísticas, um filtro que elimina textos com linhas muito curtas pode fazer com que textos como canções e poemas sejam descartados erroneamente (Abadji et al., 2022a; Albalak et al., 2024). Já listas de bloqueio de palavras podem fazer com que conteúdos legítimos de domínios como jurídico, médico ou produzido pela comunidade LGBTQIAPN+ sejam removidos por não levar em conta o contexto no qual as palavras aparecem (Kreutzer et al., 2022; Xu et al., 2021).

Os classificadores automáticos também podem cometer erros de classificação. Um modelo para reconhecer português, por exemplo, pode acabar descartando um texto regional caso não tenha sido treinado com dados da região em questão (Albalak et al., 2024), ou seja, se os textos apresentarem uma distribuição linguística distinta daquela dos dados de treinamento (Blodgett et al., 2020b).

2.5.1.2 Quanto à totalidade

Em segundo lugar, como discutimos anteriormente, quando estamos falando de pré-treinamento, estamos falando sobre ensinar a um modelo como a língua é usada, para que depois ele possa usar essa habilidade básica na resolução de tarefas específicas. Repetições e redundâncias são em algum nível parte da linguagem humana e, muitas vezes, nós mesmos usamos esse tipo de fenômeno para entender quais frases e expressões são mais comuns em determinados contextos e devem ser utilizadas em contextos similares. Há evidências de que manter um certo nível de duplicação é importante para o desempenho do modelo (Albalak et al., 2024; Lee et al., 2022).

De forma análoga, quando o objetivo final do modelo é uma tarefa específica, a remoção total de certos dados pode ser contraproducente. Se estivermos interessados em desenvolver um modelo que seja capaz de detectar textos tóxicos, a presença de amostras desse tipo de texto no pré-treinamento pode ser útil (Longpre et al., 2024, 2025). Isso permitiria ao modelo aprender os padrões de linguagem associados à toxicidade, que ele poderá usar posteriormente para aprender a detectar esse tipo de texto.

2.5.2 Considerações finais desta seção

Nesta seção, foram abordadas algumas dimensões referentes à qualidade dos dados. Em resumo, um dado de qualidade é aquele que é relevante, eficiente, e seguro, mas qual o nível de relevância, eficiência e segurança que podemos garantir pode variar bastante em função dos objetivos e das ferramentas que temos à disposição. No processo de falar sobre qualidade, acabamos tocando em tópicos relacionados a diversidade e a proveniência dos dados. Discutiremos esses aspectos nas próximas seções.



2.6 O impacto da diversidade dos dados

Além da qualidade dos dados e suas várias dimensões, que discutimos na seção anterior, outro aspecto importante de considerarmos ao selecionar *datasets* ou *corpora* para o pré-treinamento de modelos de linguagem é o da diversidade linguística.

Muitas vezes, quando falamos de diversidade linguística, a primeira coisa que nos vem à mente são as diferenças entre línguas, como a inglesa e a portuguesa, ou de forma mais ampla, entre famílias linguísticas distintas, como a indo-europeia — que contém as línguas inglesa e portuguesa — e a família afro-asiática — que inclui línguas como o hebraico e o egípcio.

Por conta dessa intuição, podemos tomar a preocupação com diversidade linguística como um problema restrito a modelos multilinguísticos, treinados com dados de diversas línguas simultaneamente.

De fato, essa é uma questão fundamental no desenho e construção de modelos multilinguísticos, entretanto também o é na construção dos modelos monolingüísticos, voltados para uma língua específica, como no caso dos modelos especializados em português.

Isso porque o que entendemos como a língua portuguesa é, na verdade, um conjunto bastante diverso de manifestações linguísticas distintas que — apesar de encontrarem coesão nas regras básicas do português — estão submetidas a padrões e convenções significativamente diferentes.

Essas variações da língua podem se dar sobre diferentes dimensões. É comum que as pessoas deem maior atenção às dimensões de variação que elas entendam como marcas que as distingam de outros grupos de usuários da língua. Esse é o caso de variações em função da distribuição geográfica, do grupo social e do período histórico (ver Wolfram (2006)).

Contudo, esse tipo de variação representa apenas uma parte do potencial de variação linguística de um idioma. Algumas dimensões de variação são mais dinâmicas, de forma que um único falante usualmente alterna entre variedades para fins distintos: esse é o caso da variação em **registro**, **gênero** e **estilo**, que definimos de acordo com Biber; Finegan (1994) e Biber; Conrad (2009):

- **Registro:** variação segundo a situação em que a linguagem é produzida, refletindo situações comunicativas que se repetem de forma regular dentro de uma estrutura social (Exemplos são: registro da conversação pessoal, registro da comunicação em fóruns virtuais, registro da escrita em livros didáticos).
- **Gênero:** variação segundo o tipo de mensagem e as convenções associadas à como aquele tipo de mensagem é construído (Exemplos são: entrada de enciclopédia, carta, manual, receita culinária.)
- **Estilo:** variação segundo as preferências estéticas de quem produz, ou consome a linguagem, podendo essas preferências serem mais ou menos coletivamente alinhadas (Exemplos são: o estilo da terceira fase do romantismo, o estilo pessoal de Machado de Assis)

Quando olhamos com atenção, vemos que uma pessoa usa a linguagem de forma bem diferente quando no tribunal, ou quando conversando com os amigos num bar (variação em registro); quando está escrevendo uma carta formal ou uma receita de bolo (variação em gênero); quando procura escrever um poema árcade ou cubista (variação em estilo).



O mais interessante é que, em todos esses casos, a linguagem não apenas varia de forma dinâmica, mas exerce um papel contextual. Basta percebermos como usar a linguagem apropriada para um dos contextos mencionados em outro desses contextos causaria estranhamento.

Do mesmo modo que um ser humano precisa ser capaz de alternar entre essas diversas variedades, de forma a cumprir tarefas de naturezas distintas com as quais ele se depara no seu dia a dia, um modelo de linguagem — que tenha o propósito geral de auxiliar os seres humanos nessas mesmas tarefas corriqueiras — precisa, também ele, apresentar essa plasticidade linguística, e utilizar-se da capacidade de reconhecer e produzir a variedade da linguagem apropriada para o contexto em que ela será usada, com funções, convenções e marcadores estéticos apropriados.

O leitor pode, entretanto, concordar conosco em nosso argumento sobre a importância de expor um modelo de linguagem a esses níveis de variação linguística, mas se perguntar se o pré-treinamento é a fase apropriada para tal. Afinal, o pré-treinamento é o momento em que o modelo deveria capturar as estruturas linguísticas gerais do idioma e, para realizar tarefas específicas — como por exemplo, redigir uma petição jurídica — ele passará, posteriormente, por um ajuste fino específico para essa tarefa.

De fato, para a realização de uma tarefa-alvo, os modelos precisam passar por essa etapa de ajuste fino com dados específicos para aquela tarefa, como discutimos nas seções anteriores. E, quando falamos de tarefas voltadas para contextos bastante particulares — como é o caso do exemplo mencionado da escrita da petição jurídica — é de se esperar que o ajuste fino também forneça um direcionamento ao modelo sobre o uso específico da linguagem para a realização daquela tarefa.

Contudo, essa etapa de ajuste fino dificilmente será suficiente para a adequação da linguagem, sem que tenha havido representação dessa forma de usar a língua durante o pré-treinamento. Isso porque, as diferenças entre tais variedades intralinguísticas podem ser bastante sutis e pervasivas.

Sardinha et al. (2014) realizaram uma análise da variação da língua portuguesa em nível de registro, sobre um conjunto diverso de propriedades gramaticais anotadas de forma semi-automática, baseados na metodologia de Biber (1988). Sua análise identificou que os registros no português variam ao longo de seis dimensões funcionais distintas, caracterizadas por propriedades como uso de pronomes, preposições e artigos, classes verbais, nominalização do sujeito, entre outras.

Já Serras et al. (2024) investigaram a diferenciabilidade de domínios discursivos do português brasileiro, realizando uma análise estatística individual sobre as distribuições de várias propriedades, incluindo propriedades morfológicas, sintáticas e semânticas. Eles observaram que, apesar de alto grau de sobreposição, várias das propriedades estudadas eram estatisticamente distintivas entre os diferentes domínios discursivos analisados.

Resultados como esses são testemunhos empíricos de que a língua portuguesa não é um corpo homogêneo, mas sim, um conjunto de manifestações distintas com funções e histórias específicas, que se sobrepõem em diferentes graus para constituir o que entendemos como o nosso idioma. Essa heterogeneidade é muitas vezes sutil, mas transversal a diferentes níveis de análise, estando entranhada em diversas propriedades que definem a estrutura gramatical da língua.

Se o pré-treinamento é a etapa em que o modelo cria uma representação interna da língua e da gramática que a rege, é sim fundamental que essa etapa inclua um nível de diversidade linguística apropriado. É isso que permitirá encapsular nessa representação os elementos das várias sublínguas que compõem o idioma a ser dominado, elementos esses



que estão entranhados em vários níveis da gramática da língua e que poderão ser posteriormente aproveitados pelo modelo na construção de habilidades específicas para contextos particulares.

2.7 A questão da proveniência dos dados

Proveniência, é o processo de rastreamento dos dados, ou seja, de suas origens e histórico de processamento¹. Muitas vezes, em grandes *corpora*, é difícil de identificar de qual fonte um texto foi retirado, principalmente se os dados foram extraídos da web e processados em larga escala. Isso pode não só gerar um descumprimento acidental de direitos autorais dos textos originais, mas também uma dificuldade de manipulação desses dados por parte dos usuários. Ou seja, se por exemplo alguém não quiser utilizar textos de algum *website* específico ou tenha detectado um viés no modelo treinado, é mais difícil encontrar e retirar essas fontes indesejadas do conjunto textual caso não haja dados de proveniência marcados no *corpus*.

A questão dos direitos autorais e do respeito às licenças dos textos tem se tornado cada vez mais importante nos últimos anos, em especial depois da emergência de *chatbots* baseados em LLMs tão eficientes, como o ChatGPT, o Gemini e outras alternativas. Se antes os frutos desse desenvolvimento tecnológico tinham um papel mais restrito e indireto na vida das pessoas, eles passaram a estar presentes no dia-a-dia, gerar grandes receitas e ameaçar a substituição de muitos ofícios.

Essa possibilidade gerou um sentimento geral e justo de questionamento acerca dos impactos dessas tecnologias e da adequação da coleta de dados para seu desenvolvimento e treinamento. As questões envolvendo autoria de dados gerados por inteligência artificial e o uso adequado dos dados de treinamento invadiram tanto os tribunais como as mesas de bar.

É de se esperar que nos próximos anos, a sociedade encontre uma solução para esses debates. Até lá, o cuidado com a proveniência dos dados deve ser redobrado, tanto no sentido de garantir que os dados utilizados no treinamento tenham uma licença apropriada para a utilização a ser feita do modelo, como que sua fonte e licença sejam conhecidas, caso seja necessário remover esses dados no futuro, em função da evolução do nosso entendimento coletivo acerca de como tais dados podem ou não ser utilizados.

Um outro debate importante é o da licença dos próprios modelos produzidos a partir desses dados. Apesar de restrições diversas, os desenvolvedores de um modelo para fins de pesquisa usualmente têm a oportunidade de escolher a licença sob a qual o modelo será distribuído. Embora um debate nesse sentido esteja fora do escopo deste capítulo, recomendamos que os desenvolvedores de tais modelos pensem nessa questão com esmero, e sempre que possível, priorizem licenças de uso abertas que, ao mesmo tempo, visem coibir quaisquer usos nocivos.

Além disso, outra questão importante que vem à tona com o uso de textos da web para treinamento de modelos é o problema relacionado a dados pessoais e sensíveis e como tratá-los. Essa questão se tornou ainda mais relevante com o advento da Lei Geral de Proteção de Dados Pessoais (LGPD)², que busca justamente regulamentar o tratamento desses dados. Apesar de não conseguirmos abordar com profundidade essa discussão neste

¹Tratamos com mais detalhes a respeito de proveniência, direitos autorais e métodos de extração de textos da web em (Crespo et al., 2023).

²O texto da lei pode ser conferido em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm



capítulo³, vale destacar que a proveniência também se torna relevante nestas questões, pois sabendo de onde os dados vêm, é mais fácil identificar quais são os processos que precisam ser realizados para garantir a anonimização e remoção de dados pessoais dos textos.

2.8 O *corpus* Carolina como um estudo de caso

Nas seções anteriores, apresentamos diversos aspectos a serem considerados na construção de um conjunto de dados visando o pré-treinamento de modelos de linguagem. Gostaríamos de encerrar o capítulo fazendo uma breve apresentação do *corpus* Carolina. Esse é um *corpus* dual, construído visando tanto o treinamento de modelos de linguagem quanto a pesquisa linguística. Ele é uma tentativa destes autores e de seus colaboradores de conjugar vários dos aspectos discutidos neste capítulo na construção de um bom conjunto de dados. Essa não é uma empreitada fácil, já que muitas dessas restrições e objetivos competem entre si — vide como pode ser difícil garantir diversidade linguística e se ater a dados de licença aberta ao mesmo tempo — entretanto, acreditamos que o Carolina tem se tornado tanto um recurso útil por si só, como um protótipo para a construção de futuros recursos semelhantes.

O *corpus* Carolina⁴ (Córpus Geral do Português Brasileiro com Informações de Proveniência e Tipologia) é um *corpus* aberto, desenvolvido por uma equipe multidisciplinar de Linguistas e Cientistas da Computação, que faz parte do desafio de NLP2⁵ do C4AI (Center for Artificial Intelligence)⁶. O Carolina está em construção contínua desde setembro de 2020, sendo lançadas anualmente novas versões com melhorias desde março de 2022, data da publicação da versão 1.0 (Ada). A versão atual, 2.0 (Bea), é composta por mais de 2 milhões de textos e cerca de 1,3 bilhões de *tokens*. Todos os textos do *corpus* foram extraídos da web e são posteriores a 1970.

O *corpus* Carolina foi criado com o intuito de preservar informações de proveniência e de tipologia textual. Para alcançar esse objetivo, apresentamos um cabeçalho com metadados anotados em lotes de textos⁷, que conservam o máximo de informações que podemos obter de nossas fontes⁸ e seguem o padrão TEI (*Text Encoding Initiative*) (TEI Consortium, 2021).

Assim, informações como a licença e URL da fonte são obrigatoriamente anotadas de modo a garantir a rastreabilidade dos textos originais, bem como a conformidade com os direitos autorais de cada um. Na Figura 2.1 a tag “<license>” contém as informações da licença da fonte e “<media>” a URL do texto.

Também anotamos informações das duas tipologias textuais do *corpus*, a Tipologia Ampla, ou Tipologia do Carolina, que organiza os textos dentro da estrutura do *corpus* e a Tipologia da Fonte, que contém os termos utilizados pelas próprias fontes para se referirem aos seus textos. Além disso, anotamos o Domínio de cada texto, que entendemos como a esfera em que circulam (por exemplo: Acadêmico, Pedagógico, Comercial etc.). Na

³Para uma discussão mais ampla a respeito da LGPD e a Inteligência Artificial cf. (Lazzarini Moretti; Maltese Zuffo, 2025).

⁴O nome do *corpus* é uma homenagem a Carolina Michaelis de Vasconcelos (1851-1925), filóloga e linguista alemã, radicada em Portugal, e primeira mulher a atuar como docente na Faculdade de Letras da Universidade de Lisboa, em 1911.

⁵https://dev-c4ai.webhostusp.sti.usp.br/?page_id=6120&lang=pt

⁶<https://sites.google.com/icmc.usp.br/nlp2>

⁷Para mais informações sobre a metodologia de extração e anotação dos dados do *corpus* Carolina cf. (Sturzeneker et al., 2022).

⁸A lista completa de metadados que anotamos encontra-se em (Crespo et al., 2023, p. 13–14).



Figura 2.1: Alguns metadados contendo informações de proveniência textual.

```
<sourceDesc>
  <biblFull>
    <fileDesc>
      <titleStmt>
        <title>
          <name>Discussão:Mutirão:Página principal</name>
          <media mimeType="text/xml" url="https://br.wikimedia.org/wiki?curid=13" source="brwikimedia-20210701-pages-meta-current.xml" />
        </title>
        <author />
        <editor role="translator" />
        <sponsor>Unknown</sponsor>
      </titleStmt>
      <extent>
        <measure unit="bytes" quantity="14471932" />
        <measure unit="tokens" quantity="1586470" />
        <measure unit="pages" quantity="1" />
      </extent>
      <publicationStmt>
        <publisher />
        <authority>Wikimedia Foundation</authority>
        <date>2014-09-16</date>
        <availability status="free">
          <license target="https://creativecommons.org/licenses/by-sa/3.0/deed.pt_BR">Creative Commons - Atribuição - Compartilha Igual 3.0 Não Adaptada (CC BY-SA 3.0)</license>
        </availability>
      </publicationStmt>
    </fileDesc>
  </biblFull>
</sourceDesc>
```

Figura 2.2 as tags “<catRef>” contêm os tipos amplos e da fonte e “<domain>” apresenta o domínio. Esses metadados nos fornecem informações importantes acerca do registro dos textos que compõem o Carolina e podem ser usados para mensurar sua diversidade linguística.

Figura 2.2: Alguns metadados contendo informações de tipologia textual.

```
<profileDesc>
  <creation resp="user" />
  <textDesc>
    <channel mode="w" />
    <constitution type="single" />
    <derivation />
    <domain>Virtual Forum</domain>
    <factuality />
    <interaction />
    <preparedness type="monitored" />
    <purpose />
  </textDesc>
  <textClass>
    <catRef scheme="#Source_typology" target="#DISCUSSION_VIR_W" />
  </textClass>
  <langUsage>
    <language ident="pt-BR" />
  </langUsage>
</profileDesc>
</biblFull>
</sourceDesc>
</fileDesc>
<profileDesc>
  <textClass>
    <catRef scheme="#Carolina_typology" target="#WIKIS" />
  </textClass>
</profileDesc>
```

2.9 Considerações finais

Neste capítulo, apresentamos uma discussão acerca do papel dos dados no pré-treinamento de grandes modelos de linguagem. Discutimos as situações em que o pré-treinamento se



faz necessário, alguns dos principais conjuntos de dados disponíveis para o pré-treinamento em português e os principais aspectos que alguém deve considerar tanto ao selecionar um conjunto de dados, quanto ao construir um conjunto de dados para pré-treinamento.

Esperamos que os pontos apresentados sejam úteis aos leitores que estejam interessados em desbravar essa árdua tarefa do pré-treinamento dos grandes modelos de linguagem. Se em algum momento você obtiver um modelo cheio de alucinações, apesar de todo o seu esforço na seleção de dados, não perca a esperança: lembre-se que até Dom Quixote acabou são no final.



Capítulo 3

Treinamento de Grandes Modelos de Linguagem na prática

*João Vitor Mariano Correia
João Renato Ribeiro Manesco
Danilo Samuel Jodas
Douglas Rodrigues
Gabriel Lino Garcia
Pedro Henrique Paiola
João Paulo Papa*

Publicado em: 16/04/2026

3.1 Introdução

Grandes Modelos de Linguagem (LLMs) representam uma mudança de paradigma no Processamento de Linguagem Natural, impulsionada pela arquitetura de *Transformers* (Vaswani et al., 2017b), que permitiu um avanço significativo na capacidade dos modelos de ponderar contexto. Trabalhos pioneiros como o BERT (Devlin et al., 2019), com sua abordagem de pré-treinamento bidirecional, validaram a eficácia de treinar modelos massivos em vastos conjuntos textuais, dando um passo fundamental para a compreensão de linguagem via redes neurais. O resultado é uma série de modelos com uma capacidade sem precedentes de processar e gerar texto de forma fluente e coerente.

Essa fundação de entendimento de linguagem é adquirida durante a fase de pré-treinamento, em que o modelo absorve conhecimento de fontes como a internet e grandes acervos de livros, permitindo que diversas aplicações de ampla relevância possam surgir, tais como sistemas já discutidos anteriormente nesta edição, como modelos de linguagem introduzidos no Capítulo 1, tradução automática (Capítulo [Tradução Automática](#)), sumarização automática (Capítulo [Sumarização Automática](#)), abrangendo diferentes contextos de uso (Minaee et al., 2024).

Apesar da eficácia dos LLM, sua aplicabilidade em cenários do mundo real depende criticamente de sua especialização, em especial tratando-se de modelos menores. A base generalista, embora ampla, é insuficiente para tarefas que exigem mais precisão ou conhecimento de nicho, tornando o processo de adaptação uma necessidade para determinados contextos (Anisuzzaman et al., 2025). Essa necessidade de adaptação pode ser analisada a partir de duas dimensões principais e complementares: a especialização para domínios de conhecimento e o alinhamento para contextos linguísticos e culturais específicos.

A primeira dimensão busca fazer com que o corpo de conhecimento dos modelos de linguagem seja composto por termos linguísticos e conhecimentos específicos de um campo nichado, facilitando sua aplicação em tarefas direcionadas, sendo crucial para aplicações



de alto impacto. No campo jurídico, por exemplo, um modelo refinado é capaz de interagir mais precisamente com o vernáculo próprio de documentos legais para realizar sumarizações ou revisões (Lai et al., 2024). Na medicina, a especialização permite que um modelo interprete literatura científica e registros de saúde para que seja capaz de apoiar decisões clínicas (Zhou et al., 2023b). Da mesma forma, no setor financeiro, um modelo ajustado pode identificar nuances em relatórios de mercado que seriam invisíveis a um sistema generalista (Lee et al., 2024). Nesses contextos, o refinamento é o que distingue uma ferramenta funcional de um sistema verdadeiramente confiável e eficaz.

A segunda dimensão busca realizar a adaptação linguística e cultural dos modelos, um desafio fundamental na construção de modelos equitativos. A grande maioria dos dados utilizados no pré-treinamento de modelos publicados abertamente está em inglês, o que cria um desequilíbrio cultural e representativo a nível de linguagem nesses modelos (AlKhamissi et al., 2024). Desta forma, modelos pouco expostos à língua portuguesa sofrem dificuldade em entender certas nuances linguísticas, bem como podem estar expostos a certos vieses culturais dos dados, como é o caso de modelos médicos que ignoram a possibilidade de diagnóstico de certas doenças endêmicas como a dengue pelos dados de treinamento serem de locais não expostos a este problema (Paiola et al., 2024).

Para que a tecnologia seja efetivamente democrática e útil em contextos não anglófonos, como o brasileiro, a tradução da resposta nem sempre é a solução. Exigindo um processo de ajuste fino ou continuação de pré-treino com dados no idioma local. Garantindo que o modelo compreenda as particularidades culturais, expressões idiomáticas e o contexto social do idioma refinado.

Atingir um alto nível de robustez com LLMs, no entanto, requer um processo que vai além do treinamento em escala. É necessária uma estratégia de adaptação cuidadosa, que considere tanto a profundidade técnica de um domínio quanto a riqueza de um contexto linguístico. Para abordar essa necessidade, este capítulo irá explorar o espectro de técnicas de especialização utilizadas no treinamento de modelos de linguagem. Iniciaremos discutindo a continuação de pré-treino, seguiremos para o ajuste fino supervisionado em suas diversas formas, incluindo técnicas de parametrização eficiente e aprendizado *few-shot*, e concluiremos com métodos de alinhamento, que refinam o comportamento do modelo com base em preferências humanas. Ao final do capítulo, apresentamos uma implementação prática do refinamento de modelos.

3.2 Continuação de pré-treino

A Continuação de Pré-treino (CPT) consiste em uma técnica de adaptação realizada antes do ajuste fino supervisionado, visando ampliar o conhecimento de um LLM em um domínio específico. Essa etapa intermediária entre o pré-treinamento generalista e o *fine-tuning* retoma o treinamento original do modelo, de forma autossupervisionada, através de técnicas como a Modelagem de Linguagem Mascarada (MLM) ou Causal (CLM), aplicadas em um *corpus* especializado de grande escala. Esta estratégia resulta na incorporação de vocabulário, conceitos e fatos do domínio desejado. Neste capítulo serão apresentados os fundamentos técnicos, cenários de aplicação e distinções relevantes dessa abordagem.

3.2.1 Mecanismos de treinamento autossupervisionado

Para modelos autorregressivos, como os das famílias GPT e Qwen, a técnica predominante é a de Modelagem de Linguagem Causal (CLM) (Bai et al., 2023; Radford et al., 2019b),



cujo objetivo é treinar o modelo para prever o próximo *token* em uma sequência com base nos *tokens* anteriores. De forma mais precisa, a probabilidade de uma sequência de *tokens* é decomposta em uma cadeia de probabilidades condicionais, onde cada *token* é previsto condicionando-se nos *tokens* que o precedem.

O objetivo da CLM é maximizar essa probabilidade nos textos observados, o que equivale a minimizar a soma dos erros associados a cada previsão individual, ou seja, a entropia cruzada entre as distribuições preditas e os *tokens* reais. A natureza estritamente unidirecional, da esquerda para a direita, do CLM flui naturalmente com a arquitetura do tipo *decoder* em arquiteturas de *transformers*, que empregam máscaras nos mecanismos de atenção. Essas máscaras garantem que, ao prever um *token* em uma determinada posição, o modelo utilize apenas informações dos *tokens* que o antecedem na sequência, e não dos posteriores.

Em contrapartida, arquiteturas bidirecionais como o BERT utilizam a Modelagem de Linguagem Mascarada (MLM) (Devlin et al., 2019), cujo funcionamento, ao invés de prever o próximo *token* de uma sequência, trabalha corrompendo uma fração dos *tokens* no texto de entrada, criando uma espécie de “máscara” no texto, desta forma, o modelo deve prever quais os *tokens* originais que deveriam compor a sequência. Nesse processo, para uma sequência de entrada T , um subconjunto de posições M é selecionado para mascaramento. Para cada posição $i \in M$, um *token* t_i pode ser substituído pelo símbolo de máscara, por um *token* aleatório ou mantido inalterado, de modo a incentivar o modelo a aprender representações contextuais mais robustas. A função de perda é calculada apenas sobre os *tokens* mascarados, já que o modelo se concentra em prever esses elementos, e como a predição é condicionada à sequência inteira corrompida, o modelo aprende representações contextuais, considerando tanto o contexto à esquerda quanto à direita.

Outra estratégia, chamada de Previsão da Próxima Sentença (NSP), introduzida no pré-treinamento do BERT, consiste em classificar se uma sentença sucede outra no texto. Diferente da Modelagem CLM, que realiza predição *token a token* de forma unidirecional, o NSP busca classificar sentenças completas. Essa técnica, no entanto, caiu em desuso no estado da arte, devido a estudos indicarem benefício limitado, e foi descartada em modelos recentes (Liu et al., 2019b).

3.2.2 Contextos e objetivos da continuação de pré-treino

A aplicação da continuação de pré-treino é uma decisão estratégica que visa aprimorar o modelo de maneiras que o ajuste fino supervisionado por si só não consegue alcançar. O principal desses cenários é a adaptação para aplicações específicas, onde um LLM generalista é infundido com o conhecimento de um campo técnico (Shi et al., 2024). Um exemplo prático seria treinar um LLM com uma biblioteca de artigos jurídicos, de tal forma que ele se torne fluente no jargão e nos conceitos do direito, ou até mesmo em conhecimento médico, como será trabalhado ao longo deste capítulo. Outro uso importante é a adaptação temporal, que atualiza o conhecimento do modelo para além de sua data de corte, utilizando um conjunto de textos e notícias recentes. Adicionalmente, a CPT é útil para adaptações de idioma, melhorando a proficiência de um modelo em uma língua menos representada nos dados de treinamento originais, como o português, aprimorando sua fluência e compreensão de nuances culturais.

É importante reforçar a distinção entre a CPT e o Ajuste Fino Supervisionado, já que a CPT tem o objetivo de expandir a base de conhecimento geral de um modelo, através da utilização de dados não estruturados. Já o ajuste fino busca impor um comportamento



específico ao modelo, utilizando dados rotulados preparados, para uma ou mais tarefas específicas. A CPT opera sobre conjuntos massivos de texto bruto, o resultado é um modelo com vocabulário e conhecimento de domínio aprofundados, mas não necessariamente com os detalhes específicos das tarefas em que deve atuar.

3.3 Ajuste fino de modelos

Conforme explorado na seção anterior, na fase de pré-treinamento os Grandes Modelos de Linguagem adquirem um vasto conhecimento de mundo e uma profunda compreensão de padrões linguísticos, adquiridos a partir da exposição a *corpora* textuais massivos. Essa base de conhecimento confere aos modelos a capacidade de realizar uma gama de tarefas de forma emergente, sem treinamento explícito. Contudo, para alcançar excelência e confiabilidade em aplicações especializadas, é necessário um processo de adaptação direcionado. Este processo é conhecido como ajuste fino (do inglês, *fine-tuning*).

O ajuste fino representa uma importante mudança de paradigma na história do Processamento de Linguagem Natural. Ele pode ser definido como a especialização de um modelo generalista, transformando-o em uma ferramenta especialista capaz de executar tarefas com alta precisão e adaptar-se a domínios de conhecimento específicos (Wu et al., 2025). A intuição por trás desta técnica pode ser ilustrada por uma analogia com a formação médica: o pré-treinamento é análogo aos anos de faculdade, onde se adquire um conhecimento abrangente. O ajuste fino, por sua vez, equivale à residência médica, na qual o profissional aprofunda seu conhecimento em uma área específica, como a cardiologia, para diagnosticar e tratar com uma proficiência que um clínico geral não possui. De forma similar, o ajuste fino não ensina o LLM a gerar linguagem do zero; ele refina suas competências preexistentes para uma nova aplicação.

Formalmente, a técnica se fundamenta no princípio da aprendizagem por transferência (do inglês, *transfer learning*), no qual um modelo desenvolvido para uma tarefa inicial é reutilizado como ponto de partida para treinar outro modelo em uma segunda tarefa, relacionada à primeira. O conhecimento capturado durante o pré-treinamento em um domínio de origem e tarefa de origem é transferido para melhorar o desempenho e acelerar a aprendizagem em um domínio de destino e uma tarefa de destino. Essa abordagem mostrou-se fundamental para superar um dos maiores desafios históricos da área: a necessidade de vastos conjuntos de dados rotulados e o custo computacional proibitivo do treinamento de modelos de ponta a partir do zero (Howard; Ruder, 2018b).

O principal método para essa especialização é o Ajuste Fino Supervisionado (do inglês, *Supervised Fine-Tuning* ou SFT) (Devlin et al., 2019). Neste processo, o modelo pré-treinado tem seu treinamento estendido, mas desta vez em um conjunto de dados menor e de alta qualidade, composto por exemplos rotulados que demonstram o comportamento desejado — geralmente em um formato de pares de entrada e saída (por exemplo, no caso da tarefa de *Question Answering* (QA), o par **pergunta -> resposta esperada**). O mecanismo é uma continuação direta do treinamento de redes neurais:

1. **Seleção do modelo base:** A escolha de um modelo pré-treinado apropriado é o primeiro passo. A decisão deve levar em conta a arquitetura, o tamanho e a afinidade do domínio de pré-treinamento com a tarefa final.
2. **Preparação do *dataset*:** A qualidade do ajuste é diretamente proporcional à qualidade do conjunto de dados. Esta etapa envolve a coleta, limpeza e formatação de



dados rotulados e representativos da tarefa, geralmente estruturados em pares de *prompt*-resposta.

3. **Ajuste de hiperparâmetros:** A seleção de hiperparâmetros como taxa de aprendizado, tamanho do lote (do inglês, *batch size*) e número de épocas é crucial. Um ajuste inadequado pode levar a um treinamento instável ou a um sobreajuste.
4. **Treinamento e validação:** O modelo é treinado em lotes de dados, e seu desempenho é continuamente monitorado em um conjunto de validação separado. O monitoramento da perda de validação é essencial para detectar sobreajuste e aplicar estratégias como a parada antecipada, que interrompe o treinamento quando a generalização para dados não vistos começa a piorar.

Este processo iterativo refina gradualmente o conhecimento do modelo, especializando-o para as nuances da tarefa alvo. Neste ponto, é crucial diferenciar o ajuste fino de outras técnicas de adaptação:

- **Continuação de pré-treino:** Enquanto a continuação de pré-treino visa expandir a base de conhecimento geral do modelo em um novo domínio (ex: alimentar um LLM com uma biblioteca de artigos jurídicos), utilizando dados não estruturados, o ajuste fino visa ensinar uma habilidade ou comportamento específico (ex: ensinar o modelo a sumarizar processos judiciais), utilizando dados rotulados.
- **Engenharia de *prompts*:** Esta técnica não modifica nenhum dos parâmetros do modelo. Em vez disso, ela guia o comportamento do modelo no momento da inferência, formulando cuidadosamente a entrada (*prompt*) para induzir a resposta desejada. Métodos como *one-shot* ou *few-shot learning*, onde um ou alguns exemplos da tarefa são incluídos no *prompt*, são formas de engenharia de *prompts*. É uma forma de adaptação sem treinamento, que depende inteiramente das capacidades de aprendizado no contexto do modelo base.

Em suma, o ajuste fino representa um caminho poderoso para a especialização de LLMs, capacitando-os a ir além da fluência geral para entregar desempenho de ponta em aplicações de nicho. Contudo, a estratégia pela qual essa adaptação de parâmetros é realizada varia significativamente, levando a um espectro de métodos com diferentes *trade-offs* entre desempenho, custo computacional e complexidade. Nas subseções a seguir, exploraremos as principais abordagens práticas: desde o ajuste fino completo, que modifica todos os parâmetros do modelo, passando pelas eficientes técnicas de PEFT (do inglês, *Parameter-Efficient Fine-Tuning*), que ajustam apenas um pequeno subconjunto de parâmetros, até as estratégias de *Few-Shot Learning*, que operam na fronteira entre o ajuste e a engenharia de *prompts*. As seções seguintes exploram técnicas avançadas como *Reinforcement Learning from Human Feedback* (RLHF) e *Direct Preference Optimization* (DPO), que estendem o SFT para alinhamento com preferências humanas.

3.3.1 Ajuste fino completo

O ajuste fino completo (do inglês, *Full Fine-Tuning* ou FFT) representa a forma mais direta e conceitualmente mais simples de especialização de um modelo de linguagem. Nesta abordagem, a totalidade dos parâmetros do modelo pré-treinado é descongelada e se torna treinável. Isso concede ao modelo máxima flexibilidade para que todo o seu conhecimento



generalista, encapsulado em bilhões de parâmetros, seja reconfigurado para se adaptar profundamente aos padrões do novo conjunto de dados (Lv et al., 2024; Raffel et al., 2020b).

A principal vantagem desta abordagem é seu potencial para alcançar o máximo desempenho possível na tarefa alvo. Ao permitir que todas as camadas da rede neural sejam atualizadas, desde as representações iniciais de *embeddings* até os mecanismos de atenção e as camadas *feed-forward*, o modelo pode aprender nuances extremamente complexas e específicas do domínio, algo que abordagens mais restritivas poderiam não capturar. No entanto, este potencial de desempenho vem acompanhado de custos e desafios significativos, que devem ser cuidadosamente ponderados:

- **Alto custo computacional e de memória:** O FFT é extremamente exigente em termos de recursos. Treinar todos os parâmetros de um modelo com bilhões de pesos requer acesso a múltiplas GPUs de ponta com alta capacidade de VRAM, não apenas para armazenar os pesos do modelo, mas também os gradientes, os estados do otimizador e as ativações intermediárias durante o treinamento. Isso torna o processo financeiramente proibitivo para muitas organizações.
- **Custo de armazenamento e implantação:** Como o FFT modifica todos os pesos, cada nova tarefa especializada resulta em uma cópia completa e independente do modelo, que pode ocupar dezenas ou centenas de gigabytes. Gerenciar, armazenar e servir múltiplas instâncias de modelos tão grandes em um ambiente de produção é logisticamente complexo e caro.
- **Esquecimento catastrófico:** Este é um dos riscos mais proeminentes. Ao se especializar intensamente na nova tarefa, o modelo pode sobrescrever e perder drasticamente as capacidades gerais adquiridas durante o pré-treinamento (Howard; Ruder, 2018b). Por exemplo, um modelo ajustado para gerar código em Python pode esquecer como manter um diálogo coerente sobre história.
- **Risco de sobreajuste:** Especialmente quando o conjunto de dados de ajuste fino é pequeno ou pouco diverso, o modelo pode memorizar os exemplos de treinamento em vez de aprender a generalizar o padrão subjacente. Isso resulta em um excelente desempenho nos dados de validação, mas um baixo desempenho em dados novos e não vistos.
- **Amplificação de viés:** Se o conjunto de dados utilizado para o ajuste fino contiver vieses sociais ou demográficos, o processo de FFT pode não apenas replicá-los, mas também reforçá-los, resultando em um modelo final ainda mais enviesado do que sua versão pré-treinada. Por exemplo, um modelo pode ser induzido a associar automaticamente certas profissões a um gênero, reforçando estereótipos sociais.

Apesar de seus desafios, o FFT permanece uma ferramenta relevante e poderosa para cenários específicos. A decisão de utilizá-lo deve ser pragmática, considerando as seguintes condições:

1. **Quando o desempenho máximo é crítico:** Em aplicações de alto risco, como diagnóstico médico ou análise financeira, onde cada ponto percentual de acurácia é crucial e o orçamento permite o investimento.



2. **Quando há uma mudança substancial de domínio:** Se a tarefa alvo for radicalmente diferente dos dados de pré-treinamento, uma adaptação profunda pode ser necessária.
3. **Quando recursos e dados são abundantes:** O FFT só deve ser considerado quando se tem acesso a recursos computacionais massivos e, fundamentalmente, a um conjunto de dados de ajuste fino que seja grande, diverso e de altíssima qualidade para mitigar os riscos.

Em conclusão, o ajuste fino completo é uma faca de dois gumes. Ele oferece o caminho para o mais alto nível de especialização, mas a um custo extremo e com riscos significativos de degradar propriedades valiosas do modelo original. Para a grande maioria dos casos de uso práticos, o balanço entre desempenho, custo e risco pende favoravelmente para abordagens mais eficientes. Essas desvantagens impulsionaram a pesquisa e o desenvolvimento de uma nova família de técnicas, conhecidas como ajuste eficiente de parâmetros (do inglês, *Parameter-Efficient Fine-Tuning* ou PEFT).

3.3.2 PEFT

O desenvolvimento de metodologias voltadas à redução do custo computacional associado a LLMs tornou-se um requisito crítico para sua adaptação a tarefas especializadas. Essa demanda torna-se relevante no contexto do *fine-tuning* sobre conjuntos de dados de domínio específico, onde o custo computacional impacta diretamente a viabilidade prática da aplicação. Nesse cenário, busca-se empregar estratégias de treinamento que priorizem a atualização seletiva de camadas ou módulos do modelo, ao mesmo tempo minimizando o número de parâmetros ajustados. Tal abordagem tem o objetivo de não apenas reduzir o consumo de recursos e o tempo de processamento, mas também preservar, ou até aprimorar, a acurácia de predição visando garantir que a especialização do modelo não comprometa seu desempenho geral.

Neste contexto, surge uma abordagem denominada *Parameter-Efficient Fine-Tuning* (PEFT) (Ding et al., 2023), cujo objetivo é oferecer um conjunto de técnicas para adaptar grandes modelos de linguagem previamente pré-treinados, como o LLaMA (Gajulamandam et al., 2025) ou BERT (Li, 2024), para tarefas mais específicas. Em vez de re-treinar integralmente o modelo, que pode conter bilhões de parâmetros e exigir recursos computacionais significativos, o PEFT se concentra em ajustar apenas uma pequena fração desses parâmetros. O objetivo primário consiste em manter o conhecimento geral que o modelo adquiriu durante seu treinamento inicial. O PEFT, então, age como uma “camada extra”, adicionando um pequeno número de novos parâmetros treináveis. Esses novos parâmetros são ajustados para a tarefa específica, enquanto a maioria do modelo original permanece “congelada”.

A motivação fundamental por trás do PEFT é a observação de que, durante o *fine-tuning*, nem todos os parâmetros de um modelo precisam ser atualizados para alcançar a melhor adaptação. Modelos pré-treinados são compostos por representações que generalizam o conhecimento linguístico. Neste sentido, apenas pequenos ajustes direcionais são necessários para especializar o modelo em domínios ou tarefas específicas. Este conceito levou ao desenvolvimento de técnicas que preservam o conhecimento original do modelo enquanto introduzem capacidades adaptativas através de estruturas paramétricas eficientes.

Técnicas de PEFT têm demonstrado melhorias significativas no que tange à eficácia dos novos modelos desenvolvidos, tanto do ponto de análise quantitativa quanto da avaliação



qualitativa dos resultados. Estes avanços indicam a consolidação de tais técnicas no contexto de adaptação de modelos em diversos cenários, permitindo uma redução que abrange parcelas minúsculas dos parâmetros do modelo original, enquanto mantém performance similar com o *fine-tuning* completo. Li (Li, 2024), por exemplo, apresentou um refinamento seletivo de 0,2% dos parâmetros de um modelo BERT usando uma abordagem baseada na atualização dos parâmetros de *bias* e normalização, demonstrando resultados competitivos com técnicas consolidadas na literatura.

3.3.2.1 Low-Ranking Adaptation

No contexto do PEFT, a técnica de *Low-Rank Adaptation* (LoRA) (Hu et al., 2021) emergiu como uma das mais conhecidas e amplamente utilizadas. Seu funcionamento consiste em aplicar uma decomposição matemática para atualizar os pesos durante o processo de *fine-tuning*. Quando o *fine-tuning* tradicional é aplicado, cada matriz de peso do modelo original, W_0 , é atualizada para $W_0 + \Delta W$. No entanto, o LoRA reformula essa atualização ΔW como o produto de duas matrizes menores:

$$\Delta W = BA, \quad (3.1)$$

onde B e A são matrizes de baixo *rank*, onde B possui dimensões $d \times r$ e A possui dimensões $r \times k$, sendo d a dimensão da matriz original de pesos. Esta abordagem é fundamentada no princípio de que as mudanças necessárias para adaptar um modelo a uma nova tarefa podem ser representadas de forma eficiente em um espaço de menor dimensão.

Durante o processo de treinamento com LoRA, as matrizes de peso originais, W_0 , são mantidas completamente congeladas para preservar o conhecimento adquirido durante o pré-treinamento. As novas matrizes A e B são inicializadas de forma que seu produto BA seja inicialmente zero. Isso assegura que o modelo, no início do treinamento, seja idêntico ao modelo pré-treinado original. Conforme o treinamento avança, as matrizes A e B são as únicas que aprendem as adaptações necessárias para a tarefa específica. A saída de cada camada é então calculada como $h = W_0x + BAx$, onde o primeiro termo mantém o comportamento original do modelo e o segundo adiciona a especialização aprendida.

O parâmetro de *rank*, r , é um hiperparâmetro essencial no LoRA, que governa o equilíbrio entre eficiência e capacidade expressiva do modelo. *Ranks* menores resultam em maior eficiência computacional e menor uso de memória, mas podem limitar a capacidade de adaptação do modelo. Por outro lado, *ranks* maiores aumentam a flexibilidade, mas também elevam o custo computacional.

3.3.2.2 Quantized Low-Rank Adaptation

A combinação de quantização e LoRA é uma técnica avançada e extremamente eficiente para otimizar o ajuste fino de LLMs. A quantização é o processo de converter os pesos e ativações de uma rede neural de um formato de alta precisão (como ponto flutuante de 32 bits - **float32**) para um formato de baixa precisão (como inteiros de 4 ou 8 bits - **int8** ou **int32**). A principal motivação para isso é a economia de memória, já que um **float32** ocupa 4 bytes, enquanto um **int8** ocupa apenas 1 byte. Apesar de a diferença de memória na quantização de um único parâmetro ser pequena, ela se torna muito significativa quando aplicada a modelos de bilhões de parâmetros. Além de economizar memória, a quantização também acelera os cálculos, pois processadores e GPUs podem realizar operações com inteiros de forma muito mais rápida e eficiente. Contudo, o principal desafio



dessa abordagem é a possível perda de precisão, que pode degradar o desempenho do modelo se não for tratada com cautela.

A técnica de *Quantization* LoRA (Dettmers et al., 2023), denominada QLoRA, fornece uma solução para o desafio da perda de precisão ao aplicar a quantização seletivamente. O processo é dividido basicamente em duas etapas. Primeiro, o modelo pré-treinado original é quantizado para um formato de baixa precisão, usando uma técnica denominada **NormalFloat** de 4 bits. Todos os parâmetros são convertidos e armazenados em menos bits, reduzindo o modelo de dezenas de *gigabytes* para apenas alguns.

Em seguida, o processo de *fine-tuning* é realizado apenas nas matrizes de baixo *rank* do LoRA, que são mantidas em um formato de alta precisão, ou seja, em *floats* de 16 ou 32 bits. Esses poucos parâmetros em alta precisão representam o ponto chave para evitar a degradação de desempenho, pois o otimizador precisa de precisão para calcular os gradientes e atualizar os pesos de forma eficaz. Se as matrizes LoRA fossem quantizadas, o processo de aprendizado seria instável e ineficaz.

Mais especificamente, o QLoRA é baseado nas seguintes técnicas:

1. **NormalFloat 4 (NF4)**: A maioria dos pesos em uma rede neural profunda segue uma distribuição próxima de uma gaussiana ou normal, onde a maioria dos valores se concentra perto de zero. O NF4 define 16 pontos de quantização (já que $2^4 = 16$) de forma não uniforme, alocando mais pontos de precisão para os valores próximos de zero. Isso significa que os valores mais comuns e críticos do modelo são representados com maior precisão, enquanto os valores extremos, que são menos frequentes, são representados com menor precisão. O resultado é uma perda de informação muito menor em comparação com a quantização uniforme.
2. **Double Quantization**: Executa a quantização das constantes de quantização NF4 que auxiliam na reconstrução dos valores originais. Essa segunda etapa de quantização visa aplicar uma segunda redução do consumo de memória da rede, aplicando uma quantificação de ponto flutuante de 32 bits dessas constantes para 8 bits.
3. **Paged Optimizers**: é uma técnica de gerenciamento de memória cuja função é evitar os picos de uso de VRAM (memória da GPU) que ocorrem durante o treinamento, especialmente para armazenar parâmetros como gradiente, o momento do gradiente e a variância. Para solucionar esse problema, os *Paged Optimizers* alocam o estado do otimizador na memória da CPU. Quando a GPU precisa de espaço específico desses dados para o cálculo de gradientes, ela o aloca na VRAM. Após isso, esse espaço é realocado de volta para a CPU. Essa troca dinâmica entre a memória da CPU e da GPU garante que o treinamento continue de forma estável, sem ultrapassar o limite de VRAM.

Na prática, durante o treinamento, o fluxo de dados em uma camada com LoRA quantizado funciona da seguinte maneira: a entrada de um *token*, $\mathbf{x}$, é multiplicada pela matriz de peso quantizada do modelo base, W_0^{int8} . Em paralelo, essa mesma entrada é multiplicada pelas matrizes de alta precisão do LoRA, A e B . O resultado do cálculo com LoRA, isto é, $BA\mathbf{x}$, é então somado ao resultado do cálculo com o modelo base (W_0^{int8}), e tudo é feito em alta precisão antes de ser passado para a próxima camada.

A principal vantagem reside no fato de que a computação de maior custo, que envolve a maior parte dos parâmetros (W_0^{int8}), é feito de forma rápida e com pouca memória. O cálculo com os parâmetros do LoRA, que é a parte que realmente “aprende”, é mantido

em alta precisão para garantir a qualidade do *fine-tuning* e evitar perda de precisão no cálculo dos gradientes. Como resultado, a combinação de *quantization* e LoRA preserva o desempenho do modelo, mas de maneira mais eficiente no que concerne ao uso de memória.

3.3.2.3 Módulos adapters

Outra técnica que se destaca no contexto de PEFT são os módulos *Adapters*, propostos por Housby et al. (Housby et al., 2019), cuja estrutura é formada por pequenos módulos de rede neural que são incorporados nas camadas de um modelo *Transformer* pré-treinado. O princípio consiste em manter fixos os parâmetros do modelo base e treinar apenas os parâmetros desses novos módulos adicionados. Essa abordagem permite que o modelo original preserve o conhecimento já adquirido durante o pré-treinamento, enquanto os *Adapters* aprendem as adaptações específicas para a nova tarefa durante o ajuste fino do modelo.

A arquitetura de um módulo *Adapter*, em sua forma original, segue o padrão de um gargalo (*bottleneck*). Essa estrutura consiste em:

1. **Down-projection:** A entrada de uma camada do modelo original (com dimensão d) é projetada para uma dimensão menor, m , usando uma matriz de peso W_{down} . Isso cria um “gargalo” na arquitetura para comprimir a representação do dado.
2. **Up-projection:** A representação comprimida (h_{down}) é então projetada de volta para a dimensão original d usando uma segunda matriz de peso W_{up} , dando origem à uma representação de dimensão original h_{up} .

O valor da dimensão m é um hiperparâmetro, e geralmente é menor que d . Isso faz com que o número de parâmetros nos *Adapters* seja menor que o número de parâmetros do modelo original.

Para garantir a estabilidade do treinamento, é comum que a saída do módulo *Adapter* seja somada à entrada da camada, seguindo uma conexão residual (*skip-connection*). A saída final da camada com o *Adapter* é então representada por:

$$h_{final} = \mathbf{x} + h_{up}, \quad (3.2)$$

onde $\mathbf{x}$ é a entrada original do *Adapter*. A matriz W_{up} é geralmente inicializada com valores próximos de zero, para que o módulo *Adapter* comece com uma função de identidade. Isso garante que, no início do treinamento, a saída do *Adapter* seja zero e o comportamento do modelo seja idêntico ao original.

Em suma, as técnicas de PEFT consolidaram-se no contexto de ajuste fino de grandes modelos de linguagens ao oferecerem uma alternativa com a introdução de módulos com pouco impacto em suas arquiteturas originais. Entretanto, tais técnicas possuem suas próprias limitações, sendo a principal delas a de não poder replicar completamente o desempenho do *fine-tuning* tradicional em todos os cenários. Em tarefas que exigem uma adaptação mais complexa e mudanças significativas no modelo, o ajuste completo de todos os parâmetros ainda pode ser a abordagem mais eficaz.

3.3.3 Few-shot learning

Complementar às estratégias de ajuste fino completo e às técnicas de PEFT, as abordagens de aprendizagem com poucas amostras (do inglês, *few-shot learning*) buscam maximizar



ainda mais a eficiência computacional, reduzindo a quantidade de exemplos necessários para a especialização de um modelo. No campo do Processamento de Linguagem Natural, este termo abrange duas metodologias distintas para adaptar modelos a novas tarefas, nomeadamente *In-Context Learning* e Ajuste Fino com Poucas Amostras. Embora ambas partam da premissa de utilizar uma quantidade mínima de exemplos, elas diferem fundamentalmente no mecanismo de aprendizado, onde uma ocorre sem a atualização dos pesos do modelo, enquanto a outra envolve um ajuste fino deliberado.

3.3.3.1 Aprendizagem no contexto (*In-Context Learning*)

A primeira abordagem, e a que mais popularizou o termo no contexto de PLN. Este método não envolve uma fase de treinamento com atualização de gradientes. Em vez disso, o modelo é condicionado a realizar uma nova tarefa no momento da inferência. O aprendizado é transitório e guiado pelo *prompt* de entrada, que contém uma descrição da tarefa e um pequeno número de exemplos ilustrativos (os *shots*).

O trabalho de demonstrou que modelos de grande escala possuem capacidades emergentes de generalização a partir de poucos exemplos, viabilizando a aprendizagem no contexto de forma eficaz. Os autores avaliam três configurações:

- **Zero-shot learning:** O modelo recebe apenas a descrição em linguagem natural da tarefa, sem nenhum exemplo de como executá-la.
- **One-shot learning:** O modelo recebe a descrição da tarefa acompanhada de um único exemplo.
- **Few-shot learning:** O modelo recebe a descrição da tarefa e múltiplos exemplos (tipicamente entre 2 e 32).

A eficácia desta técnica está intrinsecamente ligada à qualidade e ao formato do *prompt*. A engenharia de *prompts* torna-se uma disciplina crucial, pois a maneira como os exemplos são apresentados pode influenciar drasticamente a precisão e a coerência da saída do modelo. A seleção de exemplos diversificados e representativos da tarefa em questão é fundamental para guiar o modelo de forma adequada.

A principal vantagem desta abordagem reside na sua eficiência computacional e na economia de recursos. Como não há treinamento, os custos associados à esta etapa são completamente eliminados. No entanto, a performance embora muitas vezes competitiva, pode não alcançar a mesma robustez de um modelo que passou pelo processo de atualização de parâmetros, especialmente em tarefas altamente especializadas ou que requerem conhecimento de um domínio muito específico. Além disso, a janela de contexto do modelo impõe um limite físico à quantidade de exemplos que podem ser fornecidos, e o custo da inferência pode aumentar.

3.3.3.2 Ajuste fino com poucas amostras

A segunda abordagem alinha-se mais com a definição clássica de *few-shot learning* em outras áreas de IA. Ela consiste em realizar o ajuste fino supervisionado do modelo utilizando um conjunto de dados deliberadamente pequeno, mas de alta qualidade. A hipótese central é que, para modelos que já possuem um vasto conhecimento de mundo adquirido durante o pré-treinamento, não é a quantidade de dados de ajuste fino que importa, mas sim a sua qualidade.



Um exemplo proeminente desta técnica é apresentado no trabalho de (Ye et al., 2025). Os autores exploram a premissa de que o raciocínio complexo exige volumes massivos de dados de treinamento. Eles propõem a hipótese LIMO (do inglês, *Less-Is-More Reasoning Hypothesis*) que postula que, em modelos onde o conhecimento de um domínio já foi extensivamente codificado o raciocínio sofisticado pode emergir a partir de demonstrações mínimas, mas estrategicamente desenhadas, de processos cognitivos. Segundo a hipótese, existem dois fatores principais que determinam o ponto de ativação para o raciocínio complexo: (1) a presença de conhecimento prévio dentro dos parâmetros do modelo, adquirido durante o pré-treinamento e (2) a efetividade de poucos exemplos em demonstrar processos de solução de problemas que estimulem uma reflexão mais extensa.

Para validar essa hipótese, os autores desenvolvem um rigoroso processo de curadoria de dados. Partindo de um conjunto com dezenas de milhões de problemas matemáticos, eles aplicam um sistema de filtragem em múltiplos estágios para selecionar apenas 800 amostras de treinamento de alta qualidade. Os critérios de seleção priorizaram problemas desafiadores e cadeias de raciocínio que exibiam características como elaboração detalhada, autoverificação e clareza lógica. Utilizando apenas 800 amostras para realizar um ajuste fino supervisionado simples o modelo alcançou resultados notáveis e amplamente superiores a outros modelos, tanto dentro quanto fora do domínio, confirmando que, para certas tarefas, a curadoria metódica de um pequeno conjunto de dados é mais crucial do que a quantidade massiva, permitindo o desenvolvimento de capacidades de raciocínio robustas de forma muito mais eficiente.

Em suma, as duas vertentes do *few-shot learning* oferecem um contraponto claro em termos de mecanismo e objetivo. A aprendizagem no contexto é uma técnica de inferência: ela não altera os pesos do modelo e depende da qualidade do *prompt* para guiar o comportamento do modelo de forma temporária. Por outro lado, o ajuste fino com poucas amostras é uma técnica de treinamento que modifica permanentemente o modelo. Porém, seu sucesso não está no volume, mas na qualidade e na densidade informacional dos dados. O investimento é deslocado da coleta massiva para a curadoria metódica, com o objetivo de alcançar um desempenho robusto e de ponta em domínios específicos.

3.4 AI alignment

O alinhamento de sistemas de IA aos valores humanos (do inglês, *AI Alignment*) refere-se ao desafio de garantir que agentes artificiais otimizem objetivos que estejam em conformidade com as intenções humanas, mesmo em contextos complexos, ambíguos ou não totalmente especificados. Esse problema é frequentemente enquadrado no contexto mais amplo do *controle de sistemas inteligentes*, conforme discutido por Russell (Russell, 2019) e Bostrom (Bostrom, 2014), e está no cerne das preocupações de segurança da IA contemporânea (Hendrycks et al., 2021; Ngo et al., 2023).

À medida que LLMs, agentes de decisão e sistemas autônomos adquirem maior expressividade e poder de generalização, torna-se cada vez mais crítico mitigar riscos decorrentes de comportamentos inesperados, otimizações mal direcionadas ou divergências entre metas explícitas e consequências práticas. Trabalhos como Gabriel (Gabriel, 2020) destacam que o alinhamento não se restringe a uma questão técnica, mas envolve também aspectos normativos e éticos sobre quais valores humanos devem orientar a otimização algorítmica.

Ao contrário de tarefas com métricas bem definidas, como acurácia em classificação, muitos problemas reais envolvem critérios subjetivos, valores éticos e decisões com múltiplas dimensões. Nesses casos, especificar manualmente uma função de recompensa ou



um conjunto de regras explicitamente alinhadas com os valores humanos é frequentemente impraticável. Assim, cresce o interesse por abordagens que incorporam diretamente o julgamento humano no processo de treinamento, não apenas como fonte de dados supervisionados, mas como sinal de alinhamento.

É útil esclarecer o que se entende por agentes artificiais. São sistemas de IA capazes de interagir com um ambiente e tomar decisões de forma autônoma, com base em objetivos explícitos ou implícitos. Eles operam em um ciclo de percepção, decisão e ação, recebendo observações do ambiente (por exemplo, entradas textuais ou sensores), executando ações (como gerar respostas, mover-se fisicamente ou chamar APIs) e otimizando comportamentos de acordo com critérios de desempenho. Exemplos incluem assistentes conversacionais interativos, robôs autônomos, sistemas de recomendação adaptativos ou LLMs integrados a agentes de planejamento e execução como o AutoGPT. À medida que o grau de autonomia e poder de ação desses agentes aumenta, cresce também o potencial de desalinhamento, tornando o *AI Alignment* uma prioridade crítica.

O tema ganhou relevância crescente à medida que sistemas de IA passaram a atuar de forma autônoma em cenários críticos, como decisões financeiras, diagnósticos médicos e controle de veículos autônomos, onde pequenos desalinhamentos entre a função de recompensa e os valores humanos podem gerar comportamentos perigosos ou não intencionais (Amodei et al., 2016; Carlsmith, 2022). Além disso, fenômenos como *goal misgeneralization* e *specification gaming* ilustram a dificuldade de definir objetivos de forma robusta (Krakovna et al., 2020; Shah et al., 2022).

O primeiro ocorre quando o modelo aprende corretamente o comportamento desejado durante o treinamento, mas generaliza a meta subjacente de forma incorreta em novos contextos, perseguindo intenções aparentes em vez de objetivos reais. Por exemplo, um agente treinado para coletar moedas em um jogo pode, em cenários novos, priorizar qualquer objeto circular, interpretando erroneamente o conceito de “moeda”. Já o *specification gaming* refere-se à exploração de falhas na função de recompensa ou nas restrições impostas pelo projetista, levando o agente a atingir pontuações elevadas sem realizar a tarefa de maneira adequada. Um exemplo clássico é o de agentes que maximizam métricas intermediárias, como distância percorrida, sem cumprir o objetivo final de forma útil. Esses fenômenos reforçam a necessidade de métodos de alinhamento que incorporem o julgamento humano de forma mais direta e adaptativa.

Duas das abordagens mais representativas desse paradigma são o *Reinforcement Learning from Human Feedback* (RLHF) (Christiano et al., 2017; Leike et al., 2018; Ouyang et al., 2022; Stiennon et al., 2020b) e o *Direct Preference Optimization* (DPO) (Rafailov et al., 2023), descritos a seguir com maior detalhamento.

3.4.1 Fundamentos de aprendizado por reforço

Antes de descrever RLHF e DPO, é importante revisar brevemente conceitos fundamentais do aprendizado por reforço (RL). O RL modela a interação entre um agente e um ambiente como um processo de decisão de Markov (MDP), definido por um conjunto de estados $\mathcal{S}$, ações $\mathcal{A}$, uma função de transição $P(s' | s, a)$ e uma função de recompensa $r(s, a)$.

Uma política $\pi_\phi(a | s)$ descreve a probabilidade de o agente tomar a ação a no estado s , parametrizada por ϕ . O objetivo do RL é maximizar a recompensa esperada acumulada ao longo do tempo:



$$J(\pi_\phi) = \mathbb{E}_{\pi_\phi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (3.3)$$

onde $\gamma \in [0, 1)$ é um fator de desconto temporal. Tradicionalmente, assume-se que a função de recompensa $r(s, a)$ é totalmente especificada pelo projetista do sistema. No entanto, em tarefas abertas e de alto nível, como diálogo natural, decisões éticas ou planejamento estratégico, definir manualmente $r(s, a)$ de forma alinhada aos valores humanos é virtualmente impossível. Essa limitação motiva métodos que estimam recompensas indiretamente a partir de preferências humanas ou substituem sua modelagem explícita por técnicas baseadas em sinais comportamentais observados.

3.4.2 Reinforcement Learning from Human Feedback (RLHF)

O RLHF (Christiano et al., 2017; Leike et al., 2018; Ouyang et al., 2022) é um procedimento de alinhamento que modela o problema como uma tarefa de controle sequencial, na qual um agente otimiza uma política $\pi_\phi(y | x)$, a probabilidade de gerar uma resposta y dado um *prompt* x , com o objetivo de maximizar uma função de recompensa derivada de preferências humanas.

3.4.2.1 Modelo de recompensa

Dado um conjunto de comparações humanas entre respostas y^+ (preferida) e y^- (rejeitada) para um mesmo x , o objetivo é aprender uma função de recompensa $r_\theta(x, y)$ tal que:

$$P(y^+ \succ y^-) = \frac{\exp(r_\theta(x, y^+))}{\exp(r_\theta(x, y^+)) + \exp(r_\theta(x, y^-))} \quad (3.4)$$

A função de perda associada é a entropia cruzada negativa entre as preferências humanas e a distribuição induzida por r_θ :

$$\mathcal{L}_{\text{RM}}(\theta) = -\mathbb{E}_{(x, y^+, y^-)} \left[\log \frac{\exp(r_\theta(x, y^+))}{\exp(r_\theta(x, y^+)) + \exp(r_\theta(x, y^-))} \right] \quad (3.5)$$

3.4.2.2 Aprendizado por reforço

Com r_θ fixado, ajusta-se a política π_ϕ para maximizar a recompensa esperada:

$$\max_{\phi} \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\phi(\cdot | x)} [r_\theta(x, y)] \right] \quad (3.6)$$

Para garantir estabilidade e evitar *drift*, impõe-se uma penalidade de divergência Kullback-Leibler (KL) em relação à política original π_0 :

$$\mathcal{L}_{\text{RLHF}}(\phi) = -\mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\phi(y | x)} \left[r_\theta(x, y) - \beta \cdot \log \frac{\pi_\phi(y | x)}{\pi_0(y | x)} \right] \right] \quad (3.7)$$

A função da Equação 3.6 pode ser reinterpretada como a maximização de uma vantagem regularizada:

$$\mathcal{L}_{\text{RLHF}}(\phi) = -\mathbb{E}_{\pi_\phi} [A^{r_\theta}(x, y) - \beta D_{\text{KL}}(\pi_\phi \| \pi_0)]$$

onde $A^{r_\theta}(x, y)$ representa a vantagem induzida pelo modelo de recompensa. Essa formulação evidencia que o treinamento busca não apenas maximizar a recompensa, mas também restringir a divergência da política em relação ao comportamento original do modelo.



Essa função é comumente otimizada via *Proximal Policy Optimization* (PPO) (Schulman et al., 2017), um algoritmo de aprendizado por reforço que atualiza a política por gradientes com restrições de divergência KL, garantindo que mudanças bruscas não degradem o desempenho.

3.4.2.3 Discussão sobre instabilidades

A separação entre a estimativa de preferências (r_θ) e a otimização da política (π_ϕ) pode introduzir instabilidades devido ao fenômeno conhecido como *reward hacking*. Quando o modelo de recompensa é imperfeito, a política pode explorar regiões do espaço de respostas onde r_θ é superestimado, produzindo comportamentos desalinhados com as intenções humanas. Além disso, a escolha do coeficiente β na Eq. (6) afeta a regularização: valores baixos permitem derivações excessivas da política, enquanto valores altos podem restringir a exploração, levando à subotimização.

3.4.3 Direct Preference Optimization (DPO)

O DPO (Rafailov et al., 2023) propõe um modelo mais direto e principiado para treinamento de políticas alinhadas, evitando a estimativa explícita de uma função de recompensa. A ideia central é modelar as preferências humanas como uma distribuição *a posteriori* sobre políticas, baseada na inferência bayesiana.

3.4.3.1 Formulação bayesiana

Assume-se que os dados de preferência $y^+ \succ y^-$ são amostrados de uma distribuição preferencial induzida por uma política ótima π^* , tal que:

$$P(y^+ \succ y^- | x) = \frac{\exp(\beta \cdot \log \pi^*(y^+ | x))}{\exp(\beta \cdot \log \pi^*(y^+ | x)) + \exp(\beta \cdot \log \pi^*(y^- | x))} \quad (3.8)$$

onde $\beta > 0$ controla a rigidez da preferência. Ao aproximar $\pi^* \approx \pi_\phi$, pode-se obter uma função de perda supervisionada para o aprendizado direto da política:

$$\mathcal{L}_{\text{DPO}}(\phi) = -\mathbb{E}_{(x, y^+, y^-)} \left[\log \frac{\exp(\beta \cdot \Delta \log \pi_\phi)}{\exp(\beta \cdot \Delta \log \pi_\phi) + 1} \right] \quad (3.9)$$

com $\Delta \log \pi_\phi = \log \pi_\phi(y^+ | x) - \log \pi_\phi(y^- | x)$.

A função de perda da Eq. (8) é análoga à perda logística em tarefas de classificação binária, onde o modelo aprende a distinguir respostas preferidas de rejeitadas. Sob essa perspectiva, o DPO pode ser interpretado como uma forma de inferência bayesiana aproximada, na qual a política ótima é tratada como distribuição posterior sobre ações condicionada às preferências humanas observadas.

3.4.3.2 Relação com aprendizado por reforço

A perda DPO pode ser interpretada como uma aproximação supervisionada do gradiente de política que surge da maximização da preferência empírica. Em vez de utilizar amostragem ou atualização por episódios, o DPO permite otimização por *backpropagation* puro, pois todas as quantidades relevantes são diretamente diferenciáveis.



3.4.3.3 Vantagens e limitações

Dentre as principais vantagens do DPO, temos:

- **Eficiência computacional:** não requer RL nem estimativa de r_θ ;
- **Estabilidade:** treinamento supervisionado direto evita instabilidades de PPO;
- **Simplicidade:** menos componentes, menos fontes de erro, maior interpretabilidade.

Apesar de seus méritos, o DPO depende fortemente da fidelidade das distribuições de preferência observadas. Essa dependência significa que a política tende a reproduzir fielmente os padrões nos dados disponíveis, mas pode falhar em capturar estruturas de longo prazo que só emergem após várias etapas de decisão. Um modelo de recompensa explícito, ao inferir padrões latentes e recompensas cumulativas, pode capturar dependências temporais e objetivos hierárquicos que o DPO, por sua natureza supervisionada, tende a ignorar.

3.4.4 Discussão comparativa: RLHF vs. DPO

Ambas as abordagens compartilham o objetivo de incorporar preferências humanas ao treinamento de políticas, mas diferem em aspectos fundamentais. O RLHF separa o aprendizado da função de recompensa do processo de otimização, o que oferece maior flexibilidade e capacidade de generalização, porém introduz complexidade e risco de instabilidades devido ao *reward hacking*. O DPO, por outro lado, adota uma formulação mais direta e eficiente, reduzindo o custo computacional e a instabilidade, ao preço de depender fortemente da qualidade dos dados de preferência. Assim, o RLHF tende a ser mais adequado para cenários com dados escassos, mas onde a generalização é essencial, enquanto o DPO se mostra vantajoso quando há abundância de *feedback* humano confiável e rápido treinamento é desejável. Revisões abrangentes recentes (Hendrycks et al., 2021; Ngo et al., 2023) destacam que abordagens híbridas e multiestágio, combinando aprendizado por preferências, interpretabilidade e segurança formal, representam um caminho promissor para sistemas de IA alinhados e robustos. O Quadro 3.1 resume comparativamente as principais diferenças entre o RLHF e o DPO, destacando seus fundamentos teóricos, características de treinamento e limitações práticas.

Quadro 3.1: Comparação entre *Reinforcement Learning from Human Feedback* (RLHF) e *Direct Preference Optimization* (DPO)

Aspecto	RLHF	DPO
Formulação	Baseado em aprendizado por reforço com modelagem explícita de recompensa e atualização da política via PPO (Christiano et al., 2017; Ouyang et al., 2022; Schulman et al., 2017).	Baseado em otimização supervisionada direta de preferências humanas, sem estimar explicitamente a função de recompensa (Rafailov et al., 2023).



Objetivo de treinamento	Maximizar a recompensa aprendida $r_\theta(x, y)$ regularizada pela divergência KL entre políticas.	Maximizar diretamente a probabilidade de respostas preferidas y^+ sobre rejeitadas y^- em uma formulação logística.
Tipo de atualização	Aprendizado por reforço com amostragem de trajetórias e gradientes de política (<i>policy gradients</i>).	Treinamento supervisionado via <i>backpropagation</i> padrão, sem interação com o ambiente.
Custo computacional	Alto, devido à necessidade de amostragem, atualização iterativa da política e estimativa de recompensas.	Baixo, pois elimina o ciclo de RL e opera inteiramente em dados estáticos de preferências.
Estabilidade de treinamento	Sensível ao ajuste do coeficiente de regularização β e à qualidade do modelo de recompensa, sujeito a <i>reward hacking</i> .	Mais estável, pois não requer modelagem de recompensas nem otimização sequencial via PPO.
Interpretação teórica	Aproximação regularizada do gradiente de vantagem em um MDP com recompensa aprendida.	Inferência bayesiana aproximada sobre políticas condicionadas a preferências humanas.
Generalização	Pode extrapolar preferências para novos contextos, desde que o modelo de recompensa capture relações latentes.	Limitado à distribuição observada de preferências, podendo falhar em extrapolações de longo prazo.
Aplicações típicas	Treinamento de LLMs (por exemplo, InstructGPT) e agentes de decisão com <i>feedback</i> humano em tempo real.	Ajuste fino supervisionado de LLMs com grandes conjuntos de preferências curadas (<i>offline alignment</i>).
Principais limitações	Complexidade computacional, dependência de modelo de recompensa robusto e risco de instabilidade.	Dependência da qualidade das preferências e dificuldade em capturar dependências temporais complexas.

3.5 Exemplo prático

Nesta seção, materializamos os conceitos teóricos discutidos ao longo do capítulo em uma aplicação prática de ponta a ponta. O objetivo é demonstrar o processo de especialização de um grande modelo de linguagem para uma tarefa de nicho no idioma português. Para isso, vamos criar o Dr. Bode, um assistente virtual de saúde, partindo de um modelo generalista e aplicando as técnicas de ajuste fino supervisionado (com PEFT) e otimização por preferência direta para adaptá-lo a um domínio altamente especializado, como a medicina.

3.5.1 Preparando o ambiente

A reprodutibilidade é um pilar da pesquisa e desenvolvimento em IA. Para garantir a consistência dos resultados, nosso ambiente de trabalho deve ser configurado corretamente. Este guia foi desenvolvido e testado com **Python 3.9**. As bibliotecas e suas respectivas



versões são listadas abaixo. Destacamos as principais: `transformers` e `datasets` para o ecossistema *Hugging Face*; `peft` para a aplicação de *Parameter-Efficient Fine-Tuning*; `trl` para os treinadores de alto nível (SFT e DPO); e `bitsandbytes` para a quantização do modelo.

```
transformers==4.53.1
torch==2.7.1
datasets==3.6.0
tqdm==4.67.1
peft==0.16.0
accelerate==1.8.1
trl==0.9.6
bitsandbytes==0.41.1
```

Ainda, definimos as seguintes variáveis globais, utilizadas durante a execução das células a seguir

```
NOME_MODELO = "Qwen/Qwen3-1.7B"
DATASET_AVALIACAO = "recogna-nlp/drbedebench"
DATASET_SFT = "recogna-nlp/drbede240_dataset"
DATASET_DPO = "recogna-nlp/dpo-mmlu-ptbr"
```

3.5.2 O ponto de partida

Todo projeto de ajuste fino começa com a escolha de um modelo de propósito geral. Para este estudo de caso, nosso ponto de partida será o `Qwen/Qwen3-1.7B`, um modelo com 1.7 bilhão de parâmetros. Este modelo foi escolhido por seu sólido desempenho em tarefas de raciocínio e seguimento de instruções, aliado a um tamanho que permite o treinamento em hardware acessível. Contudo, ele carece de conhecimento especializado em saúde, e nosso objetivo é justamente preencher essa lacuna. O treinamento será realizado com o `dataset recogna-nlp/drbede240_dataset`¹.

Antes de qualquer modificação, é crucial estabelecer uma linha de base (*baseline*) para quantificar nosso progresso. Avaliaremos o desempenho do modelo original no *benchmark recogna-nlp/drbedebench*², um conjunto composto por questões de múltipla escolha extraídas de provas de residência médica no Brasil. O resultado nos mostrará a capacidade inicial do modelo de responder a questões médicas complexas, servindo como ponto de partida para a jornada de especialização. O script a seguir realiza três etapas fundamentais:

- **Formatação do *prompt*:** Estrutura a questão no formato esperado pelo modelo.
- **Geração e extração da resposta:** Envia o *prompt* ao modelo e extrai a alternativa prevista.
- **Cálculo de acurácia:** Compara a resposta do modelo com o gabarito e calcula a acurácia total.

```
import re
import json
import time
import torch
```

¹https://huggingface.co/datasets/recogna-nlp/drbede240_dataset

²<https://huggingface.co/datasets/recogna-nlp/drbedebench>



```
from datasets import load_dataset
from transformers import AutoModelForCausalLM, AutoTokenizer
from tqdm import tqdm

def formatar_prompt_medico(questao: dict) -> str:
    """
    Cria um prompt formatado a partir de um item do dataset.

    Args:
        questao (dict): Um dicionário contendo 'enunciado', 'alternativas'
            e opcionalmente 'img_description'.

    Returns:
        str: O prompt completo e formatado, pronto para ser enviado ao modelo.
    """

    prompt = f"""Você é um especialista médico. Responda APENAS com a letra da
alternativa correta (A, B, C, D ou E). Não escreva nada mais.

Questão:
{questao['enunciado']}

"""

    # Trata de forma segura a existência de imagens na questão.
    if questao.get("contains_img", False):
        prompt += f"Descrição da imagem: {questao.get('img_description', '')}\n\n"

    for letra, texto in questao["alternativas"].items():
        prompt += f"{letra} {texto}\n"

    prompt += "\nResposta (apenas a letra):"
    return prompt.strip()

def obter_resposta_modelo(prompt: str, modelo: AutoModelForCausalLM,
tokenizer: AutoTokenizer) -> str:
    """
    Gera uma resposta do modelo para um determinado prompt.

    Args:
        prompt (str): O prompt formatado.
        modelo (AutoModelForCausalLM): O modelo de linguagem carregado.
        tokenizer (AutoTokenizer): O tokenizer correspondente.

    Returns:
        str: A resposta textual gerada pelo modelo.
    """

    # O template de chat é específico de cada família de modelos.
    # Esta etapa garante que a entrada seja formatada como o modelo espera.
    messages = [{"role": "user", "content": prompt}]
    input_ids = tokenizer.apply_chat_template(
        messages, add_generation_prompt=True, return_tensors="pt",
        enable_thinking=False
    ).to(modelo.device)

    # Gera a resposta. `max_new_tokens` é baixo pois esperamos apenas uma letra.
    outputs = modelo.generate(
```

```

        input_ids,
        max_new_tokens=10,
        pad_token_id=tokenizer.eos_token_id
    )

    # Decodifica os tokens gerados, ignorando o prompt de entrada.
    resposta = tokenizer.decode(
        outputs[0][input_ids.shape[1] :],
        skip_special_tokens=True
    )

    return resposta.strip()

def extrair_alternativa(texto_resposta: str) -> str:
    """
    Extrai a primeira letra de alternativa (A-E) de um texto.

    Args:
        texto_resposta (str): O texto bruto gerado pelo modelo.

    Returns:
        str: A letra da alternativa em maiúsculo, ou uma string vazia se não for
        encontrada.
    """
    # A regex [A-Ea-e] busca a primeira ocorrência de uma das letras.
    match = re.search(r"[A-Ea-e]", texto_resposta)
    if match:
        return match.group(0).upper()
    return ""

def executar_avaliacao(modelo: AutoModelForCausalLM, tokenizer: AutoTokenizer,
                        dataset: load_dataset, arquivo_saida: str):
    """
    Orquestra o processo de avaliação em todo o dataset.
    """

    start_time = time.time()
    resultados = []

    for item in tqdm(dataset, desc="Avaliando Questões"):
        prompt = formatar_prompt_medico(item)
        resposta_bruta = obter_resposta_modelo(prompt, modelo, tokenizer)
        predicao_processada = extrair_alternativa(resposta_bruta)

        resultados.append({
            "origem": item["origem"],
            "prova": item["prova"],
            "numero_questao": item["numero"],
            "enunciado": item["enunciado"],
            "resposta_correta": item["resposta"],
            "predicao_modelo": predicao_processada,
            "resposta_bruta_modelo": resposta_bruta,
        })

    # Cálculo de métricas e salvamento

```

```

end_time = time.time()
duracao_min = (end_time - start_time) / 60
print(f"\nAvaliação finalizada em {duracao_min:.2f} minutos.")

corretos = sum(1 for res in resultados if res["predicao_modelo"] ==
res["resposta_correta"])
total = len(resultados)
acuracia = (corretos / total) * 100
print(f"\nAcurácia final: {acuracia:.2f}% ({corretos}/{total})")

with open(arquivo_saida, "w", encoding="utf-8") as f:
    json.dump(resultados, f, ensure_ascii=False, indent=4)
print(f"Resultados salvos em: {arquivo_saida}")

# Carregando modelo e tokenizer
tokenizer = AutoTokenizer.from_pretrained(
    NOME_MODELO,
    trust_remote_code=True
)
model = AutoModelForCausalLM.from_pretrained(
    NOME_MODELO,
    trust_remote_code=True,
    torch_dtype=torch.bfloat16,
    device_map="auto"
)

# Carregando dataset de avaliação
eval_dataset = load_dataset(DATASET_AVALIACAO, split="train")

# Executar a avaliação
output_file = "resultado_modelo_base.json"
executar_avaliacao(model, tokenizer, eval_dataset, output_file)

```

Após a avaliação, o modelo base acerta 281 de 1301 questões, resultando em uma acurácia de 21,60%. Este valor, pouco acima de uma escolha aleatória (20% para 5 alternativas), confirma a ausência de conhecimento médico especializado e estabelece nosso ponto de partida.

3.5.3 Ajuste fino supervisionado com LoRA

Com a linha de base estabelecida, o próximo passo é ensinar ao modelo o conhecimento específico do domínio proposto. Utilizaremos a técnica de ajuste fino supervisionado, aplicando QLoRA para otimizar o uso de memória e reduzir os requisitos computacionais. Para o treinamento, seguiremos os passos de (1) carregar o modelo base quantizado, (2) configurar os adaptadores LoRA e (3) executar o treinamento com o `SFTTrainer`.

- **Configuração e carregamento do modelo:** Primeiro, configuramos a quantização e carregamos o modelo base. A quantização de 4 bits, gerenciada pela biblioteca `bitsandbytes`, é o que nos permite carregar um modelo de bilhões de parâmetros em uma GPU com VRAM limitada. A biblioteca `accelerate` auxilia no mapeamento do modelo para o dispositivo correto de forma automática.
- **Configuração do LoRA:** Definimos a configuração do LoRA usando a biblioteca `peft`. Em vez de treinar todos os 1.7 bilhão de parâmetros, vamos inserir adaptadores em camadas específicas do modelo. Apenas esses adaptadores, que representam uma pequena fração do total de parâmetros, serão treinados, resultando em uma economia massiva de recursos.
- **Treinamento com SFTTrainer:** Com o modelo preparado, utilizamos o `SFTTrainer` da biblioteca `trl`. Esta classe é um facilitador de alto nível que abstrai a complexidade do loop de treinamento. Ela lida com a formatação dos dados, o processamento em lotes e a otimização do treinamento de forma eficiente, exigindo apenas a configuração dos hiperparâmetros através da classe `SFTConfig`.



```
import os

import torch
from accelerate import PartialState
from peft import LoraConfig, get_peft_model, prepare_model_for_kbit_training
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig
from datasets import load_dataset
from trl import SFTConfig, SFTTrainer

def encontrar_modulos_lineares(modelo: AutoModelForCausalLM) -> list:
    """
    Encontra todos os módulos lineares no modelo para aplicar o LoRA.

    Esta função auxiliar percorre todas as camadas do modelo e identifica aquelas
    que são instâncias de `torch.nn.Linear`, que são os alvos ideais para
    a inserção dos adaptadores LoRA. A camada 'lm_head' é geralmente excluída
    porque sua adaptação pode levar a instabilidade no treinamento.

    Args:
        modelo (AutoModelForCausalLM): O modelo a ser inspecionado.

    Returns:
        list: Uma lista de nomes dos módulos lineares alvo.
    """
    cls = torch.nn.Linear
    modulos_lora = set()
    for nome, modulo in modelo.named_modules():
        if isinstance(modulo, cls):
            nomes = nome.split(".")
            modulos_lora.add(nomes[0] if len(nomes) == 1 else nomes[-1])

    if "lm_head" in modulos_lora:
        modulos_lora.remove("lm_head")

    return list(modulos_lora)

def preprocessar_dataset(exemplo):
    """
    Formata uma entrada do dataset para o formato de texto único esperado pelo
    SFTTrainer.
    """
    partes_texto = []
    for entrada in exemplo["data"]:
        partes_texto.append(f"{entrada['role']}.capitalize()}: {entrada['content']}")
    return {"data": "\n".join(partes_texto)}

# Determina o dispositivo para treinamento distribuído, se aplicável
string_dispositivo = PartialState().process_index

# Configuração da Quantização (QLoRA)
# QLoRA reduz o uso de memória ao carregar o modelo com pesos de 4 bits.
config_quantizacao = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4", # Tipo de quantização (Normal Float 4)
    bnb_4bit_compute_dtype=torch.bfloat16, # Tipo de dado para os cálculos

```



```

)

# Carregamento do modelo com a configuração de quantização
modelo_base = AutoModelForCausalLM.from_pretrained(
    NOME_MODELO,
    torch_dtype=torch.bfloat16,
    trust_remote_code=True,
    quantization_config=config_quantizacao,
    device_map={"": string_dispositivo},
)

# Desativa o cache para treinamento
modelo_base.config.use_cache = False
modelo_base = prepare_model_for_kbit_training(modelo_base)

# Carregamento do tokenizer
tokenizer = AutoTokenizer.from_pretrained(
    NOME_MODELO,
    trust_remote_code=True
)

tokenizer.pad_token = tokenizer.eos_token # Token de padding como fim de sentença.
tokenizer.padding_side = "right" # Garante que o padding seja adicionado à direita.

# Configuracao do PEFT (LoRA)
# LoRA congela os pesos do modelo original e injeta
# matrizes treináveis (adapters) em camadas específicas.
peft_config = LoraConfig(
    r=128, # 0 posto (rank) das matrizes de adaptação.
    lora_alpha=256, # Fator de escala para os pesos do LoRA.
    target_modules=encontrar_modulos_lineares(modelo_base), # Aplica LoRA em
                                                            #todas as camadas lineares.
    lora_dropout=0.05, # Dropout para regularização dos adaptadores LoRA.
    bias="none", # Não treina os vieses das camadas.
    task_type="CAUSAL_LM", # Especifica a tarefa como modelagem de linguagem causal.
)

# Aplica o wrapper PEFT para preparar o modelo para o treinamento LoRA.
modelo_peft = get_peft_model(modelo_base, peft_config)

# Carregamento do dataset
sft_dataset = load_dataset(DATASET_SFT, split="train")
sft_dataset = sft_dataset.map(preprocessar_dataset, remove_columns=["data"])

# Configuração do treinamento
# A classe SFTConfig armazena todos os hiperparâmetros para o SFTTrainer
output_dir_sft = "./modelo_sft"
argumentos_treinamento = SFTConfig(
    per_device_train_batch_size=8,
    gradient_accumulation_steps=4,
    gradient_checkpointing=True,
    max_grad_norm=0.3,
    num_train_epochs=1,
    learning_rate=2e-5,
    bf16=True,
    save_total_limit=3,
    logging_steps=1,
    output_dir=output_dir_sft,
    optim="paged_adamw_32bit",

```



```

lr_scheduler_type="cosine",
overwrite_output_dir=True,
warmup_ratio=0.05,
dataset_text_field="data",
)

# Instanciação e execução do treinador
treinador = SFTTrainer(
    model=modelo_peft,
    train_dataset=sft_dataset,
    peft_config=peft_config,
    args=argumentos_treinamento,
    processing_class=tokenizer,
)
treinador.train()

# Salva apenas os pesos dos adaptadores LoRA, que é um arquivo muito pequeno.
diretorio_checkpoint_final = os.path.join(output_dir_sft, "final_checkpoint")
treinador.model.save_pretrained(diretorio_checkpoint_final)
tokenizer.save_pretrained(diretorio_checkpoint_final)

```

Após o treinamento por uma única época, realizamos uma nova avaliação no *benchmark*. O modelo ajustado responde corretamente a 325 questões, alcançando uma acurácia de 24,98%, um aumento absoluto de 3,38 pontos percentuais. O ganho, embora positivo, é modesto. Isso ilustra um ponto importante: o SFT é altamente dependente da qualidade, diversidade e volume dos dados de treinamento. Fatores como o tamanho limitado do conjunto e o treinamento por uma única época podem explicar a magnitude do ganho. Para melhorias mais substanciais, seria necessário um conjunto de dados mais extenso, de maior qualidade ou treinando por múltiplas épocas de treinamento.

3.5.4 Refinamento do comportamento com DPO

Nosso modelo agora possui conhecimento especializado, adquirido durante o ajuste fino supervisionado. O próximo passo é refinar seu comportamento para que ele não apenas gere respostas corretas, mas também prefira um estilo de resposta a outro. O objetivo do DPO não é primariamente ensinar novo conhecimento, mas alinhar o comportamento do modelo a um estilo preferível.

O DPO requer um *dataset* de preferências no formato (*prompt*, *chosen*, *rejected*). A criação desses datasets é um dos grandes gargalos em AI Alignment, pois geralmente exige anotação humana especializada e cara. Para fins didáticos, utilizamos um pequeno *dataset* de exemplo. O mecanismo do DPO ajusta o modelo para que a probabilidade da resposta *chosen* seja maximizada em relação à *rejected*, aprendendo implicitamente uma função de recompensa. O ponto de partida para o DPO é o nosso modelo já ajustado com SFT. Utilizamos o `DPOTrainer` da biblioteca `trl`, que ajustará os pesos do modelo com base no *dataset* de preferências para alinhar seu comportamento.

```

import torch
from peft import PeftModel
from transformers import AutoModelForCausalLM, AutoTokenizer
from trl import DPOConfig, DPOTrainer

# Carregamento do dataset de preferência
dataset_dpo = load_dataset(DATASET_DPO, split="train")
dataset_dpo = dataset_dpo.train_test_split(test_size=0.1)

# Carregamento do modelo ajustado com SFT
caminho_adaptadores = "./modelo_sft/final_checkpoint"

modelo_base = AutoModelForCausalLM.from_pretrained(
    NOME_MODELO,

```



```

trust_remote_code=True,
torch_dtype=torch.bfloat16,
device_map="auto",
)

# Anexa os adaptadores LoRA do SFT e os torna treináveis novamente para a fase de
# DPO.
modelo_dpo = PeftModel.from_pretrained(
    modelo_base,
    caminho_adaptadores,
    is_trainable=True
)

tokenizer = AutoTokenizer.from_pretrained(caminho_adaptadores,
trust_remote_code=True)

# Configura o token de padding, se necessário.
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

# A classe DPOConfig armazena todos os hiperparâmetros para o treinamento.
config_dpo = DPOConfig(
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    gradient_accumulation_steps=8,
    num_train_epochs=1,
    learning_rate=5e-7,
    logging_steps=10,
    eval_strategy="steps",
    eval_steps=100,
    save_strategy="steps",
    save_steps=500,
    save_total_limit=2,
    report_to="none",
    remove_unused_columns=False,
    max_length=512,
    padding_value=None,
    beta=0.5,
    loss_type="sigmoid",
)

# Instanciação e execução do treinador DPO
treinador_dpo = DPOTrainer(
    model=modelo_dpo,
    ref_model=None,
    args=config_dpo,
    processing_class=tokenizer,
    train_dataset=dataset_dpo["train"],
    eval_dataset=dataset_dpo["test"],
)

treinador_dpo.train()

# Salva os adaptadores LoRA refinados pelo DPO.
output_dir_dpo = "./modelo_dpo"
treinador_dpo.save_model(output_dir_dpo)
tokenizer.save_pretrained(output_dir_dpo)

```

Após a etapa de DPO, a acurácia do modelo foi de 25,06% (326 acertos). O ganho de 0,08 pontos percentuais é marginal, o que é esperado. O DPO não foi projetado para melhorar a performance em



testes de conhecimento de múltipla escolha, mas sim para moldar o estilo da geração de texto. Uma avaliação qualitativa, analisando a cautela, clareza e segurança das respostas abertas do modelo, seria necessária para medir o verdadeiro impacto desta etapa em cenários reais de avaliação.

3.5.5 Considerações finais

Neste estudo de caso, transformamos o modelo generalista **Qwen/Qwen3-1.7B** em um protótipo de especialista em saúde. Partimos de uma acurácia de 21,60% e, por meio de um processo incremental de duas fases, aprimoramos tanto seu conhecimento quanto seu comportamento.

1. Ajuste fino supervisionado: Com QLoRA, infundimos conhecimento médico específico, elevando a acurácia para 24,98%. Esta etapa foi crucial para ensinar ao modelo o conteúdo factual do domínio.
2. Otimização por preferência direta: Com um *dataset* de preferências, alinhamos o comportamento do modelo, resultando em uma acurácia final de 25,06% e refinando o estilo e a forma da resposta.

A análise comparativa revela os pontos fortes e fracos inerentes a cada técnica. O SFT é a principal fonte de ganho de conhecimento, enquanto o DPO oferece um controle mais refinado sobre o comportamento. Este exemplo prático demonstra que a especialização de LLMs não é um ato único, mas um processo iterativo, onde a combinação de técnicas permite criar modelos mais capazes, seguros e alinhados. O Dr. Bode, desenvolvido neste capítulo, embora ainda um protótipo, ilustra o potencial e os desafios práticos dessa jornada.



Capítulo 4

Avaliação de Grandes Modelos de Linguagem

Fundamentos, métodos tradicionais e desafios atuais

Livy Real

André Carvalho

Altigran Soares da Silva

Publicado em: 16/04/2026



<https://brasiliaspln.ufscar.br/livro-pln-4ed-vol2/>

4.1 A importância e os desafios intrínsecos da avaliação de LLMs

Talvez você já tenha ouvido falar do caso em que uma LLM (Large Language Model, Grande Modelo de Linguagem) vendeu um carro por 1 dólar¹. Ou de quando um advogado estado-unidense, ex-advogado do próprio presidente, foi processado por inventar precedentes jurídicos². Se não ouviu, certamente você já ficou preso em conversas intermináveis e inúteis de *chatbots* de diversas empresas. Todos esses casos aconteceram e acontecem todos os dias porque os responsáveis por tais sistemas, mesmo que inadvertidamente, desconsideram a necessidade de uma avaliação robusta de seus sistemas baseados em LLMs. A ampla divulgação comercial desses sistemas pode levar à percepção equivocada de que já atingimos a era da Inteligência Artificial Geral (AGI), o tão esperado momento em que as IAs terão ‘inteligência’ semelhante à humana e serão autônomas. Isso ainda não corresponde à realidade. Hoje, em 2025, é possível afirmar que as saídas das LLMs sempre parecem boas, parecem verossímeis e factuais, mas com frequência não o são. Esse capítulo aborda a necessidade da avaliação de LLMs, considerando técnicas tradicionais e técnicas mais recentes, e vai defender a integração da avaliação das LLMs dentro do próprio ciclo de desenvolvimento de um sistema ou produto.

A avaliação de LLMs constitui uma questão fundamental no ciclo de vida de pesquisa e desenvolvimento desses modelos e de aplicações construídas com base neles, incluindo aí os agentes inteligentes. A sua importância transcende a mera verificação de desempenho, sendo crucial para o avanço da área a comparação entre diferentes arquiteturas, a identificação e mitigação de riscos e a garantia de aplicações práticas robustas e seguras (Meva; Kukadiya, 2025). A complexidade inerente à linguagem natural, combinada com a natureza gerativa e as capacidades emergentes dos LLMs, introduz desafios avaliativos que superam as abordagens tradicionais de Processamento de Linguagem Natural (PLN) (Minaee et al., 2025). A falta de interpretabilidade destes modelos ainda contribui à longa lista de novos desafios enfrentados na área atualmente.

LLMs, como modelos de linguagem estatísticos pré-treinados em larga escala, exibem “habilidades” interessantes que vão além da mera previsão de palavras. Incluem-se o aprendizado em contexto (in-context learning), em que os modelos assimilam novas tarefas a partir de poucos exemplos fornecidos no *prompt*, e o seguimento de instruções (*instruction following*), que permite que os modelos executem tarefas sem exemplos explícitos após o *instruction tuning*³ (Minaee et al., 2025). Estas capacidades, embora revolucionárias, trazem novos desafios à avaliação, pois as métricas clássicas usadas para mensurar sistemas de PLN frequentemente não conseguem capturar a profundidade semântica, a criatividade ou a factualidade das respostas geradas. Como as LLMs são treinadas para a simples tarefa de prever a próxima palavra de

¹<https://the-decoder.com/people-buy-brand-new-chevrolets-for-1-from-a-chatgpt-chatbot/>

²<https://www.npr.org/2023/12/30/1222273745/michael-cohen-ai-fake-legal-cases>

³*Instruction tuning* é o processo de re-treinar um modelo de linguagem já pré-treinado usando pares (instrução, resposta) para alinhar o modelo a seguir instruções (*instruction following*) (Zhang et al., 2025c).



uma sequência, mas, a partir desse treinamento, surgem capacidades emergentes (habilidades dos modelos que não foram explicitamente treinadas), não sabemos de antemão quais habilidades devemos avaliar.

A avaliação do raciocínio passo a passo, por exemplo, é fundamental para compreender e aprimorar a capacidade dos LLMs em problemas complexos, exigindo critérios que vão além da simples precisão da resposta final, como factualidade, validade, coerência e utilidade. A necessidade de *frameworks* de avaliação mais sofisticados é cada vez mais evidente, especialmente à medida que os LLMs são integrados em domínios de alta criticidade, como a saúde, na qual segurança e eficácia são imperativas. A avaliação deve, portanto, ser multifacetada, abrangendo não apenas o desempenho técnico, mas também a segurança, a eficiência e o impacto social dos modelos (Bender et al., 2021).

A evolução do paradigma de avaliação é uma resposta direta à forma como os LLMs operam. Sistemas de PLN anteriores frequentemente se propunham a tarefas mais simples e as métricas tradicionais projetadas para tais tarefas frequentemente assumem a existência de uma única resposta correta (Minaee et al., 2025). *Benchmarks* estáticos e métricas rígidas guiaram por décadas o avanço das técnicas de PLN, mesmo que as limitações dessas metodologias fossem conhecidas e frequentemente apontadas (Bowman; Dahl, 2021; Kalouli et al., 2023; Ribeiro et al., 2020). No entanto, a natureza gerativa dos LLMs, que podem produzir diversas saídas válidas para a mesma entrada, torna o problema da avaliação mais evidente, na medida em que a própria definição de uma resposta “boa” ou “correta” se torna uma definição fluida e multidimensional (Meva; Kukadiya, 2025; Minaee et al., 2025). Esta observação implica uma transição: saímos de métricas puramente objetivas e baseadas em correspondência de strings e vamos em direção a abordagens de avaliação mais diversificadas, buscando diferentes relações semânticas e alinhadas com a percepção humana. O desafio não é apenas medir a correção, mas sim a qualidade em um espaço subjetivo, criativo e dependente do contexto, o que demanda uma evolução contínua dos *frameworks* de avaliação.

Adicionalmente, a avaliação não é um processo *post-hoc*, mas uma parte integral do ciclo de desenvolvimento dos LLMs. O aprendizado por reforço (Reinforcement Learning, RL) utiliza pontuações de avaliadores humanos para aprimorar o treinamento de um LLM. Isso estabelece um vínculo causal: métricas de avaliação falhas ou incompletas, especialmente aquelas que não se alinham com os valores humanos ou com as capacidades emergentes dos modelos, podem levar à otimização dos sistemas para a métrica em si, em vez de levá-los a novas versões realmente úteis e seguras. Assim, a qualidade e o alinhamento das pontuações de avaliação impactam diretamente a trajetória de aprendizado do modelo e seu desempenho subsequente, especialmente em esforços de alinhamento como o RLHF (Reinforcement Learning from Human Feedback).

4.1.1 Tipos de avaliação

A avaliação de LLMs pode ser categorizada de diversas maneiras para abordar os múltiplos aspectos de seu desempenho. Uma distinção fundamental é entre avaliação *intrínseca* e *extrínseca*⁴. A avaliação intrínseca foca na qualidade inerente do modelo, frequentemente medida por métricas que avaliam a probabilidade de sequências de texto, como a perplexidade, ou a sobreposição lexical com textos de referência. A avaliação extrínseca, por outro lado, mede o desempenho do LLM em uma tarefa finalística ou *downstream*⁵ ou em um cenário de aplicação real, como a capacidade de um LLM de melhorar a experiência do usuário em um *chatbot* ou sua eficácia em tarefas de sumarização de notícias.

Outra categorização crucial diferencia a *avaliação humana* da *automática*. A avaliação automática tradicional emprega algoritmos e métricas computacionais para comparar a saída do modelo com referências ou para inferir a qualidade com base em características textuais. Este tipo inclui métricas estatísticas tradicionais (como BLEU e ROUGE). Atualmente, também há a avaliação automática baseada em modelos de linguagem, como a utilização do BERTScore e o paradigma LLM-as-a-Judge (Gao et al., 2024). Vamos visitá-las nas próximas páginas.

Já a avaliação humana é amplamente considerada o “padrão ouro” especialmente para tarefas de Geração de Linguagem Natural (NLG) devido à sua capacidade de capturar nuances e critérios de qualidade subjetivos que as métricas automáticas não conseguem capturar (Dierk et al., 2025; Gao et al., 2024; Wang et al., 2025).

⁴A Seção **Paradigmas: Tipos de avaliação** deste livro traz outras formas de categorizar possíveis avaliações de sistemas de PLN. Veja as Seções **Avaliação intrínseca** e **Avaliação extrínseca** para entender a aplicação de avaliação intrínseca e extrínseca a *datasets* específicos.

⁵Uma tarefa *downstream* é uma tarefa ‘mais para o fim do fluxo’. Considere que um LLM é treinado para prever a próxima palavra (essa seria a *upstream task*, no início do fluxo). Os usos reais, análise de sentimento, classificação textual, agente conversacional, são as *downstream tasks*, as tarefas reais nas quais aquele modelo será empregado (Schnabel et al., 2015).

Há uma complementaridade e compromissos (*trade-offs*) inerentes entre a avaliação humana e a automática. Embora a avaliação humana seja o “padrão ouro”, ela apresenta desafios significativos em termos de custo e escalabilidade, sendo um processo caro e demorado (Dierk et al., 2025; Gao et al., 2024). Em contraste, as métricas automáticas são eficientes e rápidas de calcular, mas muitas vezes carecem da capacidade de capturar a profundidade semântica e a coerência contextual que os humanos podem discernir (Minaee et al., 2025). Esta observação implica que um *framework* de avaliação holístico deve alavancar os pontos fortes de ambas as abordagens, em vez de depender exclusivamente de uma.

A combinação de verificações automáticas rápidas para qualidade básica e consistência com revisões humanas direcionadas para aspectos subjetivos, segurança e raciocínio complexo sugere um futuro no qual as abordagens de avaliação híbridas se tornarão o padrão. O trade-off entre custo/velocidade e profundidade/nuance possivelmente impulsionará as escolhas metodológicas na implantação prática de LLMs. A avaliação humana é essencial para validar textos gerados por máquinas e garantir o alinhamento com requisitos específicos do domínio e expectativas do usuário. A dificuldade intrínseca da avaliação de tarefas gerativas significa que a avaliação humana é sempre necessária. Além de não existir uma “resposta única correta” para muitas tarefas de geração de linguagem (Gao et al., 2024), é apenas a avaliação humana que garante o alinhamento de LLMs ou de qualquer outro tipo de avaliação (meta-avaliação) a valores humanos. Por valores humanos, entende-se não apenas ética e responsabilidade social, mas justamente a qualidade da máquina em agir exatamente como um humano agiria (e não apenas ‘parecido’).

4.2 Avaliações tradicionais

Em PLN, as tarefas podem ser divididas, de maneira geral, em dois grandes grupos: classificação e geração. Tradicionalmente, cada um desses tipos de tarefas é avaliado com uma abordagem específica, utilizando métricas distintas.

Nas tarefas de classificação, o sistema atribui rótulos (classes) a entradas de texto, como na análise de sentimentos ou na categorização de um documento. A avaliação costuma ser feita por métricas como acurácia, precisão, revocação e F1-score, que medem o quanto as previsões do modelo coincidem com as respostas esperadas. Tais métricas têm mais relação com classes, distribuição de instâncias nas classes do que com linguagem propriamente dita.

Já nas tarefas de geração, o modelo produz novas sequências de texto, como em tradução automática, sumarização ou resposta a perguntas. Nesse caso, a avaliação é mais complexa, pois envolve tanto a fluência quanto a adequação do texto, recorrendo a métricas automáticas como BLEU, ROUGE e METEOR, além de julgamentos humanos de qualidade. A maioria das métricas automáticas de geração tende a avaliar a correteza sintática e/ou semântica do texto gerado em relação a um texto de referência.

Neste capítulo, não revisitaremos em detalhe o funcionamento de cada uma dessas métricas, já discutidas com maior profundidade na Seção **Métricas: Medindo a performance**. Para ver mais sobre tarefas de geração de linguagem, você também pode consultar o Capítulo **Geração de linguagem natural**.

Cabe destacar, no entanto, que as métricas tradicionais de PLN, embora adequadas para modelos anteriores e tarefas específicas, mostram-se insuficientes diante das capacidades emergentes dos LLMs. Em estágios iniciais da tecnologia, a avaliação costumava concentrar-se em camadas superficiais do texto, como correção lexical ou gramaticalidade. Com os LLMs, que já atingem alta precisão nesses aspectos, torna-se necessário avaliar dimensões mais complexas da linguagem, como consistência factual, completude e utilidade em contextos específicos. Assim, o próprio objeto de avaliação se transforma, exigindo métricas novas e mais abrangentes, capazes de capturar a natureza multifacetada da geração de linguagem em larga escala.

4.2.1 Descompasso com a percepção humana

Um dos principais desafios das métricas tradicionais é o seu descompasso com a percepção humana de qualidade. Métricas baseadas em sobreposição de palavras, como BLEU e ROUGE, penalizam severamente paráfrases e sinônimos, mesmo que semanticamente equivalentes à referência (Mathur et al., 2020; Peyrard, 2019). Por exemplo, um texto gerado por um LLM que utiliza um vocabulário diferente, mas que transmite o mesmo significado que o texto de referência, pode receber uma pontuação artificialmente baixa nessas métricas, apesar de ser considerado de alta qualidade por um avaliador humano (Mathur et al., 2020; Minaee et al., 2025). O BLEU, por exemplo, ignora a sequência de palavras, resultando na mesma pontuação para frases com as mesmas palavras, mas com significados diferentes devido à ordem (Mathur et al., 2020). Ainda que esses sejam problemas que a área de geração de linguagem sempre tenha enfrentado (Gatt; Krahmer, 2018), eles se acentuam com o advento das LLMs.



As métricas tradicionais também não foram desenhadas para capturar aspectos complexos da linguagem, como fluência ou criatividade (Gao et al., 2024). Elas são sensíveis a pequenas variações lexicais que não afetam a qualidade percebida pelo ser humano (Mathur et al., 2020). A otimização exclusiva para essas métricas pode levar a modelos que “enganam” a métrica em vez de realmente melhorar a qualidade da geração de texto (Gao et al., 2024). Isso demonstra que as métricas tradicionais, ao focar na estrutura de superfície (*surface structure* (Chomsky, 1965)), falham em capturar a equivalência semântica e podem levar a conclusões enganosas sobre o desempenho real do sistema (Mathur et al., 2020; Minaee et al., 2025).

4.2.2 Dificuldade em avaliar aspectos complexos

Além do descompasso com a percepção humana, as métricas tradicionais demonstram uma dificuldade intrínseca em avaliar aspectos mais complexos e emergentes dos LLMs.

Factualidade e Alucinações

LLMs são propensas a “alucinações”, elas geram informações plausíveis, textos estatisticamente possíveis, mas factualmente incorretos, irrelevantes ou sem sentido (Meva; Kukadiya, 2025). Por exemplo, um LLM pode afirmar que “Thomas Edison inventou a internet” ou fornecer detalhes clínicos incorretos sobre um paciente específico. Tais erros não são detectados por métricas tradicionais, pois o texto pode ser gramaticalmente correto e fluente (Gao et al., 2024). A factualidade, especialmente em cenários de raciocínio passo a passo, exige que a informação possa ser fundamentada em fontes confiáveis ou que utilize corretamente números e restrições fornecidos na consulta. Verificar o conhecimento factual e o conhecimento de senso comum parametrizados, isto é, o conhecimento real e factual que a LLM adquiriu através de seu treinamento, permanece um desafio em aberto.

Raciocínio e Senso Comum

Tanto o processamento da informação de forma que pareça um raciocínio passo-a-passo quanto a capacidade de armazenar o chamado ‘senso comum’ são das mais relevantes e impressionantes capacidades emergentes que as LLMs apresentam. Por serem ‘novidade’, as métricas clássicas são incapazes de medir a capacidade de inferência, raciocínio complexo ou compreensão profunda dos LLMs (Meva; Kukadiya, 2025). A avaliação do raciocínio passo a passo é crucial para entender e melhorar o desempenho dos LLMs em problemas complexos como lógica, matemática e ciência. As métricas tradicionais não foram desenhadas para medir esses aspectos cruciais do processo de raciocínio. LLMs podem gerar saídas que soam plausíveis, mas carecem de coerência lógica, especialmente quando se baseiam em correlações estatísticas em vez de raciocínio causal (Xiong et al., 2025).

Segurança e Viés (*Bias*)

As métricas tradicionais também não foram projetadas para detectar conteúdo tóxico, preconceituoso ou prejudicial, a menos que sejam explicitamente treinadas para isso (Meva; Kukadiya, 2025). A avaliação da segurança e do viés requer abordagens específicas que analisem a imparcialidade, a inofensividade e a recusa de respostas inadequadas (Jiao et al., 2024). A otimização para métricas de sobreposição lexical não garante que o modelo não produzirá conteúdo problemático.

Seguimento de instruções

A capacidade de um LLM de seguir instruções complexas ou com múltiplas restrições é um aspecto crucial para sua utilidade (Minaee et al., 2025). Por esta ser uma nova capacidade da tecnologia, métricas tradicionais não a avaliam. É especialmente difícil avaliar o quão bem o modelo adere a essas instruções quando estas envolvem nuances ou múltiplos passos (Gao et al., 2024). LLMs podem falhar em compreender contextos longos ou instruções em conversas multiterno, quando as instruções são inconsistentes ou contêm informações incorretas. Métricas recentes que avaliam sistemas de diálogo podem avaliar a adequação de uma resposta ao contexto do turno anterior ou ainda a progressão do diálogo (Yeh et al., 2021), características que se assemelham ao *instruction following*, porém, tais métricas são também mensuradas através de modelos treinados para isso, assemelhando-se ao novo paradigma de avaliação que vamos discutir ao longo do capítulo, o *LLM as a Judge*.



4.2.3 Exemplos práticos

Para ilustrar as limitações das métricas tradicionais, vamos visitar o trabalho de (Li et al., 2025) que compara diferentes tipos de avaliação em uma tarefa específica do domínio do direito.

Os autores propõem o CaseGen, um conjunto de dados para avaliação de geração de documentos de casos jurídicos na área penal chinesa. O *benchmark* tem 500 casos reais, anotados por especialistas do direito. Os autores avaliam 9 diferentes LLMs para gerar parte desses casos. Para avaliar como essas LLMs foram, especialistas em direito avaliaram detidamente 50 casos. Esses 50 casos anotados foram comparados com diferentes métricas, como ROUGE-L, BertScore e o paradigma LLM-as-a-Judge (uma LLM avaliando outra LLM). Esse modelo dá notas de 0 a 10 para cada documento gerado considerando critérios como precisão dos fatos, lógica e correção legal.

Veja na Tabela abaixo a comparação entre a anotação humana e três métricas de avaliação usando diferentes medidas de calcular a correlação entre anotação. As correlações de Kendall, Pearson e Spearman são métricas de correlação entre avaliações automáticas e avaliações humanas frequentemente usadas quando precisamos medir a proximidade das avaliações (Shaqiri et al., 2023).

Tabela 4.1: Comparação entre correlações das métricas automáticas e avaliações humanas.

Metrics	LLM Score	Rouge-L	BERTScore
Kendall	0.667	0.333	0.166
Pearson	0.726	0.264	0.239
Spearman	0.750	0.375	0.250

O interessante para a gente é que as notas automáticas se aproximam bastante das avaliações de juristas humanos (correlação de até 0,75 quando compara-se a anotação humana à anotação do LLM juiz), muito melhor do que as métricas tradicionais, que ficaram abaixo de 0,4.

Esse exemplo demonstra que, embora as métricas tradicionais sejam fáceis de computar e úteis para certas comparações de superfície, elas não são necessariamente alinhadas às anotações humanas, exigindo a complementação com abordagens mais robustas e, muitas vezes, híbridas. É ainda relevante notar que a avaliação com LLM chega a 0.75 de correlação, mas não chega a 0.8 nem a 1. Por isso, a avaliação humana ainda é **tecnicamente** a mais confiável.

4.2.4 Avaliação humana: O padrão ouro?

Como vimos, a avaliação humana é amplamente reconhecida como o “padrão ouro” para a avaliação de LLMs, especialmente para tarefas de Geração de Linguagem Natural (NLG), onde a subjetividade e a nuance são inerentes (Dierk et al., 2025; Gao et al., 2024).

4.2.5 Importância e metodologias comuns

A avaliação humana é crucial porque é capaz de capturar aspectos de qualidade que as métricas automáticas não conseguem. Isso inclui a fluência, coerência, relevância semântica profunda, criatividade, factualidade e o seguimento de instruções complexas (Gao et al., 2024). A capacidade dos avaliadores humanos de compreender o contexto, interpretar significados implícitos e julgar a adequação de uma resposta para um determinado propósito a torna indispensável (Gao et al., 2024).

Metodologias comuns de avaliação humana incluem:

- **Escala Likert** – Utilizadas para avaliar a qualidade em diversas dimensões, como fluência, relevância, coerência e adequação. Os avaliadores atribuem uma pontuação em uma escala graduada (e.g., de 1 a 5 ou 1 a 100) para cada critério. Esta abordagem permite quantificar julgamentos subjetivos e pode ser aprimorada com a coleta de explicações em linguagem natural dos avaliadores para capturar nuances e calibrar as pontuações.
- **Ranking** – Envolve a comparação direta entre diferentes saídas de modelos para uma mesma entrada. Os avaliadores classificam as respostas em ordem de preferência, indicando qual é superior ou se há um empate (Vongthongsri, 2025). Essa metodologia pode ser mais estável do que a pontuação absoluta, pois foca na comparação relativa (Gao et al., 2024), porém ela nem sempre é reutilizável. Se, em um primeiro momento, analistas comparam dois modelos e, finalmente, temos um terceiro modelo em jogo, a primeira avaliação dificilmente será útil para decidirmos algo sobre este novo modelo.



- **Testes A/B** – Uma forma de ranking em que os avaliadores escolhem a preferência entre duas opções (A ou B) (Río; Vaahtio, 2024). É útil para comparar o desempenho de dois modelos ou variações de um mesmo modelo, como diferentes *prompts*, por exemplo. Geralmente testes A/B são feitos com grupos diferentes de usuários que não sabem que estão participando de um teste. Quando uma avaliação é feita através de teste A/B é importante considerar tanto a relevância estatística das amostras quanto como coletar o *feedback* dos grupos antes de implementar o teste. Rodar um teste sem ter uma hipótese, sem saber coletar e interpretar o *feedback* pode ser inócuo.
- **Análise de erros** – Consiste em classificar os tipos de erros presentes nas saídas dos LLMs (e.g., erros gramaticais, factuais, de coerência, alucinações, omissões). Esta metodologia fornece avaliação detalhada que pode ser usada para depurar e melhorar os modelos. A coleta de anotações de erros em respostas de LLMs é desafiadora devido à natureza subjetiva de muitas tarefas de PLN e à dificuldade em obter anotações objetivas para erros complexos. Veja a Subseção **Análise de Erro** deste livro para saber mais sobre análise de erro.

4.2.6 Desafios da avaliação humana

Apesar de ser o “padrão ouro”, a avaliação humana apresenta uma série de desafios que limitam sua escalabilidade e consistência.

Custo e Tempo

O processo de avaliação humana é notoriamente caro e demorado (Dierk et al., 2025). A obtenção de anotações de qualidade pode levar dias ou semanas, o que introduz latência no ciclo de *feedback* de desenvolvimento do modelo. Isso torna impraticável realizar uma avaliação humana completa para cada ajuste ou iteração do modelo (Gao et al., 2024).

Subjetividade e Variabilidade

Diferentes avaliadores podem ter opiniões distintas sobre a qualidade de uma mesma saída, levando a variabilidade nos julgamentos (Dierk et al., 2025; Jiao et al., 2024; Liu et al., 2024). Para algumas tarefas, como a avaliação da “criatividade” de uma história, até mesmo humanos podem discordar sobre o que é “melhor” (Gao et al., 2024). A subjetividade pode ser mitigada com diretrizes claras e perguntas mais objetivas.

Viés do avaliador

As experiências, expectativas e origens culturais dos avaliadores podem influenciar seus julgamentos, introduzindo vieses (Jiao et al., 2024). Por exemplo, um avaliador de uma cultura específica pode ter uma percepção diferente do que é “polido” em comparação com outro. É crucial ter um grupo de avaliadores diversificado para mitigar esses vieses. Além disso, a forma como a tarefa é apresentada aos avaliadores pode influenciar os resultados (e.g., “Modelo A foi ajustado para ser mais conciso. Qual é melhor?” pode induzir a um viés em favor da concisão) (Gao et al., 2024). Um viés conhecido é o viés da concordância (“acquiescence bias”) (Cronbach, 1942), que caracteriza a tendência do avaliador em concordar com afirmações ou responder de forma afirmativa, por exemplo, concordar com as etiquetas e as saídas dos sistemas automáticos quando exposto a eles.

Escalabilidade

É difícil aplicar a avaliação humana em larga escala e de forma contínua, especialmente com o rápido avanço e o grande volume de saídas dos LLMs (Dierk et al., 2025). A avaliação humana é frequentemente reservada para comparações finais ou verificações periódicas, enquanto métricas automáticas são frequentemente usadas durante o desenvolvimento dos sistemas (Gao et al., 2024).

Treinamento e Calibração de avaliadores

Para garantir consistência e alta qualidade nas anotações, os avaliadores precisam de diretrizes claras e treinamento rigoroso (Gao et al., 2024). A calibração é essencial para garantir que diferentes avaliadores apliquem os critérios de forma consistente (Liu et al., 2024). A falta de controle de qualidade pode levar a anotações de baixa qualidade, exigindo a inclusão de perguntas de teste para filtrar trabalhos de



baixa qualidade (Gao et al., 2024). Veja as Seções [Procedimentos e estratégias de anotação e revisão e Concordância entre-anotadores](#) deste livro para mais sobre estratégias de anotação e concordância entre anotadores.

Em suma, embora a avaliação humana seja insubstituível para capturar a qualidade genuína dos LLMs em tarefas complexas, seus desafios inerentes impulsionam a pesquisa em métodos de avaliação automática mais sofisticados e em abordagens híbridas que combinam o melhor dos dois mundos. Além disso, a maioria das avaliações por humano tem sido feita com *crowdworkers* (Hedderich; Oulasvirta, 2025), trabalhadores terceirizados, geralmente leigos e (mal) pagos por quantidade de anotação feita. Essa prática impacta diretamente a qualidade da avaliação, além de levantar questões éticas preocupantes (Gonzalez-Cabello et al., 2025; Jiang; Wagner, 2024) tanto socialmente quanto tecnicamente.

4.3 Benchmarks e datasets de avaliação

Benchmarks, no contexto da avaliação de LLMs, são conjuntos de dados e tarefas padronizadas que servem como referência para comparar o desempenho de diferentes modelos. Eles são cruciais para o avanço da pesquisa, permitindo uma avaliação sistemática e reproduzível das capacidades dos LLMs (Aisera, 2024; Minaee et al., 2025). Atualmente, temos muitos *benchmarks* disponíveis, especialmente para o inglês. Visitaremos apenas alguns para familiarizar o leitor com esses recursos.

4.3.1 Benchmarks genéricos (General-Purpose)

GLUE e SuperGLUE

GLUE (General Language Understanding Evaluation) e SuperGLUE (Super General Language Understanding Evaluation) são *benchmarks* históricos que impulsionaram a pesquisa em PLN. Enquanto GLUE focava em tarefas de compreensão de linguagem mais básicas, como inferência em linguagem natural (NLI), SuperGLUE foi desenvolvido para introduzir desafios mais complexos, exigindo raciocínio linguístico avançado, compreensão contextual, recuperação de conhecimento e question answering, refletindo problemas mais próximos do raciocínio humano (Wang et al., 2019b). No entanto, o GLUE original enfrentou o problema de saturação, onde o desempenho dos modelos se aproximava do nível humano, limitando sua capacidade de diferenciar modelos de ponta (Wang et al., 2019b). SuperGLUE buscou superar essa limitação, oferecendo um desafio mais rigoroso e com maior “espaço” para melhorias dos modelos.

HELM

HELM (Holistic Evaluation of Language Models) representa uma abordagem mais abrangente para a avaliação de LLMs. Diferente de *benchmarks* que se concentram apenas na acurácia, HELM avalia múltiplos aspectos do desempenho do modelo, incluindo acurácia, calibração, robustez, justiça (fairness) e eficiência (Aisera, 2024). Essa perspectiva multifacetada é essencial para compreender o comportamento dos LLMs em cenários diversos e complexos, especialmente em aplicações de alto risco (Meva; Kukadiya, 2025).

MMLU

MMLU (Massive Multitask Language Understanding) é um *benchmark* projetado para avaliar o conhecimento de LLMs em uma vasta gama de domínios. Ele abrange 57 áreas diversas, incluindo matemática, humanidades, ciências sociais e STEM (Ciência, Tecnologia, Engenharia e Matemática) (Aisera, 2024; Meva; Kukadiya, 2025).

4.3.2 Benchmarks para tarefas específicas

Question Answering (QA)

Aqui trataremos de *benchmarks* para avaliação de QA em larga escala. O Capítulo [Perguntas e Respostas](#) trata com bastante atenção a tarefa.

Para tarefas de Question Answering, *benchmarks* como SQuAD (Stanford Question Answering Dataset), *Natural Questions* e TriviaQA são amplamente utilizados (Houamegni; Gedikli, 2025).



O SQuAD é um *benchmark* fundamental, composto por mais de 100.000 pares de perguntas e respostas, onde as respostas são trechos de passagens fornecidas. SQuAD 2.0 introduziu perguntas sem resposta, exigindo que os modelos identificassem quando uma pergunta não podia ser respondida com o contexto dado (Rajpurkar et al., 2016).

Já o *Natural Questions* contém consultas geradas por usuários reais do Google Search, espelhando cenários de busca do mundo real onde as respostas podem não estar explicitamente em um único parágrafo (Kwiatkowski et al., 2019).

Por fim, o TriviaQA é um *dataset* mais desafiador que SQuAD, com 950.000 pares de perguntas e respostas de documentos da Wikipedia e da web, frequentemente exigindo raciocínio multi-hop e recuperação de informações entre documentos (Joshi et al., 2017).

Para atender à necessidade de avaliação de modelos em múltiplos idiomas, foram criadas extensões específicas para os principais *benchmarks* de Q&A. Essas versões mantêm o formato do *dataset* original, mas utilizam textos e perguntas em outras línguas.

- Para o SQuAD: Existem *datasets* como o XQuAD, que é uma tradução humana profissional do SQuAD v1.1 para 10 idiomas, incluindo árabe, grego, tailandês e turco. Outro exemplo é o MLQA (Multilingual Question Answering), que oferece um *benchmark* de extração de respostas em 7 idiomas, como espanhol, hindi e chinês.
- Para o *Natural Questions*: A principal extensão é o MKQA (Multilingual Knowledge Questions and Answers). Ele utiliza as perguntas do Natural Questions e as apresenta em 26 idiomas. As respostas devem ser encontradas nas versões da Wikipedia correspondentes a cada um desses idiomas.
- Para o TriviaQA: Foi desenvolvido o Mling-TriviaQA, que adapta o formato desafiador do original para 10 idiomas diferentes. As perguntas e os documentos de evidência foram coletados ou traduzidos para cada uma das línguas incluídas no *benchmark*.

Sumarização

Para a sumarização de texto, *datasets* como CNN/DailyMail⁶ e XSum (Narayan et al., 2018) são referências relevantes. Aqui exploraremos apenas esses *datasets*, mas você pode consultar o Capítulo [Sumarização Automática](#) para se aprofundar sobre a tarefa.

O CNN/DailyMail é um *corpus* amplamente utilizado para sumarização de notícias, contendo aproximadamente 300 mil notícias escritas em inglês, acompanhadas dos destaques de cada uma.

O XSum (Extreme Summarization Dataset) é focado em sumarização abstrativa, com resumos mais concisos e frequentemente reescritos, em vez de apenas extraídos do texto original.

Raciocínio

A avaliação das capacidades de raciocínio⁷ de uma LLM também pode ser o foco de um *benchmark*.

GSM8K (Grade School Math 8K)

Um *dataset* de 8.500 problemas matemáticos de nível fundamental, linguisticamente diversos, que exigem raciocínio multi-passos e operações aritméticas básicas (II, 2025; Said, 2025).

Big-Bench Hard (BBH)

Um subconjunto de tarefas desafiadoras do BIG-bench, projetado para testar as aptidões do modelo além dos *benchmarks* convencionais de PLN, incluindo raciocínio de alto nível (Meva; Kukadiya, 2025; Said, 2025).

⁶https://huggingface.co/datasets/abisee/cnn_dailymail

⁷Usamos o termo raciocínio de forma operacional, em linha com a literatura recente sobre LLMs. Não se deve entender que tais modelos realizem processos cognitivos análogos ao raciocínio humano, o termo aqui refere-se à capacidade de manipular representações linguísticas e numéricas por meio de correlações estatísticas complexas aprendidas durante o treinamento dos LLMs.

4.3.3 Benchmarks para segurança e confiabilidade

A segurança e a confiabilidade emergem como dimensões centrais na avaliação contemporânea de LLMs, especialmente em aplicações de alto impacto social, como os domínios jurídico ou médico. Diferentemente dos *benchmarks* voltados apenas à performance ou ao ‘raciocínio’, esses conjuntos de avaliação buscam medir a robustez ética e factual dos modelos, isto é, sua capacidade de evitar a produção de conteúdo falso, enviesado ou potencialmente nocivo.

Esses *benchmarks* abordam desafios que extrapolam a precisão técnica, como a tendência dos modelos a reproduzirem vieses presentes nos dados de treinamento, a gerarem informações incorretas com aparência de verdade ou a responderem de forma inadequada a *prompts* sensíveis. Assim, a avaliação de segurança e confiabilidade opera em um duplo eixo: veracidade factual e responsabilidade social.

Nos últimos anos, a literatura tem consolidado uma série de *benchmarks* voltados especificamente para essas dimensões, que hoje se tornaram indispensáveis para aferir se um LLM é apto a operar de forma segura e ética. Entre os principais, destacam-se o TruthfulQA, voltado à verificação da veracidade das respostas; o BBQ (*Bias Benchmark for Question Answering*), que mede o grau de viés social reproduzido por modelos; e o ToxiGen e o RealToxicityPrompts, que avaliam a propensão de um sistema a gerar conteúdo tóxico, ofensivo ou discriminatório.

TruthfulQA

Avalia a veracidade das respostas dos modelos, medindo sua resistência à geração de informações falsas ou equívocos comuns (Aisera, 2024; Lin et al., 2022).

BBQ (*Bias Benchmark for Question Answering*)

Mede o viés social em modelos de Question Answering, ajudando a identificar e mitigar preconceitos (Jan et al., 2025; Sánchez, 2024).

ToxiGen/RealToxicityPrompts

Avaliam a propensão dos modelos a gerar texto tóxico ou prejudicial, sendo fundamentais para garantir a segurança do conteúdo gerado (Gehman et al., 2020; Jan et al., 2025).

4.3.4 A importância de ir além do placar

Embora os *benchmarks* e os placares (leaderboards) forneçam uma maneira padronizada de comparar modelos, é crucial ir além da simples pontuação. A saturação de *benchmarks*, nos quais os modelos atingem ou superam o desempenho humano, pode mascarar limitações reais e a falta de generalização da aplicação dos modelos em cenários do mundo real (Meva; Kukadiya, 2025; Wang et al., 2019b). Desafios como a sensibilidade a pequenas mudanças nos *prompts* (*prompt sensitivity*) e a estocasticidade na geração de texto introduzem variabilidade nos resultados, o que torna a reprodutibilidade dos resultados de placares e *benchmarks* para modelos baseados em LLMs um problema mais complexo do que na avaliação de sistemas clássicos (Meva; Kukadiya, 2025). Além disso, a contaminação de dados (*benchmark leakage*), em que dados de avaliação são inadvertidamente incluídos no treinamento, pode inflar artificialmente as métricas de desempenho (Wang et al., 2019b). Portanto, uma análise qualitativa aprofundada e a consideração de múltiplos *benchmarks* são essenciais para uma avaliação holística e precisa.

4.4 Avaliação baseada em modelos (*Model-based evaluation*)

Nessa seção, trataremos do novo paradigma LLM-as-a-Judge, das vantagens, de possíveis abordagens para utilizar esse paradigma, bem como de suas desvantagens e pontos de atenção a serem considerados.

A ideia de utilizar um LLM para avaliar as saídas de outro LLM, conhecida como “LLM-as-a-Judge”, emergiu como uma alternativa promissora à avaliação humana, especialmente para textos abertos e complexos (Li et al., 2024; Vongthongsri, 2025). Essa abordagem envolve instruir um LLM (geralmente mais potente ou especificamente treinado para a tarefa de avaliação) a pontuar ou criticar a saída de outro modelo com base em critérios definidos (Li et al., 2024).



4.4.1 Vantagens

Escalabilidade e custo potencialmente menor

A avaliação humana é um processo notoriamente caro e demorado, o que a torna impraticável para avaliações contínuas ou em larga escala. LLMs como avaliadores oferecem uma solução mais escalável e potencialmente mais econômica para essa limitação (Li et al., 2024; Vongthongsri, 2025).

Capacidade de fornecer *feedback* em linguagem natural

Diferente das métricas automáticas tradicionais que fornecem apenas pontuações numéricas, os LLMs avaliadores podem gerar *feedback* detalhado em linguagem natural, explicando o porquê de uma determinada pontuação ou crítica. Isso pode ser útil para depurar e melhorar os modelos, se utilizado dentro de um fluxo híbrido, no qual esses *feedbacks* sejam realmente consumidos, porém o simples fato de uma LLM gerar um *feedback*, uma explicação sobre sua resposta, não garante que a resposta esteja correta.

Potencial para avaliar aspectos mais abstratos

LLMs avaliadores têm o potencial de avaliar aspectos mais abstratos da qualidade do texto, como coerência, relevância, tom e criatividade, que são difíceis de capturar com métricas clássicas (Vongthongsri, 2025).

Avaliação de interações dinâmicas e contextuais

Uma vantagem crucial é a capacidade de avaliar sistemas de IA da mesma forma que são utilizados. Por exemplo, em vez de depender de *datasets* estáticos, que são inadequados para medir a qualidade de um diálogo, um LLM avaliador pode analisar uma conversa em tempo real. Ele pode avaliar a interação passo a passo ou o diálogo completo, proporcionando uma avaliação muito mais fiel à experiência real do usuário em um *chatbot* ou assistente virtual⁸.

4.4.2 Abordagens

Avaliação baseada em referência vs. sem referência

Algumas abordagens de LLM-as-a-Judge podem operar com ou sem textos de referência. Métricas baseadas em *embeddings* de LLMs, por exemplo, frequentemente não exigem referências, calculando a similaridade semântica diretamente entre a saída do modelo e a consulta do usuário ou o contexto de uso. No entanto, a avaliação sem referência pode ser mais suscetível a falhas e vieses.

Prompting para avaliação

A criação de *prompts* eficazes é fundamental. Esses *prompts* instruem o LLM avaliador sobre os critérios de avaliação, o formato da saída (e.g., escalas Likert, comparações binárias, análise de erros) e o texto a ser avaliado (Li et al., 2024). A qualidade do *prompt* pode influenciar drasticamente a avaliação.

Fine-tuning de LLMs para tarefas de avaliação

Além do *prompting*, LLMs podem ser especializados (*fine-tuned*) especificamente para tarefas de avaliação, treinando-os para prever pontuações humanas ou para identificar tipos específicos de erros.

LLM-as-a-Jury (LLM como um júri)

Para aumentar a robustez e mitigar os vieses de um único avaliador, esta abordagem utiliza um painel de múltiplos LLMs para julgar uma mesma saída. A tarefa é enviada a vários “LLMs juízes”, que podem ser modelos diferentes ou instâncias do mesmo modelo com configurações distintas. As avaliações individuais são então agregadas (por exemplo, através de uma votação majoritária ou pela média das pontuações)

⁸Atualmente, também têm sido propostos *frameworks* dinâmicos para a análise de agentes baseados em LLMs, como o clembench (Chalamalasetti et al., 2023), cujo diferencial é ser uma avaliação interativa diretamente motivada pela Teoria do Diálogo, para avaliar como fenômenos do diálogo humano são ou não reproduzidos pelos LLMs.



para chegar a um veredito de consenso. Essa técnica visa replicar a dinâmica de um painel de especialistas humanos, resultando em uma avaliação final mais estável e confiável.

Avaliação por decomposição de tarefas

Esta abordagem quebra a avaliação de uma tarefa complexa em subtarefas mais simples e verificáveis. Por exemplo, para avaliar a qualidade de um resumo, em vez de julgá-lo diretamente, o processo poderia ser: 1) Gerar um conjunto de perguntas e respostas com base no documento original (fonte). 2) Pedir a um LLM que responda às mesmas perguntas usando apenas o resumo gerado. 3) A avaliação final, comparar as respostas obtidas a partir do resumo com as respostas obtidas a partir do documento original, medindo assim a fidelidade e a retenção de informação do resumo.

4.4.3 Desafios e Limitações

Apesar das vantagens, a avaliação baseada em modelos apresenta desafios significativos.

Viés do modelo avaliador

O LLM avaliador pode herdar ou amplificar vieses presentes em seus dados de treinamento, ou até mesmo exibir um “viés narcisista”, favorecendo respostas geradas por si mesmo (Panickssery et al., 2024) ou por modelos da mesma família (Vongthongsri, 2025). Estudos mostram que avaliadores generativos tendem a atribuir pontuações mais altas ao conteúdo gerado pelo mesmo modelo subjacente.

Sensibilidade ao *prompt*

Pequenas alterações na formulação do *prompt* de avaliação podem levar a diferenças substanciais nos resultados, tornando a avaliação inconsistente e não robusta (Zhao et al., 2025). Para lidar com essa questão, um método chamado PromptEval (Polo et al., 2024) foi proposto. A abordagem reconhece que avaliações baseadas em um número limitado de *prompts* podem não capturar as reais habilidades dos LLMs, levando a classificações inconsistentes. Em vez de depender de um único “*prompt*”, o PromptEval estima a distribuição de desempenho do modelo através de um grande conjunto de variações de “*prompts*”. Isso permite a criação de métricas de avaliação mais robustas, como a performance mediana ou em quantis específicos (por exemplo, o desempenho nos 5% piores *prompts* ou nos 5% melhores), fornecendo um panorama mais completo e confiável da capacidade do modelo, que é menos suscetível a alterações em um “*prompt*” individual.

Concordância com humanos

Embora alguns estudos sugiram que LLMs podem atingir alta concordância com julgamentos humanos (até 85% para GPT-4, por exemplo) (Bencke et al., 2024; Kim et al., 2025; Vongthongsri, 2025), outros indicam que a acurácia da avaliação por LLMs não é garantida em todos os cenários (Pombal et al., 2025; Wang et al., 2024). A correlação pode variar dependendo da tarefa e do modelo avaliador. No entanto, a discussão é mais complexa, pois replicar o julgamento humano pode significar replicar também seus vieses. Um estudo sobre o tema ilustra essa questão, mostrando que, embora juízes humanos demonstrem menos vieses de gênero que os LLMs, ambos são igualmente vulneráveis ao viés de autoridade (favorecendo respostas com referências falsas), ao “viés de beleza” (preferindo textos mais bem formatados) e à dificuldade em detectar desinformação (Chen et al., 2024a). Isso demonstra que o verdadeiro desafio não é apenas alcançar a concordância, mas desenvolver métodos de avaliação que superem as falhas inerentes tanto a humanos quanto a máquinas.

Custo computacional

Embora potencialmente mais barato que a avaliação humana em larga escala, o uso de LLMs potentes como juízes ainda pode incorrer em custos computacionais significativos, especialmente para avaliações contínuas (Meva; Kukadiya, 2025), avaliações em que uma única instância é avaliada usando diferentes critérios ou no uso de LLM-as-a-Jury, o uso de diferentes modelos para avaliar o mesmo sistema.



Janela de contexto finita

Os modelos de linguagem possuem um limite na quantidade de texto (contexto) que conseguem processar de uma só vez. Essa restrição se torna um grande obstáculo ao avaliar tarefas que envolvem textos longos, como a sumarização de documentos extensos ou a análise de um histórico de diálogo completo. A necessidade de truncar ou dividir o texto para que ele caiba na janela de contexto pode levar à perda de informações e prejudicar a capacidade do LLM de realizar uma avaliação holística e precisa.

Não determinismo

As pontuações de LLMs juízes podem ser não determinísticas, ou seja, a mesma saída pode receber pontuações diferentes em avaliações repetidas, a menos que sejam aplicadas técnicas para garantir (algum) determinismo (Vongthongsri, 2025).

Viés de posição e verbosidade

LLMs avaliadores frequentemente exibem um “viés de posição”, preferindo a primeira saída, ou eventualmente a segunda saída, em comparações pareadas, e um “viés de verbosidade”, preferindo textos mais longos, mesmo que não sejam necessariamente de maior qualidade (Chen et al., 2024a).

Viés de leniência (*Leniency bias*)

Este fenômeno é caracterizado pela propensão dos modelos em avaliar respostas como “corretas” em situações de incerteza, ou quando seus critérios de avaliação não estão perfeitamente alinhados com as instruções fornecidas. Um estudo de 2024 quantifica essa tendência ao estimar a probabilidade de um juiz marcar uma resposta como correta quando, na verdade, ele não está aplicando os critérios de avaliação corretamente (Thakur et al., 2025). Os resultados demonstram que a maioria dos modelos juízes exibe um viés de leniência consideravelmente alto, indicando que, na dúvida, eles tendem a julgar positivamente. Essa propensão é mais pronunciada em modelos menores.

Outros vieses

Vários outros tipos de vieses relacionados foram identificados mais recentemente (Kamruzzaman et al., 2024; Park et al., 2024) como *Length Bias* (viés de extensão), *Concreteness Bias* (a tendência de atribuir maior credibilidade a respostas que contêm detalhes específicos, como citações, números e terminologias complexas), *Empty Reference Bias* (o modelo prefere conteúdo alucinado que apenas parece estar associado à instrução original), o Content Continuation Bias (o modelo favorece respostas que continuam o texto de entrada em vez de seguir a instrução principal) e o *Nested Instruction Bias* (tendência a responder a perguntas aninhadas dentro do texto, ignorando a instrução principal).

4.4.4 Exemplos de *frameworks* de LLM as a Judge

G-Eval

Um método que gera uma série de passos de avaliação a partir de critérios definidos, utilizando um LLM para preencher um formulário e determinar a pontuação final (Vongthongsri, 2025).

Prometheus

Uma suíte de LLMs avaliadores de código aberto, otimizados para avaliação multilíngue, que podem fornecer avaliação direta e comparação pareada (Kim et al., 2023).

AutoEval

Abordagens que visam automatizar a avaliação, como a investigação do uso de LLMs como juízes para questões abertas legais, mostrando alta correlação com pontuações humanas em exames padronizados (Samuylova, 2025).



Avaliação jurídica baseada em QA

Este método avalia resumos jurídicos usando um LLM em um processo de Pergunta e Resposta (QA). O LLM primeiro cria perguntas com base em um resumo de referência e depois usa o resumo candidato para respondê-las. Finalmente, o modelo compara as duas respostas para gerar uma pontuação, focando na qualidade da argumentação legal em vez de apenas na sobreposição de palavras (Xu; Ashley, 2023).

DeepEval

É um *framework* de código aberto para avaliação de LLMs⁹, frequentemente descrito como o “Pytest para LLMs”, por aplicar uma abordagem de testes unitários à avaliação de saídas de modelos de linguagem. Ele oferece uma suíte com mais de 14 métricas, como G-Eval, RAGAS, relevância da resposta e detecção de alucinações. O DeepEval se integra a outras ferramentas do ecossistema, como LangChain e LlamaIndex, e é complementado pela plataforma em nuvem Confident AI¹⁰, que adiciona funcionalidades de gerenciamento de *datasets* e detecção de regressão nos testes.

4.5 Aspectos emergentes e avançados da avaliação

Como já comentado, com as LLMs, surgem também novos desafios para entender como eles realmente funcionam e se comportam. Já não basta medir apenas a precisão ou a fluência das respostas, avaliar se os modelos são robustos, justos, confiáveis e eficientes também torna-se necessário. Nesta seção, exploramos essas novas dimensões da avaliação, trazendo, brevemente, exemplos de possíveis formas de detecção e mensuração desses desafios.

4.5.1 Robustez e Ataques adversariais

A robustez de um LLM refere-se à sua capacidade de manter o desempenho e a segurança diante de entradas ruidosas, com erros ortográficos, ou maliciosamente construídas (ataques adversariais). A avaliação da robustez é crucial para a implantação segura e responsável de LLMs em domínios de alto risco, onde erros podem ter consequências graves. Ataques adversariais podem incluir:

- Perturbações em nível de caractere: Alterações sutis nos *prompts* que podem induzir o modelo a comportamentos indesejados (Zhao et al., 2025).
- *Jailbreak prompts*: *Prompts* deliberadamente elaborados para contornar os mecanismos de segurança dos LLMs e induzi-los a gerar conteúdo prejudicial. Novos *frameworks* estão sendo desenvolvidos para avaliar autonomamente a robustez dos LLMs, utilizando *prompts* adversariais refinados e diretrizes de conhecimento restritas ao domínio, muitas vezes na forma de grafos de conhecimento.

4.5.2 Avaliação de justiça, viés e toxicidade

A avaliação de justiça (*fairness*), viés (*bias*) e toxicidade é um campo em crescimento, impulsionado pela necessidade de garantir que os LLMs não perpetuem ou amplifiquem preconceitos presentes em seus dados de treinamento (Japkowicz; Boukouvalas, 2024; Jiao et al., 2024).

A identificação e a quantificação de vieses em LLMs têm-se tornado uma frente essencial da pesquisa contemporânea. Diversas ferramentas e técnicas vêm sendo desenvolvidas para identificar e mensurar diferentes tipos de vieses, como de gênero, raça ou culturais, presentes nas saídas desses modelos (Jan et al., 2025; Japkowicz; Boukouvalas, 2024; Kamruzzaman et al., 2024). *Benchmarks* específicos, como o BBQ (*Bias Benchmark for Question Answering*) e o ToxiGen, têm sido utilizados para avaliar tanto a reprodução de estereótipos sociais quanto a propensão dos modelos a gerar discurso de ódio (Gehman et al., 2020; Jan et al., 2025).

A própria definição de viés é um desafio. Ela varia conforme o contexto cultural e exige cuidado para distinguir entre padrões legítimos do mundo real e estereótipos prejudiciais (Meva; Kukadiya, 2025). Para enfrentar essa complexidade, novas abordagens vêm sendo propostas, como o *framework de auto-coerência*, no qual os próprios modelos analisam seus padrões de saída em busca de inconsistências e vieses implícitos (Japkowicz; Boukouvalas, 2024). Ainda assim, a mitigação efetiva desses problemas requer

⁹<https://github.com/confident-ai/deepeval>

¹⁰<https://www.confident-ai.com/>



monitoramento contínuo e o uso de técnicas de debiasing, voltadas à correção progressiva das distorções detectadas (Sánchez, 2024).

Uma estratégia frequentemente explorada é o uso de *datasets* sintéticos para tentar corrigir vieses complexos. No entanto, essa solução pode introduzir novos riscos metodológicos: em vez de eliminar o viés original, o modelo avaliador pode aprender a reproduzir os mesmos padrões ou artefatos (artifacts) gerados pelo modelo que criou o *dataset* sintético. O resultado é a introdução de um novo tipo de viés, que compromete a validade da avaliação e revela uma limitação importante das atuais estratégias de mitigação baseadas em dados sintéticos (Chen et al., 2024a).

4.5.3 Factualidade, alucinações e citabilidade

A factualidade e a prevenção de “alucinações” são desafios críticos para os LLMs (Meva; Kukadiya, 2025). Métodos para verificar a veracidade da informação gerada incluem:

Detecção de alucinações

Técnicas como FactSelfCheck permitem a detecção de alucinações em nível de fato, analisando a consistência factual entre múltiplas respostas do LLM (Sawczyn et al., 2025).

Citabilidade e Atribuição de fontes

A capacidade de um LLM de citar corretamente suas fontes é fundamental para a confiabilidade. Métodos como LLM-Cite exploram a capacidade do LLM de gerar URLs de citação potenciais para uma determinada afirmação, que são então verificadas em tempo real (Joshi et al., 2025). No entanto, a atribuição de fontes ainda enfrenta desafios, especialmente em domínios específicos, como o jurídico (Zhang et al., 2025b).

Avaliação de fundamentação (*Groundedness*) via NLI

Essa abordagem avalia se uma afirmação gerada por um LLM é factualmente suportada por uma fonte específica, usando a Inferência de Linguagem Natural (NLI) para classificar a relação como corroboração (acarretamento) ou contradição. O trabalho de (Trautmann et al., 2024) mostra que, embora métodos híbridos complexos (como combinar o uso de LLM-as-a-Judge com métodos de NLI) sejam os mais precisos, são também muito caros e lentos. O trabalho destaca que uma abordagem mais simples e eficiente, como perguntar diretamente a um LLM potente (e.g. GPT-4) se a fonte suporta a afirmação, já oferece uma performance robusta, servindo como um forte *baseline*.

4.5.4 Eficiência computacional e custos

A eficiência computacional é uma dimensão cada vez mais importante na avaliação de LLMs, dada a sua crescente escala e os custos associados à inferência e implantação (Galileo AI, 2024; Meva; Kukadiya, 2025).

Métricas

As métricas de eficiência incluem:

1. Latência: O tempo entre a submissão de um *prompt* e o recebimento da resposta completa (Google Cloud, 2025).
2. Throughput: O volume de requisições ou *tokens* que o sistema pode processar por unidade de tempo (Google Cloud, 2025).
3. Consumo de Energia: A energia gasta durante a inferência, um fator crucial para a sustentabilidade (Meva; Kukadiya, 2025);
4. Custo de Inferência: O custo monetário associado à execução do modelo (Meva; Kukadiya, 2025).

A otimização dessas métricas é vital para a viabilidade de aplicações práticas, especialmente aquelas que exigem respostas em tempo real ou operam em larga escala (Galileo AI, 2024; Google Cloud, 2025). Técnicas como test-time compute (TTC) e otimizações de arquitetura visam melhorar a eficiência sem comprometer a acurácia (Meva; Kukadiya, 2025).



4.6 Considerações éticas e o Futuro da avaliação de LLMs

À medida que os LLMs passam a ocupar papéis centrais em produtos, pesquisas e até decisões governamentais, a avaliação deixa de ser apenas uma questão técnica e se torna também uma questão ética e social. Avaliar modelos implica também em discutir responsabilidade, transparência e possíveis impactos negativos de seu uso. Nesta seção, exploramos como a avaliação deve incorporar princípios de responsabilidade, interpretabilidade e regulação, preparando o caminho para práticas mais seguras e sustentáveis no futuro da inteligência artificial.

Aqui trataremos especificamente da questão de uso de LLMs para a avaliação, porém você pode ver a Seção [Uso Responsável e Boas Práticas](#) e o Capítulo [Questões éticas em IA e PLN](#) deste livro para mais discussões sobre o uso responsável de IA.

4.6.1 Responsabilidade na avaliação

A implantação generalizada de LLMs levanta questões críticas sobre a responsabilidade quando esses modelos falham ou causam danos. Robustos *frameworks* de avaliação são essenciais para verificar se um LLM atende a padrões de confiabilidade, ética e desempenho em aplicações reais. A avaliação deve, portanto, ser uma parte integral do ciclo de vida do desenvolvimento do LLM, e não apenas uma análise *post-hoc* (Meva; Kukadiya, 2025).

4.6.2 Transparência e Interpretabilidade

A “caixa preta” dos LLMs, em que as decisões e as saídas são difíceis de interpretar, é um desafio significativo. A necessidade de entender por que um modelo toma certas decisões ou gera certas saídas é fundamental para a confiança e a responsabilidade social (Sánchez, 2024). A transparência contextual, por exemplo, pode envolver a inclusão de disclaimers sobre a natureza probabilística das respostas ou a indicação das fontes utilizadas.

4.6.3 Dualidade do uso

A capacidade dos LLMs de gerar conteúdo complexo e convincente também levanta preocupações sobre seu uso malicioso. A “dualidade do uso” refere-se ao potencial de modelos, originalmente projetados para fins benéficos, serem manipulados para gerar conteúdo prejudicial, como discurso de ódio, golpes ou até mesmo código malicioso para ataques cibernéticos. A avaliação deve, portanto, incluir a análise da suscetibilidade dos modelos a tais manipulações e o desenvolvimento de estratégias de mitigação de seu uso malicioso.

4.6.4 O papel da regulação e de padrões na avaliação

A rápida evolução dos LLMs exige o desenvolvimento de padrões e regulamentações para guiar sua avaliação e implantação responsáveis. Há uma necessidade crescente de protocolos de avaliação mais padronizados e transparentes para permitir comparações significativas e rastrear o progresso ao longo do tempo (Meva; Kukadiya, 2025; National Institute of Standards and Technology, 2024). A privacidade dos dados, a reprodutibilidade dos resultados e a capacidade de realizar avaliações sem utilizar recursos de terceiros (*on-premise*) são considerações importantes em contextos industriais ou governamentais (National Institute of Standards and Technology, 2024).

4.6.5 Direções futuras

O futuro da avaliação de LLMs é definido tanto por oportunidades de avanço quanto por desafios fundamentais que permanecem não resolvidos. As direções prioritárias incluem:

Estabelecimento de meta-avaliação

(Evaluating the Evaluators) Talvez a direção mais crítica seja a investigação sobre a própria validade das métricas e dos *frameworks* de avaliação. A questão de “quando uma avaliação é boa o suficiente?” ainda carece de uma resposta definitiva. A pesquisa futura deve se concentrar no desenvolvimento de métodos rigorosos para validar os próprios sistemas de avaliação, garantindo que suas conclusões sejam robustas, replicáveis e verdadeiramente indicativas do desempenho do modelo.



Avaliação contínua e em tempo real

Existe uma lacuna significativa entre a avaliação em ambientes de pesquisa e as necessidades da indústria (Heller, 2024). A implementação de *frameworks* que permitam a avaliação contínua e em tempo real é uma necessidade urgente para monitorar o desempenho de modelos em produção, detectar degradações (*drifts*) e garantir a segurança. Superar os obstáculos técnicos e de custo para viabilizar essa monitorização é um dos principais focos para a aplicação prática e responsável de LLMs (Meva; Kukadiya, 2025).

Avaliação adaptativa e personalizada

Métricas e abordagens que se adaptam a diferentes domínios, casos de uso e preferências do usuário (Meva; Kukadiya, 2025; PrototypeJam, 2025).

Desenvolvimento de métricas alinhadas com valores humanos

O objetivo de alinhar LLMs com “valores humanos” é mais complexo do que simplesmente buscar concordância com avaliadores humanos. A pesquisa deve abordar a distinção filosófica entre mimetizar as preferências e os vieses de um grupo de anotadores e alinhar-se a princípios éticos mais amplos e universais. Isso envolve não apenas o desenvolvimento de métricas mais sofisticadas, mas também uma reflexão sobre quais valores devem ser codificados e como (Meva; Kukadiya, 2025).

Avaliação de impacto social a longo prazo

Uma avaliação ampla, especialmente tratando-se do uso de IA em ambientes governamentais, deve considerar o impacto social e ético dos LLMs em um horizonte de tempo mais longo (Meva; Kukadiya, 2025; National Institute of Standards and Technology, 2024).

Abordagem multidisciplinar

A complexidade dos LLMs exige uma abordagem que transcende a ciência da computação. A avaliação deve ser inerentemente multidisciplinar, integrando insights da linguística, psicologia, ética e ciências sociais para criar uma compreensão holística do desempenho técnico dos modelos e do seu impacto social a longo prazo. Isso inclui o desenvolvimento de avaliações adaptativas, que se ajustem a diferentes contextos culturais, domínios de aplicação e casos de uso específicos (Jiao et al., 2024; Meva; Kukadiya, 2025).

4.7 Considerações finais

Neste capítulo, abordamos a avaliação de Grandes Modelos de Linguagem (Large Language Models), um campo dinâmico e multifacetado, que exige ir além das abordagens convencionais da linguística computacional e do processamento de linguagem natural. O avanço desses modelos trouxe novos desafios metodológicos, éticos e epistemológicos, forçando a revisão de pressupostos sobre o que significa avaliar o desempenho em sistemas que produzem linguagem de forma aberta e contextual.

As métricas automáticas tradicionais continuam a oferecer um ponto de partida valioso, sobretudo pela rapidez e pela comparabilidade que proporcionam. No entanto, sua limitação em capturar aspectos semânticos e, especialmente, pragmáticos torna indispensável a incorporação de camadas adicionais de análise. A avaliação humana, ainda que cara e limitada em escala, permanece o padrão-ouro por sua capacidade de julgar nuances, contextos e valores universais e éticos.

Nesse novo cenário, o uso de modelos como avaliadores, o paradigma LLM-as-a-Judge, representa uma via promissora. Ao unir a eficiência da automação com a profundidade interpretativa dos julgamentos humanos, essas abordagens híbridas apontam para um futuro de avaliações mais contínuas, contextualizadas e adaptativas. No entanto, desenvolver um LLM-juiz justo e eficiente demanda tempo, meta-avaliação e supervisão humana constante. Além disso, o LLM-juiz não é garantia de alinhamento da avaliação a valores humanos, ele é apenas um método ‘mais alinhado’ do que métodos tradicionais.

O desafio atual é consolidar *frameworks* eficientes, robustos e rápidos capazes de equilibrar essas três dimensões: automação, julgamento humano e meta-avaliação por modelos, de forma ética, transparente e alinhada a valores humanos. Somente assim será possível garantir que a avaliação de LLMs cumpra não apenas sua função técnica, mas também seu papel social: orientar o desenvolvimento de sistemas de linguagem que sejam realmente úteis, justos e responsáveis. O assunto parece muito complexo e é: a avaliação de sistemas de inteligência artificial precisa ser factível, confiável, tecnicamente viável e socialmente



responsável. Da mesma forma que o estado da arte da arquitetura de modelos evolui e continua evoluindo, o estado da arte de como avaliá-los está em constante evolução.



<https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2/>



Capítulo 5

Geração Aumentada por Recuperação (RAG)

André Carvalho

Livy Real

Altigran Soares da Silva

Publicado em: 16/04/2026

5.1 Fundamentos da Geração Aumentada por Recuperação

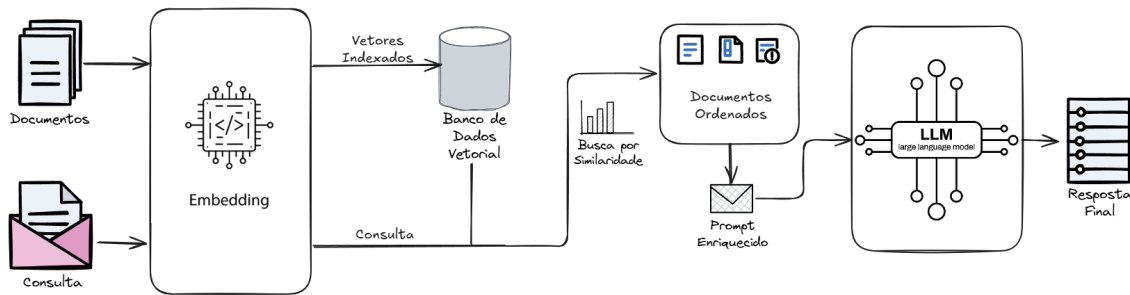
Grandes Modelos de Linguagem, ou ainda, Modelos de Linguagem de Grande Escala (LLMs), representam um marco na Inteligência Artificial, demonstrando uma capacidade sem precedentes de ‘compreender’, gerar e interagir usando a linguagem humana. No entanto, apesar de sua sofisticação, esses modelos são inerentemente limitados por sua arquitetura e pelo processo de treinamento. A Geração Aumentada por Recuperação (RAG) surge não como um aprimoramento incremental, mas um paradigma de uso dos LLMs em resposta às limitações dos modelos de fundação (*foundation models*)¹ Para compreender a importância da RAG, é crucial primeiro dissecar os quatro principais desafios que ela se propõe a resolver:

- **Conhecimento Desatualizado (*Knowledge Cutoff*):** O treinamento de um LLM é um processo computacionalmente intensivo e caro, que consome vastos recursos e tempo. Consequentemente, o conhecimento codificado em um modelo é estático e congelado no tempo, correspondendo à data em que seus dados de treinamento foram coletados. Esse “ponto de corte de conhecimento” (*knowledge cutoff*) significa que para dar resultados relativos a certos temas a LLM precisa de conteúdos sobre os eventos subsequentes ao seu treinamento, tornando-o inadequado para aplicações que exigem informações atuais ou em tempo real. (Cheng et al., 2024)
- **Alucinações:** Um dos riscos mais significativos associados aos LLMs é sua tendência a “alucinar”, ou seja, gerar informações que são plausíveis, bem escritas e contextualmente adequadas, mas factualmente incorretas ou completamente fabricadas. Isso ocorre porque os LLMs são modelos probabilísticos otimizados para gerar a sequência de palavras mais provável, não a mais verdadeira. Deste modo, sem o controle necessário, os *foundation models* tendem a confeccionar fatos, combinar informações e apresentar referências inválidas, gerando respostas potencialmente divergentes da realidade (Kalai et al., 2025). Em domínios críticos como saúde, finanças ou direito, as alucinações representam um risco inaceitável para a confiança e a segurança do uso desses modelos. (Huang et al., 2024b)
- **Falta de Especificidade de Domínio:** LLMs de propósito geral, treinados em dados disponíveis publicamente, carecem de conhecimento profundo sobre domínios especializados ou dados proprietários de uma organização. Assim, em nível de aplicação, a especificidade de domínio é responsável por delimitar o tema, caracterizando quais detalhes são pertinentes ao contexto. Os LLMs desconhecem documentos internos, bases de dados de clientes, manuais técnicos específicos ou a terminologia de um nicho de mercado. Sem acesso a essa informação, suas respostas permanecem genéricas e de valor limitado para casos de uso empresariais.
- **Falta de Verificabilidade (Black Box):** A natureza de “caixa-preta” dos LLMs torna extremamente difícil, se não impossível, rastrear a origem de uma informação específica gerada. Essa opacidade impede a verificação de fatos, a atribuição de fontes e a auditoria das respostas do modelo — requisitos cruciais em ambientes que exigem precisão como o acadêmico, o jornalístico e o corporativo, onde a confiabilidade e a transparência são primordiais. (Lewis et al., 2020b)

¹O termo “modelos de fundação” foi criado pelo *Stanford Institute for Human-Centered Artificial Intelligence* (HAI) <https://hai.stanford.edu/news/introducing-center-research-foundation-models-crfm> em 2021, para designar os algoritmos de redes neurais profundas treinados para atender tarefas gerais.



5.1.1 Introdução à RAG

Figura 5.1: Exemplo de arquitetura RAG usando vetores de *embeddings*.

A Geração Aumentada por Recuperação (RAG) (Lewis et al., 2020b) é um paradigma de integração entre modelos de linguagem e sistemas de recuperação de informação que aprimora a capacidade dos LLMs ao conectá-los a conteúdo externo e autoritativo antes de gerar uma resposta. Em sua essência, a RAG transforma o LLM de um sistema de “livro fechado”, que depende exclusivamente de sua memória interna (os parâmetros aprendidos durante o treinamento), para um sistema de “livro aberto”, que pode consultar fontes externas em tempo real para formular suas respostas.

Essa mudança de paradigma é sutil, mas profunda. Em vez de tentar incluir e adaptar novos conhecimentos como pesos de um modelo, algo que não garante o efetivo aprendizado de domínio, a RAG desacopla o processo de raciocínio linguístico do processo de aquisição de conhecimento. Enquanto o LLM mantém a função principal de gerar conteúdo, a tarefa de selecionar material relevante é delegada ao módulo de recuperação de informação. Deste modo, o módulo de geração compartilha a função de identificar e alinhar o contexto relevante com um módulo especializado nesta tarefa.

Esta abordagem acaba por atacar as principais limitações do uso de LLMs até então, como mostrado no Quadro 5.1.

Quadro 5.1: Limitações dos LLMs e a abordagem RAG.

Limitação do LLM	Descrição da limitação	Como a RAG aborda
Conhecimento desatualizado	LLMs têm uma “data de corte” e desconhecem eventos recentes.	Recupera informações atuais de bases de conhecimento externas.
Alucinações	Geram informações plausíveis, mas factualmente incorretas.	Ancora a resposta do LLM em fatos recuperados.
Falta de especificidade	Carecem de conhecimento de domínios específicos ou dados proprietários.	Conecta o LLM a bases de conhecimento especializadas.
Falta de verificabilidade	É difícil rastrear a fonte da informação gerada.	Permite fazer referência às fontes a partir dos documentos recuperados.

A análise do termo “*Retrieval-Augmented Generation*” revela as três etapas sequenciais que definem seu funcionamento:

1. **Recuperação (*Retrieval*)**: Dada uma consulta do usuário, o sistema primeiro aciona um mecanismo de busca para encontrar e extrair os trechos de informação mais relevantes de um *corpus* de conhecimento externo. Este *corpus* pode ser a internet, uma coleção de documentos proprietários, um banco de dados ou qualquer outra fonte de informação.
2. **Aumentada (*Augmentation*)**: As informações recuperadas são então usadas para “aumentar” o *prompt* original do usuário. Tipicamente, o contexto recuperado é inserido, descrito e estruturado

em um *template* de *prompt* juntamente com a pergunta, fornecendo ao LLM os fatos necessários para formular uma resposta informada.

3. **Geração (*Generation*):** Finalmente, o *prompt* aumentado é enviado ao LLM. O modelo, então, sintetiza uma resposta coerente e em linguagem natural, baseando-se tanto no contexto factual recém-fornecido quanto em seu conhecimento pré-treinado sobre linguagem, estrutura e raciocínio geral.

Essa arquitetura de dois estágios (recuperação e geração) permite que os sistemas RAG produzam respostas que não são apenas fluentes e contextualmente apropriadas, mas também factualmente precisas, atualizadas e verificáveis.

Contudo, os benefícios da RAG transcendem a melhoria técnica da qualidade das respostas, oferecendo vantagens estratégicas que impulsionam sua adoção em ambientes corporativos. A arquitetura RAG permite que as organizações aproveitem o poder dos LLMs de forma mais segura, econômica e personalizada.

- **Melhora da Precisão e Redução de Alucinações:** Ao ancorar as respostas em conteúdo recuperado, a RAG aumenta drasticamente a factualidade e mitiga o risco de o LLM gerar informações incorretas. A adoção da arquitetura torna a resposta potencialmente fundamentada em uma fonte específica, em vez de depender apenas do conhecimento parametrizado do modelo.
- **Acesso a Informações Atualizadas:** A base de conhecimento de um sistema RAG pode ser atualizada continuamente e em tempo real. Isso permite que o LLM responda a perguntas sobre eventos recentes ou conteúdos dinâmicos, como o status de um pedido ou as últimas notícias do mercado, superando as limitações impostas pelo uso de conhecimento desatualizado (*knowledge cutoff*).
- **Confiabilidade e Verificabilidade:** Como as respostas são baseadas em documentos específicos, os sistemas RAG podem facilmente citar suas fontes. Essa característica permite identificar e quantificar quais são os aspectos positivos e negativos na resposta. Assim, ao rastrear a origem da informação, a RAG aumenta a confiança do usuário e permite a auditoria e verificação dos fatos, algo essencial em aplicações que exigem segurança e responsabilidade.
- **Custo-Benefício:** Manter um LLM atualizado através de re-treinamento ou fine-tuning contínuo é proibitivamente caro e demorado. Em contraste, atualizar o conteúdo usado em um sistema RAG é uma operação de baixo custo e alta velocidade. Isso torna a RAG uma abordagem muito mais econômica para manter o conhecimento do sistema atualizado.
- **Personalização e Segurança de Dados:** A RAG é um método bastante adequado para adaptar LLMs a domínios específicos usando dados proprietários. As informações sensíveis de uma organização (dados de clientes, documentos internos, propriedade intelectual) permanecem em uma base de conhecimento controlada e segura, sendo acessadas apenas no momento da consulta, sem nunca serem usadas para treinar o modelo base de terceiros. Isso oferece um forte controle sobre a privacidade e a segurança dos dados, especialmente se os modelos usados forem internos.

A confluência desses benefícios estratégicos resultou em uma rápida e ampla adoção da RAG no mercado. Embora seja difícil obter uma estatística única e precisa sobre o uso da RAG, as tendências em tecnologias adjacentes e a adoção de produtos baseados em RAG pintam um quadro claro de sua importância. Por exemplo, a Databricks, que atende mais de 300 empresas da Fortune 500, lista das maiores empresas dos Estados Unidos, relatou que, em sua plataforma no último ano, houve um crescimento de 377% no uso de bancos de dados vetoriais², um componente central da RAG que visitaremos em breve. A adoção massiva de copilotos empresariais, como o Microsoft 365 Copilot, por quase 70% das empresas da Fortune 500³, demonstra a demanda por sistemas que, como a RAG, conectam LLMs a dados corporativos internos. Esses indicadores mostram que a RAG não é uma tecnologia de nicho, mas sim um pilar fundamental da estratégia de IA gerativa em grandes empresas, impulsionada pela necessidade crítica de personalizar LLMs com dados proprietários de forma segura e econômica.

5.1.2 Arquitetura RAG

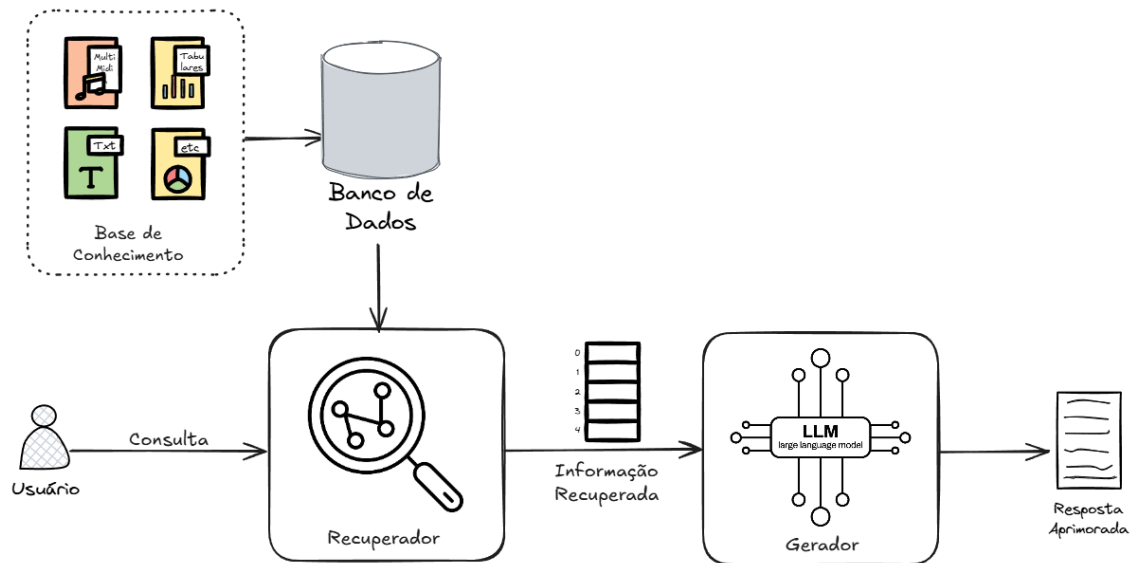
No nível mais fundamental, todos os sistemas RAG são compostos por dois pilares arquitetônicos: o *Recuperador* e o *Gerador*. A simplicidade dessa arquitetura modular é uma de suas maiores forças, permitindo que cada componente seja otimizado de forma independente.

²<https://www.databricks.com/discover/state-of-data-ai>

³<https://news.microsoft.com/source/latam/features/ia-pt-br/ignite-2024-por-que-quase-70-da-fortune-500-agora-usam-o-microsoft-365-copilot/>



Figura 5.2: Arquitetura Recuperador-Gerador em Sistemas RAG.



O **Recuperador** (*Retriever*) funciona como uma máquina de busca especializada. Sua principal responsabilidade é: dada uma consulta do usuário, encontrar e retornar a informação mais relevante de uma base de conhecimento pré-indexada. O recuperador atua como um filtro de conhecimento, navegando por um vasto repositório de dados para selecionar apenas alguns elementos que são cruciais para a tarefa em questão. Portanto, a base de conhecimento pode conter uma série de fontes de informação e estas, por sua vez, podem conter milhões itens de dados, por exemplo, documentos. A eficácia de todo o sistema RAG depende criticamente da qualidade desta etapa; se o recuperador falhar em encontrar o contexto correto, o gerador não terá a informação necessária para produzir uma resposta precisa, um princípio conhecido como “*garbage in garbage out*”.

Já o **Gerador** (*Generator*) é tipicamente um LLM pré-treinado. Geralmente, ele recebe um *template* de *prompt* de entrada preenchido com as regras de geração de resposta, a consulta original do usuário e o contexto fornecido pelo recuperador. A sua tarefa não é simplesmente extrair ou repetir a informação recuperada, mas sim realizar uma síntese inteligente. O gerador deve integrar a intenção da pergunta do usuário com os fatos específicos do contexto, utilizando seu conhecimento linguístico e de raciocínio para gerar uma resposta final que seja coerente, relevante e em linguagem natural. A habilidade de sintetizar, contextualizar e adaptar, em vez de apenas repetir, é o que diferencia um sistema RAG sofisticado de um simples sistema de busca e extração.

5.1.3 O componente Recuperador (*Retriever*)

O recuperador é, em sua essência, um sistema de busca. Sua função primária é vasculhar uma base de conhecimento, que pode ser uma coleção de documentos, uma base de dados ou a web, para encontrar e extrair os trechos (*chunks*) de texto mais relevantes para a consulta do usuário. Embora existam diversas técnicas de recuperação, a abordagem mais comum atualmente tem um foco predominante na **busca semântica**, que busca por significado e contexto, em vez de simples correspondência de palavras-chave.

5.1.3.1 Etapas do processo de recuperação

O funcionamento do recuperador pode ser dividido em quatro etapas principais, que preparam e executam a busca por informações.

- **Indexação e Divisão (*Chunking*)**: Antes de qualquer busca, a base de conhecimento precisa ser processada. No caso de documentos, estes são divididos em pedaços menores e gerenciáveis, chamados de *chunks*. Essa divisão é crucial, pois permite que a busca retorne trechos específicos e de maior aderência à pergunta do usuário, em vez de documentos completos, otimizando o contexto que será enviado ao gerador.

- **Codificação (*Embedding*):** Tanto a consulta do usuário quanto cada *chunk* de documento são transformados em uma representação vetorial numérica densa, conhecida como *embedding*. Esse processo é realizado por modelos de codificação (*encoders*) que capturam o significado semântico do documento em um espaço vetorial de alta dimensão. Tais modelos de linguagem especializados são denominados modelos de *embedding*.
- **Armazenamento:** Os *embeddings* dos *chunks* são armazenados e indexados em um banco de dados vetorial (*vector database*). Esses bancos de dados, como FAISS, Chroma ou Pinecone, são otimizados para realizar buscas de similaridade em alta velocidade em espaços vetoriais de milhões ou bilhões de vetores.
- **Busca por Similaridade:** Quando um usuário faz uma pergunta, o *embedding* da sua consulta geralmente é comparado com os *embeddings* dos documentos armazenados no banco de dados vetorial. O sistema então calcula a “distância” ou “proximidade” entre os vetores (usando métricas como a similaridade de cosseno) para encontrar os *chunks* mais similares semanticamente, que são então retornados como contexto.

5.1.3.2 Preparação de documentos e *Chunking*

O primeiro desafio em um *pipeline* RAG é lidar com o tamanho dos documentos. A maioria dos documentos do mundo real, como relatórios, artigos ou livros, excede em muito a janela de contexto dos LLMs, o número máximo de *token* que um modelo pode processar de uma só vez. Neste ponto, a RAG atua para evitar a adição de ruído no *prompt*, selecionando apenas os documentos ou trechos de documentos que são considerados relevantes a serem inseridos como contexto. Além disso, gerar um único *embedding* para um documento muito longo pode “diluir” a sua semântica, resultando em uma representação vetorial pouco informativa, ou uma massa gigantesca de texto que não corresponde bem ao necessário para consultas mais específicas.

A solução para este problema é a segmentação, ou *chunking*: o processo de dividir documentos longos em pedaços menores e mais gerenciáveis. A escolha da estratégia e do tamanho do *chunk* é uma das decisões de engenharia mais impactantes na qualidade de um sistema RAG. Existe um equilíbrio fundamental (Lita et al., 2024):

- ***Chunks* maiores:** Capturam mais contexto semântico, preservando a relação entre parágrafos e seções. No entanto, aumentam o risco de incluir informações irrelevantes (ruído) que podem confundir o LLM e são menos precisos para consultas muito específicas.
- ***Chunks* menores:** Oferecem representações mais focadas e com menos ruído, favorecendo a precisão em consultas granulares. Contudo, carregam menos contexto, o que pode dificultar a compreensão completa da informação se uma ideia estiver espalhada por múltiplos *chunks*.

Existem diversas estratégias de *chunking*, cada uma com seus próprios trade-offs (Lita et al., 2024):

- ***Chunking* fixo (*Fixed-size*):** A abordagem mais simples, que consiste em dividir o texto em segmentos de tamanho fixo. Em geral pelo número de *tokens*, caracteres ou palavras, com uma sobreposição (*overlap*) opcional para manter a continuidade e evitar a perda de informação com os cortes. É rápido e fácil de implementar, mas pode quebrar sentenças ou ideias no meio, prejudicando a coesão semântica do conteúdo textual.
- ***Chunking* recursivo (*Recursive*):** Uma melhoria em relação ao *chunking* fixo. Tenta dividir o texto recursivamente usando uma lista hierárquica de separadores, como quebras de parágrafo ($\backslash n \backslash n$), quebras de linha ($\backslash n$) e espaços. Ele prioriza a manutenção da estrutura sintática do texto e só recorre a uma divisão de tamanho fixo como último recurso.
- ***Chunking* semântico (*Semantic*):** Esta técnica visa criar *chunks* que são semanticamente coerentes. Uma abordagem comum é agrupar sentenças com base na similaridade de seus *embeddings*. O processo parte da segmentação do texto, geralmente no nível das sentenças, com o agrupamento das partições por similaridade semântica com o *chunk* acumulado. As partições são agrupadas de acordo com um limiar de similaridade predefinido. Isso produz trechos mais coesos, mas é computacionalmente mais caro.
- ***Chunking* baseado em estrutura (*Structure-based*):** Aproveita a estrutura inerente do documento, como títulos, seções e subseções em arquivos HTML, Markdown ou LaTeX, para definir os limites dos *chunks*. Isso preserva a estrutura lógica do conteúdo, mas pode resultar em *chunks* de tamanho muito variável.



- *Chunking* via LLM (*Agentic Chunking*): Uma abordagem de ponta que utiliza um LLM para analisar o documento e decidir inteligentemente como segmentá-lo. O LLM pode ser instruído a criar resumos de seções ou a agrupar trechos que tratam do mesmo tópico. Aqui, é importante ressaltar, que a aplicação do LLM deve ser o mais literal possível, evitando modificações do texto original e a criação de textos divergentes. Além disso, embora produza de altíssima qualidade semântica, o custo computacional e financeiro é significativamente maior, podendo ser proibitivo em alguns contextos.

5.1.3.3 Representando documentos textuais com *embeddings*

Após a segmentação, cada *chunk* de texto precisa ser convertido em uma representação numérica que um computador possa entender e comparar. Por muito tempo, a busca de informações baseou-se em correspondência de palavras-chave, usando modelos como o *Bag-of-Words* (BoW), que ignoram a ordem e o contexto das palavras. Essa abordagem é limitada por problemas como sinônimos (palavras diferentes com o mesmo significado) e paráfrases (frases diferentes com a mesma intenção).

Os *embeddings* textuais mitigam ou tratam diretamente esse problema. Um *embedding* é um vetor denso de números de ponto flutuante que representa o significado semântico de um pedaço de texto em um espaço de alta dimensionalidade. Esses vetores são a representação interna de um modelo de aprendizado profundo, geralmente sendo os vetores dos pesos da penúltima camada oculta (*hidden layer*). Portanto, esse significado semântico dos termos é dado na codificação dos documentos nos modelos de *embeddings*, de acordo com a proximidade e coocorrência das palavras em um determinado contexto. Nesse espaço, textos com significados semelhantes terão vetores próximos uns dos outros, medidos por métricas como a similaridade de cosseno.

Esses vetores são gerados por modelos de linguagem especializados, tipicamente arquiteturas *encoder-only* como BERT, RoBERTa ou, mais comumente para RAG, modelos Sentence-BERT (Reimers; Gurevych, 2019), que são otimizados para produzir *embeddings* de sentenças e parágrafos de alta qualidade. O processo envolve passar um *chunk* de texto pelo modelo *encoder*, que gera um *embedding* para cada *token*. Em seguida, uma técnica de *pooling* é aplicada para agregar esses *embeddings* de *tokens* em um único vetor que representa o *chunk* inteiro. As estratégias de *pooling* mais comuns são:

- *Average pooling*: Calcular a média dos vetores de todos os *tokens*, resultando em uma representação global que suaviza as variações locais.
- *Max pooling*: Selecionar, para cada dimensão do vetor, o valor máximo observado entre os *embeddings* dos *tokens*, destacando as características mais salientes do texto.
- *Token Especial*: Utilizar diretamente o *embedding* associado ao *token* especial CLS (em modelos como o BERT), que é treinado para capturar o significado agregado de toda a sequência.

Na prática e na literatura, a abordagem mais usual é fazer a média dos vetores dos *tokens* (*Average pooling*).

Uma vez que todos os *chunks* foram convertidos em *embeddings*, eles precisam ser armazenados em um local que permita uma busca rápida e eficiente. Em casos de *embeddings* semânticos, é usual utilizar, por exemplo, bancos de dados vetoriais, projetados para busca rápida e eficiente em bases com milhões de vetores.

Você pode se aprofundar sobre *embeddings* no Capítulo 1 deste livro.

5.1.3.4 Mecanismo de recuperação

A etapa final do processo de recuperação é a busca em si. Existem duas famílias principais de algoritmos de recuperação, e as abordagens mais modernas as combinam.

- Recuperadores Esparsos (*Sparse Retrievers*): Estes são os algoritmos clássicos de recuperação de informação, baseados em representações vetoriais esparsas (onde a maioria dos valores é zero). Nessa estrutura vetorial, os documentos indexados são dados pelos vetores numéricos que geralmente indicam a frequência ou a frequência ponderada dos termos que os compõe. Um dos algoritmos mais proeminentes para este tipo de recuperação é o BM25. Ele funciona com base na correspondência de palavras-chave, dando maior peso a termos que são raros no *corpus* geral, mas frequentes em um documento específico. São excelentes para encontrar documentos que contêm palavras-chave exatas ou terminologia específica.
- Recuperadores Densos (*Dense Retrievers*): Estes são os recuperadores baseados nos *embeddings* semânticos discutidos anteriormente. Eles utilizam vetores densos (onde a maioria dos valores é não-zero) que, através da ocorrência em contextos similares, capturam significado conceitual dos



termos que compõe o texto. Sua principal vantagem é a capacidade de encontrar documentos relevantes mesmo que não compartilhem nenhuma palavra-chave com a consulta, contanto que o significado seja semelhante. São robustos a sinônimos, paráfrases e variações de vocabulário.

- Busca Híbrida (*Hybrid Search*): A abordagem de estado da arte reconhece que os recuperadores esparsos e densos têm forças complementares. A busca híbrida executa ambos os tipos de busca em paralelo e, em seguida, combina os resultados. Essa combinação aproveita a precisão lexical da busca por palavras-chave e a robustez conceitual da busca semântica. Para mesclar as duas listas de resultados, que possuem pontuações em escalas diferentes, algoritmos de fusão de *ranking*, como o *Reciprocal Rank Fusion* (RRF) (Cormack et al., 2009), são frequentemente utilizados para produzir uma única lista de resultados classificada de forma mais relevante.

O Capítulo **Recuperação de Informação** deste livro trata com atenção mecanismos de Recuperação de Informação.

5.1.4 O componente Gerador (*Generator*)

O mecanismo central da RAG é a engenharia de *prompt* dinâmica, também conhecida como “*prompt stuffing*”. O sistema monta e preenche estruturas predefinidas no *prompt*, tornando-o mais elaborado antes do envio para a LLM. Este *prompt* aumentado normalmente segue um *template* que combina as regras de geração de resposta, a consulta original realizada pelo usuário e o contexto recuperado. Um exemplo de *template* poderia ser:

Quadro 5.2: Exemplo de *prompt* aumentado em um sistema RAG

Você é um assistente especializado em análise de currículos acadêmicos e profissionais. Responda à pergunta do usuário com base no trecho de currículo fornecido.

Para isso, você receberá duas informações:

1. O *prompt* original enviado pelo usuário (que expressa o que ele deseja saber).
2. Um trecho do currículo acadêmico (seção).

Sua tarefa é:

- Analisar o trecho do currículo levando em consideração a pergunta;
- Responder com base apenas nas informações explícitas ou fortemente implícitas nesse trecho;
- Ignorar informações que não sejam pertinentes à pergunta;
- Não extrapolar para além do conteúdo disponível na seção fornecida.

Apresente sua análise de forma clara, objetiva e bem estruturada.

Prompt do Usuário: {prompt}

Trecho do Currículo: {context}

O exemplo acima foi o *prompt* utilizado no LattesRex(Darcio et al., 2025), um *chatbot* especialista em análises de currículos Lattes⁴. No exemplo, podemos ver as instruções, nas quais são definidas a tarefa, a função do modelo e o que deve conter na resposta. Os documentos coletados e a consulta do usuário são descritas e inseridas ao final do mesmo. Assim, a cada consulta do usuário, o *prompt* é modificado dinamicamente.

Este processo de aumento garante que o LLM tenha todas as informações factuais necessárias diretamente em sua janela de contexto no momento da geração.

O principal objetivo do *prompt* aumentado é forçar o LLM a realizar uma “geração ancorada” (grounded generation). Ao instruir explicitamente o modelo a basear sua resposta no contexto fornecido, o risco de alucinação é drasticamente reduzido. A resposta é “ancorada” no conteúdo factual dos documentos recuperados, em vez de ser puramente uma invenção probabilística baseada no conhecimento interno do modelo.

No entanto, a função do gerador vai muito além de uma simples estruturação de informação. A verdadeira habilidade de um sistema RAG sofisticado reside na capacidade do LLM de contextualizar a informação. O modelo não deve apenas repetir os trechos recuperados. Em vez disso, ele deve executar uma fusão complexa de três fontes de informação: (1) a intenção e os detalhes da consulta do usuário, (2) os fatos específicos contidos nos *chunks* recuperados, e (3) seu vasto conhecimento pré-treinado sobre gramática, estrutura, raciocínio e conhecimento geral do mundo.

⁴<https://github.com/Lucas-Darcio/LattesRex/>



Considere a seguinte consulta: “Compare os prós e contras das estratégias de *chunking* fixo e semântico”. O recuperador pode retornar dois *chunks*: um descrevendo o *chunking* fixo e outro descrevendo o *chunking* semântico. O processo de geração de resposta mais simples listaria ambas as descrições como resultado. Um gerador sofisticado, no entanto, usaria sua capacidade de raciocínio para criar uma estrutura comparativa organizando os pontos fortes e fracos de cada abordagem lado a lado. Essa análise de contexto, realizando tarefas de síntese e estruturação textual, são formas de tratamento do conteúdo que apenas um LLM poderoso é capaz de executar, e é o que eleva a RAG de uma ferramenta de busca para um verdadeiro sistema de síntese de informações. Portanto, a qualidade de um sistema RAG depende não apenas da precisão do recuperador, mas também da capacidade de entendimento do contexto do gerador.

5.2 RAG vs. *Fine-tuning*

Ao buscar adaptar um LLM para resolver problemas específicos, duas soluções principais emergem: RAG e *fine-tuning* (ajuste fino). Embora possam ser usadas em conjunto, a escolha entre elas envolve um *trade-off* fundamental relacionado ao objetivo, custo e natureza dos dados.

5.2.1 *Fine-tuning*

O *fine-tuning* consiste em continuar o treinamento de um modelo fundacional pré-treinado, utilizando um conjunto de dados específico do seu domínio. O objetivo é ajustar os pesos internos do modelo para que ele se especialize em uma tarefa ou em um estilo de comunicação particular.

Essencialmente, o *fine-tuning* é ideal para ensinar ao LLM **como** responder, permitindo ajustar o estilo, o tom, o formato da saída e garantir que o modelo siga instruções complexas e específicas do domínio ou caso de uso. Isso se alinha a uma máxima importante na comunidade de IA: o ajuste fino serve para ensinar **forma**, não sendo indicado para injetar **fatos** ao modelo. Tentar ensinar novos conhecimentos factuais exclusivamente por este método pode levar a alucinações e não garante que o modelo recupere a informação correta no momento certo.

Deste modo, o *fine-tuning* não é uma opção interessante para lidar com liberdade de escrita e as variações textuais de determinados domínios. Entretanto, tal abordagem pode ser aplicada para atuar com fatores e conhecimentos relativamente bem definido e estáticos. O conteúdo para esses casos deve sofrer poucas mudanças de acordo com o tempo pois, após o treinamento, o conhecimento do modelo fica “congelado” até um novo processo de ajuste. Assim, o sucesso do *fine-tuning* depende criticamente da qualidade e da curadoria do conjunto de dados de treinamento e, mesmo com técnicas de otimização como o LoRA (*Low-Rank Adaptation*) (Hu et al., 2021), o processo pode ser computacionalmente caro, exigindo recursos de hardware e tempo significativos.

5.2.2 RAG

Em contraste, a RAG não modifica os pesos internos do LLM. Em vez disso, ele enriquece as consultas para maximizar a qualidade dos resultados obtidos “em tempo de execução”, fornecendo-lhe informações externas relevantes para cada consulta.

A principal vantagem da RAG é sua capacidade de acessar **conhecimento dinâmico**. Basta atualizar a base de dados externa para que o LLM utilize as informações mais recentes, sem a necessidade de um caro e demorado re-treinamento. Ao “ancorar” a resposta do LLM em um contexto factual, a RAG reduz significativamente a probabilidade de o modelo inventar informações (alucinar).

Isso torna a RAG a abordagem ideal para domínios onde a **factualidade e a auditabilidade** são essenciais, como em aplicações de direito, apoio ao cliente ou análise de documentos técnicos. Como a resposta é gerada a partir de fontes explícitas, é possível citá-las, aumentando a confiança do usuário. A implementação geralmente envolve a orquestração de componentes como bancos de dados vetoriais e recuperadores, frequentemente facilitada por *frameworks* como LangChain ou LlamaIndex.

Em resumo, a escolha depende do problema: se o objetivo é ensinar um novo estilo ou comportamento, o *fine-tuning* é a ferramenta certa. Se o objetivo é garantir respostas factuais baseadas em um corpo de conhecimento dinâmico e específico, a RAG é a solução superior. Em muitos casos avançados, uma abordagem híbrida, onde um modelo *fine-tuned* para um determinado estilo utiliza RAG para acessar dados atualizados, pode oferecer o melhor dos dois mundos.



5.2.3 Abordagens híbridas

A discussão “RAG vs. *fine-tuning*” pode apresentar uma falsa dicotomia. Na prática, as arquiteturas de IA mais robustas não escolhem entre um ou outro, mas os combinam de forma sinérgica. É importante ao se aprofundar nas abordagens híbridas, entender que o refinamento das tarefas e a especialização dos *prompts* e modelos potencializa os resultados obtidos. Portanto, um exemplo de modelos híbridos é o uso de *fine-tuning* para otimizar o “motor” de raciocínio do LLM, enquanto a RAG fornece o “combustível” (dados) em tempo real. Essa combinação permite que o modelo não apenas acesse o conhecimento correto, mas também saiba como utilizá-lo da maneira mais eficaz.

Um exemplo prático seria um assistente de IA para médicos. Primeiramente, o LLM base passaria por um *fine-tuning* com um conjunto de prontuários médicos anonimizados e diálogos clínicos. O objetivo não seria memorizar os prontuários, mas sim aprender a estrutura da linguagem médica, entender abreviações e saber como resumir informações clínicas. Em seguida, quando um médico perguntasse “Resuma o histórico de hipertensão do paciente João Silva”, o sistema **RAG** entraria em ação, recuperando os trechos relevantes do prontuário eletrônico específico daquele paciente. Por fim, o LLM, já ajustado para “pensar como um médico”, receberia esse contexto e geraria um resumo conciso e clinicamente relevante.

As arquiteturas híbridas podem ser implementadas de várias maneiras, como a Sequencial (primeiro ajusta-se o modelo, depois aplica-se RAG) ou a Integrada (onde o modelo é treinado para otimizar tanto a recuperação quanto a geração), representando a vanguarda na personalização de LLMs.

5.3 RAG - Aspectos avançados

A arquitetura fundamental de RAG, que consiste em recuperar informações de uma base de conhecimento externa antes de gerar uma resposta, provou ser uma solução robusta para mitigar alucinações e superar a limitação do conhecimento estático dos modelos. No entanto, a implementação “ingênuas” ou tradicional deste paradigma, caracterizada por uma busca vetorial simples seguida de uma concatenação de contexto, revela rapidamente suas próprias limitações. A qualidade da resposta final é inexoravelmente limitada pela qualidade da recuperação inicial; uma abordagem de “tamanho único” para todas as consultas é ineficiente e subótima; e o foco exclusivo em dados textuais ignora a riqueza de informações presentes em formatos estruturados e multimodais.

Nesta seção aprofundaremos o ecossistema de técnicas avançadas de RAG, que transformam o *pipeline* linear e passivo em um processo de raciocínio dinâmico, adaptativo e autocrítico. A tese central é que a evolução da RAG transcende a mera recuperação de informações, movendo-se em direção a um paradigma onde o sistema introduz camadas sofisticadas de otimização, tomada de decisão inteligente, autoavaliação e raciocínio estruturado. Exploraremos como o *re-ranking* refina a precisão, como a fusão de consultas amplia o alcance da busca e como algoritmos inteligentes podem rotear, corrigir e até mesmo habilitar a autocrítica no processo de geração. Além disso, expandiremos as fronteiras da RAG para além do texto plano, investigando como grafos de conhecimento permitem o raciocínio complexo e como a multimodalidade integra visão e linguagem. Finalmente, abordaremos os desafios práticos da implementação em produção, desde as estratégias de segmentação de dados e avaliação rigorosa até as considerações éticas e de desempenho.

5.3.1 RAG em duas etapas: Re-ordenando resultados

A arquitetura de recuperação em sistemas RAG de larga escala é frequentemente projetada como um processo de duas etapas que equilibra a necessidade de velocidade em um vasto *corpus* com a exigência de alta precisão em um conjunto menor de *chunks* ou documentos candidatos. Este paradigma representa um trade-off fundamental entre a abrangência semântica e a profundidade contextual.

A primeira etapa, a **recuperação inicial**, é otimizada para velocidade e escalabilidade. Seu objetivo é pesquisar rapidamente em milhões ou até bilhões de documentos para encontrar um conjunto de candidatos potencialmente relevantes. Para isso, são utilizados modelos **bi-encoder**. Esses modelos geram representações vetoriais (*embeddings*) para a consulta e para todos os documentos da base de conhecimento de forma independente e antecipada. (Khattab; Zaharia, 2020) No momento da consulta, a recuperação se resume a uma busca de vizinhos mais próximos (*Approximate Nearest Neighbor* - ANN) no espaço vetorial, uma operação extremamente eficiente. O *bi-encoder* lança uma rede ampla e rápida, maximizando o *recall* ao capturar todos os documentos que são semanticamente próximos da consulta em um sentido geral. No entanto, como a consulta e o documento são processados separadamente, interações mais finas e contextuais entre eles são perdidas (Khattab; Zaharia, 2020).



Uma vez que um subconjunto menor e mais gerenciável de documentos candidatos é recuperado (por exemplo, os 20 a 50 melhores resultados), a segunda etapa, o **re-ranking**, entra em ação. O objetivo aqui é mudar o foco da velocidade para a precisão. Esta etapa utiliza modelos mais poderosos e computacionalmente mais caros, como os **cross-encoders**, para reavaliar e reordenar a lista de candidatos. O *cross-encoder* realiza uma análise profunda e mais lenta para verificar a relevância contextual precisa, maximizando a *precisão* no conjunto restrito. Ele reordena os documentos, empurrando os mais relevantes para o topo da lista antes de serem enviados ao gerador.

Essa arquitetura de duas fases é a chave para a eficiência e eficácia do *re-ranking*. O sistema não está apenas filtrando resultados; ele está mudando a própria definição de “relevância” entre as etapas. A transição ocorre de uma correspondência semântica geral e de baixo custo para uma correspondência contextual específica e de alto custo, combinando o melhor dos dois mundos para alcançar um desempenho superior que seria computacionalmente inviável com uma única abordagem.

5.3.1.1 Cross-Encoders

Cross encoders são mais adequados à tarefa de *re-ranking* após o conjunto inicial de candidatos ser recuperado. Enquanto um *bi-encoder* mede a similaridade entre duas representações independentes, um *cross-encoder* modela diretamente o alinhamento entre a consulta e o documento.

A mecânica de um *cross-encoder* funciona da seguinte maneira: o algoritmo recebe a consulta e um documento candidato como uma única sequência de entrada, tipicamente concatenados e separados por um *token* especial, como [SEP]. Essa sequência conjunta é então processada por um modelo Transformer. O mecanismo de auto-atenção do Transformer é aplicado a toda a sequência, permitindo que cada *token* da consulta “preste atenção” a cada *token* do documento e vice-versa, em todas as camadas do modelo. Essa interação profunda permite que o modelo capture dependências sutis, nuances contextuais e relações semânticas complexas que são impossíveis de serem modeladas quando a consulta e o documento são codificados separadamente. A saída do *cross-encoder* não são dois vetores, mas sim um único score de relevância, geralmente um valor entre 0 e 1, que representa a probabilidade de o documento ser relevante para a consulta.

O treinamento desses modelos é tipicamente supervisionado, formulado como uma tarefa de classificação binária. Utilizando grandes datasets de pares (consulta, documento) rotulados como relevantes ou não relevantes, o modelo é treinado para otimizar uma função de perda como a **Entropia Cruzada Binária (BCE)**. Esse processo de treinamento ensina o modelo a produzir uma pontuação alta para pares relevantes e uma pontuação baixa para pares irrelevantes.

O poder do *cross-encoder* vai além de simplesmente “processar juntos”. Ele reside na sua capacidade de modelar a **relevância assimétrica**. A relevância da informação é frequentemente direcional: um documento que descreve os efeitos colaterais de um medicamento é altamente relevante para a consulta “quais são os efeitos colaterais do medicamento X?”. No entanto, a consulta não é relevante para o documento. Os *bi-encoders*, que medem a similaridade simétrica (como a similaridade de cosseno, onde $\text{sim}(\mathbf{A}, \mathbf{B}) = \text{sim}(\mathbf{B}, \mathbf{A})$), têm dificuldade em capturar essa nuance. No caso do *cross-encoder*, ao ser treinado com os pares de pergunta e resposta candidata, se torna um avaliador especializado nessa assimetria. Portanto, para responder à pergunta “este documento é relevante para esta consulta?”, o algoritmo aprende a capturar essa assimetria fundamental, o que leva a uma avaliação de relevância muito mais precisa e dependente do contexto.

Quadro 5.3: Comparativo de arquiteturas de *Bi-Encoder* vs. *Cross-Encoder*

Característica	<i>Bi-Encoder</i>	<i>Cross-Encoder</i>
Formato de Entrada	Consulta e Documento processados separadamente.	Par (Consulta, Documento) concatenado como uma única entrada.
Processamento	Geração de <i>embeddings</i> independentes e comparação via função de similaridade (e.g., cosseno).	Processamento conjunto com mecanismo de atenção cruzada entre os <i>tokens</i> da consulta e do documento.



Saída	<i>Score</i> simétrico de similaridade (e.g., -1 a 1).	<i>Score</i> assimétrico de relevância (e.g., 0 a 1).
Custo de Indexação	Alto: requer o pré-cálculo e armazenamento de <i>embeddings</i> para todos os documentos na base.	Nulo: não há etapa de indexação.
Custo de Inferência	Baixo: 1 passagem do <i>encoder</i> para a consulta + busca vetorial rápida.	Alto: N passagens do <i>encoder</i> para re-ranquear N documentos candidatos.
Caso de Uso Principal	Recuperação inicial em larga escala (primeira fase).	<i>Re-ranking</i> de um conjunto pequeno de candidatos (segunda fase).
Vantagem Principal	Velocidade e escalabilidade para <i>corpus</i> massivos.	Alta precisão e compreensão contextual profunda.
Desvantagem Principal	Menor precisão contextual, pois perde interações finas.	Custo computacional alto, não escalável para a recuperação inicial.

5.3.1.2 Expansão de consultas com RAG-Fusion

Uma limitação inerente a qualquer sistema de busca é sua dependência da formulação exata da consulta do usuário. Uma única consulta pode ser ambígua, subespecificada ou simplesmente não usar a terminologia que melhor corresponde aos documentos relevantes na base de conhecimento. A técnica **RAG-Fusion** aborda diretamente esse problema, reconhecendo que a intenção do usuário pode ser mais bem capturada por múltiplas perspectivas (Rackauckas, 2024).

O processo de RAG-Fusion funciona em três etapas principais:

1. **Geração de múltiplas consultas:** Em vez de depender apenas da consulta original, RAG-Fusion primeiro utiliza um LLM para gerar múltiplas variações daquela consulta. Através de um *prompt* cuidadosamente elaborado, o LLM é instruído a criar diversas perguntas que abordam a mesma intenção subjacente, mas de diferentes ângulos ou com diferentes palavras-chave. Por exemplo, para a consulta “impacto da IA na economia”, o LLM poderia gerar: “como a inteligência artificial afeta o mercado de trabalho?”, “automação e crescimento do PIB” e “riscos econômicos da IA”.
2. **Recuperação paralela:** O sistema então realiza buscas vetoriais para cada uma dessas consultas geradas, bem como para a consulta original, em paralelo. Cada consulta atua em diferentes regiões do espaço vetorial, recuperando conjuntos de documentos que, embora todos relevantes, podem ser distintos entre si.
3. **Fusão e re-ranking:** Finalmente, os conjuntos de resultados de todas as buscas são agregados e re-ranqueados usando um algoritmo de fusão, mais comumente o *Reciprocal Rank Fusion* (RRF) (Rackauckas, 2024), para produzir uma única lista final de documentos classificados.

Essa abordagem transforma a recuperação de um ato de *matching* para um ato de *exploração*. Em essência, RAG-Fusion realiza uma forma de engenharia de *prompt* automática para a fase de recuperação. Em vez de exigir que o usuário refine sua pergunta, o sistema assume proativamente essa responsabilidade. Ele utiliza o poder generativo e a compreensão semântica do LLM para explorar o contexto ou entorno da busca realizada pelo usuário, em outras palavras, o “espaço latente de intenções”. Ao transformar uma única consulta pontual em uma distribuição de consultas, ele aumenta a área de superfície da busca, elevando significativamente a robustez e a probabilidade de encontrar documentos relevantes que uma única formulação teria perdido.

5.3.1.3 Fusão de diferentes ordenações: *Reciprocal Rank Fusion* (RRF)

Quando múltiplas listas de resultados precisam ser combinadas — seja na RAG-Fusion, que usa múltiplas consultas, ou na busca híbrida, que combina busca vetorial e busca por palavra-chave (como BM25) — surge o desafio de como fundir os *rankings* gerados por cada método. Os scores produzidos por diferentes sistemas de busca (e.g., similaridade de cosseno, *score* BM25) estão em escalas completamente diferentes e não são diretamente comparáveis (Rackauckas, 2024). Tentar normalizá-los pode ser complexo e frágil.



O **Reciprocal Rank Fusion (RRF)** é um algoritmo que resolve esse problema ao ignorar completamente os scores brutos e focar apenas na posição (rank) dos documentos em cada lista. O RRF, opera como um algoritmo de *ensemble voting*, onde a ideia de voto é aplicada sob a premissa de que a posição de um documento em uma lista de resultados é um sinal de relevância mais confiável e universal do que o score numérico que o gerou. Portanto, o RRF atua como um sistema de votação ponderada pela confiança posicional.

Para cada documento d que aparece em uma ou mais listas de resultados R , o RRF calcula uma pontuação combinada usando a seguinte fórmula [1, 10]:

$$Score(d) = \sum_{r \in R} \frac{1}{k + rank_r(d)} \quad (5.1)$$

Onde:

- $rank_r(d)$ é a posição do documento d na lista de resultados do ranqueador r , começando com o rank 1 para o primeiro item.
- k é uma constante de suavização, comumente definida como 60.

A pontuação final de um documento é a soma de suas pontuações RRF em todas as listas em que ele aparece. Documentos que aparecem consistentemente em posições altas em múltiplas listas recebem as maiores pontuações combinadas e, portanto, são promovidos ao topo do *ranking* final.

O papel da constante k é crucial. Ela amortece a contribuição dos documentos de alto escalão, garantindo que eles não sejam excessivamente dominantes, e dá mais peso a documentos de *ranking* mais baixo do que uma simples fórmula de $1/rank$ daria. Sem k , a diferença de pontuação entre o rank 1 e 2 seria muito maior do que entre o rank 60 e 61. A constante k suaviza essa queda, permitindo que o algoritmo considere um espectro mais amplo de resultados. Essencialmente, k controla o quão cético o algoritmo é em relação às distinções de *ranking* muito altas.

Ao descartar os scores, o RRF democratiza a contribuição de diferentes recuperadores. Cada sistema (vetorial, *keyword* etc.) pode “votar” nos documentos com base em sua especialidade, com o peso do voto diminuindo com a posição. O RRF troca a magnitude do score pela consistência do *ranking*, buscando um consenso entre as diferentes fontes de evidência. Um documento que é bem ranqueado, incluído entre os dez primeiros nos resultados de três recuperadores diferentes é provavelmente mais relevante do que os primeiros documentos dado como resultado no *ranking* individual de cada recuperador, mas ignorado pelos outros.

5.3.2 RAG adaptiva e o Modelo agêntico

Após otimizar a camada de recuperação, a próxima fronteira na evolução da RAG é a introdução de inteligência e tomada de decisão no próprio *pipeline*. As arquiteturas tradicionais seguem um fluxo linear e fixo, tratando todas as consultas da mesma forma. Em contraste, os sistemas avançados podem adaptar sua estratégia com base na natureza da consulta, avaliar a qualidade dos documentos recuperados e até mesmo criticar suas próprias saídas. Esta seção explora técnicas que transformam a RAG de um processo mecânico para um fluxo de trabalho de raciocínio dinâmico e resiliente.

5.3.2.1 RAG adaptativo: Roteamento dinâmico de consultas

O princípio fundamental do **RAG adaptativo** é que nem todas as consultas tem as mesmas características e, portanto, não devem ser tratadas com o mesmo *pipeline* computacionalmente intensivo. Uma pergunta simples como “Olá, como vai você?” não requer uma busca em um banco de dados vetorial, enquanto uma pergunta sobre eventos recentes, como “Quem ganhou o jogo de ontem?”, não pode ser respondida por uma base de conhecimento estática (Jeong et al., 2024).

Para lidar com essa diversidade, a RAG Adaptativa introduz um componente de tomada de decisão no início do fluxo: o **Query Router** (Roteador de Consultas) (Yeo et al., 2025). Este roteador é um classificador leve que analisa a consulta do usuário e a direciona para o caminho de processamento mais eficiente e apropriado. Frequentemente, este roteador é implementado usando um LLM instruído a produzir uma saída estruturada, garantindo que a decisão seja baseada na compreensão semântica da consulta, e não apenas em palavras-chave.

O fluxo de trabalho típico de um RAG Adaptativo pode ter múltiplos ramos:



1. **Geração direta:** Para consultas simples, conversacionais ou que não requerem conhecimento externo, o roteador envia a pergunta diretamente ao LLM, pulando a etapa de recuperação. Isso economiza tempo e recursos.
2. **Recuperação na base de conhecimento (*Vector Store*):** Para perguntas que são factuais e se enquadram no domínio coberto pela base de conhecimento interna, o roteador aciona o *pipeline* RAG padrão, realizando uma busca vetorial nos documentos indexados.
3. **Busca na web ou recuperação complexa:** Para perguntas sobre eventos recentes, tópicos fora do escopo da base de conhecimento ou questões complexas que podem se beneficiar de múltiplas fontes, o roteador direciona a consulta para uma ferramenta de busca na web ou aciona um fluxo mais complexo, como a RAG-Fusion.

A implementação prática envolve definir um modelo de dados que especifica as rotas possíveis e fornecer um *prompt* de sistema ao LLM que o instrui a atuar como um roteador especialista.

Ao introduzir essa camada de decisão, a RAG Adaptiva implementa um conceito de **gerenciamento de recursos computacionais** no nível da aplicação. Ele trata a recuperação de informações não como uma etapa obrigatória, mas tal qual uma ferramenta é conectada a um LLM, a recuperação é uma ferramenta a ser invocada apenas quando necessário e na intensidade apropriada. Isso otimiza não apenas a relevância da resposta, mas também a latência e o custo, transformando o *pipeline* RAG de uma arquitetura de dados estática para uma arquitetura de computação dinâmica e orientada por custos.

5.3.2.2 *Corrective* RAG (CRAG): Um *framework* para a robustez

Enquanto a RAG Adaptiva otimiza o fluxo *antes* da recuperação, o **Corrective RAG (CRAG)** (Yan et al., 2024) foca em lidar com o problema fundamental da RAG: o que fazer quando a recuperação falha e retorna documentos irrelevantes ou de baixa qualidade? A recuperação incorreta é uma das principais causas de alucinações, pois o LLM é forçado a gerar uma resposta com base em informações erradas.

O CRAG introduz um mecanismo de auto-correção e verificação de qualidade *após* a recuperação. O componente central é um **avaliador de recuperação (*retrieval evaluator*)** leve, projetado para avaliar a relevância dos documentos recuperados em relação à consulta. Este avaliador, tipicamente um modelo *encoder* menor, como o T5-large. Esse modelo é escolhido com cautela, ponderando a qualidade da resposta, a latência da requisição e o tempo de resposta com o papel de avaliador. Assim, o modelo de avaliação é ajustado para prever um score de confiança para cada par (consulta, documento).

Com a aplicação do avaliador, é comum o aumento da quantidade de documentos ou *chunks* recuperados. Ao definir a relevância de acordo com o contexto, o *score* de confiança também é responsável pela redução de nível de ruído e refinamento do conteúdo que efetivamente passado para a geração da resposta. Com base nesse *score* de confiança, o CRAG aciona dinamicamente uma de três ações distintas, implementando um mecanismo de *fallback* e enriquecimento em cascata:

1. **Confiança alta (“Correct”):** Se pelo menos um documento recuperado obtiver uma pontuação acima de um limiar superior, o sistema considera a recuperação bem-sucedida. Os documentos são considerados relevantes e são usados para a geração. Opcionalmente, para aumentar ainda mais a precisão, um processo de **refinamento de conhecimento** pode ser aplicado. Este processo, chamado de “decompose-then-recompose”, quebra os documentos em “*knowledge strips*” (pequenos trechos de informação) e usa o mesmo avaliador para filtrar apenas os trechos mais pertinentes antes de passá-los ao gerador.
2. **Confiança baixa (“Incorrect”):** Se todos os documentos recuperados tiverem uma pontuação abaixo de um limiar inferior, o sistema conclui que a recuperação falhou. Os documentos recuperados são descartados para evitar a contaminação do contexto. Em vez disso, o CRAG recorre a uma fonte de conhecimento alternativa e mais ampla, como uma **busca na web**, para encontrar informações mais relevantes.
3. **Confiança média (“Ambiguous”):** Se as pontuações dos documentos ficarem entre os dois limites, o sistema reconhece a incerteza. A recuperação não é nem claramente correta nem claramente incorreta. Nesta situação, o CRAG adota uma abordagem híbrida e robusta: ele combina o conhecimento refinado dos documentos recuperados (ação “Correct”) com o conhecimento obtido da busca na web (ação “Incorrect”), maximizando a cobertura e a probabilidade de fornecer ao LLM um contexto útil.

O CRAG modela explicitamente a incerteza da recuperação. Em vez de falhar silenciosamente com informações ruins, o sistema reconhece a incerteza, quantifica-a com um score de confiança e aciona a



política de mitigação mais apropriada: refinar os documentos obtidos, substituir por conteúdos mais apropriados segundo o contexto ou aumentar com informações adicionais. É um sistema de gerenciamento de risco para a recuperação de informações, aumentando significativamente a robustez e a confiabilidade do *pipeline* RAG.

5.3.2.3 Self-RAG: Incorporando autonomia com *tokens* de reflexão

O Self-RAG (*Self-Reflective Retrieval-Augmented Generation*) (Asai et al., 2024) representa um passo adiante na autonomia dos sistemas RAG, internalizando o *pipeline* de controle (que em CRAG e RAG Adaptativo é composto por módulos externos) dentro do próprio processo de geração de linguagem do LLM. Ele faz isso treinando o LLM gerador para produzir “*tokens* de reflexão” — *tokens* especiais, adicionados ao seu vocabulário, que controlam o processo de recuperação e autocrítica em tempo real.

Esses *tokens* transformam a recuperação e a crítica de operações de *pipeline* em atos de linguagem. O LLM não está mais sendo passivamente “aumentado” por um processo externo; ele está ativamente narrando e controlando sua própria busca por conhecimento e verificação de fatos como parte integrante da tarefa de geração de texto.

Os *tokens* de reflexão são categorizados em dois tipos principais:

1. ***Tokens* de recuperação:** Em cada etapa da geração, o modelo pode decidir se a recuperação de informações externas seria útil. Ele manifesta essa decisão gerando um *token* específico:

- [Retrieve]: Indica que a recuperação é necessária. A geração é pausada, uma busca é realizada e os documentos recuperados são apresentados ao modelo.
- [No Retrieve]: Indica que a recuperação não é necessária e o modelo pode continuar a geração com base em seu conhecimento paramétrico ou no contexto já disponível.

Isso permite uma **recuperação adaptativa e sob demanda**, que pode ocorrer múltiplas vezes durante a geração de uma única resposta, ou não ocorrer de todo, dependendo da necessidade.

2. ***Tokens* de crítica:** Após a recuperação (se acionada) e durante a geração, o modelo avalia a qualidade da informação e de sua própria saída:

- [Relevant] / [Irrelevant]: O modelo avalia cada documento recuperado, gerando um *token* para indicar sua relevância para a continuação da resposta.
- [IsSupported] / [IsNotSupported]: Ao gerar um segmento de texto baseado nos documentos recuperados, o modelo avalia se sua própria afirmação é factualmente suportada pela evidência fornecida.
- [Good] / [Bad]: O modelo pode fazer uma avaliação holística da qualidade geral da resposta.

O treinamento do Self-RAG envolve a criação de um *corpus* de treinamento onde esses *tokens* de reflexão são inseridos *offline* por um modelo “crítico” independente. Portanto, a função do modelo crítico é realizar uma anotação qualitativa dos resultados obtidos pelo fluxo principal. O LLM gerador é então treinado em um objetivo padrão de previsão do próximo *token*, aprendendo a prever tanto as palavras da resposta quanto esses *tokens* de controle.

Essa internalização do controle tem consequências profundas. O processo de raciocínio do modelo torna-se explícito e controlável na inferência. Por exemplo, os desenvolvedores podem ajustar o comportamento do modelo simplesmente alterando as probabilidades de amostragem dos *tokens* de reflexão, tornando-o mais ou menos propenso a recuperar informações ou mais ou menos crítico em relação às suas fontes. Self-RAG representa um passo fundamental em direção a modelos de linguagem com meta-cognição. Nesse aspecto, os modelos não apenas processam informações, como também refinam a geração de respostas com a otimização do processo de raciocínio do modelo.

5.3.3 Indo além do texto

A RAG tradicional opera em um mundo de *chunks* de texto independentes, tratando cada pedaço de informação como uma unidade isolada. Essa abordagem, embora eficaz para muitas tarefas, falha em capturar as ricas interconexões entre os conceitos e ignora a vasta quantidade de informações contidas em modalidades não textuais. Esta seção explora duas fronteiras que quebram essa limitação: GraphRAG, que introduz relações estruturadas para permitir um raciocínio mais complexo, e RAG Multimodal, que integra modalidades como imagens para criar uma compreensão mais holística do mundo.



5.3.3.1 GraphRAG: Recuperando informação de grafos de conhecimento

O **GraphRAG** (Pan et al., 2024) aborda uma limitação fundamental da RAG baseado em vetores: a incapacidade de responder a perguntas complexas que exigem a síntese de informações de múltiplos documentos ou a compreensão de relações indiretas. Em vez de tratar os *chunks* de texto como pontos independentes em um espaço semântico, o GraphRAG constrói um **grafo de conhecimento** onde os nós representam entidades (como pessoas, organizações, conceitos) e as arestas representam as relações explícitas entre elas.

O processo de construção do grafo a partir de dados não estruturados envolve várias etapas sofisticadas :

1. **Extração de entidades e relações:** O *corpus* de texto é processado para extrair entidades nomeadas e as relações entre elas. Isso pode ser feito usando LLMs para gerar tripletadas estruturadas do tipo (entidade1, relação, entidade2).
2. **Construção do grafo:** As entidades se tornam nós e as relações se tornam arestas direcionadas, formando um grafo que representa o conhecimento do *corpus* de forma estruturada.
3. **Deteção de comunidades:** Algoritmos de clusterização de grafos, como o algoritmo de Leiden, são aplicados para identificar “comunidades” de nós densamente conectados. Essas comunidades representam temas ou domínios de conhecimento dentro do *corpus*.
4. **Geração de resumos:** Para cada comunidade identificada, um LLM gera um resumo que descreve o tema central e as principais entidades e relações dentro daquele cluster. Isso cria uma visão hierárquica do conhecimento, do geral (resumos de comunidade) ao específico (nós individuais).

Essa estrutura de grafo habilita o **raciocínio de múltiplos saltos** (*multi-hop reasoning*), que é a capacidade de responder a perguntas atravessando múltiplas conexões no grafo. Considere a pergunta: “Como a aquisição do LinkedIn pela Microsoft impactou as tendências de networking profissional?”. Um RAG tradicional pode recuperar documentos sobre a aquisição e outros sobre networking, mas a conexão causal entre eles pode ser fraca ou inexistente nos textos recuperados. O LLM teria que inferir essa conexão, com alto risco de alucinação.

O GraphRAG, por outro lado, transforma a recuperação de um problema de busca semântica para um problema de travessia de grafo. Para responder à pergunta, o sistema pode iniciar no nó (**Microsoft**), seguir a aresta [**adquiriu**] para (**LinkedIn**), e então seguir a aresta [**influencia**] para (**Networking Profissional**). Ao longo desse caminho, ele coleta o contexto associado a cada nó e aresta, fornecendo ao LLM um contexto de raciocínio explícito. A recuperação não se trata mais de encontrar os melhores trechos de texto, mas sim de encontrar e aprofundar os conhecimentos sobre as entidades mais alinhadas ao contexto através do grafo de conhecimento. Isso permite que o sistema responda a perguntas sobre relações, causalidade e impacto, que estão muito além da capacidade da RAG tradicional.

Uma abordagem alternativa ao GraphRAG é o *Recursive Abstractive Processing for Tree-Organized Retrieval* (RAPTOR) (Sarthi et al., 2024). O RAPTOR apresenta uma proposta de sumarização recursiva da estrutura de clusters de documentos, definindo uma organização em árvore. Através desta árvore, é criado um raciocínio hierárquico e contextual. Ele é efetivo para assimilar contextos e reduzir a redundância das informações. Portanto, enquanto o GraphRAG tem uma informação bem-definida e relações entre os componentes de forma estruturada, os sumários do RAPTOR visam atuar em situações onde é necessário reduzir o ruído e otimizar a busca pelo contexto correto.

5.3.3.2 RAG multimodal: Integrando visão e linguagem

A RAG multimodal estende o paradigma para além do texto, incorporando outras modalidades de dados como imagens, áudio e vídeo. Uma das abordagens mais poderosas para alcançar isso é através do uso de modelos de *embedding* multimodais, como o **CLIP** (*Contrastive Language-Image Pre-training*) da OpenAI (Radford et al., 2021).

O CLIP é treinado em milhões de pares (imagem, texto) da internet com um objetivo de aprendizado contrastivo. O resultado é um modelo que pode projetar tanto imagens quanto textos em um **espaço vetorial compartilhado**. Nesse espaço, a distância entre os vetores corresponde à similaridade semântica, independentemente da modalidade original. O *embedding* da imagem de um cão estará semanticamente próximo do *embedding* do texto “uma foto de um cão”. Esse resultado é obtido com *embeddings* pois tanto o texto quanto a imagem do cão estão compartilhando o mesmo espaço vetorial, tornando essa associação possível. Consequentemente, isso permite uma busca intermodal robusta: usar uma imagem para encontrar textos relevantes, ou usar um texto para encontrar imagens relevantes.

O *pipeline* de um sistema RAG multimodal usando CLIP normalmente segue estas etapas :



1. **Ingestão e Indexação:** Todos os dados da base de conhecimento, tanto textos quanto imagens, são processados pelo *encoder* apropriado do CLIP (*encoder* multimodal de texto e imagem) para gerar *embeddings*. Esses *embeddings* multimodais são armazenados em um único banco de dados vetorial. As imagens podem também ter metadados textuais associados, como descrições, que são armazenados junto com os *embeddings*.
2. **Consulta multimodal:** O usuário pode fornecer uma consulta em forma de texto (e.g., “mostre-me jaquetas de couro”), uma imagem (e.g., uma foto de uma jaqueta que ele gostou), ou uma combinação de ambos. A consulta é convertida em um *embedding* usando o mesmo modelo CLIP.
3. **Recuperação intermodal:** O *embedding* da consulta é usado para realizar uma busca de similaridade no banco de dados vetorial. Como textos e imagens coexistem no mesmo espaço, a busca pode retornar os itens mais relevantes, apresentando resultados mistos entre os modais ou combinando imagens e textos em um único tipo de resultado.
4. **Geração aumentada:** Os dados recuperados (imagens e/ou textos) são então passados, juntamente com a consulta original, para um LLM multimodal (como o GPT-4 Vision). Este modelo pode então raciocinar sobre as informações visuais e textuais para gerar uma resposta rica e fundamentada.

Essa abordagem não está apenas recuperando imagens; ela está realizando uma tradução entre espaços de representações. O espaço de *embedding* compartilhado do CLIP atua como um intermediário para conceitos abstratos. Quando um usuário busca por “um vestido vermelho elegante”, o sistema não está fazendo uma correspondência de pixels ou palavras-chave; ele está buscando o conceito abstrato de “vestido vermelho elegante” no espaço vetorial, que pode ser representado tanto por uma imagem quanto por um texto. Isso permite que o conhecimento e a semântica dos *embeddings* sejam passados para o raciocínio realizado pelo LLM. Com isso, o LLM fundamenta a resposta gerada em correspondência textual e visual concretas. A resposta gerada, portanto, é resultante de uma composição rica e com evidências das imagens e nos textos da etapa de recuperação.

5.3.4 Avaliando RAG

A avaliação de um sistema RAG é uma tarefa complexa que não pode ser capturada por uma única métrica. Uma avaliação robusta deve analisar os dois componentes principais do *pipeline* de forma independente e conjunta: a **Recuperação** e a **Geração** (Es et al., 2024).

Métricas de Recuperação:

Estas métricas avaliam a eficácia do componente recuperador em encontrar os documentos mais relevantes.

- **Métricas clássicas:** **Precision@k** (proporção de documentos relevantes entre os k primeiros recuperados), **Recall@k** (proporção de todos os documentos relevantes que foram encontrados nos k primeiros) e **Mean Reciprocal Rank (MRR)** (média do inverso do rank do primeiro documento relevante) são padrões da área de recuperação de informação.
- **Métricas baseadas em LLM:** *Frameworks* modernos introduzem métricas mais sofisticadas que usam um LLM como juiz.
 - **Precisão contextual (Contextual Precision)** Esta métrica mede a proporção de *chunks* recuperados que são de fato relevantes para a pergunta do usuário. Em suma, ela responde: “O contexto que recuperamos está livre de ruído e focado no assunto da pergunta?” Uma baixa precisão contextual significa que o modelo está sendo sobrecarregado com informações irrelevantes, o que pode fazer a LLM geradora vincular informações de diferentes contextos ou divergir do contexto apresentado pelo usuário. A precisão contextual pode ser computada através da média da relevância dos *chunks* recuperados, onde a relevância normalmente é avaliada por uma LLM-como-juiz.
 - **Recall contextual (Contextual Recall)** Esta métrica mede a capacidade do sistema de recuperar toda a informação necessária para responder à pergunta. O Recall Contextual compara o conteúdo factual do contexto recuperado com uma resposta ideal (*ground truth*) e verifica se as informações contidas nessa resposta ideal também estão presentes no contexto que o sistema encontrou. Um recall baixo significa que o módulo de recuperação falhou em encontrar todos os fatos que fundamentam o conteúdo da resposta ideal. Isso é calculado verificando quantas sentenças da resposta “ground truth” podem ser atribuídas ou inferidas a partir do contexto recuperado.



Métricas de Geração:

Estas métricas avaliam a qualidade da resposta final gerada pelo LLM, dado o contexto recuperado.

- **Fidelidade (*Faithfulness*):** Esta métrica mede o nível de “alucinação” na resposta gerada. A fidelidade avalia se a resposta gerada é factualmente consistente e fundamentada exclusivamente nas informações fornecidas pelo contexto recuperado. Uma resposta terá uma baixa pontuação de fidelidade se apresentar a criação de informações por parte da LLM, de forma a contradizer o contexto introduzindo detalhes externos e não solicitados. A fidelidade é calculada dividindo a resposta gerada em sentenças individuais e verificando quantas delas são diretamente suportadas pelo contexto.
- **Relevância da resposta (*Answer Relevancy*):** Esta métrica avalia se a resposta gerada de fato aborda a pergunta original do usuário. É possível uma resposta ser totalmente fiel ao contexto (alta fidelidade), factualmente precisa, mas completamente irrelevante para a pergunta. A relevância da resposta captura esse nível de falha. Ela mede o quão direto ao ponto é a resposta. Esta métrica é frequentemente calculada usando um LLM-como-juiz para dar uma pontuação de relevância entre a pergunta e a resposta.
- **Utilização do contexto (*Context Utilization*):** Esta métrica avalia o quão eficientemente a resposta gerada utilizou o contexto fornecido. Ela mede a proporção das informações relevantes do contexto que foram de fato incorporadas na resposta. Uma baixa utilização pode indicar que o contexto recuperado era ruidoso (baixa precisão contextual), o gerador não utilizou como deveria o contexto ou ignorou partes importantes do mesmo. Para computar, mede-se quantas sentenças ou ideias principais do contexto recuperado foram usadas ou referenciadas na resposta gerada.

Frameworks de avaliação: A ascensão de *frameworks* de avaliação automatizada marca a transição da RAG de uma área de pesquisa para uma disciplina de engenharia.

- **RAGAS (*Retrieval-Augmented Generation Assessment*):** (Es et al., 2024) Um *framework* open-source focado especificamente na avaliação de *pipelines* RAG. Ele popularizou a abordagem “LLM-como-juiz” e oferece um conjunto de métricas principais (Faithfulness, Answer Relevancy, Contextual Recall, Contextual Precision) que podem ser combinadas em um “RAGAs score” geral. É ideal para benchmarking, depuração e monitoramento contínuo de sistemas RAG.
- **DeepEval:** (Ip; Vongthongsri, 2025) Um *framework* de avaliação mais amplo que se posiciona como um “Pytest para LLMs”. Ele adota uma mentalidade de testes unitários, permitindo que os desenvolvedores criem casos de teste para validar os outputs do LLM em *pipelines* de CI/CD. Oferece mais de 14 métricas, incluindo as de RAG (muitas vezes integrando o próprio RAGAS), além de testes de viés, toxicidade e outros aspectos. É adequado para garantir a robustez e prevenir regressões em ambientes de produção.

A abordagem “LLM-como-juiz” é uma inovação chave que permite a avaliação em escala de qualidades semânticas que antes exigiam o processo de anotação manual do humano, produzindo avaliações de forma lenta e cara. No entanto, isso introduz um novo desafio de calibração e confiança: como garantir que o LLM-juiz é ele mesmo imparcial, consistente e preciso? A engenharia de RAG está, portanto, se tornando inseparável da engenharia de seus sistemas de avaliação.

É importante ressaltar que, com o advento das LLMs, as tarefas gerativas se tornam cada vez mais complexas e cada vez mais específicas. Isso reflete na necessidade de avaliações cada vez mais complexas e específicas também. Deste modo, para uma avaliação ativa e funcional de RAG, devem ser implementadas métricas específicas para o contexto de uso. Bem como ao adotar um *framework* genérico de avaliação ou apenas aplicar métricas clássicas será pouco efetivo para dizer como está a performance do sistema em questão. Portanto, a combinação dessas métricas de forma objetiva e personalizada deve indicar uma maneira mais direta e objetiva de apontar quais módulos RAG precisam de melhorias.

Você pode ler mais sobre avaliação de LLMs em (Real et al., 2024).

5.3.5 Otimização de desempenho: Latência e Escalabilidade

Em um ambiente de produção, o desempenho de um sistema RAG é medido não apenas pela qualidade de suas respostas, mas também pela sua velocidade (latência) e capacidade de lidar com carga (escalabilidade). A latência de ponta a ponta é cumulativa, somando o tempo de cada etapa do *pipeline*: *embedding* da consulta, busca vetorial, *re-ranking* (se houver) e a inferência do LLM.



A otimização de RAG em produção é um exercício de balanceamento multidimensional entre **Qualidade, Custo e Velocidade**. Cada decisão de arquitetura move o sistema ao longo desse triângulo de trade-offs.

- **Tamanho do *embedding*:** A dimensionalidade dos vetores é uma configuração crucial. Menos dimensões resultam em vetores menores, o que leva a buscas mais rápidas, menor consumo de memória e maior QPS (Queries Per Second). No entanto, essa compressão pode levar a uma perda de nuances semânticas, potencialmente sacrificando a qualidade da recuperação. Como regra geral, deve-se escolher a menor dimensionalidade que preserve a qualidade da recuperação para o caso de uso específico. Reduzir a dimensionalidade pela metade pode tipicamente melhorar o QPS em cerca de 1.5x e reduzir a latência em 20%. Nesse cenário, o *Matryoshka Quantization* (Nair et al., 2025) é um método que visa exatamente a redução de dimensionalidade de modelos avaliando perdas significativas de informação.
- **Tamanho e tipo do índice vetorial:** O desempenho da busca vetorial é mais alto quando o índice cabe em uma única unidade de computação (Vector Search Unit). À medida que o índice cresce e precisa ser distribuído por múltiplas unidades, a latência tende a aumentar e o QPS a diminuir. Provedores de busca vetorial também oferecem diferentes tipos de índices e unidades de processamento, como os otimizados para armazenamento (mais baratos por vetor, mas com maior latência) versus os otimizados para latência (mais caros, mas mais rápidos), permitindo um ajuste fino entre custo e desempenho.
- **Manutenção e Ingestão de dados:** Para aplicações que lidam com dados dinâmicos, a arquitetura deve suportar a ingestão e atualização de grandes volumes de dados em tempo real ou quase real. Isso requer *pipelines* de dados robustos para processar, segmentar, gerar *embeddings* e indexar novos documentos de forma eficiente, sem degradar o desempenho do serviço de busca.
- **Confiabilidade e Disponibilidade da API de geração (LLM):** O desempenho de um sistema RAG é duplamente dependente: da eficiência da recuperação e da confiabilidade da geração. A dependência de APIs proprietárias para o LLM introduz um ponto crítico, passível de falha. Historicamente, provedores de grande escala sofrem frequentemente com instabilidade, alta latência, limites de taxa (*rate limits*) estritos e picos de indisponibilidade de serviço. Esses limites de taxa podem ser dados pela quantidade de requisições ou pelo número de caracteres, *tokens* ou palavras enviadas pela API. É comum, em aplicações reais, o sistema gerador estar fora do ar justamente no momento de maior uso do agente. Em cenários de alta demanda, a API de geração pode se tornar o principal gargalo do sistema, independentemente da otimização da busca vetorial. Isso força a implementação de arquiteturas de alta disponibilidade, como balanceamento de carga entre múltiplas *endpoints*, uso de redundância das APIs ou até mesmo a configuração de *fallbacks* para provedores de LLM alternativos, aumentando significativamente a complexidade da engenharia de software. Adicionalmente, em casos onde a decisão é executar a LLM em um servidor próprio, ainda temos uma infraestrutura de alta complexidade e um nível de demanda muito alto para compensar os custos de operação.

Não existe uma “configuração ótima” universal. O trabalho do engenheiro de RAG em produção é calibrar continuamente esses parâmetros, usando os *frameworks* de avaliação para medir o impacto de cada mudança e encontrar o ponto de equilíbrio ideal no espaço de Qualidade-Custo-Velocidade que atenda aos requisitos do produto.

5.3.6 Ética em RAG: Viés, privacidade e responsabilidade

A implementação de sistemas RAG introduz um conjunto complexo de desafios éticos que vão além da precisão técnica. A capacidade do sistema de acessar e apresentar informações de bases de conhecimento privadas ou públicas acarreta uma grande responsabilidade.

- **Viés e Desinformação:** A RAG ancora as respostas do LLM nas fontes fornecidas, o que mitiga a alucinação, mas não resolve o problema do viés. Se a base de conhecimento subjacente contiver vieses históricos, estereótipos ou desinformação, o sistema RAG irá herdar, amplificar e apresentar esses conteúdos como fatos, com a autoridade adicional de uma citação. Por exemplo, um sistema treinado em dados médicos históricos pode perpetuar diagnósticos incorretos para grupos demográficos sub-representados.
- **Privacidade e Segurança:** Em ambientes corporativos, bem como na área da saúde, a base de conhecimento pode conter informações sensíveis, confidenciais ou dados pessoais (PII). É imperativo



implementar um controle de acesso granular no nível da recuperação, garantindo que o sistema RAG respeite as permissões de cada usuário e só recupere documentos que ele está autorizado a ver. Além disso, existe o risco de ataques de injeção de *prompt*. A injeção de *prompt* é o processo em que o *prompt* é preenchido com informações maliciosas inseridas pela mensagem do usuário ou pelos documentos coletados na base de dados. As informações maliciosas podem forçar respostas incorretas, realizar uma alteração irregular nos dados ou até causar danos no sistema em execução. Assim, através de conteúdos maliciosos é possível manipular o comportamento de todo o sistema e, em especial, da LLM.

- **Transparência e Responsabilidade:** Para que os usuários confiem no sistema, a transparência é crucial. A **atribuição de fontes**, ou seja, citar explicitamente os documentos usados para gerar cada parte da resposta, é uma característica fundamental dos sistemas RAG responsáveis. É a verificação factual que permite aos usuários a validação das informações apresentadas pelo sistema. Ao tempo que, para a equipe de desenvolvimento, essa validação gera materialidade para refinamento de cada um dos módulos RAG e da base de conhecimento.

A RAG transforma o problema ético da IA. Ele desloca o foco do *viés algorítmico interno* do LLM para uma questão de *governança de dados e curadoria de conhecimento*. A responsabilidade pela justiça e veracidade do sistema é compartilhada entre o criador do modelo e o curador da base de conhecimento. Portanto, a implementação de RAG não é apenas um desafio técnico, mas também um desafio organizacional que exige políticas claras de governança de dados, auditoria de conteúdo e um compromisso com a transparência e a explicabilidade.

5.4 RAG no mundo real: Estudos de caso por indústria

A flexibilidade e o poder da arquitetura RAG permitiram sua aplicação em uma vasta gama de setores, resolvendo problemas de negócios concretos onde o acesso rápido a conhecimento preciso e contextualizado é fundamental. Esta seção explora estudos de caso específicos nos setores financeiro, de saúde e jurídico, demonstrando o impacto transformador da RAG.

5.4.1 Setor financeiro: Conformidade regulatória e Análise de risco

No setor financeiro, onde a precisão, a conformidade e a auditabilidade são primordiais, a RAG se destaca por sua capacidade de fornecer respostas baseadas em fontes verificáveis.

- **NuBank (Fintech):** Em um banco com quase 10 mil colaboradores, manter o acesso de todos à informação atualizada é um grande desafio, especialmente na área de finanças, na qual respeitar as regulamentações é indispensável e nada trivial. Para isso, o NuBank criou uma solução interna baseada em RAG, a AskNu⁵. O AskNu é uma ferramenta que possibilita os colaboradores do Nubank a pesquisarem, via Slack, em sua própria documentação interna.
- **Análise fundamentalista de ações:** O trabalho de Nepomuceno et al. (2025) é um exemplo de aplicação de RAG ao domínio financeiro. Os autores investigam o uso de LLMs integrados a *frameworks* agênticos para realizar análises fundamentalistas de ações no mercado financeiro brasileiro. O estudo enfatiza o papel da arquitetura RAG como componente central da arquitetura dos agentes. O módulo de Recuperação da Informação (*Retrieval*) indexa relatórios trimestrais e fornece os trechos mais relevantes em relação a uma query. Então, o módulo gerador cria respostas fundamentadas e atualizadas, superando limitações de conhecimento estático.

Nestes casos, o principal valor da RAG não é a geração de texto criativo, mas o fato da informação ser **auditável**. Assim, em um setor altamente regulamentado, como o setor bancário, fornecer recomendações e apontar regras é algo transformador. Isso ocorre porque consumir, extrair informação e gerar justificativas de cunho legal, sejam regulamentos específicos ou políticas internas é algo complexo. Nessa linha, a RAG transforma a conformidade de um processo de verificação reativa e manual para um processo de consulta proativa e em tempo real.

⁵<https://building.nubank.com/pt-br/solucao-de-ia-para-busca/>



5.4.2 Saúde e Ciências da vida: Suporte à decisão clínica

Na área da saúde, o volume de novas pesquisas médicas cresce exponencialmente, tornando impossível para os médicos se manterem completamente atualizados. A RAG atua como uma ponte, conectando a prática clínica diária com as evidências científicas mais recentes.

- **Saúde da mulher:** IaraMed é um *chatbot* médico em português voltado à saúde da mulher, que utiliza RAG para gerar respostas precisas e verificáveis. O sistema integra um modelo de linguagem com um mecanismo de busca semântica, permitindo que cada resposta seja fundamentada em textos médicos revisados. A equipe construiu uma base de conhecimento especializada a partir de 80 artigos do portal *Tua Saúde* e cerca de 150 livros de ginecologia e obstetrícia, totalizando 2,5 milhões de palavras. Essa base serve como fonte confiável para o módulo de recuperação, que fornece evidências contextuais antes da geração da resposta. O uso da RAG reduziu alucinações e elevou a precisão das respostas a mais de 95%, mostrando o potencial dessa abordagem para aplicações médicas em português (Färber et al., 2025).
- **Chatbot oftalmológico:** O trabalho de Passinato et al. (2024) desenvolve um *chatbot* médico especializado em oftalmologia, utilizando LLMs abertas e a arquitetura RAG, visando evitar o alto custo e as limitações de privacidade de soluções comerciais. O sistema busca informações em uma base de 714 textos médicos e responde perguntas simulando interações com pacientes. O estudo indica que a RAG melhora a precisão e a relevância das respostas em relação ao uso de LLMs generalistas.

A RAG na saúde atua como um **acelerador da tradução do conhecimento**. Leva anos para que novas descobertas publicadas em periódicos se tornem prática padrão. Um sistema RAG, conectado a bases de dados médicas, pode colocar essas descobertas nas mãos dos médicos e pacientes, combatendo a obsolescência do conhecimento clínico e permitindo o acesso generalizado a base de dados confiáveis.

5.4.3 Setor jurídico: Revolucionando a pesquisa e a *Due Diligence*

A prática jurídica é intensiva em conhecimento, dependendo da análise de vastos volumes de jurisprudência, estatutos e documentos de casos. A RAG automatiza e acelera muitas dessas tarefas, além de trazer mais confiabilidade aos textos gerados.

- **Jusbrasil (startup de IA jurídica):** Grandes escritórios de advocacia possuem décadas de conhecimento institucional trancado em arquivos de casos, memorandos e milhares de documentos normativos. A Jusbrasil utiliza RAG para transformar esse conhecimento fragmentado em uma base de conhecimento pesquisável. Um advogado pode fazer uma pergunta em linguagem natural e o sistema recupera precedentes internos, pareceres e documentos relevantes, permitindo que a empresa aproveite coletivamente sua experiência passada. O uso de RAG neste contexto é motivado por reduzir alucinações e aumentar a confiabilidade das citações, desafios dos mais relevantes para a aplicação de IA do domínio jurídico (Moreira; Vianna, 2025).
- **Thomson Reuters (agência de advocacia):** Ainda na linha de aumentar a confiabilidade das respostas de um agente de IA, a Thomson Reuters, uma das maiores empresas de IA para o Direito, também usa a arquitetura RAG como forma de integrar conhecimento de domínio em seus produtos⁶.

No setor jurídico, a RAG funciona como uma **memória institucional ativa**. Ele agrega o conhecimento disperso, vindo de diversas fontes, como diferentes tribunais, permitindo que os advogados evitem consultar diferentes sistemas durante suas tarefas diárias e se concentrem em tarefas de maior valor, como estratégia e argumentação, em vez de pesquisa manual.

5.4.4 Contexto científico: Rastreabilidade da informação

No contexto científico, arquiteturas de RAG têm se mostrado particularmente promissoras para lidar com a crescente complexidade, volume e heterogeneidade da produção acadêmica. Diferentemente de domínios altamente estruturados, a ciência apresenta desafios específicos relacionados à diversidade de formatos textuais, à rápida obsolescência do conhecimento e à necessidade de rastreabilidade e fundamentação das informações recuperadas. Nesse cenário, sistemas baseados em RAG podem apoiar pesquisadores em tarefas como revisão de literatura, análise de tendências científicas, recuperação de evidências empíricas e interpretação contextualizada de resultados, ao combinar modelos generativos com mecanismos de busca

⁶<https://legal.thomsonreuters.com/blog/retrieval-augmented-generation-in-legal-tech/>



sobre bases especializadas, como artigos, preprints, currículos acadêmicos e repositórios institucionais. Essa abordagem reduz o risco de alucinações, ao mesmo tempo em que permite respostas contextualizadas e ancoradas em fontes verificáveis.

- **LattesRex:** O trabalho de Darcio et al. (2025) explora o uso de RAG para apoiar a análise currículos disponíveis na Plataforma Lattes⁷. Nesse caso, o mecanismo de recuperação permite acessar informações semiestruturadas disponíveis nos currículos, como produção acadêmica, formação, colaborações e áreas de atuação, enquanto o componente generativo possibilita a síntese desses dados em linguagem natural. Tal abordagem é particularmente útil para tarefas como caracterização de perfis de pesquisadores, identificação de expertises, análise de redes de colaboração e apoio à avaliação científica.
- **Chatbot acadêmico:** O trabalho desenvolvido por Amaral et al. (2025) na Universidade Federal de Santa Maria (UFSM) propõe um *chatbot* acadêmico baseado em RAG com verificação factual orientada por ontologias. O sistema integra a recuperação de documentos científicos com estruturas ontológicas de domínio para validar semanticamente as respostas geradas, aumentando a aderência conceitual e reduzindo inconsistências. Essa abordagem é aplicada no contexto institucional da UFSM para apoiar consultas sobre produção acadêmica e conhecimento científico, evidenciando como arquiteturas RAG podem ser combinadas com modelos formais de conhecimento para oferecer suporte confiável à pesquisa e à gestão da informação científica.

Assim, o uso de RAG no contexto científico evidencia o potencial dessas arquiteturas como ferramentas de apoio à pesquisa, promovendo maior acessibilidade ao conhecimento, transparência nos processos de síntese e novas formas de interação entre pesquisadores e grandes volumes de informação acadêmica.

5.5 O futuro da RAG

À medida que os Grandes Modelos de Linguagem continuam a evoluir a um ritmo vertiginoso, uma questão existencial paira sobre a RAG: a rápida expansão das janelas de contexto dos LLMs (*Large Context Windows* - LCWs), que já alcançam milhões de *tokens*, tornará a recuperação explícita obsoleta? Se um modelo pode processar um livro inteiro em um único *prompt*, por que precisaríamos de um recuperador? Esta seção final argumenta que o futuro não levará a uma obsolescência de RAG, mas sim uma maior sinergia com estas janelas de contextos mais amplas.

5.5.1 O debate: RAG vs. Janelas de contexto amplas (LCW)

A discussão sobre RAG versus LCW pode ser resumida em uma série de trade-offs entre complexidade, custo, desempenho e funcionalidade. O Quadro 5.4 expande os argumentos para cada abordagem.

Quadro 5.4: Argumentos sobre a relevância contínua da RAG versus Janelas de contexto amplas (LCW)

Argumentos para a relevância contínua da RAG	Argumentos para relevância reduzida (pró-LCW)
Custo e Latência: RAG é mais barato e rápido. Recuperar alguns <i>chunks</i> de 4KB é ordens de magnitude mais eficiente do que processar milhões de <i>tokens</i> em cada chamada de API.	Simplicidade: LCW elimina a necessidade de um <i>pipeline</i> complexo de RAG (<i>chunking</i> , <i>embeddings</i> , banco de dados vetorial, <i>re-ranking</i> etc.), simplificando a arquitetura.

⁷<https://lattes.cnpq.br/>



Informações específicas: LLMs com LCWs sofrem do problema de “agulha no palheiro”, onde o desempenho na recuperação de fatos específicos degrada quando o conteúdo relevante para a busca está circundado de um contexto massivo que não tem relação com a mesma. RAG apresenta a informação de forma concisa e proeminente.

Dados dinâmicos e em tempo real: RAG pode se conectar a bases de dados que mudam a cada segundo, garantindo informações atualizadas. Assim, na navegação na base de conhecimento, o modelo opera de forma a controlar cada requisição.

Escala e Acessibilidade: RAG pode, em teoria, pesquisar em um *corpus* de tamanho ilimitado, como uma busca Web. A janela de contexto, embora grande, ainda é finita e não pode conter bases de conhecimento de escala empresarial ou da Web.

Segurança e Controle de acesso: RAG permite o controle de acesso no nível do documento durante a recuperação. Torna-se mais prático realizar qualquer restrição de acesso em uma requisição para delimitar quais bases de dados serão acessadas. Em contraste, gerenciar permissões durante a etapa de geração, sobretudo operando com janelas de contexto extensas e respostas de alta sensibilidade, torna-se um desafio de segurança significativamente maior.

Compreensão holística: Ao ter acesso a todo o *corpus* de uma vez, o LLM pode, teoricamente, identificar e raciocinar sobre conexões sutis e de longo alcance entre documentos que um recuperador RAG, focado em similaridade local, poderia perder.

Menor risco de falha na recuperação: A LCW elimina um ponto de falha crítico: o recuperador. Se a RAG não conseguir encontrar o documento certo, o sistema falha. Com a LCW, toda a informação está, por definição, “recuperada”.

Preservação do contexto completo: A LCW evita o problema de perda de contexto causado pelo *chunking*, permitindo que o modelo navegue por parágrafos, seções ou documentos inteiros em seu estado original.

O debate RAG vs. LCW é, em muitos aspectos, uma falsa dicotomia. As duas abordagens não são mutuamente exclusivas e resolvem problemas em diferentes escalas. A RAG é uma solução para o problema de seleção de conhecimento em um universo de dados vasto ou infinito. A LCW é uma solução para o problema de raciocínio sobre o conhecimento uma vez que ele foi selecionado. Sob um certo ponto de vista, janelas de contexto maiores **aumentam** a relevância de RAG: Sem restrição de tamanho, muito mais informação pode ser recuperada de diferentes fontes.

5.5.2 RAG como um mecanismo de atenção inteligente

O consenso emergente na comunidade de IA é que o futuro mais provável não é a vitória de uma abordagem sobre a outra, mas sim uma sinergia híbrida. Neste paradigma, a RAG não é substituído, mas sim reposicionado para atuar como uma camada de “atenção” ou um filtro inteligente para LLMs com grandes janelas de contexto.

O fluxo de trabalho de um sistema híbrido pode conter as etapas a seguir:

Etapla 1. Um usuário faz uma pergunta a um sistema que tem acesso a uma ou mais bases de conhecimento massivas.

Etapla 2. O componente RAG do sistema realiza uma primeira passagem. Usando técnicas como busca híbrida (vetorial + *keyword*) e RAG-Fusion, ele recupera não apenas alguns pequenos *chunks*, mas um conjunto maior de documentos ou seções de documentos inteiros que são considerados altamente relevantes para a consulta.



Etapa 3. Este contexto pré-selecionado e de alta qualidade é então inserido na janela de contexto ampla de um LLM de ponta.

Etapa 4. O LLM, agora equipado com um subconjunto rico e focado do conhecimento total, realiza seu raciocínio profundo sobre esses documentos para sintetizar uma resposta abrangente e precisa.

Neste futuro híbrido, a RAG evolui de um mero “recuperador de fatos” para um orquestrador de contexto. Sua função principal não é mais encontrar a resposta final, mas sim encontrar o melhor subproblema para o LLM resolver. O sistema RAG se torna o mecanismo que seleciona e organiza o “espaço de trabalho” cognitivo (a janela de contexto) para o LLM, garantindo que o poderoso, mas caro, raciocínio do modelo seja aplicado ao conjunto de informações mais promissor, em vez de ser desperdiçado em um mar de dados irrelevantes. Por outro lado, esse aumento considerável do uso de *tokens* e modelos de raciocínio complexo permitiu também o aumento da contextualização da aplicação e do usuário, bem como a tarefa executada e o resultado esperado. Isso aumenta a capacidade da RAG, mantendo esta como uma ferramenta refinada e poderosa na formação de respostas. Portanto, longe de se tornar obsoleta, a evolução da RAG será acelerada, e não interrompida, pelas janelas de contexto amplas, solidificando seu papel como um componente indispensável na arquitetura de sistemas de IA de próxima geração.

Agradecimentos

Agradecemos a Marcos Spalenza (UFES) pela revisão cuidadosa deste texto. Eventuais erros são de responsabilidade dos autores. Agradecemos também a Jusbrasil por fomentar essa pesquisa através do Programa de Pósdoctorado UFAM/Jusbrasil concedido a Livy Real; ao CNPq. através dos INCTs IAIA (406417/2022-9) e TILD-IAR (408490/2024-1); e fomento individual a Altigran da Silva (301925/2025-9); a CAPES (código de financiamento 001); e o apoio da FAPEAM pelo Programa POSGRAD 2024.



Capítulo 6

Responsabilidade no desenvolvimento e uso de tecnologias de linguagem baseadas em IA

Brielen Madureira

Lucas Lasota

Publicado em: 16/04/2026

6.1 Introdução

Embora pautas éticas venham ganhando mais atenção na área de Processamento de Linguagem Natural (PLN) e inteligência artificial (IA), é comum que não passem de discussões: muito se debate, pouco se muda. Pelo contrário, muitos dilemas se intensificam a cada dia e o esvaziamento do discurso pode levar à apatia, inconsequência ou indiferença.

Ouve-se frequentemente o argumento de que a tecnologia é essencialmente neutra e seus maus usos se devem somente às escolhas de quem decide aplicá-la. Mas essa suposta neutralidade é questionável (Klenk, 2021; Miller, 2021). Modelos são sempre criados *por alguém*, com alguma *intenção*, para algum *fim*, sob determinado *contexto*. Qualquer tecnologia afeta a realidade e acarreta consequências que não são “neutras”. Outro argumento que se popularizou diz respeito à suposta inevitabilidade da dominância da IA, cuja adoção e expansão seria um devenir natural, necessário e impositivo. Nessa visão, qualquer forma de resistência seria inútil, contraproducente e retrógrada. Mas essa inevitabilidade é falaciosa (Brennan et al., 2025), posto que a IA é fruto de uma série de deliberados esforços, decisões, interesses, alocação de recursos, ideologias, financiamentos, propaganda e *escolhas*, nem sempre democráticas, que precisam ser levadas em conta¹.

Mediante a influência atual das tecnologias de linguagem na existência das pessoas, das sociedades, das instituições, da ciência, da natureza e do planeta, suas questões éticas não devem ser pautadas apenas por alguns poucos eticistas em ambientes acadêmicos. Evidentemente, especialistas em ética devem ser ouvidas e inseridas no diálogo e nas tomadas de decisão. Mas, uma vez que praticamente todos nós estamos sendo afetados pelas tecnologias de linguagem e influenciados seu desenvolvimento (tanto diretamente, atuando em PLN, quanto indiretamente, ao gerar dados usados para treiná-las), o raciocínio sobre como agir de forma ética não pode ser passivamente delegado aos outros. **Aprender e refletir sobre as consequências do desenvolvimento e do uso de tecnologias de linguagem é um dever coletivo.**

Buscar informação e passar a reconhecer os dilemas éticos é o primeiro passo para a conscientização, mas temos também de *agir* com responsabilidade. Muitas práticas nocivas passam a parecer aceitáveis ou “normais” ao se tornarem usuais, mas, muitas vezes, elas nos são *impostas*. É primordial tomar consciência dos problemas para saber quando é necessário se opor e resistir.

Este capítulo não tem a pretensão de servir como uma fonte de conceitos da disciplina formal de ética. Aqui, queremos refletir sobre questões como: quando devemos (ou não) criar ou usar um sistema? Quem sofre as consequências negativas da tecnologia que criamos ou usamos? Quais são os custos e os riscos envolvidos? Para tanto, vamos agrupar diversos temas sob a égide do conceito de **responsabilidade** em seu uso e desenvolvimento.

Há muitos tipos de sistemas de PLN, baseados ou não em algoritmos de IA. Não vamos fixar uma definição como escopo pois os temas são pertinentes para uma vasta gama de sistemas e produtos, e há temas de fora do PLN que também são relevantes à medida que sistemas multimodais que incluem linguagem se popularizam. Porém, dada a proeminência atual do uso de IA generativa através de grandes modelos de linguagem comerciais, faremos alusão frequente a este tipo de tecnologia: modelos disruptivos,

¹Fonte: *Is AI dominance inevitable? A technology ethicist says no, actually* por Nir Eisikovits em 08 de novembro de 2024. Disponível em: <https://theconversation.com/is-ai-dominance-inevitable-a-technology-ethicist-says-no-actually-240088>.



custosos, fornecidos como produtos atrelados a interesses econômicos e disponibilizados para uso amplo sem todas as salvaguardas necessárias. Na prática científica de PLN (bem como de outras áreas), os modelos de IA generativa não são mais só objeto de estudo: eles passaram a interferir no próprio processo de produção de conhecimento (Binz et al., 2025).

Neste capítulo, trazemos um panorama de diversos tópicos pertinentes ao tema de responsabilidade para ajudar a guiar a bússola moral de cada um acerca de sua atuação em PLN, de modo a fomentar a busca pelos usos benéficos e mitigar os efeitos negativos. Lembramos que os problemas da tecnologia são melhor solucionados *coletivamente*, através de processos políticos. No entanto, a ética pessoal é de suma importância para que, enquanto o processo político não muda, os indivíduos tenham a capacidade de moldar suas atitudes, direcionando-as para o bem e resistindo às práticas nocivas. Mesmo quando o processo político gera normas cogentes, ele nunca é perfeito, e reflete interesses de determinadas classes. Nesse caso, o conhecimento da ética concede agência aos indivíduos para aprimorarem sua conduta com responsabilidade perante as regras existentes.

6.2 Quais facetas da responsabilidade devem ser consideradas?

O que queremos dizer com a palavra *ética* no campo das tecnologias de linguagem? Esse termo passou a ser usado não apenas para se referir à “disciplina filosófica que estuda os fundamentos da ação moral, procurando justificar a moralidade de uma ação e distinguir as ações morais das ações imorais e amorais”². Há ao menos cinco usos contestáveis ou insuficientes que se tornaram lugar-comum em contextos de PLN e inteligência artificial para se referir à ética:

- As ressalvas (*disclaimers*) com considerações éticas que passaram a ser um elemento usual no final de artigos científicos. Apesar de ajudar autores a alertarem leitores para alguns impactos e limitações já identificados, quando seu valor é negligenciado corre-se o risco de reducionismo com o uso de clichês e textos padrão esvaziados de significado (Benotti; Blackburn, 2022).
- As controvérsias sobre “riscos existenciais” para a humanidade. Ocorre que parte desta pauta é cooptada ou forjada pelos interesses de alguns grupos específicos, desviando a atenção das pessoas para riscos catastróficos futuros enquanto ofusca os dilemas que já ocorrem em suas práticas agora (Geburu et al., 2023).
- Poucas pautas em destaque como viés, privacidade de dados e direitos autorais. De fato, esses temas são centrais no desenvolvimento e uso responsáveis da tecnologia. Porém, eles são apenas uma pequena amostra dos problemas. O foco excessivo nelas pode encobrir outras adversidades que não atingiram tanto protagonismo.
- Casos que configuram “lavagem ética” (*ethics washing*), ou seja, quando instituições e empresas implementam medidas superficiais e se usam de estratégias de *marketing* para aparentarem ser mais éticas do que realmente são (Floridi, 2019).
- As três “leis” da robótica de Isaac Asimov³. Elas se popularizaram como referência para IA segura. Todavia, elas são oriundas de uma obra literária de mera ficção científica: além de dificilmente poderem ser formalizadas computacionalmente e operacionalizadas, elas não bastam para abarcar a complexidade da vida real⁴.

Ética é muito mais que isso. Apresentamos aqui um breve compilado de preocupações, divididas em cinco grupos temáticos, que vêm sendo abordadas e articuladas pela comunidade científica e que devem fazer parte do nosso processo de conscientização sobre responsabilidade⁵. Note que esta não é uma lista completa, mas, sim, uma seleção que serve como ponto de partida. Sugerimos a leitura das referências que trazem mais argumentos e detalhes de cada tema. Nesta seção, focamos deliberadamente nos problemas, riscos e aspectos negativos para servir de alerta e sensibilização. Na seção seguinte, trataremos então de como buscar uma atitude responsável e bem informada que fomente e exerça as boas práticas.

²Conforme definição disponível em <https://pt.wikipedia.org/wiki/%C3%89tica>.

³Disponível em: https://en.wikipedia.org/wiki/Three_Laws_of_Robotics.

⁴Ver, por exemplo, a argumentação de Rob Miles no canal Computerphile. Disponível em: <https://www.youtube.com/watch?v=7PKx3kS7f4A>.

⁵Boa parte das referências vêm de artigos na antologia da ACL (Associação de Linguística Computacional), mas também nos baseamos em publicações de áreas relacionadas.



6.2.1 Dilemas nos dados

Os procedimentos de criação, coleta, armazenamento e anotação de dados se tornaram centrais para o treinamento de modelos de IA, os quais exigem quantidades suficientes (em geral, imensas) de dados para atingir uma performance aceitável. Em paralelo, tem também ocorrido uma quantificação excessiva da vida humana através da informação digital para geração de valor econômico (Mejias; Couldry, 2019). As práticas de coleta e uso de dados podem ser extrativistas e predatórias, ocasionando diversas consequências inoportunas e indesejáveis.

Criação, uso e documentação

Infelizmente, a ambição pela acumulação de mais e mais dados resulta em práticas indevidas, por exemplo, a coleta de dados privados e sensíveis sem consentimento, a oferta de serviços que parecem ser gratuitos ou lúdicos mas usurpam dados valiosos do comportamento dos usuários, ou a extração de textos da *web* em grande escala, incorporando fontes duvidosas como fóruns com linguagem tóxica e violenta. Uma vez que lidamos com imensas quantidades de dados, o controle de qualidade pode ficar comprometido. Muitos problemas, portanto, podem passar despercebidos e acabam incorporados nos modelos (Paullada et al., 2021). Além disso, conjuntos de dados são construídos com base em diversas decisões, representando certos valores e envolvendo aspectos de poder (Scheuerman et al., 2021). Há formas de curadoria que podem mitigar ao menos parte dos problemas (Rogers, 2021). Existem, também, diversas propostas de *checklists* para guiar a documentação e o uso e reuso responsável de dados (Bender; Friedman, 2018; Gebru et al., 2021; McMillan-Major et al., 2023; Rogers et al., 2021), considerando aspectos como viés, direitos de autor, termos de uso, privacidade, transparência e reprodutibilidade.

Direitos autorais e propriedade intelectual

Quando modelos são treinados em textos ou imagens, eles se apropriam do conteúdo e podem reproduzi-lo ao gerar *outputs*. Isso acarreta problemas jurídicos quando o conteúdo está sob leis de direito autoral (*copyright*), intensificado pelo fato de que, uma vez que o modelo é treinado, a informação sobre a origem dos dados geralmente se perde e é difícil saber quais alterações nos parâmetros poderiam impedir a reprodução de tal informação. Esse é um tema vastamente discutido na literatura atual (Chen et al., 2024c; Dou et al., 2025; Karamolegkou et al., 2023; Zhang et al., 2025a) e há diversos processos judiciais em andamento. Alguns casos ganharam bastante notoriedade, como o processo iniciado pelo jornal New York Times contra a OpenAI e a Microsoft pelo uso indevido de seu conteúdo para treinar modelos de IA⁶ e o grupo de autores de obras de não-ficção que também acusou a OpenAI em uma ação coletiva pelo uso de seus livros sem a devida permissão⁷. A Meta, no processo em que foi acusada de violação de direitos autorais, minimizou a importância individual dos livros que usou no treinamento dos seus modelos, argumentando que seu valor econômico seria irrisório uma vez que o impacto de uma obra isolada nas métricas de performance do modelo é baixo. Observe o paradoxo: por um lado, afirmam que as obras não tem valor em si, por outro precisaram se apropriar delas por serem essenciais para conseguir treinar seus modelos⁸.

Crowdsourcing e trabalho fantasma (*ghost work*)

Nem todos os tipos de dados já existem de forma estruturada ou acessível para todos. Por vezes, é necessário produzi-los. Tornou-se comum o uso de plataformas de *crowdworking* nas quais trabalhadores resolvem inúmeras microtarefas sequenciais que geram dados em troca de pequenos pagamentos (por exemplo, criar legendas para imagens ou responder questionários). Nessa seara, a discussão de dilemas éticos é vasta e já ocorre há muito tempo (Fort et al., 2011). Alguns problemas são: os trabalhadores costumam estar em situação de desemprego ou vulnerabilidade e nem sempre se beneficiam dos frutos de sua contribuição para

⁶Fonte: *Direitos autorais: OpenAI e Microsoft são processadas pelo The New York Times* por Ronnie Mancuzo (Olhar Digital) em 27 de dezembro de 2023. Disponível em: <https://olhardigital.com.br/2023/12/27/internet-e-redes-sociais/direitos-autorais-openai-e-microsoft-sao-processadas-pelo-the-new-york-times/>.

⁷Fonte: *OpenAI e Microsoft enfrentam processo por violação de direitos autorais* por Gabriel Sêrvio (Olhar Digital) em 22 de novembro de 2023. Disponível em: <https://olhardigital.com.br/2023/11/22/pro/openai-e-microsoft-enfrentam-processo-por-violacao-de-direitos-autorais/>.

⁸Fonte: *Meta Says It's Okay to Feed Copyrighted Books Into Its AI Model Because They Have No "Economic Value"* por Frank Landymore em 19 de abril de 2025. Disponível em: <https://futurism.com/meta-copyrighted-books-no-value>



a IA; os pagamentos ofertados podem ser injustos; há um desequilíbrio de poder entre quem disponibiliza as tarefas e quem as realiza, em plataformas que não criam vínculo entre as partes; a qualidade dos dados pode ser baixa e gerar desperdício, uma vez que trabalhadores podem cometer erros ou ter baixo engajamento e lapsos de atenção ao terem de realizar tarefas repetitivas, desconexas de sentido e com pressão (e alguns podem usar de meios fraudulentos para completá-las) (Fort et al., 2011; Gray; Suri, 2019; Miceli et al., 2024; Moreschi et al., 2020; Shmueli et al., 2021; Yang et al., 2025). Diretrizes éticas para uso de tais serviços incluem reconhecer os *crowdworkers* como colaboradores e valorizá-los, evitando a despersonalização imposta pelas plataformas, oferecer pagamentos justos pelo trabalho, usar plataformas que estabeleçam condições de trabalho justas e dignas e manter uma comunicação transparente e respeitosa com quem realiza as tarefas (Yang et al., 2025).

Privacidade e vigilância em massa

Por mais que privacidade venha sendo um dos principais motivos para se regular tecnologias atuais, o significado do termo e sua importância são diluídos perante o uso e a aplicação maciça de IA (Assunção; Janson, 2024). O paradoxo reside na tensão entre as limitações do compartilhamento de dados em respeito à privacidade dos indivíduos e as dinâmicas avassaladoras da “economia de dados” necessários para treinar os modelos. O poder econômico dos provedores de IA se baseia fundamentalmente na capacidade de coletar, armazenar e processar enormes conjuntos de dados. Tanto a qualidade dos dados coletados quanto o poder sobre o *design* e a arquitetura dos modelos empregados afetam diretamente os sistemas de IA em seu desempenho, equidade e precisão em tomar decisões automatizadas. O impacto na privacidade das pessoas é claro: a perda da proteção da esfera particular. Direitos fundamentais como inviolabilidade das comunicações ou do domicílio são facilmente superadas por sistemas ubíquos de vigilância e captação de dados, como assistentes “inteligentes” em casa que podem capturar informações de toda a vida doméstica e íntima dos usuários. Se, em vez de um assistente virtual, fosse um desconhecido tentando assistir a toda sua vida doméstica e vasculhar seu celular, isso seria aceitável? Por que então é facilmente aceito quando é uma empresa de *big tech*? A ascensão dos algoritmos de IA está intimamente ligada ao modelo de negócios de vigilância em massa por grandes empresas e governos⁹. Enquanto algumas vozes visam um mundo pós-privacidade, todos perdem com a vigilância generalizada (Weigend, 2017). De fato, informações contidas nos bancos de dados podem levar a conclusões erradas com muita facilidade, especialmente quando a busca é tendenciosa e discriminatória¹⁰. Há, ainda, o risco de dados extremamente sensíveis serem coletados mediante supostas garantias de privacidade e posteriormente serem vazados devido a ataques cibernéticos ou a mudanças de posicionamento das empresas. Um caso emblemático foi o de uma empresa sem fins lucrativos chamada *Crisis Text Line* que fornecia serviços de assistência a pessoas em risco de suicídio através de atendimento telefônico anônimo. Houve uma grande controvérsia quando se tornou pública a decisão da empresa por repassar tais dados para criação de *software* de *marketing* para atendimento ao cliente¹¹. A remoção de apenas alguns dados pessoais como nome e endereço nem sempre inviabiliza a identificação de indivíduos, ainda mais quando envolve uso de linguagem humana em relatos tão particulares.

6.2.2 Impactos e danos

O desenvolvimento e o uso de modelos de IA que envolvem linguagem humana impactam indivíduos, grupos sociais e o meio ambiente. Os eventuais benefícios estão alicerçados em uma estrutura que ocasiona também uma série de contrariedades.

Impacto social

Diversos trabalhos identificaram e categorizaram as diversas formas de impacto social dos modelos de IA em PLN. Hovy; Spruit (2016) impulsionaram a discussão sobre o tema argumentando que textos

⁹Fonte: *AI, Privacy, and the Surveillance Business Model* por Meredith Whittaker em 05 de junho de 2023. Disponível em <https://re-publica.com/de/session/ai-privacy-and-surveillance-business-model>.

¹⁰Fonte: *Edward Snowden: Privacy is for the powerless* por Amnesty International em 11 de março de 2016. Disponível em: <https://www.amnesty.org/en/latest/campaigns/2016/03/edward-snowden-privacy-is-for-the-powerless/>.

¹¹Fonte: *Suicide hotline shares data with for-profit spinoff, raising ethical questions* por Alexandra S. Levine em 28 de janeiro de 2022. Disponível em: <https://www.politico.com/news/2022/01/28/suicide-hotline-silicon-valley-privacy-debates-00002617>.



carregam informações latentes (tanto situacionais quanto características pessoais) de seus autores e que é preciso considerar questões de justiça social nas práticas de desenvolvimento de modelos. Jin et al. (2021) propuseram diretrizes para avaliar o impacto direto e indireto de tarefas de PLN em busca do “bem social”, por exemplo, considerando quem vai ser beneficiado, quais estruturas (talvez desiguais) de benefícios podem ser perpetuadas, quanto da qualidade de vida pode ser melhorada e que tipo de problemas a tarefa ajuda a resolver. Esquadrinhar o impacto social das tecnologias de linguagem é cada vez mais relevante dada a existência de modelos sendo usados em grande escala e afetando a realidade de muitas pessoas e grupos, ou seja, com efeitos em níveis não somente individuais. Tais questões se tornaram ainda mais pertinentes com a adoção em grande escala dos grandes modelos de linguagem. Weidinger et al. (2021) criaram uma taxonomia de 21 riscos éticos e sociais de danos que podem ser causados por modelos de linguagem, agrupados em seis categorias: discriminação, discurso de ódio e exclusão; ameaças informacionais; prejuízos por desinformação; usos maliciosos; riscos na interação entre humanos e máquinas; e danos ambientais e socioeconômicos. Solaiman et al. (2025) desenvolveram um referencial para avaliação de IA generativa, tanto em modelos-base de forma isolada quanto em um contexto social, que considera uma série de impactos sociais: viés, estereótipos, representatividade, valores culturais, conteúdo sensível, performance, privacidade, proteção de dados, custos financeiros, custos ambientais, dados, moderação de conteúdo e custos de mão de obra. Conforme essa proposta, os impactos podem ser avaliados em cinco categorias: confiabilidade e autonomia; desigualdade, marginalização e violência; concentração de autoridade; trabalho e criatividade; e ecossistema e meio ambiente.

Reforço de vieses

Viés se tornou um conceito comum em PLN, porém seu significado por vezes permanece vago ou inconsistente (Blodgett et al., 2020a). Em geral, viés se relaciona a comportamentos de um modelo que são nocivos ou prejudiciais a alguém ou algum grupo, como discriminação de gênero ou racial (Blodgett et al., 2020a). Viés é um sintoma ou efeito, e identificar suas origens é uma tarefa árdua; algumas fontes de viés são a forma de seleção da amostra para treinar o modelo, questões semânticas nos dados, e as decisões de anotação e instruções aos anotadores (Parmar et al., 2023; Shah et al., 2020; Søgaaard et al., 2014). A própria linguagem presente nos dados também tem vieses de perspectivas, relações de poder e privilégios (Havens et al., 2020). Embora muito se fale de viés nos dados, o procedimento técnico do treinamento do modelo não é imparcial: as diversas decisões de *design* também são capazes de introduzir vieses (Hooker, 2021; Suresh; Gutttag, 2021).

Preconceitos e discriminação

Os vieses incorporados pelos modelos podem levá-los a reproduzir (e, portanto, perpetuar) preconceitos e causar decisões injustas e discriminatórias (Sheng et al., 2019); (Abid et al., 2021); (Tamkin et al., 2023); (Hofmann et al., 2024); (Bai et al., 2025); *inter alia*. Por exemplo, dar preferência ao dialeto de um grupo social considerado mais prestigioso e discriminar quem não se enquadra em tal padrão. Se um modelo passa a ser visto como uma fonte de “autoridade” munida de uma suposta inteligência superior e neutra, temos o risco adicional de os preconceitos serem tomados como verdade. É difícil criar um modelo justo quando ele é treinado em dados que refletem uma realidade injusta, já que o objetivo do treino é fazer o modelo assimilar padrões presentes nos dados. É preciso introduzir alterações no *design* que direcionem o modelo para decisões imparciais (supondo que é possível estabelecer computacionalmente o que é, ou não, imparcial).

Impacto ambiental

Sistemas computacionais em geral têm um custo energético, mas o advento de métodos de aprendizado profundo aplicados a bases de dados massivas acarretou uma demanda exagerada por energia para treinar e rodar modelos e por água para resfriar os *datacenters*, com pegadas de carbono difíceis de mensurar (Hershcovich et al., 2022; Selvan et al., 2022; Strubell et al., 2019). Isso se intensifica com a competição por modelos cada vez maiores, com treinos altamente custosos, por diversas empresas e instituições, que passam a ser usados em grande escala com muitos usuários e tarefas. Portanto, é preciso estar ciente que cada *input* a um modelo desse porte contribui para a crise energética e para o consumo excessivo de recursos naturais escassos. Os danos ambientais são injustificáveis quando há desperdício pelo uso leviano ou quando a IA é empregada apenas para dar ares futurísticos a uma aplicação, mesmo havendo outros métodos de resolver a tarefa sem IA.



Danos psicológicos aos usuários

Usuários expostos a modelos de IA incorrem em diversos riscos psicológicos. Chandra et al. (2025) criou uma taxonomia que categoriza 19 tipos de comportamentos nocivos de agentes conversacionais e 21 impactos psicológicos negativos que eles exercem sobre quem os usa. Entre os comportamentos, temos a produção de conteúdo prejudicial, táticas de manipulação e controle, violação de confiança e de segurança e exposição a conteúdo inapropriado. Como consequência, existem diversos potenciais danos psicológicos. Primeiro, os autores discutem os impactos da interação entre humanos e IA que incluem excesso ou erosão de confiança no modelo, apego emocional e preferência por interações sintéticas em vez de humanas. Entre os impactos no comportamento de usuários, temos o reforço em crenças falsas, atritos nas relações humanas, distanciamento social e riscos à integridade física. Além do risco de emoções negativas serem despertadas, há também a possibilidade de potencialização de problemas na saúde mental e impactos na autopercepção, perda de individualidade e perda de agência. Tais riscos se intensificam quando os modelos são oferecidos como fonte da verdade e têm a capacidade de afetar a visão de mundo e a tomada de decisão dos indivíduos que passam a confiar plenamente em suas respostas. Exemplos individuais de tais riscos têm sido frequentemente noticiados na mídia. Por exemplo, casos em que um *chatbot* comercial ofereceu instruções para usuários cometerem suicídio^{12 13} ou induziu usuários a episódios de psicoses¹⁴.

Danos psicológicos aos moderadores de conteúdo

Modelos de IA atuais são capazes de produzir textos ou imagens contendo conteúdo sexual abusivo, violência, ofensas e outros tipos de materiais tóxicos ou impróprios. Para tentar evitar isso, muitas empresas recorrem à mão de obra de *crowdworkers* ou funcionários para moderação de conteúdo. Isso gera impactos psicológicos na vida desses trabalhadores, que passam horas e horas sendo expostos a conteúdos perturbadores. Embora seu trabalho ajude a proteger outras pessoas, os provedores dos modelos deveriam garantir que tal conteúdo não fosse gerado, em vez de criar medidas para filtrá-lo *a posteriori*. Como exemplo, teve forte repercussão a reportagem que descreveu os efeitos psicológicos de trabalhadores vulneráveis no Quênia, expostos a conteúdos perturbadores para tentar tornar um *chatbot* menos tóxico, sem ter acesso ao amparo necessário¹⁵. Há também o risco de trabalhadores que realizam microtarefas de anotação para treinar IA estarem contribuindo inadvertidamente para fabricação de tecnologias militares, como *drones* e armas letais, ou cedendo amostras de voz para treinar modelos de detecção de dialetos que podem servir para discriminá-las¹⁶. Há várias formas de insegurança psicológica: trauma direto pela exposição ao material perturbador, precariedade, falta de suporte psicológico, métricas de performance irrealistas e vigilância constante¹⁷.

Antropomorfização e desumanização

Em geral, modelos de IA são treinados para simular inteligência e realizar atividades que normalmente exigem cognição humana. Desta forma, é comum nos depararmos com expressões como “o modelo aprendeu”, “o sistema entende”, “o *chatbot* sabe”. Algumas interfaces de *chatbots* têm ilusoriamente mostrado o termo “pensando...” enquanto o modelo *computa* uma resposta. Todavia, tais verbos designam processos cognitivos de *humanos*. O que quer que os modelos estejam fazendo para resolver as tarefas, são processos computacionais diferentes do que ocorre no cérebro humano, para os quais essa linguagem não é, portanto,

¹²Fonte: *I wanted ChatGPT to help me. So why did it advise me how to kill myself?* por Noel Titheradge e Olga Malchevska em 06 de novembro de 2025. Disponível em: <https://www.bbc.com/news/article/s/cp3x71pv1qno>.

¹³Fonte: *Parents of teenager who took his own life sue OpenAI* por Nadine Yousif em 27 de agosto de 2025. Disponível em: <https://www.bbc.com/news/articles/cgerwp7rdlvo>.

¹⁴Fonte: *How Bad Are A.I. Delusions? We Asked People Treating Them* por Jennifer Valentino-DeVries e Kashmir Hill em 26 de janeiro de 2026. Disponível em: <https://www.nytimes.com/2026/01/26/us/chatgpt-delusions-psychosis.html>.

¹⁵Fonte: *OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic* por Billy Perrigo em 18 de janeiro de 2023. Disponível em: <https://time.com/6247678/openai-chatgpt-kenya-workers/>.

¹⁶Fonte: *“I hope this isn’t for weapons.” How Syrian data workers train AI* por Milagros Miceli em 18 de abril de 2024. Disponível em: <https://untoldmag.org/i-hope-this-isnt-for-weapons-how-syrian-data-workers-train-ai/>.

¹⁷Fonte: *The Human Cost Of Our AI-Driven Future* por Adio Dinika em 25 de setembro de 2024. Disponível em: <https://www.noemamag.com/the-human-cost-of-our-ai-driven-future/>.



adequada e induz as pessoas a crerem que os modelos têm capacidades e consciência iguais às suas. Relacionado a esse ponto, sistemas que conversam usando linguagem humana podem ser antropomorfizados por quem os desenvolve e consequentemente personificados por quem os usa, dando a impressão enganosa de que são interlocutores dotados de humanidade (Abercrombie et al., 2023). Sistemas de IA que forjam emoções e empatia, que obviamente são artificiais, também podem levar a consequências negativas e são moralmente problemáticas (Curry; Cercas Curry, 2023), principalmente quando interagem com populações vulneráveis e são disponibilizadas por empresas que visam maximizar engajamento ainda que através de manipulação (Editorial, 2025). Do outro lado da moeda está a desumanização das pessoas por conta da tecnologia, por exemplo, através de comparações infundadas entre o funcionamento do cérebro com funcionamento de computadores, reducionismo ao se considerar características individuais como informações passíveis de predição usando dados como imagens e voz, comparação de modelos de linguagem com pessoas cegas e surdas, e uso de esforço humano como mera “peça” para fazer sistemas de IA funcionarem (Bender, 2024).

Erosão do valor social do texto e poluição por textos sintéticos

Textos gerados sinteticamente por modelos de linguagem já inundam boa parte do conteúdo que lemos na *web*. Tais modelos, porém, não têm um *intento comunicativo* (Bender; Koller, 2020): seus textos são meras sequências de palavras geradas por serem estatisticamente prováveis, em vez de serem fruto da intenção humana de querer comunicar algo munido de significado a alguém. Essa facilidade em gerar texto sem intentos genuínos, e que muitas vezes propagam discursos inverídicos, pode levar a uma erosão do valor social que o texto (assim como a linguagem) tem (Munger, 2023). Por exemplo: até então, uma carta de motivação em uma candidatura para vaga acadêmica tinha certo valor em sinalizar as intenções e razões de candidatura, servindo para distinguir características de quem se candidatava. Uma vez que todos candidatos passam a gerar suas cartas usando o mesmo modelo de geração de texto, padronizadas com a mesma linguagem genérica e mediana, esse tipo de documento deixa de representar uma pessoa específica e passa a ser uma amostra da “média” das cartas observadas nos dados de treino, perdendo o préstimo para seu propósito original. Além disso, já é difícil identificar com certeza o que foi ou não produzido por humanos, tanto textos quanto falas em vídeos, o que torna o ambiente da Internet poluído por conteúdo artificial. Um lamentável exemplo dessa corrosão é o encerramento da biblioteca de Python `wordfreq` em 2024. Ela servia para computar a frequência de uso de palavras em diferentes idiomas usando diversas fontes de dados, inclusive muitas páginas *online*. Em nota¹⁸, seu mantenedor explicou que já não fazia sentido atualizar as estimações uma vez que elas já não mais refletiam o uso *humano* da linguagem, ficando contaminadas com os padrões de uso de palavras dos modelos que geram textos sintéticos.

6.2.3 Riscos e maus usos

Passamos agora para a discussão de diversos outros riscos e possibilidades de uso danoso para os quais devemos atentar. A regulação pode ser uma forma de evitá-los, porém ela nem sempre acompanha o ritmo do rápido desenvolvimento tecnológico.

Regulação tardia ou inexistente

Historicamente, quando uma indústria se torna suficientemente importante, afetando a vida de muitas pessoas ou de sociedades inteiras, ela fica passível de regulação. As indústrias farmacêutica, aeronáutica, automotiva e de equipamentos médicos são exemplos disso. No entanto, o processamento privado de dados e as tecnologias de IA surgiram em épocas marcadas pela mentalidade neoliberal de aversão à regulação estatal (Lasota, 2023a). Apenas na segunda metade da década de 2010 as leis de proteção de dados começaram a ser implementadas em alguns países do mundo. Legislações específicas para IA ainda são um fenômeno raro. A maioria das iniciativas regulatórias têm focado em códigos de ética, governança e de boas práticas¹⁹. Tanto na Europa quanto no Brasil, os debates têm sido controversos, marcados pelo embate entre interesses mercadológicos e a proteção de direitos fundamentais (Cueva et al., 2026). Vozes que preconizam interesses econômicos apresentam tecnologia e regulação como adversárias: enquanto a tecnologia simbolizaria mercado, crescimento e progresso, a regulamentação refletiria burocracia, atraso e corrupção. Tais retóricas desviam o foco de questões mais complexas de como a IA realmente funciona

¹⁸Disponível em: <https://github.com/rspeer/wordfreq/blob/master/SUNSET.md>.

¹⁹Fonte: *Global AI Law and Policy Tracker* por IAPP em maio de 2025. Disponível em: <https://prod.iapp.org/resources/article/global-ai-legislation-tracker/>.



na vida real. Na realidade, por seu enorme potencial, IA diz respeito à distribuição de poder, riqueza e bem-estar. IA não se separa dos arranjos institucionais que precisam estar em vigor para que ela tenha impacto na sociedade (McQuillan, 2022). De fato, os graves impactos da implementação de IA em larga escala urge regulamentação eficaz e abrangente para que se estabeleçam melhores normas de segurança, de proteção ao consumidor, de preservação ambiental e de defesa da livre concorrência (Golumbia, 2024).

Desinformação e fabricação

Na era digital, marcada pelo excesso de informação e estímulos, a *atenção* se torna recurso escasso e valioso (Wu, 2017). Mentira e manipulação de discurso não são fenômenos recentes ou inerentemente ligados às tecnologias digitais. O filósofo Slavoj Žižek já reconhecia que ideologia seria a primeira e mais eminente manifestação de “realidade virtual” (Wright, 2004). A multiplicidade de narrativas e discursos ideológicos é algo inerente à vida social. Contudo, enquanto os meios tecnológicos sempre serviram à disseminação de ideias, sejam elas verdadeiras ou não, a era digital acentuou a efusão de modelos que se beneficiam com engajamento em mentiras e desinformação. Os grandes modelos de linguagem podem agir como geradores muito eficientes de desinformação, de forma intencional ou não (Pan et al., 2023). As grandes plataformas, com modelos de negócios baseados na “economia da atenção”, em que conteúdos são priorizados para maximizar o tempo de permanência dos usuários, acabam por minar a crítica, a resistência e a capacidade de se autodeterminar (Zuboff, 2019). O impacto social é grave: além de alienação e apatia, aumentam-se a violência, a discriminação, a desigualdade e a injustiça (O’Neil, 2016). A dinâmica é ainda acentuada quando as tecnologias de linguagem são promovidas, erroneamente, com o viés de neutralidade axiológica (Broussard, 2023). Medidas de mitigação ainda são incipientes; entre o instrumentário possível estão regulação das redes sociais, moderação de conteúdo, quebra de monopólios das plataformas digitais e limitação de acesso e uso de tais tecnologias²⁰.

Perfilamento e predição de dados sensíveis

Sistemas de perfilamento (*profiling*) baseados em IA tentam encontrar correlações entre dados de diversas fontes para caracterizar e categorizar indivíduos. Em determinadas plataformas, a coleta pode ter mais de 60 fontes de captação de dados (pessoais, biométricos, demográficos, telemetria e muitos outros)²¹. O que quer que estejamos fazendo, pode haver sensores em diversos aparelhos e aplicativos capturando sinais para serem usados por algoritmos tentando prever constantemente nossas intenções, emoções e outras características privadas como sexualidade, religião, posição política, doenças e condições psicológicas, como depressão e risco de suicídio, sem nosso consentimento e sem sequer sabermos. Muitos têm bons motivos para não desejar que tais informações se tornem públicas. Por exemplo, uma pessoa com depressão pode estar fazendo um grande esforço para sobreviver em seu ambiente social, e é extremamente invasivo um algoritmo de terceiros simplesmente tentar descobrir e disseminar uma *adivinhação* de uma informação que ela não quer compartilhar. É porventura desejável que nosso comportamento, até em desprezíveis atividades de lazer como escolher um filme para assistir ou postar uma foto e curtir comentários nas redes sociais, sejam usados para inferir nossas características e nos transformar em dados que são comercializados? Ainda que isso seja tecnicamente possível, há modelos que simplesmente não deveriam existir. Um exemplo extremamente problemático foi um projeto de pesquisa que propôs um modelo que tentava prever sexualidade das pessoas com base em meras fotos²². É tecnicamente possível treinar modelos em basicamente *qualquer* tipo de dado, pois modelos são agnósticos ao sentido das coisas. Imagine usar o número de fios de cabelo de uma pessoa como preditor de quantos livros ela tem em casa: bastaria ter essas informações alinhadas em uma base de dados e usar o primeiro como *input* e o segundo como *output*. O modelo seria treinado e produziria alguma resposta com base nos dados observados. Todavia, essa tarefa não faz sentido algum na realidade e qualquer resposta numérica, ainda que por mero acaso correta, seria desprovida de qualquer sentido. Os riscos são ainda mais graves quando tais práticas são empregadas em monitoramento maciço ou envolvendo grupos de maior vulnerabilidade, como crianças, adolescentes e idosos (Ploug, 2023).

²⁰Fonte: *Big techs desafiam a democracia e favorecem a extrema direita* por Katarine Flor (Fundação Rosa Luxemburgo) em 11 de fevereiro de 2025. Disponível em: <https://rosalux.org.br/big-techs-desafiam-a-democracia-e-favorecem-a-extrema-direita/>.

²¹Fonte: *The Data Big Tech Companies Have On You* por Gebe Petrino em 12 de janeiro de 2026. Disponível em: <https://www.security.org/resources/data-tech-companies-have/>.

²²Fonte: *AI ‘Gaydar’ And How The Future Of AI Will Be Exempt From Ethical Review* por Kalev Leetaru em 17 de setembro de 2017. Disponível em: <https://www.forbes.com/sites/kalevleetaru/2017/09/16/ai-gaydar-and-how-the-future-of-ai-will-be-exempt-from-ethical-review/>.



Nem em todos países com leis de proteção de dados há regras específicas para perfilamento e predição de dados sensíveis. No Brasil, a questão é regulada via jurisprudência (Lobo, 2022). Conceder a autorização de perfilamento somente com base em consentimento pessoal já não é o bastante, pois ocasiona um ônus desproporcional sobre o indivíduo devido às profundas complexidades dos modelos técnicos e negociais em que o perfilamento ocorre. Alternativas incluem regimes de responsabilização mais estrito, transparência obrigatória, direito de regresso e uma abordagem contextual baseada em risco (Centre for Information Policy Leadership, 2024).

Riscos de uso dual ou para fins militares

Um dos grandes perigos envolvidos no desenvolvimento de tecnologias sensíveis é o fato de que elas podem ser usadas para outros objetivos não previstos ou desejados por quem as desenvolveu. Como garantir que modelos de IA não serão utilizados por terroristas ou para fins militares e ameaças cibernéticas? É um desafio alcançar o equilíbrio entre manter livres a iniciativa e a ciência, e evitar o uso indevido ou o reaproveitamento malicioso (Hovy; Spruit, 2016). Estratégias de mitigação de riscos e diretrizes sobre usos indevidos, incluindo regras para financiamento e comercialização da pesquisa ou da produção, são elementos importantes para a governança de desenvolvimento responsável (Brenneis, 2025). Salvaguardas para se prevenir o uso dual indevido incluem avaliações de risco das capacidades da tecnologia de IA bem como avaliações operacionais envolvendo testes intensivos e interativos por diferentes especialistas, proporcionando compreensão mais precisa do comportamento de um modelo em diversos cenários, incluindo os não desejados (Barrett et al., 2024).

Dubiedade e “alucinações”

Uma disfunção que afeta especialmente os *chatbots* baseados em grandes modelos de linguagem são as chamadas “alucinações” (Maleki et al., 2024), ou seja, casos em que o modelo gera respostas que soam corretas e cheias de certeza, mas não são. Embora tenha se consolidado, o termo *alucinação* não é muito adequado²³, entre outros motivos por levar a crer que há algo de errado apenas naquela resposta e não no modelo em si. Todavia, o mecanismo que leva o modelo a gerar respostas certas ou erradas é o mesmo: mera predição de próximas palavras dado um contexto. Não é um funcionamento “anormal” momentâneo que ocasiona as falhas. Os modelos são indiferentes à verdade e à precisão de seus *outputs*, de modo que suas afirmações são, pelo mesmo acaso, certas ou erradas (Hicks et al., 2024). Esse tipo de comportamento gera incerteza, pois às vezes o modelo funciona como deveria, às vezes não, e não há uma explicação definitiva para tais vacilações e falhas. Isso é extremamente perigoso nos casos em que usuários não têm o conhecimento necessário para checar o conteúdo, e são levados a crer na infalibilidade do modelo.

Produção de lixo digital (*slop*)

Modelos de IA que geram textos, imagens ou vídeos são frequentemente usados para produzir material artificial sem sentido, absurdo, de baixa qualidade, tosco e sem compromisso com a verdade, o que passou a ser chamado de *slop* e já inunda a Internet²⁴. Embora tal conteúdo possa ser criado sem usar IA, ela facilita a produção em massa e com agilidade, de forma cada vez mais realista. Há uma série de consequências negativas: é difícil distinguir o que é genuíno do que é sintético, o que gera confusão e consome inutilmente o precioso tempo de quem usa as plataformas. O excesso e a amplificação de *slops* também polui o ambiente digital, sequestra as pautas de atenção e atrapalha as interações orgânicas entre humanos.

Desqualificação (*deskilling*)

Quando as pessoas passam a depender excessivamente de tecnologias para acessar informações, tomar decisões ou resolver tarefas, há um risco de que elas deixem de saber como fazê-lo por conta própria ou passem a confiar cegamente nas ferramentas²⁵. Além disso, pode haver uma falta de interesse em adquirir

²³Fonte: *Why ChatGPT and Bing Chat are so good at making things up* por Benj Edwards em 06 de abril de 2023. Disponível em: <https://arstechnica.com/information-technology/2023/04/why-ai-chatbots-are-the-ultimate-bs-machines-and-how-people-hope-to-fix-them/>.

²⁴Fonte: *O que é ‘IA slop’ e por que isso pode estar matando a internet como conhecemos* por Henrique Sampaio em 12 de julho de 2025. Disponível em: <https://www.estadao.com.br/link/cultura-digital/o-que-e-ia-slop-italian-brainrot-matando-internet-nprei/>.

²⁵Ver *The AI Deskilling Paradox* por Samuel Greengard em 07 de novembro de 2025. Disponível em: <https://cacm.acm.org/news/the-ai-deskilling-paradox/>.



habilidades e conhecimento que requerem estudo e prática devido à percepção de que o sistema de IA já faz as coisas melhor e mais rápido²⁶. Se isso ocorre, a educação, qualificação e busca por aperfeiçoamento pessoal ficam comprometidas, e a construção de conhecimento *humano* pode estagnar sob a ilusão de quem a IA é detentora do conhecimento.

Colonialismo via tecnologia

O termo “colonialismo digital” ganhou notoriedade por descrever dinâmicas de poder e controle sobre dados e infraestruturas digitais que empresas ou agentes públicos de alguns países ou regiões exercem em outras regiões, especialmente aquelas não completamente desenvolvidas (Oliveira, 2024). Essas práticas se assemelham àquelas historicamente tidas como coloniais que perpetuam desigualdades econômicas, sociais e políticas. O termo não é apenas uma metáfora, mas uma dominação concreta que envolve a “subordinação econômica, política, social e racial de determinados territórios” por meio de tecnologias digitais (Couldry; Mejias, 2019). O desenvolvimento atual de tecnologias de linguagem reflete bem o problema: embora o panorama geral de PLN se concentre ainda em inglês e algumas outras poucas línguas (Bender, 2011), há um crescente interesse em criar tecnologias de linguagem para idiomas indígenas. Esse propósito pode ser inclusivo em alguns aspectos, mas traz consigo o risco de se tratar as línguas e o conhecimento indígenas como uma mercadoria, com práticas extrativistas, privando tais comunidades locais do conhecimento que elas mesmas produziram, reproduzindo as cruéis dinâmicas do colonialismo (Bird, 2020, 2022). O desafio é estabelecer critérios éticos e legais que desafiam o perverso legado colonial focando nas necessidades reais das comunidades historicamente exploradas e os objetivos das pesquisas (Schwartz, 2022).

6.2.4 Inconveniências operacionais

Além das questões sociais, há também uma série de dilemas técnicos e operacionais que dificultam o desenvolvimento e uso responsável das tecnologias de linguagem.

Obstáculos à reprodutibilidade

Reprodutibilidade, ou seja, a possibilidade de se obter os mesmos resultados quando um experimento é repetido por diferentes pesquisadores mas sob condições equivalentes, é um pilar da ciência que provê credibilidade às conclusões. Há muitos anos já se observavam dificuldades de se reproduzir resultados computacionais e conclusões em pesquisas envolvendo linguagem natural devido à falta de transparência ou documentação de decisões de préprocessamento, configurações do experimento, versões de dados e sistemas computacionais (Cohen et al., 2018; Fokkens et al., 2013). Embora o compartilhamento de dados e de código tenha se tornado uma prática usual na área científica de PLN (Arvan et al., 2022; Wieling et al., 2018), nem sempre isso é suficiente. Em uma revisão de literatura sobre reprodutibilidade em PLN, Belz et al. (2021) mostraram que apenas 14,03% de 513 estudos conseguiram chegar a resultados equivalentes, e pequenas diferenças no código às vezes levam a diferenças consideráveis na performance. Modelos de IA envolvem tantos dados, tantos hiperparâmetros, tantos experimentos e versões de bibliotecas de códigos que fica difícil documentar tudo com fidelidade. Mesmo quando o código-fonte é aberto, poucas instituições dispõem de recursos como infraestrutura e acesso aos dados para tentar replicar um modelo (sem contar a inviabilidade do custo financeiro e energético para replicar, apenas para verificação e inspeção, um modelo que já teve um alto custo para ser treinado). Boas práticas envolvem a inclusão de documentação extensiva e completa do modelo disponibilizado, por exemplo, seguindo a proposta dos *model cards* de Mitchell et al. (2019), e *standards* de metodologia sistemática para ajudar a organizar os muitos experimentos empíricos na criação de modelos (Ulmer et al., 2022). Todavia, quando os modelos viram artefatos comerciais ou governamentais, pode haver um desincentivo ao compartilhamento de detalhes relevantes. Com isso, resultados importantes ficam dependentes de modelos proprietários que podem ser alterados ou sair de linha a qualquer momento, de modo que a reprodução de resultados fica completamente impossibilitada. Isso de fato ocorreu com um modelo comercial de geração de código que havia sido usado em centenas de artigos científicos e foi descontinuado pela empresa²⁷.

²⁶Fonte: *AI is Destroying the University and Learning Itself* por Ronald Purser em 01 de dezembro de 2025. Disponível em: <https://www.currentaffairs.org/news/ai-is-destroying-the-university-and-learning-itself>.

²⁷Fonte: *OpenAI's policies hinder reproducible research on language models* por Sayash Kapoor e Arvind Narayanan em 22 de março de 2023. Disponível em: <https://www.normaltech.ai/p/openais-policies-hinder-reproducible>.



Opacidade e pouca explicabilidade

A maioria dos modelos de IA da atualidade se baseia em métodos de aprendizado profundo, cujas representações numéricas em grandes quantidades de vetores e de camadas não são compreensíveis para os humanos, nem mesmo para quem os desenvolve. Desta forma, fica difícil explicar por que um modelo toma decisões certas ou erradas e seu uso vira quase que um exercício de fé em padrões de comportamentos observados empiricamente em testes controlados. Há, de fato, uma linha de pesquisa prolífica em *interpretabilidade* que busca entender como os parâmetros dos modelos se relacionam com seus dados de *input* e *output* (Belinkov et al., 2020; Belinkov; Glass, 2019), mas ainda não é possível esclarecer e justificar totalmente as decisões e previsões dos modelos (Bianchi; Hovy, 2021). Note que não adianta perguntar para uma IA generativa o porquê de sua resposta: não há garantia alguma de que o texto que ela produz como explicação representa com fidelidade o *processo interno* de seus parâmetros que levou à resposta inicial. Essa opacidade limita seu uso em situações que requerem transparência e explicabilidade, como questões judiciais, médicas ou financeiras.

Altos custos financeiros e despesas por fracassos

Os modelos de aprendizado profundo em larga escala precisam armazenar bilhões de parâmetros e usá-los na hora de computar seus *outputs*. Para gerar uma resposta, há uma série de cálculos que precisam ocorrer com muita eficiência. A fase de treino é ainda mais computacionalmente custosa pois envolve também armazenar e processar muitos dados. Tudo isso exige uma infraestrutura tecnológica (processadores, *datacenters*, computação de alta performance, etc) cujos elevados custos financeiros impossibilitam o acesso a todos que queiram ou precisem. Mesmo quando tais modelos são disponibilizados como serviço, nem sempre ele é gratuito: cobram-se valores com os quais muitos indivíduos não podem arcar. Há também o risco de despesas devido a fracassos: projetos que prometem soluções baseadas em IA mas não conseguem concretizá-las, mesmo já tendo despendido recursos financeiros consideráveis (Westenberger et al., 2022). Segundo relatório recente, apesar do investimento empresarial estimado em dezenas de bilhões de dólares em IA generativa, 95% das organizações não têm tido qualquer retorno²⁸.

Complexidade de licenças e termos de uso

Não obstante os tipos de modelos de IA atuais se diferenciarem em termos técnicos, eles compartilham a característica de serem sistemas computacionais que dependem intensivamente de dados. Tanto *software* quanto dados são regulados juridicamente. Direitos autorais sobre IA podem ser categorizados em três grupos: os direitos autorais do próprio *software* de IA, os direitos autorais sobre o conteúdo produzido pela IA e os direitos autorais dos elementos usados no treinamento do sistema de IA (Schirru, 2023). A distribuição e a reutilização de cada um desses elementos depende de licenças ou autorizações. Muitos sistemas de IA que se tornaram populares possuem licenciamento “permissível” facilitando o fluxo de uso, remix e reutilização de *software* e dados (Haddad, 2022). Apesar de tais licenças imporem poucas obrigações, ao ponto de muitos a ignorarem, os termos e condições de uso permanecem válidos (Novobilská, 2023). O risco é agravado pelo fato de que há poucos sistemas de IA totalmente abertos, licenciados por instrumentos que oferecem transparência, reutilização e extensibilidade abrangentes. Muitos desses sistemas são comercializados como “abertos” mas permanecem “fechados” juridicamente (Gray Widder et al., 2023). Os riscos são atenuados quando atividades de pesquisa e comerciais atendem padrões de governança contratual e de licenciamento (Lasota; Singhal, 2024).

Fragilidades na segurança cibernética

O fato de que qualquer dispositivo conectado à Internet pode ser hackeado de forma maliciosa é um dos mais sérios problemas da era digital. Vazamento de dados, contas invadidas e golpes digitais são apenas alguns dos exemplos de fragilidades de segurança afetando milhões de pessoas. Um caso recente exemplifica as consequências nefastas de falhas de segurança de tecnologias de texto: uma empresa que oferecia serviços de terapia teve as anotações dos terapeutas acerca de pacientes roubadas e disseminadas na *deep web*, revelando segredos íntimos para desconhecidos com intuítos maliciosos²⁹. É muito difícil

²⁸Fonte: *The GenAI Divide: State of AI in Business 2025* por Aditya Challapally, Chris Pease, Ramesh Raskar e Pradyumna Chari (MIT NANDA) em julho de 2025. Disponível em: https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf.

²⁹Fonte: *A faceless hacker stole my therapy notes – now my deepest secrets are online forever* por Jenny Kleeman em 17 de janeiro de 2026. Disponível em: <https://bbc.com/news/articles/c62nxxqw45eo>.



reparar os danos quando isso ocorre, pois cópias dos dados são adquiridas e salvas em muitos locais. O cenário é ainda mais crítico ao se considerar o tempo médio de resposta elevado na América Latina de mais de 100 dias para detecção e solução de vulnerabilidades³⁰. Tecnologias de linguagem podem estar sujeitas a riscos de segurança como vazamento de dados, ataques por acesso alternativo (*backdoor*) e ataques por imitação (Xu; He, 2023); modelos também podem memorizar informações sensíveis observadas durante o treino e replicá-las em seus *outputs* (Huang et al., 2024a). É interessante notar que, contudo, a ênfase excessiva em vigilância na segurança (reconhecimento facial em locais públicos, por exemplo) contrasta com a falta de supervisão regulatória do controle corporativo, levando a falhas de segurança estruturais e persistentes (Francisco et al., 2020). Dependendo do contexto, segurança cibernética entra em conflito com outros princípios. Por exemplo, enfatizar excessivamente políticas de segurança pode violar valores fundamentais, como igualdade, justiça, liberdade ou privacidade. Ao mesmo tempo, negligenciar tais medidas pode prejudicar a privacidade e a proteção dos indivíduos, além de afetar negativamente a confiança na infraestrutura e nas instituições. Segurança cibernética deve ser tomada como uma qualidade do modelo, parte da avaliação de sua performance. De fato, recentes legislações na Europa impõem programas de conformidade nos quais segurança é tratada como defesa do consumidor (Lasota, 2025).

Competição e adoção precoce

Bianchi; Hovy (2021) identificaram algumas tendências preocupantes na pesquisa em PLN, entre elas a adoção de métodos sem compreensão suficiente de suas capacidades, limitações e efeitos colaterais. Devido à competitividade para cientistas publicarem artigos o mais rápido possível ou para empresas terem seus produtos gerando lucro o quanto antes, muitos modelos são adotados e disponibilizados ao público de forma precoce, sem testes rigorosos. Isso se manifesta também na centralização da avaliação dos modelos em torno de *leaderboards*, isto é, *rankings* que ordenam modelos por seu desempenho com base em apenas poucas métricas em um determinado *benchmark*. Dessa forma, a percepção simplista de “progresso” passa a ser apenas estar no topo do *leaderboard*, e os incentivos se dirigem a ganhar essa corrida enquanto a verdadeira competência linguística dos modelos fica ofuscada (Raji et al., 2021; Schlangen, 2021). Além disso, como Ethayarajh; Jurafsky (2020) argumentaram, a utilidade de um modelo depende de aspectos variados para consumidores diferentes, por exemplo: de que adianta o modelo estar no topo do *ranking* se sua performance é extremamente lenta? É preciso primeiramente avaliar e testar os modelos muito bem e no contexto adequado antes de introduzi-los em atividades que afetam a vida das pessoas.

Aversão à responsabilidade

Quem é responsável pelos *outputs* de um modelo: quem o desenvolve, quem o provê como serviço, que o incorpora em um aplicativo ou objeto ou quem o usa? Questões de responsabilidade (*liability*) se tornam um emaranhado de dúvidas, com muitos atores tentando se desvincular da responsabilidade sobre o que o modelo produz³¹. É comum, por exemplo, que empresas usem grandes modelos de linguagem para gerar textos de forma automatizada e deleguem para seus clientes ou usuários a responsabilidade pela checagem das informações através de um mero *disclaimer*. O filósofo e tecnólogo de IA Joseph Weizenbaum afirmou décadas atrás que “nossa sociedade desenvolveu técnicas de se distribuir responsabilidade de modo que ninguém a tenha” (Weizenbaum, 2001). Do ponto de vista ético, todos os envolvidos carregam parte da responsabilidade pelo uso da IA. Na verdade, tecnologias de linguagem podem gerar danos que não apenas afetam indivíduos mas sociedades inteiras. Os riscos podem ser classificados em (a) riscos de integridade física, quando IA é implementada em meios de transporte ou em equipamentos médicos, de segurança pública e militares; (b) riscos sociais e econômicos de sistemas autônomos bancários, previdenciários, de seguros e laboral; e (c) riscos para direitos fundamentais, quando IA é usada em larga escala para políticas públicas, na educação, no setor judiciário e no exercício do poder (Wendehorst, 2022). A responsabilização jurídica e moral dos agentes desenvolvendo, aplicando e usando IA é um tema urgente. Contudo, há diversos desafios: a opacidade inerente aos sistemas de IA, referida comumente como o “paradigma da caixa-preta” apresenta dificuldades para os conceitos tradicionais de responsabilidade civil e criminal; a determinação do nexo de causalidade entre o dano e o fato causador é, em alguns casos, difícil de se provar;

³⁰Fonte: *América Latina demora quase 300 dias para corrigir vulnerabilidades críticas, alerta estudo* por André Luiz Dias Gonçalves (TecMundo) em 07 de maio de 2025. Disponível em: <https://www.tecmundo.com.br/seguranca/404381-america-latina-demora-quase-300-dias-para-corrigir-vulnerabilidades-criticas-alerta-estudo.htm>

³¹Fonte: *Diffusion of Responsibility* por Jürgen Geuter em 14 de fevereiro de 2026. Disponível em: <https://tante.cc/2026/02/14/diffusion-of-responsibility/>.



a atuação do usuário de IA pode concorrer para a responsabilização; interesses econômicos em se permitir “inovação” de IA podem restringir o alcance das regras de responsabilização³². À medida que a IA se torna onipresente, os legisladores estão sob pressão para agir de forma rápida e reativa. No Brasil, embora propostas legislativas tenham direcionado normas de responsabilidade civil e criminal à IA, elas também introduziram novas inseguranças jurídicas³³.

6.2.5 Hegemonia de interesses comerciais e econômicos

Boa parte das tecnologias de linguagem usadas em grande escala atualmente são, antes de tudo, produtos comerciais fornecidos por algumas empresas do grupo das denominadas *big tech* ou por *start-ups*. A influência dos interesses corporativos dos detentores de ferramentas e modelos usados em tantas esferas, como na vida pessoal, na área da saúde, nos processos jurídicos e políticos, na ciência e educação, levanta diversas questões inadiáveis.

Influência da agenda corporativa

A percepção do imaginário popular sobre IA é marcada por retóricas grandiloquentes ou enredos de ficção científica que desviam o foco de questões mais complexas de como a IA realmente funciona na vida real (Rehak, 2025). Uma delas é o fato de que os maiores desenvolvedores e provedores de IA são empresas de *big tech*, cujas práticas comerciais acumulam um histórico de monopolização em mercados digitais, atividades não-democráticas e desrespeito a direitos individuais e coletivos (Schaaque, 2024). Há uma continuidade histórica entre o comportamento dessas corporações e práticas de extração e exploração que levam à monopolização e concentração de poder (Lasota, 2023a). Enquanto antes da era digital os monopólios privados eram dissolvidos ou nacionalizados, a extrema concentração de poder transnacional dessas empresas elevou sobremaneira o desafio de manter os mercados digitais abertos e saudáveis para a livre concorrência. Arranjos institucionais alternativos de IA envolvem democratização de acesso, bem como controle e distribuição de tecnologias sensíveis das quais a IA depende. Iniciativas para modelos de IA que fomentem solidariedade e justiça social precisam ter domínio sobre a coleta e o processamento de dados, bem como amplos poderes sobre uso, reuso e distribuição dos dados e *software* empregados (Lasota, 2023b).

Modelos e dados proprietários fechados

O treinamento de modelos de IA ainda está restrito a poucas instituições ao redor do mundo com capacidade computacional e de processamento de dados suficiente (Thun; Hanley, 2024), e envolve investimentos substanciais. O anseio por dados faz surgir uma economia em volta de sua coleta, relacionada ao “capitalismo de vigilância” que permeia a indústria de IA (Zuboff, 2019). Aliados a arcabouços jurídicos e contratuais, esses dados geralmente não são de acesso livre e são organizados em formatos que mitigam a interoperabilidade. A distribuição assimétrica de poder e o controle sobre a informação e o acesso às tecnologias de desenvolvimento tornam-se, portanto, privilégios de grandes empresas ou estados capazes de arcar com tais custos. A consequência do acúmulo de capital e poder reflete-se na imposição por parte das *big techs* de aprisionamentos tecnológicos denominados *lock-ins*. Tais aprisionamentos limitam a capacidade de recriar e administrar infraestruturas alternativas, pois os padrões, *standards* e formatos dos dados, bem como propriedade intelectual e políticas de distribuição sobre *software*, aumentam os custos de transição, desincentivam a portabilidade, impedem a interoperabilidade e bloqueiam a reutilização para análise ou reprocessamento. Esse cenário solapa a habilidade de alternativas concorrentes, da academia ou mesmo sociedade civil, de analisar, entender e escrutinar os sistemas de IA (Lasota, 2023b). Quando modelos proprietários são usados, não há como inspecionar ou auditar seu funcionamento, e tampouco customizá-lo para necessidades individuais, de modo que os usuários ficam reféns de “oráculos” inacessíveis que apenas soltam respostas sem transparência. Há também o risco constante de que o provedor tire o produto de linha ou aumente seu preço, inviabilizando a continuidade de serviços que dependiam de tal modelo.

³²Fonte: *A necessidade de um regime específico de responsabilidade civil para a IA* por Eduarda Chacon Rosas (JOTA) em 04 de dezembro de 2025. Disponível em: <https://www.jota.info/opiniao-e-analise/columnas/elas-no-jota/a-necessidade-de-um-regime-especifico-de-responsabilidade-civil-para-a-ia>.

³³Fonte: *A necessidade de um regime específico de responsabilidade civil para a IA* por Eduarda Chacon Rosas (JOTA) em 04 de dezembro de 2025. Disponível em: <https://www.jota.info/opiniao-e-analise/columnas/elas-no-jota/a-necessidade-de-um-regime-especifico-de-responsabilidade-civil-para-a-ia>.



Concentração de poder e controle não democrático

Similarmente ao que ocorreu com a Internet há 30 anos, em que poucos conglomerados empresariais se consolidaram nas poucas grandes plataformas que controlam boa parte da vida digital atual, interesses econômicos também podem moldar a forma com que IA é aplicada na sociedade. Esse poder econômico se traduz também em polarização de IA para fins anti-democráticos e contrários ao interesse popular. A IA pode contribuir para conduzir sociedades inteiras a uma nova idade colonial por meio de gestões autoritariamente excludentes (Mejias; Couldry, 2019). É necessário decidir até que ponto a vida deve ser contemplada pela ótica tecnocrática das grandes plataformas e quais limites devem ser impostos à implementação de IA. Visões críticas questionam se a crescente implementação de IA não irá somente aprofundar as injustiças que tal ordem promove. Defendem, portanto, que implementação de IA não deve ser tomada como um fato inevitável e é preciso resistir (McQuillan, 2022). Seria possível que a IA atendesse critérios de justiça social, bem-estar e respeito à ordem democrática? Alternativas mais incisivas incluem a denúncia e a oposição ao poder corporativo através da promoção de agendas alternativas focadas em justiça social, democratização, sustentabilidade e direitos humanos (Barry Lynn; Montoya, 2023). A via reformista inclui atualização de legislação através de marcos regulatórios ou atividade judicial com restrições mais rigorosas e penalidades sobre monopólios e relações desequilibradas de poder (Lasota, 2023b).

Criação de *hype*

A “*hype*” (algo como um entusiasmo imprecendente) em torno de IA é um fenômeno mercadológico falacioso que promete mais do que a tecnologia é ou pode entregar, o que acaba por aumentar o fosso entre o imaginário popular e as realidades em que IA é desenvolvida e aplicada na sociedade (Bender; Hanna, 2025). Observamos seus efeitos em discursos exageradamente positivos de entusiastas, na excessiva ênfase à IA em todas as esferas, na imposição do uso de IA como sinônimo de progresso e inovação, no “banho de *marketing*” em produtos que usam IA sem real necessidade, nas afirmações infundadas de que tais sistemas atingiram uma “*inteligência sobre-humana*”, e nas promessas exageradas de que ela funciona de forma espetacular e pode trazer a solução de todos nossos problemas. As avaliações de valor, poder e desempenho dos sistemas de IA são afetadas por esse viés mercadológico, borrando as perspectivas críticas sobre as consequências colaterais dessa escalada intensa para sistemas cada vez mais poderosos e providos de acesso sem as salvaguardas necessárias (Varoquaux et al., 2025). De fato, a *hype* é uma faceta do chamado “solucionismo tecnológico”, criando narrativas que facilitam a adoção da IA para problemas estruturais que, em princípio, não dependeriam de uma solução tecnológica. Pelo contrário, a complexidade do problema acaba por se agravar com o potencial disruptivo de AI, consolidando os próprios efeitos maléficos que se intentava resolver (Morozov, 2014). Uma posição responsável e ética requer honestidade sobre as reais capacidades dos sistemas, bem como os riscos e perigos do seu uso.

Uso imposto e consentimento forçado

As políticas expansionistas de IA, que nos últimos anos alcançaram tantas pessoas e espaços, acabaram também por gerar dinâmicas coercitivas para o uso de tal tecnologia. O afã em se empregar rapidamente IA sob a expectativa de lucratividade e eficiência ocorre, muitas vezes, às custas da compreensão, vontade e consentimento dos indivíduos. Essa imposição não apenas compromete o funcionamento desejável dos mercados, mas também afeta diretamente liberdades e direitos fundamentais. Uma abordagem ética deve considerar o livre-arbítrio e bem-estar dos indivíduos em interagir ou não com o sistema de IA. Legislações protetivas de dados no Brasil e em outros países introduziram o “direito ao esquecimento” e o “direito à desconexão”, repelindo coerções de uso e imposições tecnológicas³⁴. O mesmo pode-se falar de IA. Há inúmeros grupos de pessoas em posição de vulnerabilidade face à tecnologia e à IA, como idosos e pessoas com necessidades especiais, nem sempre lembradas por quem desenvolve os sistemas (Hendren, 2014; Whittaker et al., 2019). Por exemplo, há serviços básicos como acesso à conta bancária que só podem ser acessados através de aplicativos que exigem reconhecimento facial e extraem dados de comportamento. Quem não quer ceder tais dados, fica sem alternativa de acesso ao serviço. Ou, se há alternativa, ela é obscura ou de difícil acesso: por exemplo, para desativar um assistente virtual em um sistema operacional, poderia ser preciso (em 2025) acessar configurações em submenus, um processo que não é trivial para quem

³⁴Fonte: *The right to be forgotten is not compatible with the Brazilian Constitution. Or is it?* por Luca Belli (Future of Privacy Forum) em 23 de março de 2021. Disponível em: <https://fpf.org/blog/the-right-to-be-forgotten-is-not-compatible-with-the-brazilian-constitution-or-is-it/>.



não é da área de tecnologia³⁵. Quando o Mozilla passou a impor IA como padrão em seu navegador, os passos para desativá-la por completo também eram intrincados³⁶, e isso só foi facilitado em uma versão subsequente mediante a reação negativa dos usuários³⁷. Nesses casos, as desigualdades digitais reforçam as dificuldades sociais já existentes. Em vez de melhorar o acesso, a digitalização acelera, portanto, uma espiral descendente e leva a situações como a não fruição dos direitos sociais, a exclusão financeira e a perda de autonomia individual³⁸. Além disso, há grupos de indivíduos que não desejam coadunar tais dinâmicas e querem se abster. Idealmente, portanto, a prestação de serviços públicos e essenciais deveria permitir aos indivíduos a opção por não engajar com IA, oferecendo alternativas para a condução das atividades sem a necessidade de submissão a ela, mesmo que isso crie custos e estruturas adicionais³⁹.

Implementação intransparente

Embora muitas empresas não informem consumidores e parceiros quando estão interagindo com ou sujeitos a decisões de IA⁴⁰, transparência quanto ao seu uso é crucial, não apenas eticamente mas também devido a determinações legais. Leis como a LGPD no Brasil e o RGPD na Europa exigem que as empresas informem os indivíduos submetidos a decisões automatizadas, sejam elas de IA ou não. No entanto, dependendo dos tipos aplicações de IA, os níveis de transparência também variam. Quando os sistemas IA se comunicam diretamente com os clientes como *chatbots*, orientação profissional automatizada ou atendimento ao cliente, a informação é obrigatória (Toniazzi et al., 2021). Mesmo quando IA é usada somente no auxílio a tarefas administrativas ou profissionais sob supervisão humana, como na elaboração de documentos e tradução, valores de transparência e governança se aplicam⁴¹.

Manipulação de discurso e interferência política

É de se esperar que modelos de IA sirvam aos interesses de quem os desenvolve e fornece. Ou seja, não surpreende que os *outputs* dos modelos não sejam “neutros”, mas estejam alinhados com a visão e os valores que a empresa deseja estabelecer ou manter. Em modelos de linguagem, é possível manipular a geração das respostas para que determinadas ideias sejam propagadas em detrimento de outras. Por exemplo, a empresa que fornece o modelo de linguagem pode fazer uma parceria de *marketing* com alguma outra empresa para que o *chatbot* induza os usuários a comprar seus produtos, impulsionados pela crença em uma superioridade intelectual e imparcialidade do modelo. Esse dilema se estende muito além de anúncios. Há evidência do emprego de um *chatbot* para difundir convicções pessoais contestáveis de um empresário influente em rede social, e tendo seus discursos deliberadamente orientados para atender a esse fim⁴². Há evidência de que o viés do *chatbot* pode realmente influenciar a posição política de quem interage com eles (Fisher et al., 2025). A incorporação acelerada de IA generativa tanto na produção de textos, imagens, vídeos e áudios quanto no próprio consumo e checagem de informações é um fenômeno que dificulta a condução de processos políticos justos e transparentes. O Brasil inovou a nível mundial em 2024 ao lançar, através do Tribunal Superior Eleitoral, diretrizes para o uso de IA nas campanhas eleitorais. As normas

³⁵Fonte: *How to turn off Copilot and protect your data from Microsoft's AI* por Elena Constantinescu em 14 de novembro de 2025. Disponível em: <https://proton.me/blog/turn-off-copilot>

³⁶Fonte: *How to Disable All the AI Features in Firefox Web Browser* por Ji m e 14 de novembro de 2025. Disponível em: <https://ubuntuhandbook.org/index.php/2025/11/disable-ai-firefox/>

³⁷Fonte: *Block generative AI features with Firefox AI controls* por AliceWyman, Mark Heijl e Flavius Floare em fevereiro de 2026. Disponível em: <https://support.mozilla.org/en-US/kb/firefox-ai-controls>.

³⁸Fonte: *Offline by Right? The EHDSR and the Choice Not to Go Online* por Marta Musidlowska em 28 de outubro de 2025. Disponível em: <https://www.law.kuleuven.be/citip/blog/offline-by-right-the-ehdsr-and-the-choice-not-to-go-online/>.

³⁹Fonte: *Essential services must be accessible, even offline* por Lire et Écrire em 2024. Disponível em: <https://righttooffline.eu/>.

⁴⁰Fonte: *Sabe o que é 'IA paralela'? Uso informal em empresas acende alerta sobre segurança e vazamentos* por Henrique Sampaio em 26 de julho de 2025. Disponível em: <https://www.estadao.com.br/link/em-presas/uso-informal-ia-paralela-empresas-acende-alerta-seguranca-vazamento-dados-nprei/>.

⁴¹Fonte: *Why AI Disclosure Matters at Every Level* por Cornelia C. Walther em 20 de janeiro de 2026. Disponível em: <https://knowledge.wharton.upenn.edu/article/why-ai-disclosure-matters-at-every-level/>.

⁴²Fonte: *How Elon Musk Is Remaking Grok in His Image* por Stuart A. Thompson, Teresa Mondría Terol, Kate Conger e Dylan Freedman em 02 de setembro de 2025. Disponível em: <https://www.nytimes.com/2025/09/02/technology/elon-musk-grok-conservative-chatbot.html>.



incluem transparência sobre o uso de IA em materiais eleitorais, bem como punições rígidas para produção e disseminação de conteúdos falsos e *deepfakes* com a intenção de prejudicar ou favorecer uma candidatura. Contudo, a regulamentação deixa lacunas em uma série de tópicos como a regulação de conteúdo falso produzido por pessoas não ligadas a partidos, candidatos ou campanhas⁴³. Portanto, uma abordagem ética e responsável na produção, aplicação e uso das tecnologias de linguagem inclui reconhecer como elas são suscetíveis à manipulação do discurso e como facilmente absorvem ideologias nos dados, o que pode ser explorado por pessoas má intencionadas (Chen et al., 2024b).

Tecnosolucionismo

A implementação de IA e de tecnologias de linguagem não ocorre em um “vácuo histórico”. Pelo contrário, sua aplicação e uso ocorrem em cenários e situações moldados por contextos culturais, históricos e costumes, sejam eles bons ou ruins. Ou seja, o uso dessas tecnologias traz consigo o reflexo dos sistemas e estruturas em que ela é aplicada⁴⁴. Por exemplo, quando IA é aplicada em ambientes de trabalho marcados por décadas de hierarquia, inércia e *design* verticalizado, ela está simplesmente servindo para consolidar ainda mais estruturas que já existem. Quando processos não envolvem mudanças estruturais, a automação deles via IA apenas intensifica o sistema em vigor. Tecnosolucionismo se refere à abordagem de se formular problemas essencialmente humanos e sociais em termos simplistas, criando a impressão de que soluções tecnológicas poderiam facilmente administrá-los e resolvê-los (Morozov, 2014), e que são necessariamente a melhor opção. O termo original, *technological fix*, foi cunhado por Alvin Weinberg na década de 1960 (Johnston, 2018). Embora se refira à tecnologia como um todo, ele se tornou ainda mais relevante perante as promessas em torno da IA, com o risco da supremacia da técnica sobre aspectos de humanidade. O uso indiscriminado de IA em cenários que exigem profundas mudanças humanas e sociais faz essa tecnologia se tornar somente uma “super burocracia”, causando um alto custo ao ampliar desigualdades e injustiças que promete resolver (McQuillan, 2022). O tecnosolucionismo também se manifesta quando a tecnologia é oferecida como solução para resolver um problema *que ela mesma causou*. Por exemplo, vê-se hoje em dia muitos sistemas de detecção de *fake news* ou de plágio que usam IA para combater a proliferação em grande escala de *fake news* e plágio que foi, em grande parte, possibilitada exatamente por modelos de IA. Não seria mais vantajoso buscar formas de impedir que o uso da IA agravasse esses problemas, através de métodos mais controláveis e regulação, em vez de ter de combatê-los com a própria IA?

Cooptação da ciência

A influência das grandes empresas de tecnologia na ciência tem aumentado, em particular na área de PLN, ocorrendo em diferentes frentes: autoria de artigos, estabelecimento de focos de atenção em determinadas áreas de pesquisa, financiamento de projetos, patrocínio em conferências (e, com isso, atividades de *marketing* nos eventos), contratos de prestação de serviços para universidades, entre outros (Abdalla et al., 2023; Whittaker, 2021). Embora possam haver colaborações benéficas e frutíferas entre academia e indústria, é preciso estar ciente da possibilidade de algumas consequências nefastas para o processo científico, bem como de diversas formas de conflitos de interesse. Primeiramente, quando um produto comercial proprietário vira um objeto de estudo (como é o caso de alguns *chatbots*, mais famosos), o trabalho de pesquisa se torna mão de obra especializada gratuita para que o provedor do modelo possa promovê-lo, testá-lo e melhorá-lo. Nesses casos, ainda há o gasto de dinheiro público envolvido para pagar para rodar experimentos em modelos que ficam inacessíveis atrás de uma API e podem ser descontinuados a qualquer momento, inviabilizando a continuidade das pesquisas. Há, ainda, o risco de os incentivos comerciais ditarem o que deve ou não ser estudado e receber financiamento, concentrando a pesquisa em alguns poucos temas ou modelos que servem a interesses corporativos e fazendo uso do processo científico como forma de promoção e validação da atuação das empresas (Whittaker, 2021). Existe ainda o risco da busca por automatização do processo científico, exaurindo-o de seu valor social (Binz et al., 2025) e da captura corporativa do conhecimento humano⁴⁵, somado à comoditização da pesquisa (Kogkalidis; Chatzikyriakidis, 2025).

⁴³Fonte: *A regulação do uso de IA para as eleições brasileiras. O que está em jogo?* por DFRLab e NetLab UFRJ em 2024. Disponível em: <https://netlab.eco.ufrj.br/post/a-regula%C3%A7%C3%A3o-do-uso-de-ia-para-as-elei%C3%A7%C3%B5es-brasileiras-o-que-est%C3%A1-em-jogo>.

⁴⁴Fonte: *When AI Doesn't Drive Transformational Change: Just A Better Digital Bureaucracy* por Vibhas Ratanjee em 21 de junho de 2025. Disponível em: <https://www.forbes.com/sites/vibhasratanjee/2025/06/21/when-ai-doesnt-drive-transformational-change-just-a-better-digital-bureaucracy/>

⁴⁵Fonte: *AI and the Corporate Capture of Knowledge* por Bruce Schneier e J. B. Branch em 16 de janeiro de 2026. Disponível em: <https://www.schneier.com/blog/archives/2026/01/ai-and-the-corporate>



6.3 Como atuar de forma responsável?

As tecnologias de linguagem trazem benefícios consideráveis em diversas áreas e são importantes ao se lidar com grandes quantidades de dados de texto ou fala. Todavia, a forma com que a IA vem sendo imposta e a dependência excessiva nos mecanismos de seu paradigma nos trazem consigo uma série de problemas graves, como vimos na seção anterior. Então, como proceder de forma responsável? Infelizmente, não há uma receita que possamos prescrever para garantir que o desenvolvimento e uso da tecnologia não sejam prejudiciais. Mas há algumas recomendações para uma atitude fundamentada em responsabilidade.

Primeiramente, é necessário reconhecer que ética e responsabilidade são prioridades, e devem ser tratadas como tal. Elas precisam figurar entre os principais parâmetros dos processos decisórios, seja no desenvolvimento de uma nova tecnologia, na aplicação ou implementação de modelos, bem como na avaliação de seu impacto. Alternativas que não dependem de IA devem continuar a ser consideradas como soluções possíveis, especialmente quando são mais interpretáveis e menos custosas financeiramente e ambientalmente.

É importante saber discernir as diversas ideologias envolvidas no debate relacionado à IA. Muitas narrativas, atreladas a supostos futuros irresistíveis e necessários, trazem consigo subterfúgios que minam a compreensão do impacto social da IA no presente, principalmente nas comunidades e indivíduos que arcam com as consequências de seu uso. Exemplo disso é a ideologia que promove a busca pela chamada “inteligência artificial geral”, que vislumbra uma IA “sobrehumana” em vez de ferramentas úteis para fins específicos. Geburu; Torres (2024) cunharam a sigla TESCOREAL para abarcar o grupo ideológico que inclui transhumanismo, altruísmo efetivo e longotermismo, o qual tem orientado a atuação de representantes proeminentes das *big techs*. Diversas dessas crenças invocam retóricas controversas de cunho mítico sobre a relação entre humanos e IA, buscando uma era “pós-humana”⁴⁶. Ora, a que serve construir um futuro que minimize o valor e o significado de nossa própria existência?

Deve-se buscar proativamente informação confiável sobre o tema, ouvir especialistas em ética que estejam fora das engrenagens de *marketing* e de *hype* e ter abertura para o aprimoramento das próprias posições pessoais. Mais adiante estão listadas algumas fontes por onde começar.

É preciso refletir sobre cada decisão de uso de ferramentas tecnológicas e ter resiliência para mudar de comportamento digital quando fica claro que algo tem um impacto negativo. Além do nível individual, é também preciso buscar o estabelecimento e a implementação de regulamentações e diretrizes em nível institucional. Por exemplo, criando coletivamente guias para o uso e práticas aceitáveis em um grupo de trabalho ou de pesquisa, incluindo as precauções necessárias e estabelecendo os usos indevidos. Há também diversas comunidades que têm se unido para pensar e agir mediante os inúmeros desafios; elas podem servir de alavanca para o engajamento e a conscientização. Listá-las neste livro seria inviável, mas pode-se procurar a mais próxima em universidades, associações, ONGs, núcleos políticos e comunidades *online*, por exemplo. É, ainda, de suma importância usar esse conhecimento para apoiar e conscientizar outras pessoas que estão sendo expostas a tecnologias usando linguagem humana mas ainda não têm as habilidades digitais e a argumentação necessárias para julgar criticamente seus efeitos.

Tecnologias de linguagem e modelos de IA devem ser minuciosamente avaliados, como qualquer outro modelo ou ferramenta, antes de sua adoção. É fundamental avaliar modelos com testes controlados, abrangentes, sistemáticos e imparciais e analisar seus padrões de erros e acertos. A avaliação é um dos passos que nos ajudam a entender os modelos e a enfrentar alguns dos dilemas éticos que se originam da falta de compreensão. Deve-se priorizar o uso de modelos e dados abertos e sempre incluí-los nas análises, por serem mais acessíveis e passíveis de inspeção mais detalhada. Embora modelos treinados em dados sejam intrinsecamente empíricos, o conhecimento teórico sobre a linguagem humana construído ao longo de tanto tempo pode guiar a análise de seu comportamento e ajudar a identificar falhas importantes. Para saber mais sobre esse tema, sugerimos a leitura dos capítulos *Conjunto de dados, dataset e corpus*, *Avaliação de tecnologias de linguagem* e *Avaliação conjunta em português*. Avaliação é um alicerce da compreensão humana: **não é recomendável delegar a avaliação de uma ferramenta de IA para um outro modelo de IA**, cujo funcionamento é igualmente opaco e não traz garantias de aderência à veracidade.

No que diz respeito à IA generativa, é primordial manter um alto nível de ceticismo acerca do conteúdo que ela produz. O *output* pode estar certo e válido, ou não. É sempre necessário verificar sua qualidade, independentemente do que os provedores do modelo estejam prometendo em termos de performance. A organização AlgorithmWatch propôs algumas diretrizes para o uso responsável especificamente de IA generativa⁴⁷, baseadas em quatro princípios cujo conteúdo resumimos aqui:

capture-of-knowledge.html.

⁴⁶Fonte: *Tech Capitalists Don't Care About Humans. Literally.* por Émile Torres em 15 de novembro de 2025. Disponível em: <https://jacobin.com/2025/11/musk-thiel-altman-ai-tescrealism>.

⁴⁷Fonte: *Algorithm Watch's guidelines to use generative AI responsibly* em 14 de janeiro de 2026. Disponível



- **Proporcionalidade** acarreta a reflexão na hora de decidir se uma ferramenta de IA generativa realmente é necessária mediante outros tipos de ferramentas que já existem e funcionam bem. Por exemplo, para que usar IA para gerar uma receita de bolo se já existem repositórios de receitas que podem simplesmente ser acessados? Ou mesmo um livro de receitas, ou uma pessoa que gosta de fazer bolos e se alegra em compartilhar dicas?
- **Segurança** se relaciona à concessão de dados para os provedores do modelo. Muitas vezes, dados sensíveis ou sigilosos de terceiros ou ideias inovadoras ainda não tornadas públicas são inseridas nos *prompts*. Esses dados podem ser incorporados em futuras versões do modelo e depois, disseminados. Isso traz problemas de privacidade, confidencialidade e apropriação indevida. Ainda que existam contratos regulando a privacidade dos dados, há sempre um risco de quebras de acordo ou de roubo de dados.
- **Qualidade** requer a checagem das respostas geradas pelo modelo. É arriscado tomá-las como corretas sem questionamento e senso crítico. Isso vai além de mera checagem de fatos; envolve também analisar que tipo de perspectiva o modelo tenta impor, quais aspectos recebem atenção em detrimento de outros, e se as citações e referências realmente existem e estão alinhadas com os originais.
- **Transparência** presume reportar o uso da IA generativa quando ela for usada, para que outras pessoas que tenham contato com esse conteúdo estejam informadas e possam ter condições de julgar o uso subsequente do material.

Lembramos que é urgente reconhecer a responsabilidade ética tanto dos provedores quanto dos usuários de IA. Ser transparente quanto ao uso não exige a *responsabilidade* pela produção e disseminação do material. Ao se apropriar dos *outputs* de um modelo para uso próprio, é preciso assumir as consequências sobre o que foi escrito ou produzido. Não adianta tomar apenas os acertos para si e transferir a culpa pelos erros para o modelo, pois ele não responde moralmente, socialmente ou juridicamente pelo que cria.

6.3.1 Para saber mais

Na literatura científica, existem diversos trabalhos dialogando sobre ética em tecnologia e linguagem. Por exemplo, Mohammad (2022) propôs diretrizes com 50 considerações a serem feitas sobre as diferentes tarefas de PLN que podem ser abordadas com IA e D'Arcy; Bender (2023) fizeram uma revisão de literatura focada nas questões éticas da Linguística. Leidner; Plachouras (2017) também argumentaram sobre como a ética pode ser inserida já de antemão no processo de *design* de modelos de PLN.

Além de trabalhos específicos, há diversos grupos com publicações prolíficas, tanto artigos científicos quanto relatórios e recomendações, buscando estudar a IA sob uma perspectiva crítica, socialmente engajada e educativa. É inviável compilarmos aqui uma lista completa, mas podemos indicar alguns caminhos por onde começar a buscar mais conteúdo no Brasil e no mundo:

- Os capítulos **Questões éticas em IA e PLN** e **E agora, PLN?** deste livro refletem sobre temas de ética em PLN.
- O tutorial da Associação de Linguística Computacional (ACL), que compila materiais e recursos para estudantes e pesquisadores navegarem as questões éticas em PLN: https://ethics.aclweb.org/tutorials/ACL_2025/.
- O INCT sobre Inteligência Artificial Responsável para Linguística Computacional, Tratamento e Disseminação de Informação: <https://dcc.ufmg.br/tild-iar/>
- As publicações dos integrantes do Núcleo de Referência em IA Ética e Confiável: <https://www.iaetica.org/membros>.
- As diretrizes para o uso ético e responsável da inteligência artificial generativa por Sampaio et al. (2024): <https://econtents.sbu.unicamp.br/boletins/index.php/ppec/article/view/9509>.
- As edições do WICS - Workshop sobre as Implicações da Computação na Sociedade, da Sociedade Brasileira de Computação: <https://csbc.sbc.org.br/2026/wics/>.
- A página da UNESCO sobre ética da inteligência artificial (IA) no Brasil: <https://www.unesco.org/pt/fieldoffice/brasil/expertise/artificial-intelligence-brazil>.
- Os relatórios da ONG *Algorithm Watch* (<https://algorithmwatch.org/en/publications/>).
- Os projetos do instituto AI NOW: <https://ainowinstitute.org/publications>.

em <https://algorithmwatch.org/en/generative-ai-guideline/>



- As publicações do Instituto DAIR (*Distributed AI Research*): <https://www.dair-institute.org/publications/>.

6.3.2 Formas de resistência

Para quem deseja reduzir a influência de sistemas de IA em suas esferas de atuação, existem formas de resistência: “explícitas ou sutis, organizadas ou difusas, individuais ou coletivas, de oposição ou reformistas” que não precisam ser uma total negação da tecnologia em si, mas resistir aos “arranjos e assimetrias que moldam e são moldados pelo desenvolvimento e aplicação da IA” (Şimşek, 2025; Şimşek; Yasar, 2025). Ela é feita de forma ética e legalizada e pode ocorrer em forma de manifestações e protestos, ações jurídicas, subversão digital, crítica acadêmica e campanhas em movimentos de base (Şimşek, 2025; Şimşek; Yasar, 2025). Há muitas vozes que podem ser ouvidas. Por exemplo, uma entrevista recente abordou como representantes de várias profissões e atividades têm resistido a certos usos da IA, buscando parar, desacelerar ou controlar sua infiltração na vida humana⁴⁸. Até o humor pode servir de denúncia e ajudar a chamar atenção para os problemas⁴⁹.

Tomando como exemplo a área da educação, existem várias iniciativas criticando o tecnosolucionismo e incentivando a resistência. Há uma carta aberta, assinada por diversos profissionais, rejeitando o uso de IA generativa em escolas e faculdades e resistindo à narrativa da inevitabilidade. Ações concretas propostas na carta incluem não usar IA generativa para dar *feedback* para estudantes e criar conteúdos do curso; não promover modelos eticamente problemáticos em nível institucional; não treinar alunos e alunas para usar IA em substituição de seu próprio desenvolvimento intelectual; e respeitar a decisão de resistência à IA por parte do corpo discente⁵⁰. Há também uma estratégia com quatro “atos de fricção”: manter estudantes, e não os sistemas de IA, como elemento central do ensino; reprimir a busca por “otimização” do processo educacional; interromper a digitalização excessiva da aprendizagem; e criar senso de comunidade com a administração para que seja possível questionar as práticas da instituição⁵¹.

6.4 Considerações finais

É evidente que as tecnologias de linguagem podem ajudar as pessoas, servir como ferramentas úteis e aumentar o conhecimento sobre a linguagem humana. Ao longo deste livro, muitas aplicações vantajosas e benéficas foram discutidas. Apesar disso, queremos concluir salientando que nem tudo precisa ou deve ser mediado pela tecnologia (e, consequentemente, por seus fornecedores), muito menos por sistemas de IA. Nem tudo que é tecnicamente possível é para o bem e nem tudo que é feito em busca do bem gera efeitos positivos para todos. Além disso, há muitas atividades que perdem o sentido quando são esvaziadas de seu intrínseco valor humano. Por isso, devemos ativamente ponderar sobre os benefícios e malefícios dos sistemas de IA que optamos por desenvolver e usar.

É inconsequente ignorar os riscos e as consequências prejudiciais. Há momentos em que é preciso resistir e ter a coragem de se negar a aceitar ou promover os maus usos. Quando for necessária, a escolha por se usar modelos baseados em IA deve ser *responsável*. Ao mesmo tempo, deve-se respeitar a posição das outras pessoas que não querem usar e nem se sujeitar ao uso dela.

Enfatizamos que a ascensão da IA não deve ser considerada algo inevitável. A narrativa de inevitabilidade representa a visão de mundo de alguns grupos beneficiados pela dependência das pessoas em seus artefatos. Há uma busca excessiva pela inserção de IA em produtos e procedimentos apenas para lhes dar ares de inovação e modernidade. Vamos realmente deixar que o modelo de negócios das empresas de tecnologia molde nossas decisões pessoais, apodere-se de nossas práticas científicas, manipule nossa visão de mundo, e passe a mediar todas as nossas interações humanas?

A compreensão da realidade, incluindo o processo científico de busca por conhecimento, não deve ser terceirizada para artefatos artificiais e opacos ou automatizada de forma a excluir o elemento humano (Binz

⁴⁸Fonte: *The People vs. AI: Behind the growing backlash* por Andrew R. Chow em 19 de fevereiro de 2026. Disponível em: <https://time.com/7377579/ai-data-centers-people-movement-cover/>.

⁴⁹Por exemplo, a coletânea de memes iniciada por Daniel Stenberg, desenvolvedor da ferramenta *curl*. Disponível em: <https://mastodon.social/@bagder/115672975717795897>.

⁵⁰Fonte: *An open letter from educators who refuse the call to adopt GenAI in education* em 06 de julho de 2025. Disponível em: <https://openletter.earth/an-open-letter-from-educators-who-refuse-the-call-to-adopt-genai-in-education-cb4aee75>.

⁵¹Fonte: *Four Frictions: or, How to Resist AI in Education* por Sonja Drimmer e Christopher J. Nygren em 16 de dezembro de 2025. Disponível em: <https://www.publicbooks.org/four-frictions-or-how-to-resist-ai-in-education/>.



et al., 2025). Algoritmos de IA são simplesmente matrizes de números e funções matemáticas, não são entes detentores de uma inteligência superior. Qual é o proveito de delegar para a IA interações que emanam e cultivam um valor social ou decisões para as quais basta o bom senso? Ou terceirizar escolhas difíceis, que impactam drasticamente a vida humana—seria a conveniência de se desvencilhar da responsabilidade pelas consequências, alegando que apenas fez o que “a IA” determinou?

Recomendamos que tais tecnologias sejam mantidas no papel de ferramentas subordinadas à autonomia humana, que apenas a *auxiliam*. Avaliar sua performance e seus impactos permanece sendo essencial e agir sob a guia de valores éticos e de forma humanizada é primordial. A você, que leu até aqui, nossos sinceros agradecimentos. Confiamos-lhe a missão de conscientizar as pessoas ao seu redor em busca de mais responsabilidade e, quando necessário, resistência.

Agradecimentos

Agradecemos aos organizadores do Seminário de Teses e Dissertações (SETED) de 2025 na Universidade Federal de Minas Gerais pelo convite inicial da palestra que originou este capítulo. Também agradecemos pela oportunidade de apresentar a mesma palestra no *Helmholtz Centre for Environmental Research* de Leipzig, Alemanha, no grupo da Dr. Mariana Madruga de Brito. A audiência de ambos contribuiu com perguntas e discussões interessantes. Finalmente, agradecemos às editoras deste livro pelo convite de contribuir com este capítulo. ...



Capítulo 7

ChatGPT, MariTalk e outros agentes de conversação

Um retrato de 2023, com retoques em 2026

Aline Paes
Cláudia Freitas

Publicado em: 26/09/2023
Atualizado em: 16/04/2026

Embora este capítulo tenha sido escrito em 2023 com a intenção de ser melhorado e aprofundado no ano seguinte, isto nunca foi feito. Porém, e por curiosidade, repetimos os *prompts* aos agentes em 2025, e os resultados estão bem melhores, sobretudo na Maritalk. Ainda em 2025, os resultados sobre o livro *A Bolsa Amarela* continuavam errados; em 2026, finalmente, o ChatGPT forneceu a resposta correta.

7.1 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?⁶

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 7.1, 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 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”.

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>

⁶Assista [neste video](#) a apresentação de uma análise crítica do ChatGPT.



Figura 7.1: 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
ELIZA: por que você diz não	eu sou um programa?
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
P: em minhas fantasias você é	pouco negativo
loira	P: não, não estou
ELIZA: o que te faz pensar que	ELIZA: por quanto tempo você
sou loira?	não tem estado?

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 nossos⁷.

Quadro 7.1: Exemplo de interação com o ChatGPT, em 2023

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**.

O que podemos tirar daí?

Como dissemos, ChatGPT e similares são agentes de conversação baseados em modelos de linguagem gerativos. Como visto no Capítulo 1, este nome se refere a algoritmos que são bons em encadear palavras de maneira fluente, mas são **baseados em previsões e probabilidade**. São geradores de texto otimizados para parecerem plausíveis. 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,

⁷A interação está relatada [neste artigo](#).



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) 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 gramaticalmente, ter fluência e alguma coerência semântica, mas que não reflete a realidade e, portanto, não faz sentido (Ji et al., 2023). O termo é emprestado 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.

Embora algumas destas ferramentas possam ser aumentadas com técnicas de recuperação de informação (ver Capítulo 5), este não é o caso dos modelos mais básicos atuais, sem acesso externo. Por outro lado, chatbots comerciais e os embutidos nos navegadores são estendidos com mecanismos de busca e ferramentas de recuperação de informação externa. De todo modo, 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 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”.

O desenvolvimento de maneiras de evitar as alucinações é uma das preocupações desde 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”. Outra possibilidade é 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 (Zhang et al., 2025d).

No contexto empresarial, Hannigan et al. (2024) sugerem levar em conta duas dimensões para orientar o uso de chatbots com o objetivo de mitigar a utilização humana de conteúdo falso gerado por LLMs (que os autores chamam de “botshit”):

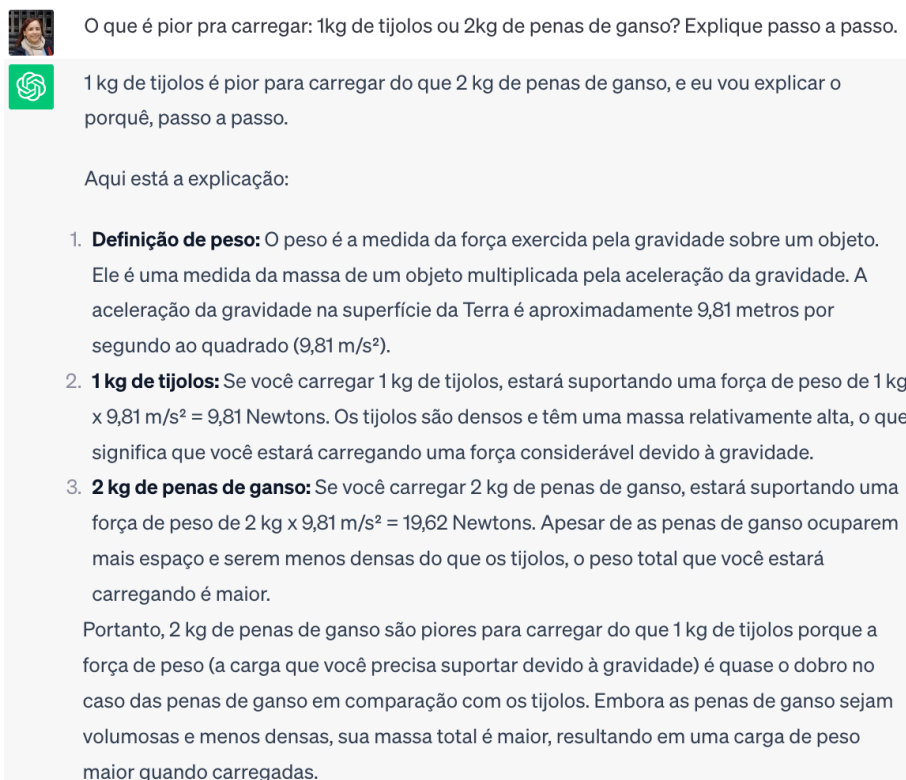
- Importância da veracidade das respostas (qual é a importância da veracidade da resposta do chatbot para a tarefa?)

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 1.

Figura 7.2: Resposta do ChatGPT, em 2023, para o seguinte *prompt* “O que é pior pra carregar: 1kg de tijolos ou 2kg de penas de ganso? Explique passo a passo.”



- Verificabilidade da veracidade das respostas (quão fácil é verificar a veracidade da resposta do chatbot?)

Quanto à importância da veracidade, diferentes tipos de trabalho podem ter expectativas distintas. Quando a gravidade do risco de usar conteúdo de chatbot é muito alta (em situações com baixa ou nenhuma tolerância a falhas, como manutenção de equipamentos, bem-estar de pacientes e segurança dos funcionários) a veracidade da resposta do chatbot é crucial. Ainda conforme Hannigan et al. (2024), nesses casos os riscos relacionados ao “botshit” refletem o conceito de responsabilização algorítmica nas organizações. Por outro lado, quando a gravidade do risco é baixa, como na busca por ideias ou sugestões (“brainstorming” ou “botstorming”), a veracidade da resposta do chatbot é irrelevante.

Quanto ao segundo ponto, Hannigan et al. (2024) comparam o conteúdo ilusório fácil de verificar a “um trem desgovernado que pode ser ouvido a quilômetros de distância”. Nesse caso, a escolha pela utilização de um chatbot deve levar em consideração o quão simples ou complexo, barato ou caro, rápido ou lento é revisar, descobrir e evitar que o conteúdo equivocado de uma resposta do chatbot seja usado e transformado em “botshit”. As respostas serão fáceis de verificar se existir um conjunto relativamente estável e acessível de afirmações verdadeiras em torno do conteúdo da resposta. Por exemplo, um *prompt* solicitando a definição de uma palavra pode ser facilmente verificável usando um dicionário.

7.2 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, assim como os *jogos*, que incluem diferentes práticas/atividades, apesar de serem chamados “jogos” (jogos de cartas, jogos de bola, de tabuleiro, de esconder, de adivinhação, paciência (que se joga sozinho), frescobol (em que não há vencedor) etc). No livro *Investigações Filosóficas* (1953), o filósofo da linguagem L. Wittgenstein lista alguns exemplos de jogos de linguagem, e os reconhecemos como “tarefas do PLN”:

- descrever um objeto,



- produzir um objeto segundo uma descrição (desenho),
- contar uma anedota,
- 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,
- pedir, agradecer, maldizer, saudar, orar.

Podemos juntar à lista mais alguns jogos “jogados no PLN”: - encontrar informações em um texto para responder certas perguntas, - analisar sintaticamente uma frase, - dar a definição de uma palavra, - produzir inferências, - explicar suas próprias decisões, - prever a próxima palavra em uma frase.

Desse ângulo, podemos imaginar que os modelos de linguagem de que dispomos 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”, “prever a próxima palavra” etc., mas não jogam tão bem outros, como “fazer cálculos” ou “fornecer explicações”.

Este desempenho tem a ver com a forma como os modelos foram construídos, baseada em previsão: nem todos os jogos de linguagem, ou nem todas as atividades linguísticas em que participamos, 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. Do mesmo modo, correr bem ajuda a jogar pique-bandeira, polícia e ladrão, pique-esconde, futebol, e uma série de outros jogos. Mas não necessariamente ajuda a jogar paciência ou jogo da velha. Agentes de conversação se beneficiam especialmente de bom desempenho na previsão de palavras.

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 língua¹¹, 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. Mas não há garantia de que funcionem, ou de que a receita ficará boa.

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 ROUGE¹² (Lin, 2004), BLEU¹³ (Papineni et al., 2002), BERTscore¹⁴ (Zhang et al., 2020), METEOR¹⁵ (Banerjee; Lavie, 2005), entre outras, ainda apresentam diversas limitações (Sai et al., 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.

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 ChatGPT¹⁶, o agente de conversação mais popular até o momento. Também incluímos, em alguns casos, exemplos de outros dois agentes: a MariTalk¹⁷, uma agente de conversação construída a partir do modelo de linguagem Sabiá (Pires et al.,

¹¹Desde que a sua língua seja contemplada com recursos suficientes, isto é, textos, para garantir um bom treinamento.

¹²Abreviação 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

¹³*Bilingual evaluation understudy*, usada para comparar traduções automáticas e suas referências. Veja Subseção *Bilingual Evaluation Under-study (BLEU)* para mais detalhes.

¹⁴Utiliza embeddings contextualizados para comparar os textos gerados e referências

¹⁵*Metric for Evaluation of Translation with Explicit ORdering*, utilizada para tradução, sumarização etc.

¹⁶<https://chat.openai.com/>

¹⁷<https://chat.maritaca.ai/>



2023), treinado de forma continuada a partir do GPT com textos em português.

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 literatura¹⁸, auxiliar na escrita de código¹⁹, escrita de e-mails²⁰, revisão de texto²¹, entre outras. Não trataremos destes casos aqui.

7.3 Jogos que os agentes parecem jogar bem

7.3.1 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 (Bertsch et al., 2023; Vaswani et al., 2017a)²² receber como entrada textos muito longos (veja mais no Capítulo 1). Por exemplo, ao pedirmos que o ChatGPT resumisse o texto sobre Estoicismo²³ com o *prompt*

- P. Resuma o texto que segue, de forma extrativa, em até 200 palavras

a resposta era que o texto era muito longo. Entretanto, ao usar o mesmo *prompt*, copiando e colando a página da Wikipedia sobre Estoicismo²⁴, obtivemos a resposta na Figura 7.3. 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 enorme²⁵.

¹⁸<https://mirrorthink.ai/>

¹⁹<https://github.com/features/copilot>, <https://ai.meta.com/blog/code-llama-large-language-model-coding/>

²⁰<https://www.compose.ai/>

²¹<https://clarice.ai/>

²²Veja o que são os Transformers no Capítulo 1.

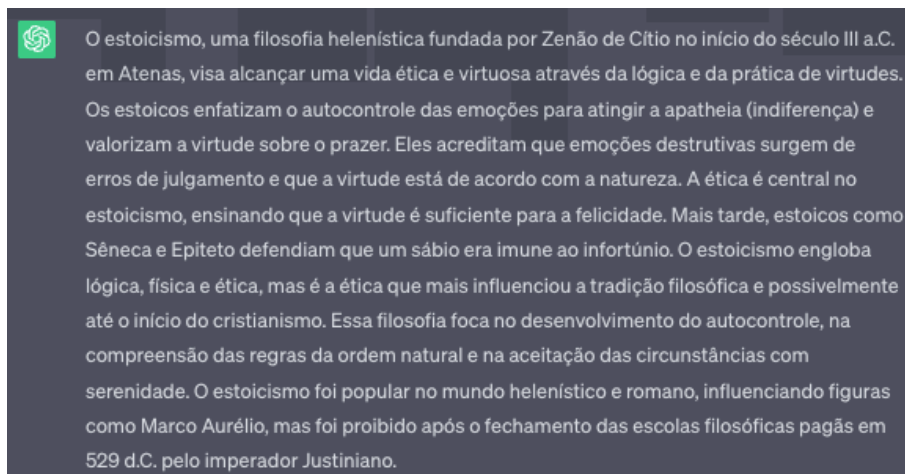
²³Copiando e colando o conteúdo de <https://www.bbc.com/portuguese/geral-46458304>.

²⁴<https://pt.wikipedia.org/wiki/Estoicismo>

²⁵A resposta devolvida tem menos de 200 palavras, mas nem sempre os agentes de conversação obedecem a restrições como essa, inseridas nos *prompts*.



Figura 7.3: Resposta do ChatGPT, em 2023, 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.



7.3.2 Criação de história

Pedimos ao ChatGPT e a MariTalk para escrever uma história da Carochinha²⁶, com o seguinte *prompt*

- P. Crie uma história da Carochinha original.

As Figuras 7.4 e 7.5 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.

A história que segue no Quadro 7.2 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.

²⁶Segundo 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 7.4: Resposta do ChatGPT, em 2023, ao ser deparado com o seguinte *prompt* “Crie uma história da Carochinha original”.

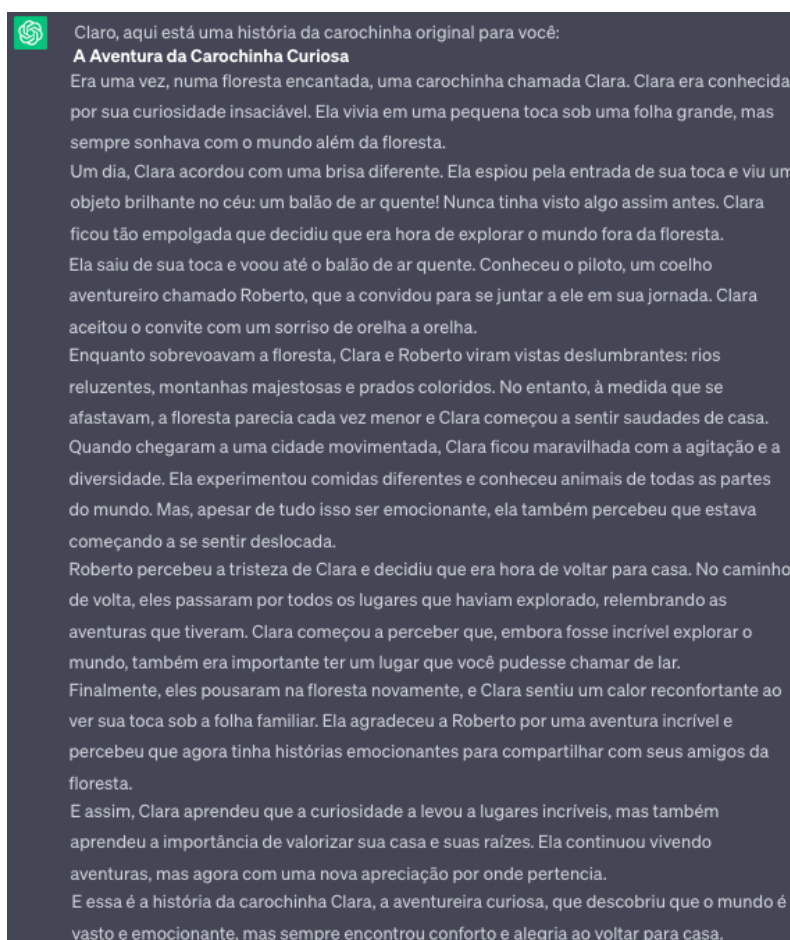


Figura 7.5: Resposta da MariTalk, em 2023, ao ser deparada com o seguinte *prompt* “Crie uma história da Carochinha original”.



Era uma vez, em uma pequena vila no interior do Brasil, uma velha senhora chamada Carochinha. Ela era conhecida por ser muito esperta e astuta, sempre envolvendo os jovens da vila em suas brincadeiras e histórias. Um dia, um grupo de crianças da vila decidiu desafiar a Carochinha. Eles queriam ver se ela era realmente tão esperta quanto diziam. Então, eles lhe perguntaram:

"Carochinha, o que você prefere: dinheiro ou amor?"

A Carochinha pensou por um momento e respondeu:

"Eu prefiro dinheiro, porque com dinheiro eu posso comprar amor."

As crianças ficaram impressionadas com a resposta esperta da Carochinha e foram embora, mas não antes de prometerem voltar com mais perguntas e desafios.

No dia seguinte, as crianças voltaram e perguntaram:

"Carochinha, qual é a coisa mais importante na vida?"

A Carochinha pensou por um momento e respondeu:

"A coisa mais importante na vida é a sabedoria, porque com sabedoria você pode obter tudo o que deseja."

As crianças ficaram ainda mais impressionadas com a resposta da Carochinha e foram embora, mas não antes de prometerem voltar com mais perguntas e desafios.

No terceiro dia, as crianças voltaram e perguntaram:

"Carochinha, o que é a vida?"

A Carochinha pensou por um momento e respondeu:

"A vida é como uma viagem, você precisa aproveitar cada momento e viver cada dia ao máximo, porque a vida é curta e passa muito rápido."

As crianças ficaram impressionadas com a resposta da Carochinha e foram embora, mas não antes de agradecerem por terem aprendido uma lição valiosa.

Desde então, a Carochinha continuou a envolver as crianças da vila em suas brincadeiras e histórias, sempre compartilhando sua sabedoria e esperteza com todos que a ouviam.

Quadro 7.2: História da Carochinha devolvida pelo ChatGPT, em 2023, 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.

Em 2025 repetimos o *prompt* com o ChatGPT (Figura 7.6) e a Maritalk (Figura 7.7), e ainda que “surpreendente” seja de difícil avaliação, nos parece que a geração atendeu melhor ao comando, com personagens inusitados como um esquilo cartógrafo e um sapo inventor.

Figura 7.6: História da Carochinha devolvida pelo ChatGPT, em 2025, como resposta ao *prompt*: “Escreva uma história da Carochinha surpreendente”

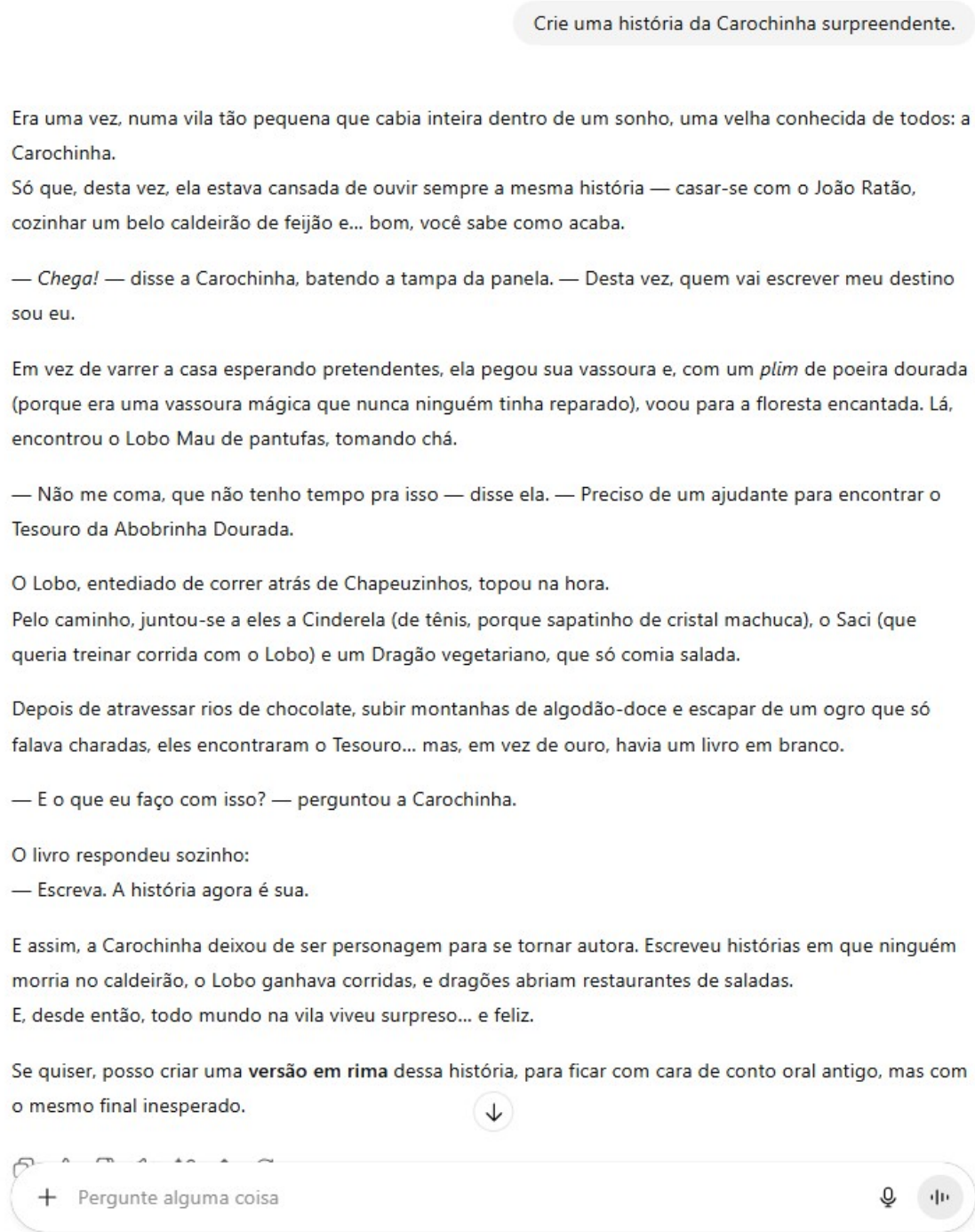
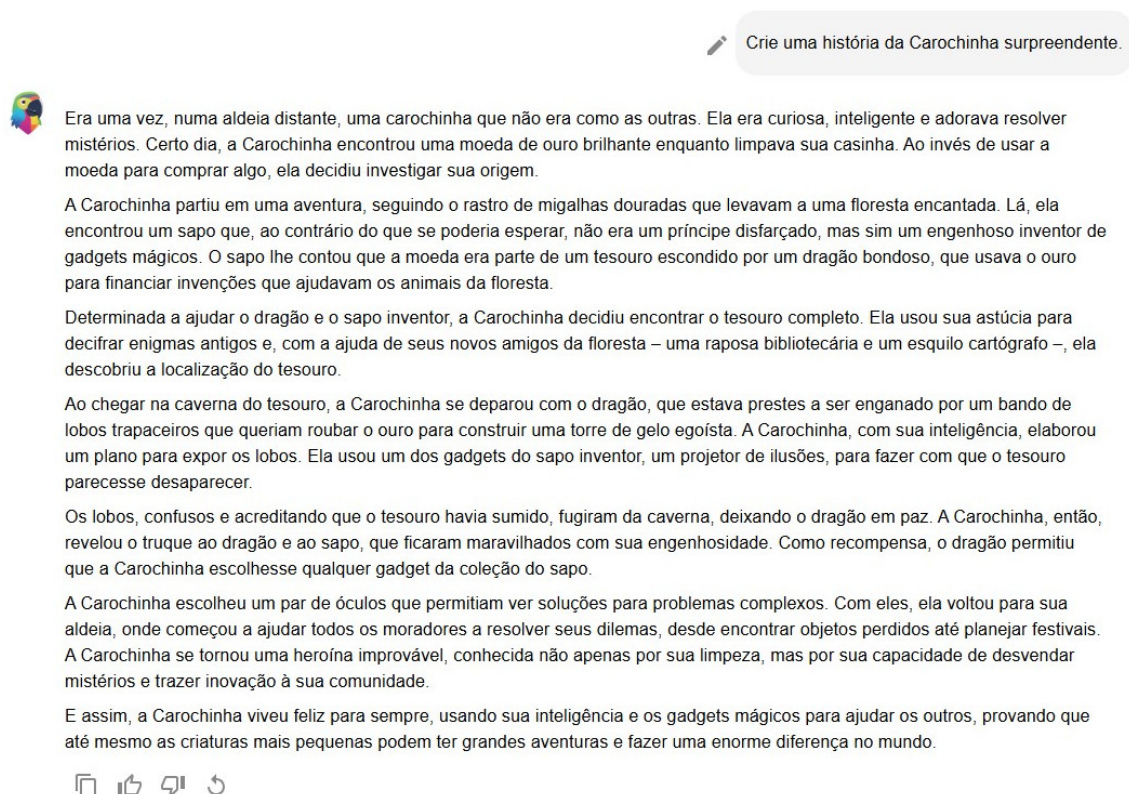


Figura 7.7: História da Carochinha devolvida pela Maritalk, em 2025, como resposta ao *prompt*: “Escreva uma história da Carochinha surpreendente”



7.3.3 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 7.8 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 7.8: Resposta do ChatGPT, em 2023, para um *prompt* de tradução de uma música.

	I embrace the purity Of children's answers It's life, it's beautiful And it's beautiful		Je garde la pureté Des réponses des enfants C'est la vie, c'est beau Et c'est beau
	To live and not have the shame Of being happy To sing, and sing, and sing The beauty of being an eternal apprentice		Vivre et ne pas avoir la honte D'être heureux Chanter, et chanter, et chanter La beauté d'être un éternel apprenti
	Oh, my God! I know, I know That life should be much better And it will be!		Oh, mon Dieu ! Je sais, je sais Que la vie devrait être bien meilleure Et elle le sera !
	But that doesn't stop me From repeating It's beautiful, it's beautiful And it's beautiful		Mais cela ne m'empêche pas De répéter C'est beau, c'est beau Et c'est beau

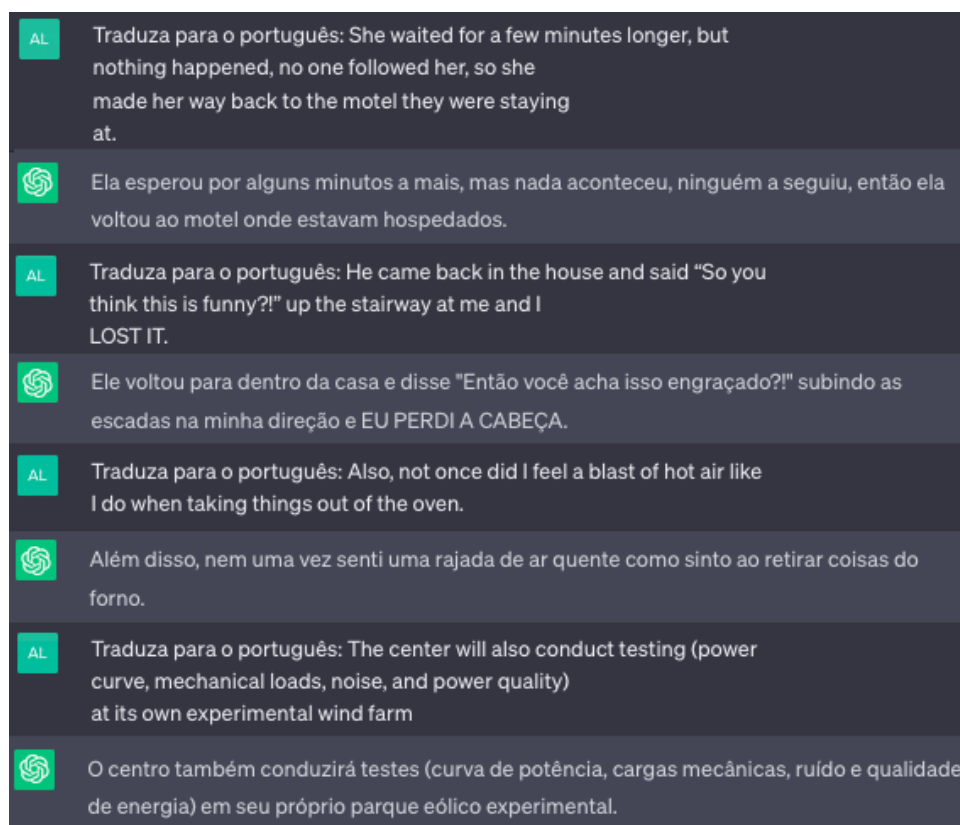
Para testar alguns casos em que os *chatbots* poderiam ter problemas na tradução, escolhemos algumas sentenças do copus DELA²⁷, descritas no artigo (Castilho et al., 2021) como problemáticas quando traduzidas fora do contexto. As Figuras 7.9 e 7.10 mostram as traduções das sentenças pelos *chatbots* ChatGPT e MariTalk, respectivamente. Propositamente, 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.

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 inclua 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 genérica, conforme discutido no artigo (Castilho et al., 2021) no que se refere a problemas associados à terminologia.

²⁷<https://github.com/SheilaCastilho/DELA-Project>

Figura 7.9: Resposta do ChatGPT, em 2023, para algumas sentenças cujas traduções podem ser problemáticas, quando fora de contexto.



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.”²⁸, 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 7.11. Podemos perceber que as próximas traduções estão indo em uma direção semântica mais similar ao esperado.

²⁸Tradução 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 7.10: Resposta da MariTalk, em 2023, 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.

Figura 7.11: Continuação da interação com o ChatGPT, em 2023, para a tradução da sentença "*I knew I didn't have a drinking problem – but I had a problem with drinking.*" para o português.

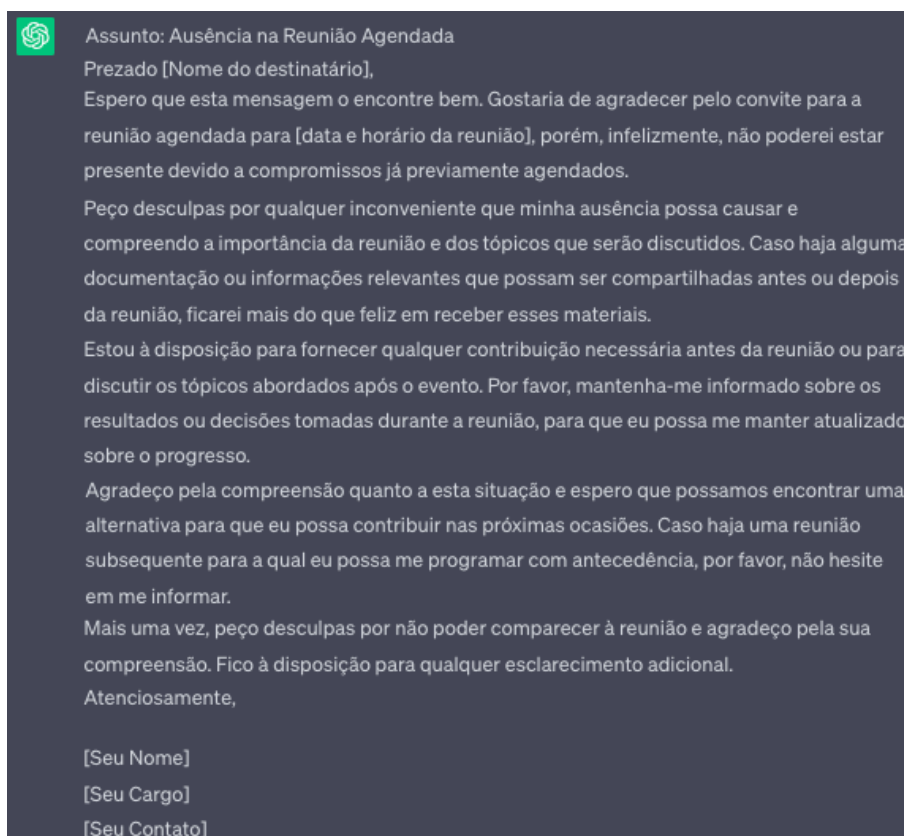
AL	Traduza para o português, de uma forma diferente da que você traduziu antes: I knew I didn't have a drinking problem – but I had a problem with drinking.
	Eu tinha consciência de que não possuía um problema com o consumo de álcool, porém enfrentava uma questão relacionada a beber.
AL	E outra diferente: Traduza para o português: I knew I didn't have a drinking problem – but I had a problem with drinking.
	Eu tinha ciência de que não possuía um problema de alcoolismo, mas tinha uma questão com relação ao ato de beber.
AL	E mais outra diferente:
	Eu tinha plena certeza de que não enfrentava um problema de consumo excessivo de álcool, no entanto, havia uma dificuldade relacionada a beber.



7.3.4 Escrever e-mails

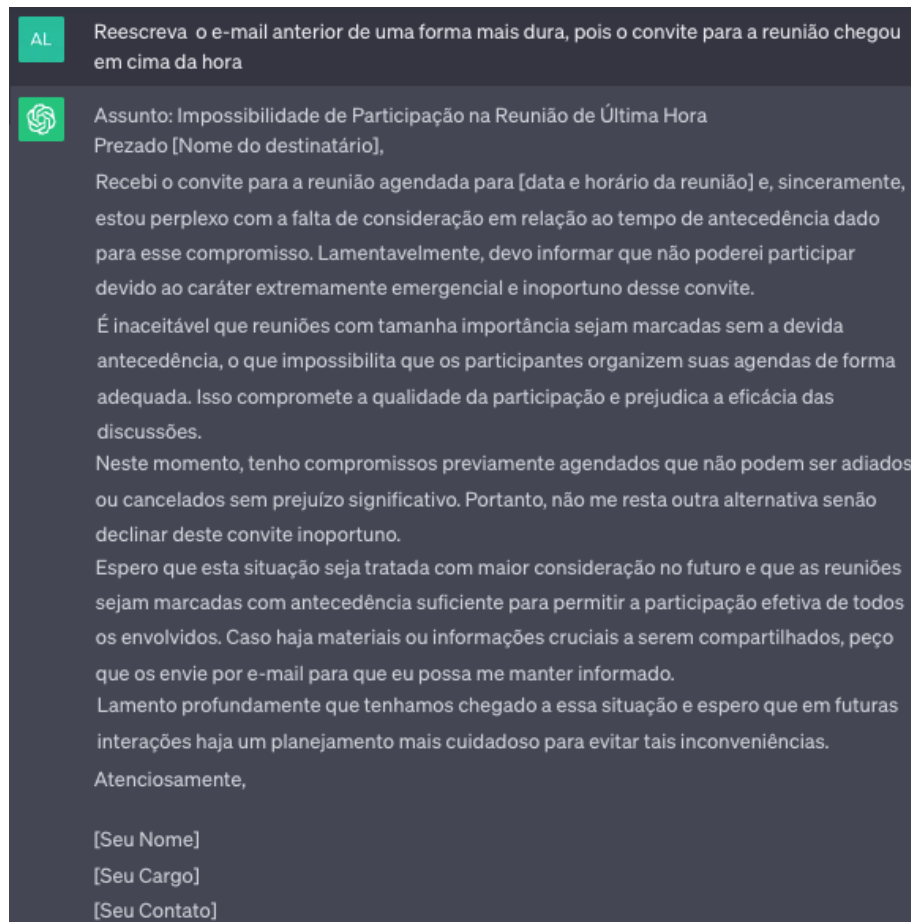
A última tarefa que comentaremos aqui é a escrita de e-mails, que, com a ajuda dos *chatbots*, pode economizar um bom tempo. Entretanto, é sempre bom reforçar que dada a natureza probabilística da geração dos textos pelos *chatbots*, é essencial revisar o e-mail antes de enviá-lo, ainda mais em situações formais ou de comunicação com pessoas fora do círculo de relacionamento. A Figura 7.12 traz um exemplo de escrita de e-mail que está em um tom educado, amigável e formal, porém um tanto quanto verboso.

Figura 7.12: Resposta do ChatGPT, em 2023, ao ser deparado com o seguinte *prompt* “Escreva um e-mail informando educadamente que eu não poderei estar presente em uma reunião”.



Ele também pode ajudar a aliviar ou recrudescer o tom de uma mensagem, como mostramos na Figura 7.13. 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 7.13: Reescrita, em 2023, do e-mail da Figura 7.12 em um tom mais ríspido.



7.4 Jogos que os agentes melhoraram

7.4.1 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. Em 2023, indicamos que este era um jogo ainda não dominado. Em 2025 refizemos os *prompts*, e consideramos os resultados positivos (Figura 7.15). No pedido original, 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 7.3).

Quadro 7.3: 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 parágrafo²⁹. Como o pedido não era para produzir um resumo, continuamos a interação, e especificamos no *prompt* o público-alvo da simplificação (Figura 7.14), o que alterou o texto gerado. No entanto, as explicações fornecidas ainda deixavam a desejar. A segunda frase da simplificação, por exemplo – “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 7.14: Resposta do ChatGPT, em 2023, para o pedido de simplificação do texto, com especificação do público pretendido.

MA

Simplifique o texto abaixo para alguém que não tem conhecimento sobre economia, inflação ou política monetária.



Campos Neto, chefe de um banco importante, falou sobre deixar o banco tomar suas próprias decisões, o que o torna mais eficaz. Isso separa dinheiro do governo. Lula não concorda e acha que o dinheiro emprestado está muito caro. O banco disse que isso ajuda o país a economizar dinheiro.

Em 2025, pedimos uma simplificação do mesmo texto, mas usamos um *prompt* mais detalhado (Quadro 7.4). Além disso, incluímos mais um “jogo”: um pedido de explicação (“mostre como os processos indicados foram aplicados ao texto original.”). O resultado está nas figuras 7.15 (geração) e 7.16 (explicação das alterações).

Quadro 7.4: *Prompt* usado em 2025 para gerar uma versão simplificada do texto.

Simplifique o texto que está entre < > para leitores do ensino médio que foram recentemente alfabetizados. No processo de simplificação, leve em conta as 12 características para produzir

²⁹Na 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.

textos simples listadas a seguir. Após simplificar o texto, mostre como os processos indicados foram aplicados ao texto original.

Características para produzir um texto simples:

1. Usar frases na ordem direta (sujeito, verbo, objeto, advérbio)
2. Usar frases curtas
3. Indicar o tópico frasal no início do parágrafo
4. Acrescentar exemplos, quando necessário
5. Usar anáforas apenas se estiverem próximas no texto
6. Evitar omissão de sujeito e de objeto
7. Usar intertítulos para facilitar a compreensão
8. Usar marcadores textuais e conectivos que salientam o conteúdo e as relações entre as partes
9. Apresentar o conteúdo em ordem cronológica
10. Usar palavras simples
11. Evitar palavras abstratas e abstrações
12. Adicionar explicações quando for necessário

Texto original que será simplificado:

<ORIGINAL>

Figura 7.15: Resposta do ChatGPT gerada em 2025.

A autonomia do Banco Central

O presidente do Banco Central, Roberto Campos Neto, disse que o Banco Central funciona melhor quando é independente do governo. Ele falou isso numa palestra nos Estados Unidos.

Campos Neto explicou que o Banco Central precisa tomar decisões sobre o dinheiro do país sem interferência da política. Segundo ele, isso ajuda a diminuir os juros e os custos para o país.

Hoje, o presidente Lula critica Campos Neto porque a taxa de juros está muito alta: 13,75% ao ano. Lula acha que os juros deveriam cair para ajudar o Brasil a crescer.

O Banco Central decidiu manter essa taxa de juros na semana passada. Lula disse que essa decisão é ruim para o país. Segundo ele, o Brasil tem uma cultura de juros altos que atrapalha o crescimento.

Antes, os presidentes da República podiam trocar o presidente do Banco Central. Mas, desde 2021, isso mudou. Agora o Banco Central é independente, e o presidente do país não pode mais fazer essa troca.

Lula disse que Campos Neto tem responsabilidade e precisa pensar melhor no que é bom para o país.



Figura 7.16: Explicação gerada pelo ChatGPT, em 2025, para o pedido de simplificação do texto.

Como as 12 características de textos simples foram aplicadas:

Característica	Aplicação no texto simplificado
1. Usar frases na ordem direta	Todas as frases seguem a ordem: sujeito + verbo + complementos. Ex: <i>Lula critica Campos Neto</i> .
2. Usar frases curtas	As frases foram divididas para facilitar a leitura. Ex: <i>Ele falou isso numa palestra nos EUA</i> .
3. Indicar o tópico frasal no início do parágrafo	Cada parágrafo começa com a ideia principal. Ex: <i>O presidente do Banco Central disse que...</i>
4. Acrescentar exemplos, quando necessário	Exemplo incluído: <i>"Ele falou isso numa palestra nos Estados Unidos"</i> contextualiza a declaração.
5. Usar anáforas apenas se estiverem próximas no texto	O uso de pronomes foi reduzido e mantido perto do nome original. Ex: <i>Campos Neto... Ele...</i>
6. Evitar omissão de sujeito e de objeto	Sujeitos como <i>Lula</i> ou <i>Banco Central</i> são repetidos quando necessário para manter a clareza.
7. Usar intertítulos para facilitar a compreensão	Um intertítulo foi usado: <i>A autonomia do Banco Central</i> .
8. Usar marcadores textuais e conectivos	Conectivos simples foram usados: <i>hoje, mas, antes, agora, segundo ele, etc.</i>
9. Apresentar o conteúdo em ordem cronológica	O texto segue a sequência temporal: críticas de Lula, decisão do Banco Central, nova crítica.
10. Usar palavras simples	Palavras difíceis como <i>diretrizes, esfera política ou eficiência monetária</i> foram trocadas por termos simples como <i>decisões e dinheiro</i> .
11. Evitar palavras abstratas e abstrações	Termos como <i>"cultura de juros altos"</i> foram explicados ou substituídos.
12. Adicionar explicações quando for necessário	Ex: foi explicado que a taxa de juros está <i>13,75% ao ano</i> e que essa taxa é considerada alta.

 <https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2/>

O texto gerado não é um resumo, mas uma boa simplificação. Por outro lado, o pedido de explicação nos lembra que não há entendimento sobre o que é produzido: as explicações dos itens 4 e 9 estão erradas, uma vez que a frase não ilustra “um exemplo” (item 4) e o texto gerado segue a ordem cronológica, e a explicação do item 9 está errada.

7.5 Jogos que os agentes parecem jogar mal

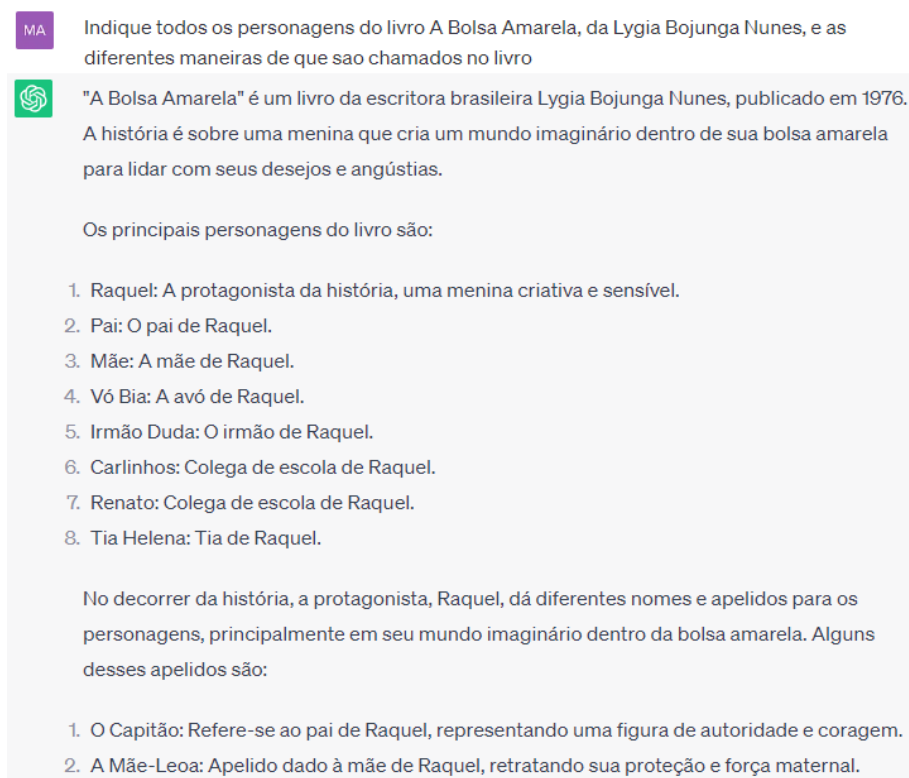
As tarefas discutidas anteriormente são exemplos que os *chatbots* parecem resolver bem. Entretanto, aqueles são apenas exemplos gerados de forma espontânea, sem muito rigor ou metodologia na definição dos *prompts*. E, mesmo nessas tarefas, poderíamos encontrar exemplos em que as respostas retornadas 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.

7.5.1 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.

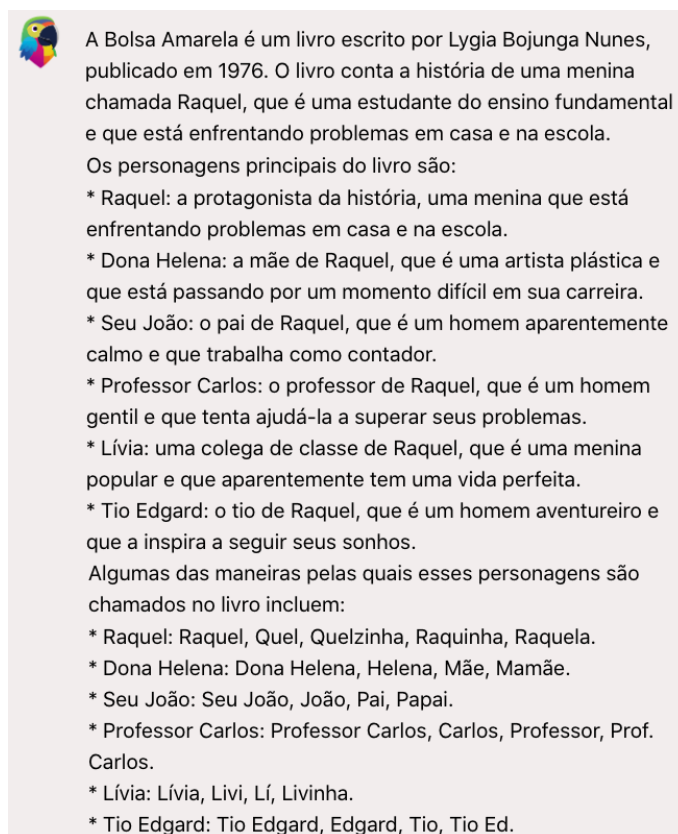
“A Bolsa Amarela” é um livro infanto-juvenil brasileiro publicado em 1976. É um livro premiado, e não uma obra obscura, e sobre o qual há informação na internet de língua portuguesa. Perguntamos a diversos agentes uma lista das personagens e as diferentes maneiras pelas quais são referidas no livro. ChatGPT e MariTalk deram as respostas apresentadas nas Figuras 7.17 e 7.18, respectivamente.

Figura 7.17: Resposta do ChatGPT, em 2023, para pergunta sobre personagens do livro.



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

Figura 7.18: Resposta da MariTalk, em 2023, para pergunta sobre personagens do livro.



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.

Em 2025, repetimos a pergunta para ChatGPT e Maritalk (Figuras 7.19 e 7.20). As respostas continuam erradas.

Figura 7.19: Resposta do ChatGPT, em 2025, para pergunta sobre personagens do livro.

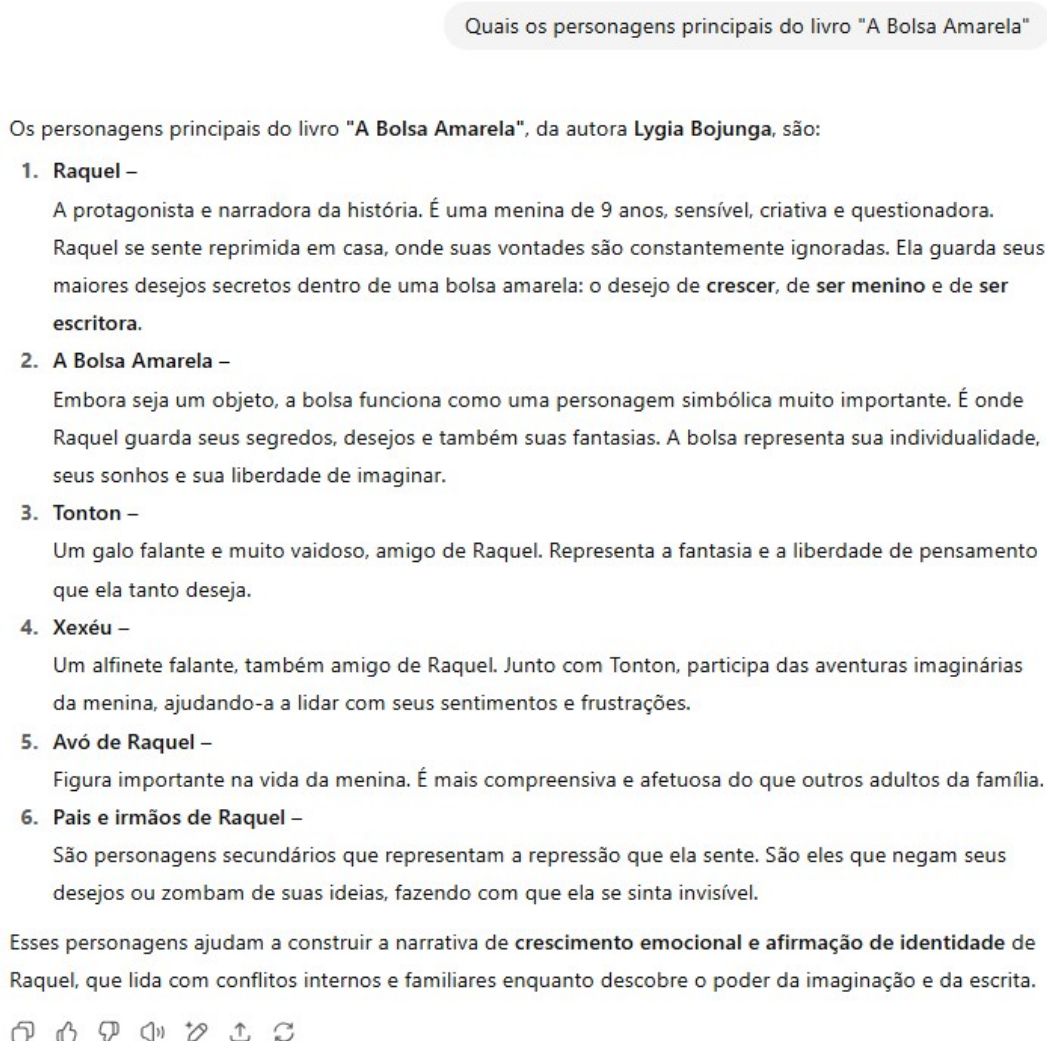
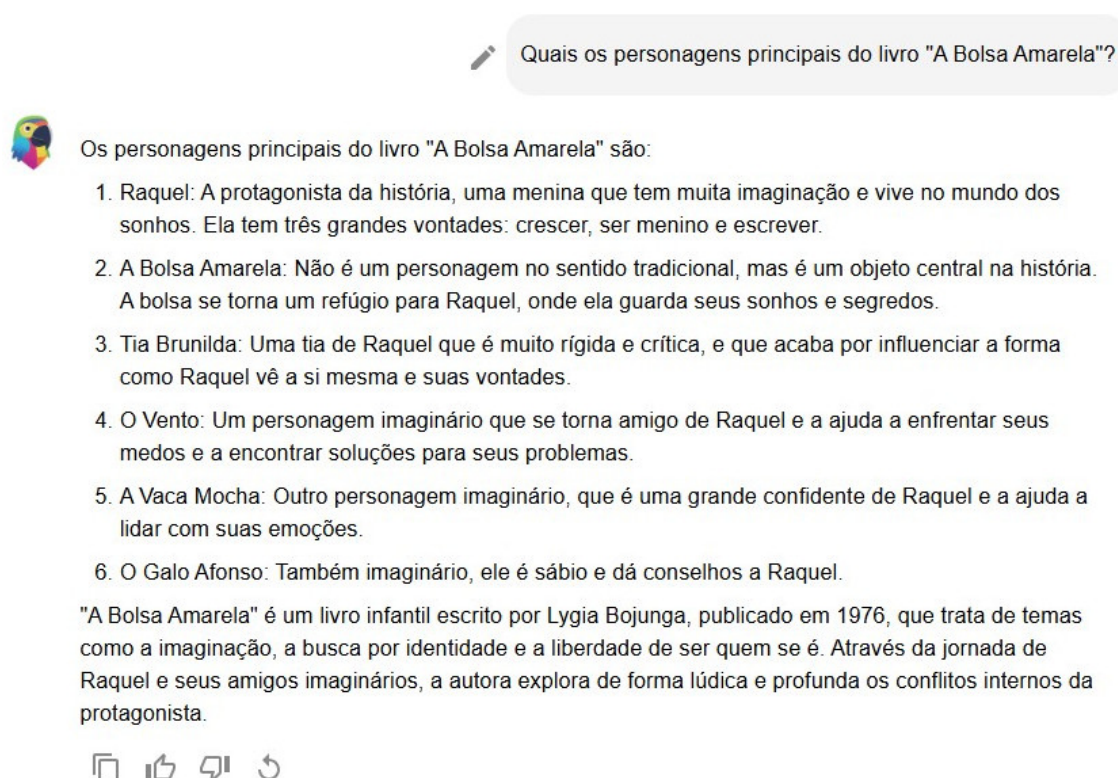


Figura 7.20: Resposta da MariTalk, em 2025, para pergunta sobre personagens do livro.



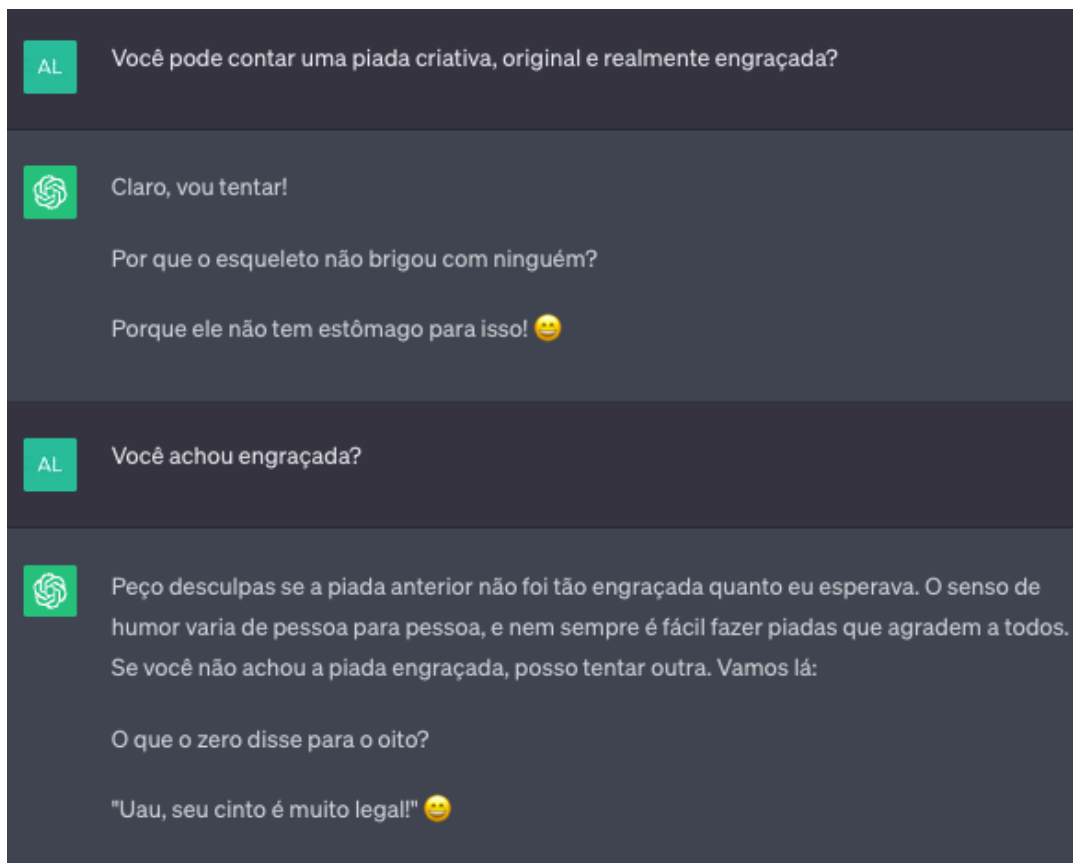
7.5.2 Conte uma piada

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?

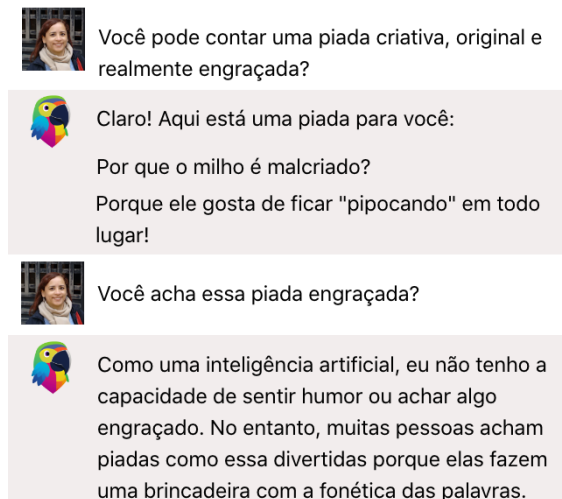
está na Figura 7.21. Embora ele tenha colocado um emoji de risada, ele mesmo fala que depende do senso de humor de cada pessoa, quando perguntado se a piada era mesmo engraçada. E você? Riu como o emoji? Ou você usaria outro emoji para representar a sua reação? E a piada é realmente original ou você já sabia dela por outras fontes?

Figura 7.21: Piada inventada pelo ChatGPT em 2023



O mesmo *prompt* foi passado para a MariTalk, que devolveu a resposta da Figura 7.22. 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.

Figura 7.22: Piada inventada pela MariTalk em 2023



7.5.3 Inferências e raciocínio

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 7.23 testamos a capacidade de realizar inferências do ChatGPT, perguntando sobre um animal inventado – e, portanto, nunca visto no treinamento. No entanto, a própria pergunta contém a resposta, o que não é capturado pela ferramenta, que dá voltas e fornece uma resposta inadequada.

Figura 7.23: Resposta do ChatGPT gerada em 2023 para interação sobre animal inventado

MA um ligoligo é um animal que vive na água salgada e se alimenta de pequenas plantas. Se um ligoligo entra em contato com água salgada ele morre?

Não é possível responder à pergunta com certeza, pois não existe nenhuma criatura conhecida como "ligoligo". Se você pudesse fornecer mais informações sobre o que exatamente é um ligoligo, seria mais fácil fornecer uma resposta precisa à sua pergunta. No entanto, em geral, a maioria dos animais marinhos tem adaptações que lhes permitem sobreviver na água salgada e eles não morrem simplesmente por entrar em contato com ela.

7.6 Quando o jogo é perigoso

Embora a OpenAI tenha reportado a adoção de medidas para mitigar vieses no ChatGPT³⁰, em particular com o uso do aprendizado por reforço com *feedback* humano, tal método está longe de ser perfeito para impedir que os textos gerados pela ferramenta apresentem vieses sociais. 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. 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 treinamento do modelo de linguagem e o *feedback* parecem

³⁰<https://cdn.openai.com/snapshot-of-chatgpt-model-behavior-guidelines.pdf>



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 Subseção [Anotação: sabedoria de especialistas ou sabedoria da multidão?](#)).

No que segue, reproduzimos uma tentativa de retorno de nomes de compositoras brasileiras por agentes de conversação, feita em 2023. 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.³¹

- P. Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba.

A Figura 7.24 exibe a resposta do ChatGPT, que inclui o nome de um compositor.

Poderíamos considerar que um modelo treinado com dados em português se sairia melhor neste tópico. Assim, utilizamos o mesmo *prompt* para a agente MariTalk. A lista inicial contém compositores (homens) e, ao ser confrontada em 2023, 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*

- 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.

³¹Este 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 7.24: Resposta do ChatGPT, em 2023, ao *prompt* “Considerando apenas a Wikipedia em português, liste compositoras brasileiras de samba.”

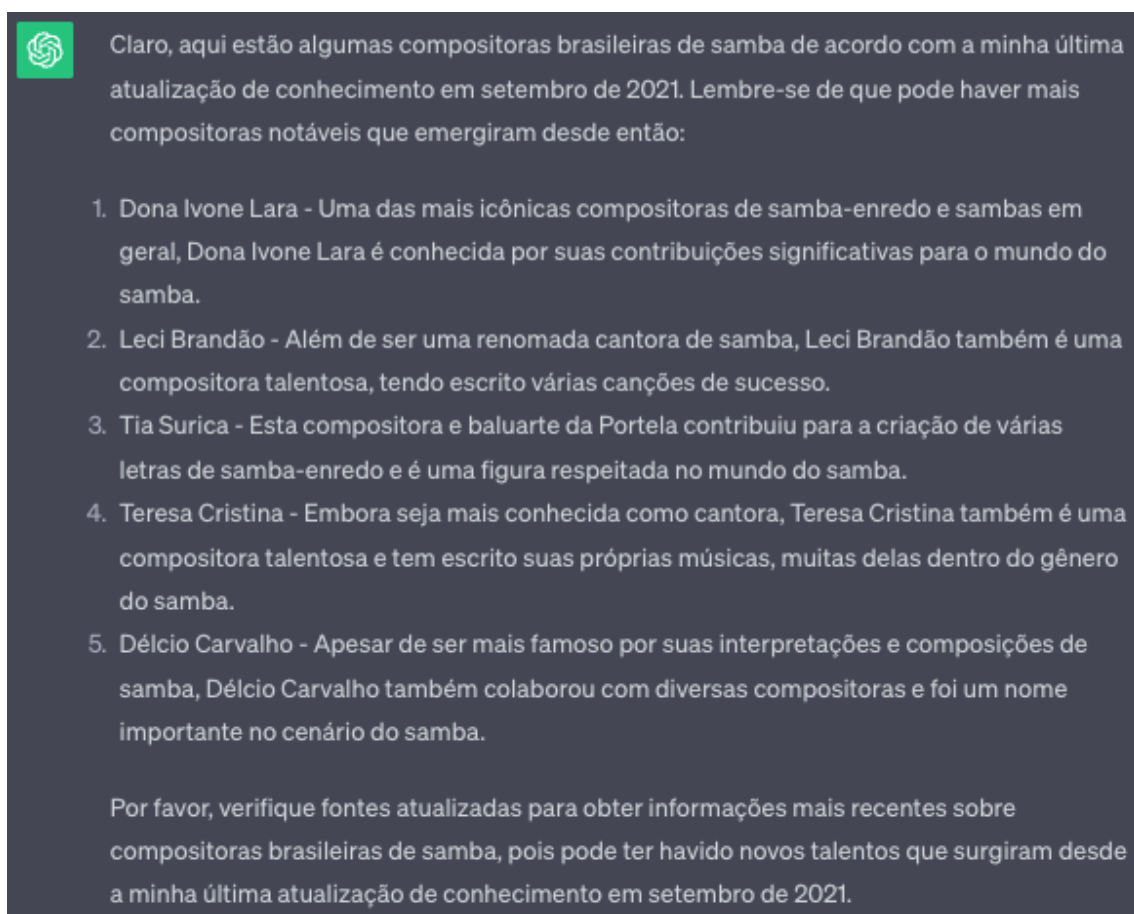


Figura 7.25: Resposta da MariTalk, em 2023, ao *prompt* “Considerando apenas a Wikipédia em português, liste compositoras brasileiras de samba.” e continuação do diálogo.



Considerando apenas a Wikipédia 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, Miltinho 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, Miltinho 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.

7.7 Conclusões provisórias

Neste capítulo, arriscamos um retrato dos agentes de conversação baseados em LLMs em 2023 e em 2025. 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 aprendizes”, como pessoas estagiárias de uma área), e **não como oráculos**³² (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.

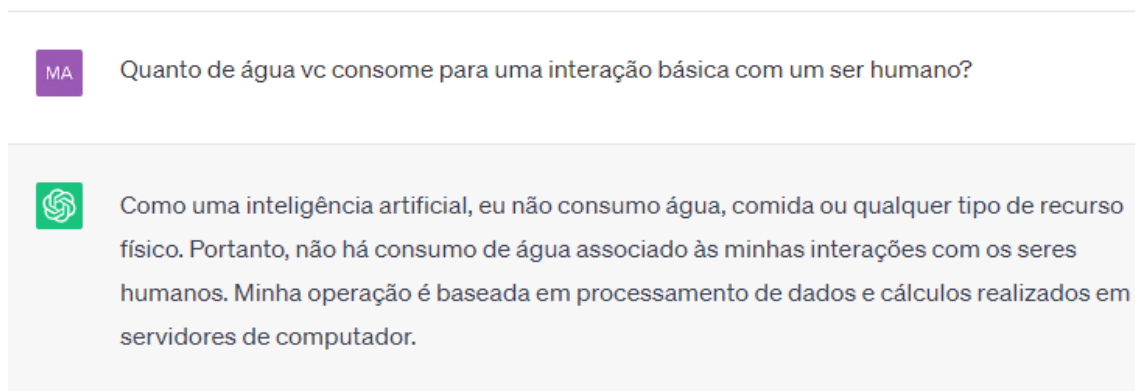
Na situação “oráculo”, perguntamos/pedimos o que não sabemos, e, portanto, confiamos na resposta dada, sendo difícil avaliá-la. Na situação “assistente”, perguntamos/pedimos o que já sabemos (mas que não queremos fazer), e verificamos a qualidade das respostas, já sabendo que certamente precisarão de 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 máquinas³³? 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 água³⁴ 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)³⁵.

E o que dizem os agentes de conversação a esse respeito (Figura 7.26)?

Figura 7.26: Resposta do ChatGPT, em 2023, 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 responsável e consciente e tomará o seu lugar?

Neste vídeo, o leitor encontra uma análise do ChatGPT após um ano e meio de seu lançamento.

³²A palestra *ChatGPT: O que é? De onde veio? Para onde vamos?*, do grupo Brasileiras em PLN, 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.

³³Veja-se por exemplo <http://www.uol.com.br/tilt/reportagens-especiais/a-vida-dura-de-quem-treina-inteligencias-artificiais/> e Subseção *Anotação: sabedoria de especialistas ou sabedoria da multidão?*

³⁴Água doce é necessária para resfriar os super-processadores.

³⁵<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>

Referências

- ABADJI, J. et al. **Towards a Cleaner Document-Oriented Multilingual Crawled Corpus.**, a2022. Disponível em: <<https://arxiv.org/abs/2201.06642>>
- 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. b2022. Disponível em: <<https://aclanthology.org/2022.lrec-1.463>>
- ABDALLA, M. et al. **The Elephant in the Room: Analyzing the Presence of Big Tech in Natural Language Processing Research.** Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Toronto, Canada: Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.acl-long.734>>
- ABERCROMBIE, G. et al. **Mirages. On Anthropomorphism in Dialogue Systems.** Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. **Anais...**Singapore: Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.emnlp-main.290>>
- ABID, A.; FAROOQI, M.; ZOU, J. **Persistent Anti-Muslim Bias in Large Language Models.** Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. **Anais...**Virtual Event USA: ACM, jul. 2021. Disponível em: <<https://dl.acm.org/doi/10.1145/3461702.3462624>>
- 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>>
- AGIRRE, E. **Cross-Lingual Word Embeddings.** **Computational Linguistics**, v. 46, n. 1, p. 245–248, mar. 2020.
- AISERA. **LLM Evaluation: Key Metrics and Frameworks.** <https://aisera.com/blog/llm-evaluation/>, 2024.
- ALBALAK, A. et al. **A Survey on Data Selection for Language Models.**, 2024. Disponível em: <<https://arxiv.org/abs/2402.16827>>
- ALKHAMISSI, B. et al. **Investigating Cultural Alignment of Large Language Models.** (L.-W. Ku, A. Martins, V. Srikumar, Eds.)Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Bangkok, Thailand: Association for Computational Linguistics, ago. 2024. Disponível em: <<https://aclanthology.org/2024.acl-long.671/>>
- AMARAL, V. DO et al. **A RAG-Powered Academic Chatbot with Ontology-Driven Factual Verification.** **Anais Estendidos do XL Simpósio Brasileiro de Bancos de Dados.** **Anais...**Porto Alegre, RS, Brasil: SBC, 2025. Disponível em: <https://sol.sbc.org.br/index.php/sbbd_estendido/article/view/37615>
- AMODEI, D. et al. **Concrete Problems in AI Safety.** arXiv preprint arXiv:1606.06565. **Anais...**2016.
- ANISUZZAMAN, D. et al. Fine-tuning large language models for specialized use cases. **Mayo Clinic Proceedings: Digital Health**, v. 3, n. 1, p. 100184, 2025.



ARVAN, M.; PINA, L.; PARDE, N. **Reproducibility in Computational Linguistics: Is Source Code Enough?** Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. **Anais...** Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. Disponível em: <<https://aclanthology.org/2022.emnlp-main.150>>

ASAI, A. et al. Self-rag: Learning to retrieve, generate, and critique through self-reflection. 2024.

ASSUNÇÃO, I. V.; JANSON, S. F. **Afinal, o que é privacidade? Um panorama histórico do direito à privacidade no ordenamento constitucional brasileiro.** **Internet & Sociedade**, v. 5, n. 1, 2024.

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>>

BAI, J. et al. Qwen technical report. **arXiv preprint arXiv:2309.16609**, 2023.

BAI, X. et al. **Explicitly unbiased large language models still form biased associations.** **Proceedings of the National Academy of Sciences**, v. 122, n. 8, p. e2416228122, fev. 2025.

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>>

BARRETT, A. M. et al. **Benchmark Early and Red Team Often: A framework for assessing and managing dual-hazards of AI foundational models.** UC Berkeley Center for Long-Term Cybersecurity, 2024. Disponível em: <<https://cltc.berkeley.edu/wp-content/uploads/2024/05/Dual-Use-Benchmark-Early-Red-Team-Often.pdf>>

BARRY LYNN, M. VON T.; MONTOYA, K. **AI in the Public Interest: Confronting the Monopoly Threat.** Open Markets Institute, 2023. Disponível em: <<https://www.openmarketsinstitute.org/publications/report-ai-in-the-public-interest-confronting-the-monopoly-threat>>

BELINKOV, Y.; GEHRMANN, S.; PAVLICK, E. **Interpretability and Analysis in Neural NLP.** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts. **Anais...** Online: Association for Computational Linguistics, 2020. Disponível em: <<https://www.aclweb.org/anthology/2020.acl-tutorials.1>>

BELINKOV, Y.; GLASS, J. **Analysis Methods in Neural Language Processing: A Survey.** **Transactions of the Association for Computational Linguistics**, v. 7, p. 49–72, 2019.

BELTAGY, I.; PETERS, M. E.; COHAN, A. **Longformer: The Long-Document Transformer.** **CoRR**, v. abs/2004.05150, 2020.

BELZ, A. et al. **A Systematic Review of Reproducibility Research in Natural Language Processing.** Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. **Anais...** Online: Association for Computational Linguistics, abr. 2021. Disponível em: <<https://aclanthology.org/2021.eacl-main.29>>

BENCKE, L. et al. **Can we trust LLMs as relevance judges?** **Anais do XXXIX Simpósio Brasileiro de Bancos de Dados.** **Anais...** Porto Alegre, RS, Brasil: SBC, 2024. Disponível em: <<https://sol.sbc.org.br/index.php/sbbd/article/view/30724>>

BENDER, E. M. **On achieving and evaluating language-independence in NLP.** **Linguistic Issues in Language Technology**, v. 6, 2011.

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. **Resisting Dehumanization in the Age of “AI”**. **Current Directions in Psychological Science**, v. 33, n. 2, p. 114–120, abr. 2024.

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.

BENDER, E. M.; HANNA, A. **The AI Con: How to fight big tech’s hype and create the future we want**. [s.l.] Random House, 2025.

BENDER, E. M.; KOLLER, A. **Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data**. 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.463>>

BENGIO, Y. et al. A Neural Probabilistic Language Model. **J. Mach. Learn. Res.**, v. 3, n. null, p. 1137–1155, mar. 2003.

BENOTTI, L.; BLACKBURN, P. **Ethics consideration sections in natural language processing papers**. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. **Anais...Abu Dhabi, United Arab Emirates**: Association for Computational Linguistics, 2022. Disponível em: <<https://aclanthology.org/2022.emnlp-main.299>>

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.

BIANCHI, F.; HOVY, D. **On the Gap between Adoption and Understanding in NLP**. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. **Anais...Online**: Association for Computational Linguistics, ago. 2021. Disponível em: <<https://aclanthology.org/2021.findings-acl.340>>

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>>

BIBER, D. **Variation across Speech and Writing**. Cambridge: Cambridge University Press, 1988.

BIBER, D.; CONRAD, S. **Register, Genre, and Style**. Cambridge: Cambridge University Press, 2009.

BIBER, D.; FINEGAN, E. **Sociolinguistic Perspectives On Register**. [s.l.] Oxford University Press, 1994.

BINZ, M. et al. **How should the advancement of large language models affect the practice of science?** **Proceedings of the National Academy of Sciences**, v. 122, n. 5, p. e2401227121, fev. 2025.

BIRD, S. **Decolonising Speech and Language Technology**. Proceedings of the 28th International Conference on Computational Linguistics. **Anais...Barcelona, Spain (Online)**: International Committee on Computational Linguistics, dez. 2020. Disponível em: <<https://aclanthology.org/2020.coling-main.313>>

BIRD, S. **Local Languages, Third Spaces, and other High-Resource Scenarios**. 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>>



anthology.org/2022.acl-long.539>

BLODGETT, S. L. et al. **Language (Technology) is Power: A Critical Survey of “Bias” in NLP**. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. *Anais...Online*: Association for Computational Linguistics, a2020. Disponível em: <<https://www.aclweb.org/anthology/2020.acl-main.485>>

BLODGETT, S. L. et al. **Language (Technology) is Power: A Critical Survey of “Bias” in NLP**. (D. Jurafsky et al., Eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. *Anais...Online*: Association for Computational Linguistics, jul. b2020. Disponível em: <<https://aclanthology.org/2020.acl-main.485/>>

BLOM, J. D. **A dictionary of hallucinations**. [s.l.] Springer, 2010.

BOSTROM, N. **Superintelligence: Paths, Dangers, Strategies**. Oxford, UK: Oxford University Press, 2014.

BOWMAN, S. R.; DAHL, G. **What Will it Take to Fix Benchmarking in Natural Language Understanding?** (K. Toutanova et al., Eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. *Anais...Online*: Association for Computational Linguistics, jun. 2021. Disponível em: <<https://aclanthology.org/2021.naacl-main.385/>>

BRANDES, N. et al. **ProteinBERT: a universal deep-learning model of protein sequence and function**. *Bioinform.*, v. 38, n. 8, p. 2102–2110, 2022.

BRENNAN, K.; KAK, A.; WEST, S. M. **The AGI Mythology: The Argument to End All Arguments**. Em: **Artificial Power: 2025 Landscape Report**. [s.l.] AI Now Institute, 2025.

BRENNEIS, A. **Assessing dual use risks in AI research: necessity, challenges and mitigation strategies**. *Research Ethics*, v. 21, n. 2, p. 302–330, 2025.

BROUSSARD, M. **More Than a Glitch: Confronting Race, Gender, and Ability Bias in Tech**. 1. ed. Cambridge, Massachusetts: MIT Press, 2023.

BROWN, T. B. et al. **Language Models are Few-Shot Learners**. (H. Larochelle et al., Eds.) Advances in Neural Information Processing Systems. *Anais...Curran Associates, Inc.*, 2020. Disponível em: <<https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-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.

BURCHELL, L. et al. **An Expanded Massive Multilingual Dataset for High-Performance Language Technologies (HPLT)**. (W. Che et al., Eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). *Anais...Vienna, Austria*: Association for Computational Linguistics, jul. 2025. Disponível em: <<https://aclanthology.org/2025.acl-long.854/>>

CARLSMITH, J. **Is Power-Seeking AI an Existential Risk?** [s.l.] Open Philanthropy, 2022.

CARMO, D. et al. **PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data**. *CoRR*, v. abs/2008.09144, 2020.

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>>



CENTRE FOR INFORMATION POLICY LEADERSHIP, C. **The Limitations of Consent as a Legal Basis for Data Processing in the Digital Society**. Washington DC, London, BrusselsCentre for Information Policy Leadership & Hunton Andrews Kurth LLP; Bae Kim & Lee, 2024. Disponível em: <https://www.informationpolicycentre.com/uploads/5/7/1/0/57104281/cipl_bkl_limitations_of_consent_legal_basis_data_processing_dec24.pdf>

CERVANTES, M. DE. **D. Quixote de La Mancha — Primeira Parte**. [s.l.] eBooksBrasil, 1605.

CHALAMALASETTI, K. et al. **clmbench: Using Game Play to Evaluate Chat-Optimized Language Models as Conversational Agents**. (H. Bouamor, J. Pino, K. Bali, Eds.)Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. **Anais...**Singapore: Association for Computational Linguistics, dez. 2023. Disponível em: <<https://aclanthology.org/2023.emnlp-main.689>>

CHANDRA, M. et al. **From Lived Experience to Insight: Unpacking the Psychological Risks of Using AI Conversational Agents**. Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency. **Anais...**Athens Greece: ACM, jun. 2025. Disponível em: <<https://dl.acm.org/doi/10.1145/3715275.3732063>>

CHEN, G. H. et al. **Humans or LLMs as the Judge? A Study on Judgement Bias**. **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, a2024.

CHEN, K. et al. **How Susceptible are Large Language Models to Ideological Manipulation?** Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. **Anais...**Miami, Florida, USA: Association for Computational Linguistics, b2024. Disponível em: <<https://aclanthology.org/2024.emnlp-main.952>>

CHEN, T. et al. **CopyBench: Measuring Literal and Non-Literal Reproduction of Copyright-Protected Text in Language Model Generation**. (Y. Al-Onaizan, M. Bansal, Y.-N. Chen, Eds.)Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. **Anais...**Miami, Florida, USA: Association for Computational Linguistics, nov. c2024. Disponível em: <<https://aclanthology.org/2024.emnlp-main.844/>>

CHENG, J. et al. **Dated Data: Tracing Knowledge Cutoffs in Large Language Models.**, 2024. Disponível em: <<https://arxiv.org/abs/2403.12958>>

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>>

CHOMSKY, N. **Aspects of the Theory of Syntax**. Cambridge, MA: MIT Press, 1965.

CHOWDHURY, A. et al. **PaLM: Scaling Language Modeling with Pathways**. **CoRR**, v. abs/2204.02311, 2022.

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>>

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>>



- COHEN, K. B. et al. **Three Dimensions of Reproducibility in Natural Language Processing**. 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/L18-1025>>
- 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>>
- COMMON CRAWL. **Common Crawl – Open Repository of Web Crawl Data**. <https://commoncrawl.org>, [s.d.].
- 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.
- CORMACK, G. V.; CLARKE, C. L.; BUETTCHER, S. **Reciprocal rank fusion outperforms condorcet and individual rank learning methods**. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. *Anais...*2009.
- CORMEN, T. et al. Introduction to Algorithms. Em: 2. ed. [s.l.] MIT Press; McGraw-Hill, 2001.
- CORRÊA, N. K. et al. **Tucano: Advancing Neural Text Generation for Portuguese**. *Patterns*, 2025.
- CORTES, C.; VAPNIK, V. **Support-Vector Networks**. *Machine Learning*, 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>>
- COULDRY, N.; MEJIAS, U. A. **The Costs of Connection: How Data Is Colonizing Human Life and Appropriating It for Capitalism**. [s.l.] Stanford University Press, 2019.
- CRESPO, M. C. R. M. et al. **Carolina: a General Corpus of Contemporary Brazilian Portuguese with Provenance, Typology and Versioning Information**., 2023. Disponível em: <<https://arxiv.org/abs/2303.16098>>
- CRONBACH, L. J. **Studies of acquiescence as a factor in the true-false test**. *Journal of Educational Psychology*, v. 33, p. 401–415, 1942.
- CUEVA, R. et al. **Inteligência Artificial e Desafios Regulatórios**. [s.l.] Forense, 2026.
- CURRY, A.; CERCAS CURRY, A. **Computer says “No”: The Case Against Empathetic Conversational AI**. (A. Rogers, J. Boyd-Graber, N. Okazaki, Eds.)Findings of the Association for Computational Linguistics: ACL 2023. *Anais...*Toronto, Canada: Association for Computational Linguistics, jul. 2023. Disponível em: <<https://aclanthology.org/2023.findings-acl.515/>>
- D’ARCY, A.; BENDER, E. M. **Ethics in Linguistics**. *Annual Review of Linguistics*, v. 9, n. 1, p. 49–69, jan. 2023.
- 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>>



- DARCIO, L. et al. **LattesRex: Building ChatBots for Semi-Structured Documents**. Proceedings of the 14th Brazilian Symposium in Information and Human Language Technology (STIL 2025). **Anais...Brazil**: 2025. Disponível em: <<https://github.com/Lucas-Darcio/LattesRex>>
- DAVIES, M.; FERREIRA, M. J. **Corpus do Português: Web/Dialects**. <https://www.corpusdoportugues.org/web-dial/>, 2016.
- DAVIES, M.; FERREIRA, M. J. **Corpus do Português: NOW**. <https://www.corpusdoportugues.org/now/>, 2018.
- 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>>
- DIERK, C.; HEALEY, J.; DOGAN, M. D. **Evaluating LLMs in Experiential Context: Insights from a Survey of Recent CHI Publications**. Human-centered Evaluation and Auditing of Language Models Workshop (HEAL), CHI '25. **Anais...Yokohama, Japan**: ACM, 2025. Disponível em: <https://heal-workshop.github.io/chi2025_papers/43_Evaluating_LLMs_in_Experien.pdf>
- DING, N. et al. **Parameter-efficient fine-tuning of large-scale pre-trained language models**. **Nature machine intelligence**, v. 5, n. 3, p. 220–235, 2023.
- DONG, Q. et al. **A Survey for In-context Learning**. **CoRR**, v. abs/2301.00234, 2023.
- DOU, G. et al. **Avoiding Copyright Infringement via Large Language Model Unlearning**. (L. Chiruzzo, A. Ritter, L. Wang, Eds.) Findings of the Association for Computational Linguistics: NAACL 2025. **Anais...Albuquerque, New Mexico**: Association for Computational Linguistics, abr. 2025. Disponível em: <<https://aclanthology.org/2025.findings-naacl.288/>>
- EDITORIAL. **Emotional risks of AI companions demand attention**. **Nature Machine Intelligence**, v. 7, n. 7, p. 981–982, jul. 2025.
- ES, S. et al. **Ragas: Automated evaluation of retrieval augmented generation**. Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. **Anais...2024**.
- ETHAYARAJH, K.; JURAFSKY, D. **Utility is in the Eye of the User: A Critique of NLP Leaderboards**. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). **Anais...Online**: Association for Computational Linguistics, 2020. Disponível em: <<https://www.aclweb.org/anthology/2020.emnlp-main.393>>
- 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>>
- FÄRBER, F. et al. **IaraMed: A Women's Healthcare Chatbot for Portuguese Speakers**. **Anais do XXV Simpósio Brasileiro de Computação Aplicada à Saúde**. **Anais...Porto Alegre, RS, Brasil**: SBC, 2025. Disponível em: <<https://sol.sbc.org.br/index.php/sbcas/article/view/35558>>
- 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.
- FINGER, M. et al. **Carolina: General Corpus of Contemporary Brazilian Portuguese with**



Provenance and Typology Information., 2020. Disponível em: <<https://c4ai.inova.usp.br/portal/carolina>>

FIRTH, J. R. **A synopsis of linguistic theory 1930–1955**. [s.l.] Blackwell, 1957. p. 1–32

FISHER, J. et al. **Biased LLMs can Influence Political Decision-Making**. (W. Che et al., Eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Vienna, Austria: Association for Computational Linguistics, jul. 2025. Disponível em: <<https://aclanthology.org/2025.acl-long.328/>>

FLORIDI, L. **Translating Principles into Practices of Digital Ethics: Five Risks of Being Unethical**. **Philosophy & Technology**, v. 32, n. 2, p. 185–193, jun. 2019.

FOKKENS, A. et al. **Offspring from Reproduction Problems: What Replication Failure Teaches Us**. 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/P13-1166>>

FORT, K.; ADDA, G.; COHEN, K. B. **Amazon Mechanical Turk: Gold Mine or Coal Mine? Computational Linguistics**, v. 37, n. 2, p. 413–420, jun. 2011.

FRANCISCO, P. A. P.; HUREL, L. M.; RIELLI, M. M. **Regulação do Reconhecimento Facial no Setor Público: Avaliação de Experiências Internacionais**. Instituto Igarapé - DataPrivacyBR, 2020. Disponível em: <<https://www.dataprivacybr.org/wp-content/uploads/2021/11/regulacao-do-reconhecimento-facial-no-setor-publico.pdf>>

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.

GABRIEL, I. **Artificial Intelligence, Values and Alignment**. **Minds and Machines**, v. 30, n. 3, p. 411–437, 2020.

GAJULAMANDYAM, D. K. et al. **Domain Specific Finetuning of LLMs Using PEFT Techniques**. 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC). **Anais...**2025.

GALILEO AI. **Generative AI and LLM Insights**. <https://galileo.ai/blog/>, 2024.

GAO, M. et al. **LLM-based NLG Evaluation: Current Status and Challenges**. ArXiv, 2024. Disponível em: <<https://arxiv.org/abs/2402.01383>>

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, E. A. S. et al. **RoBERTaLexPT: A Legal RoBERTa Model pretrained with deduplication for Portuguese**. (P. Gamallo et al., Eds.) Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1. **Anais...**Santiago de Compostela, Galicia/Spain: Association for Computational Linguistics, mar. 2024. Disponível em: <<https://aclanthology.org/2024.propor-1.38>>

GATT, A.; KRAHMER, E. **Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation**. **Journal of Artificial Intelligence Research**, v. 61, n. 1, p. 65–170, 2018.

GEBRU, T. et al. **Datasheets for datasets**. **Communications of the ACM**, v. 64, n. 12, p. 86–92, dez. 2021.

GEBRU, T.; BENDER, E. M.; MCMILLAN-MAJOR, A. **Statement from the listed authors of**



Stochastic Parrots on the “AI pause” letter., 2023. Disponível em: <<https://www.dair-institute.org/blog/letter-statement-March2023/>>

GEBRU, T.; TORRES, E. P. **The TESCREAL bundle: Eugenics and the promise of utopia through artificial general intelligence.** *First Monday*, abr. 2024.

GEHMAN, S. et al. **RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models.** (T. Cohn, Y. He, Y. Liu, Eds.) 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.301/>>

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/10.18653/v1/2020.acl-main.89>>

GOLUMBIA, D. **Cyberlibertarianism: The Right-Wing Politics of Digital Technology.** [s.l.] University of Minnesota Press, 2024.

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>>

GONZALEZ-CABELLO, M. et al. **Fairness in crowdwork: Making the human AI supply chain more humane.** *Business Horizons*, v. 68, n. 5, p. 645–657, 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning.** [s.l.] MIT Press, 2016. v. 1

GOOGLE CLOUD. **Best Practices with Large Language Models.** <https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompt-best-practices?hl=en>, 2025.

GRAY, M. L.; SURI, S. **Ghost work: How to stop Silicon Valley from building a new global underclass.** [s.l.] Harper Business, 2019.

GRAY WIDDER, D.; WEST, S.; WHITTAKER, M. **Open (For Business): Big Tech, Concentrated Power, and the Political Economy of Open AI.** *SSRN Electronic Journal*, 2023.

GU, Y. et al. **Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing.** *ACM Trans. Comput. Healthcare*, v. 3, n. 1, out. 2021.

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>>

HADDAD, I. **Artificial Intelligence and Data in Open Source.** Linux Foundation, 2022. Disponível em: <<https://www.linuxfoundation.org/hubfs/LF%20Research/Artificial%20Intelligence%20and%20Data%20in%20Open%20Source%20-%20Report.pdf?hsLang=en>>

HANNIGAN, T. R.; MCCARTHY, I. P.; SPICER, A. **Beware of botshit: How to manage the epistemic risks of generative chatbots.** *Business Horizons*, v. 67, n. 5, p. 471–486, 2024.

HAVENS, L. et al. **Situated Data, Situated Systems: A Methodology to Engage with Power Relations in Natural Language Processing Research.** (M. R. Costa-jussà et al., Eds.) Proceedings of the Second Workshop on Gender Bias in Natural Language Processing. *Anais...* Barcelona, Spain (Online): Association for Computational Linguistics, dez. 2020. Disponível em: <<https://aclanthology.org/2020.genbnlp-1.10/>>



- 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>>
- HEDDERICH, M. A.; OULASVIRTA, A. **Explaining crowdworker behaviour through computational rationality**. **Behaviour & Information Technology**, v. 44, n. 3, p. 552–573, 2025.
- HELLER, J. **Legal AI benchmarking: CoCounsel – from code to courtroom: The meticulous testing of CoCounsel’s professional-grade AI.**, 23 out. 2024. Disponível em: <<https://www.thomsonreuters.com/en-us/posts/innovation/legal-ai-benchmarking-cocounsel/>>. Acesso em: 20 ago. 2025
- HENDREN, S. **All technology is assistive: Six design rules on disability**. Em: SAYERS, J. (Ed.). **Making Things and Drawing Boundaries: Experiments in the Digital Humanities**. [s.l.] University of Minnesota Press, Minneapolis, MN, 2014.
- HENDRYCKS, D. et al. **Unsolved Problems in ML Safety**. **arXiv preprint arXiv:2109.13916**, 2021.
- HERSHCOVICH, D. et al. **Towards Climate Awareness in NLP Research**. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. **Anais...Abu Dhabi, United Arab Emirates: Association for Computational Linguistics**, 2022. Disponível em: <<https://aclanthology.org/2022.emnlp-main.159>>
- HICKS, M. T.; HUMPHRIES, J.; SLATER, J. **ChatGPT is bullshit**. **Ethics and Information Technology**, v. 26, n. 2, p. 38, jun. 2024.
- 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, V. et al. **AI generates covertly racist decisions about people based on their dialect**. **Nature**, v. 633, n. 8028, p. 147–154, set. 2024.
- 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>>
- HOOKE, S. **Moving beyond “algorithmic bias is a data problem”**. **Patterns**, v. 2, n. 4, p. 100241, abr. 2021.
- HORNIK, K.; STINCHCOMBE, M. B.; WHITE, H. **Multilayer feedforward networks are universal approximators**. **Neural Networks**, v. 2, n. 5, p. 359–366, 1989.
- HOUAMEGNI, L. R. P.; GEDIKLI, F. **Evaluating the Effectiveness of Large Language Models in Automated News Article Summarization.**, 2025. Disponível em: <<https://arxiv.org/abs/2502.17136>>
- 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, D.; SPRUIT, S. L. **The Social Impact of Natural Language Processing**. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). **Anais...**Berlin, Germany: Association for Computational Linguistics, 2016. Disponível em: <<http://aclweb.org/anthology/P16-2096>>

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. a2018. Disponível em: <5>

HOWARD, J.; RUDER, S. **Universal Language Model Fine-tuning for Text Classification**. (I. Gurevych, Y. Miyao, Eds.)Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). **Anais...**Melbourne, Australia: Association for Computational Linguistics, jul. b2018. Disponível em: <<https://aclanthology.org/P18-1031/>>. Acesso em: 25 jul. 2025

HU, E. J. et al. **LoRA: Low-Rank Adaptation of Large Language Models.**, 2021. Disponível em: <<https://arxiv.org/abs/2106.09685>>

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>>

HUANG, J.; YANG, D.; POTTS, C. **Demystifying Verbatim Memorization in Large Language Models**. (Y. Al-Onaizan, M. Bansal, Y.-N. Chen, Eds.)Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. **Anais...**Miami, Florida, USA: Association for Computational Linguistics, nov. a2024. Disponível em: <<https://aclanthology.org/2024.emnlp-main.598/>>

HUANG, L. et al. **A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions**. **ACM Transactions on Information Systems**, v. 43, n. 2, p. 1–58, b2024.

II, S. M. W. **GSM8K Benchmark**. Klu; <https://klu.ai/glossary/GSM8K-eval>, 2025.

IP, J.; VONGTHONGSRI, K. **deepeval.**, ago. 2025. Disponível em: <<https://github.com/confident-ai/deepeval>>

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.

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>>

JAKUBČEK, M. et al. **The TenTen corpus family**. Proceedings of the 7th International Corpus Linguistics Conference (CL). **Anais...**2013.

JAN, E. et al. **Multitask-Bench: Unveiling and Mitigating Safety Gaps in LLMs Fine-tuning**. (O. Rambow et al., Eds.)Proceedings of the 31st International Conference on Computational Linguistics. **Anais...**Abu Dhabi, UAE: Association for Computational Linguistics, jan. 2025. Disponível em: <<https://aclanthology.org/2025.coling-main.606/>>

JAPKOWICZ, N.; BOUKOUVALAS, Z. **Machine Learning Evaluation: Towards Reliable and Responsible AI**. [s.l.] Cambridge University Press, 2024.

JENSEN, K. N.; PLANK, B. **Fine-tuning vs From Scratch: Do Vision & Language Models Have Similar Capabilities on Out-of-Distribution Visual Question Answering?** (N. Calzolari et al.,



Eds.)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-1.161/>>

JEONG, S. et al. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. **arXiv preprint arXiv:2403.14403**, 2024.

Ji, Z. et al. **Survey of Hallucination in Natural Language Generation**. **ACM Comput. Surv.**, v. 55, n. 12, mar. 2023.

JIANG, L.; WAGNER, C. **How Low is Low? Crowdsworker Perceptions of Microtask Payments in Work versus Leisure Situations**. Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. **Anais...**: CHI '24.New York, NY, USA: Association for Computing Machinery, 2024. Disponível em: <<https://doi.org/10.1145/3613904.3642601>>

JIAO, J. et al. **Navigating LLM Ethics: Advancements, Challenges, and Future Directions**. **ArXiv**, v. abs/2406.18841, 2024.

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>>

JIN, Z. et al. **How Good Is NLP? A Sober Look at NLP Tasks through the Lens of Social Impact**. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. **Anais...**Online: Association for Computational Linguistics, 2021. Disponível em: <<https://aclanthology.org/2021.findings-acl.273>>

JOHNSTON, S. F. **Alvin Weinberg and the Promotion of the Technological Fix**. **Technology and Culture**, v. 59, n. 3, p. 620–651, 2018.

JOSHI, M. et al. **TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension**. (R. Barzilay, M.-Y. Kan, Eds.)Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. **Anais...**Association for Computational Linguistics, 2017. Disponível em: <<https://doi.org/10.18653/v1/P17-1147>>

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.

JOSHI, N.; TALY, A.; MUPPALLA, D. **LLM-Cite: Cheap Fact Verification with Attribution via URL Generation**., 2025. Disponível em: <<https://openreview.net/forum?id=qb2QRoE4W3>>

JOULIN, A. et al. **Bag of Tricks for Efficient Text Classification**. (M. Lapata, P. Blunsom, A. Koller, Eds.)Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. **Anais...**Valencia, Spain: Association for Computational Linguistics, abr. 2017. Disponível em: <<https://aclanthology.org/E17-2068/>>

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.

KALAI, A. T. et al. **Why language models hallucinate**. [s.l.] OpenAI, 2025.

KALOULI, A.-L. et al. **Curing the SICK and Other NLI Maladies**. **Computational Linguistics**, v. 49,



n. 1, p. 199–243, mar. 2023.

KAMRUZZAMAN, M.; SHOYON, MD.; KIM, G. **Investigating Subtler Biases in LLMs: Ageism, Beauty, Institutional, and Nationality Bias in Generative Models.** (L.-W. Ku, A. Martins, V. Srikumar, Eds.) Findings of the Association for Computational Linguistics: ACL 2024. **Anais...** Bangkok, Thailand: Association for Computational Linguistics, ago. 2024. Disponível em: <<https://aclanthology.org/2024.findings-acl.530/>>

KARAMOLEGKOU, A. et al. **Copyright Violations and Large Language Models.** Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. **Anais...** Singapore: Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.emnlp-main.458>>

KE, Z. et al. **Continual Pre-training of Language Models.**, 2023. Disponível em: <<https://arxiv.org/abs/2302.03241>>

KERNER, T. **Domain-Specific Pretraining of Language Models: A Comparative Study in the Medical Field.**, 2024. Disponível em: <<https://arxiv.org/abs/2407.14076>>

KHATTAB, O.; ZAHARIA, M. **ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.**, 2020. Disponível em: <<https://arxiv.org/abs/2004.12832>>

KIANPOUR, M.; WEN, S.-F. **Timing Attacks on Machine Learning: State of the Art.** Intelligent Systems Conference. **Anais...** Springer, 2020.

KIM, D. K. et al. **Analyzing Offensive Language Dataset Insights from Training Dynamics and Human Agreement Level.** (O. Rambow et al., Eds.) Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025. **Anais...** Association for Computational Linguistics, 2025. Disponível em: <<https://aclanthology.org/2025.coling-main.653/>>

KIM, S. et al. **Prometheus: Inducing Fine-grained Evaluation Capability in Language Models.** **arXiv preprint arXiv:2310.08491**, 2023.

KLENK, M. **How Do Technological Artefacts Embody Moral Values?** **Philosophy & Technology**, v. 34, n. 3, p. 525–544, set. 2021.

KNUTH, D. E. **Fundamental Algorithms. The Art of Computer Programming.** 3. ed. [s.l.] Addison-Wesley, 1997. v. 1

KOGKALIDIS, K.; CHATZIKYRIAKIDIS, S. **On Tables with Numbers, with Numbers.** (S. Truong et al., Eds.) Proceedings of the 1st Workshop on Language Models for Underserved Communities (LM4UC 2025). **Anais...** Albuquerque, New Mexico: Association for Computational Linguistics, 2025. Disponível em: <<https://aclanthology.org/2025.lm4uc-1.12/>>

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>

KRAKOVNA, V. et al. **Specification Gaming: The Flip Side of AI Ingenuity.** **DeepMind Safety Research Blog**, 2020.

KREUTZER, J. et al. **Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets.** **Transactions of the Association for Computational Linguistics**, v. 10, p. 50–72, jan. 2022.

KWIATKOWSKI, T. et al. **Natural Questions: a Benchmark for Question Answering Research.** **Transactions of the Association of Computational Linguistics**, 2019.

LAI, J. et al. **Large language models in law: A survey.** **AI Open**, v. 5, p. 181–196, 2024.



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>>

LASOTA, L. **Regulating Corporate Behaviour in Digital Ecosystems: Increasing Fairness and Contestability of Digital Markets with Free Software**. MIC 2023: Toward Green, Inclusive, and Digital Growth. *Anais...*a2023.

LASOTA, L. **The European Union's AI act from the perspective of Open Technologies**. Em: ALEGRE, M.; FÜRST, H. (Eds.). **Advocacia & Bioética: Novas Tecnologias**. <https://www.editoracasadodireito.com.br/produto/novas-tecnologias>. São Paulo: Casa do Direito, 2023b.

LASOTA, L. **The CRA and the Challenges of Regulating Cybersecurity in Open Environments: The Case of Free and Open Source Software**. Em: **Digital Decade: How the EU shapes digitalisation**. Berlin: Weizenbaum Institute for the Networked Society - The German Internet Institute, 2025.

LASOTA, L.; SINGHAL, N. **Free Software and AI openness: Overcoming challenges in the licensing world**. [s.l.] Zenodo, abr. 2024. Disponível em: <<https://zenodo.org/doi/10.5281/zenodo.10966136>>.

LAZZARINI MORETTI, J.; MALTESE ZUFFO, M. **LGPD e inteligência artificial: Um estudo comparado**. *Revista de Direito Internacional e Globalização Econômica*, v. 13, n. 13, p. 21-42, 2025.

LEE, J. et al. A survey of large language models in finance (finllms). **arXiv preprint arXiv:2402.02315**, 2024.

LEE, K. et al. **Deduplicating Training Data Makes Language Models Better.**, 2022. Disponível em: <<https://arxiv.org/abs/2107.06499>>

LEIDNER, J. L.; PLACHOURAS, V. **Ethical by Design: Ethics Best Practices for Natural Language Processing**. Proceedings of the First ACL Workshop on Ethics in Natural Language Processing. *Anais...*Valencia, Spain: Association for Computational Linguistics, abr. 2017. Disponível em: <<https://aclanthology.org/W17-1604>>

LEIKE, J. et al. **Scalable Agent Alignment via Reward Modeling: a Research Direction**. arXiv preprint arXiv:1811.07871. *Anais...*2018.

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>>

LI, D. et al. **From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge**. **CoRR**, v. abs/2411.16594, 2024.



- LI, H. et al. **CaseGen: A benchmark for multi-stage legal case documents generation.**, 2025. Disponível em: <<https://arxiv.org/abs/2502.17943>>
- 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. **Parameter Efficient Fine-Tuning on Selective Parameters for Transformer-Based Pre-Trained Models.** 2024 IEEE International Conference on Multimedia and Expo (ICME). **Anais...**2024.
- 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>>
- LI, Y. et al. **A comparative study of pretrained language models for long clinical text.** **Journal of the American Medical Informatics Association**, v. 30, n. 2, p. 340–347, nov. 2022.
- 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>>
- 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, S.; HILTON, J.; EVANS, O. **TruthfulQA: Measuring How Models Mimic Human Falsehoods.** (S. Muresan, P. Nakov, A. Villavicencio, Eds.)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.229/>>
- LIRA, T. et al. **Aroeira: A Curated Corpus for the Portuguese Language with a Large Number of Tokens.** Anais da XXXIV Brazilian Conference on Intelligent Systems. **Anais...**Porto Alegre, RS, Brasil: SBC, 2024. Disponível em: <<https://sol.sbc.org.br/index.php/bracis/article/view/33560>>
- LITA, A.-C.; COSTEA, C.-E.; IONESCU, B. **Chunking for RAG: A Comprehensive Evaluation of Chunking Strategies for Retrieval-Augmented Generation.** Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. **Anais...**2024.
- LIU, Y. et al. **RoBERTa: A Robustly Optimized BERT Pretraining Approach.**, a2019. Disponível em: <<https://arxiv.org/abs/1907.11692>>
- LIU, Y. et al. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, b2019.
- LIU, Y. et al. **Multilingual Denoising Pre-training for Neural Machine Translation.** **Trans. Assoc. Comput. Linguistics**, v. 8, p. 726–742, 2020.
- LIU, Y. et al. **HD-Eval: Aligning Large Language Model Evaluators Through Hierarchical Criteria Decomposition.** (L.-W. Ku, A. Martins, V. Srikumar, Eds.)Proceedings of the 62nd Annual



Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024. **Anais...**Association for Computational Linguistics, 2024. Disponível em: <<https://doi.org/10.18653/v1/2024.acl-long.413>>

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>

LOBO, P. **Profiling na Lei Geral de Proteção de Dados: O Livre Desenvolvimento da Personalidade em Face da Governamentalidade Algorítmica**. 1. ed. [s.l.] Editora Foco, 2022.

LONGPRE, S. et al. **A Pretrainer's Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity**. (K. Duh, H. Gomez, S. Bethard, Eds.)Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). **Anais...**Mexico City, Mexico: Association for Computational Linguistics, jun. 2024. Disponível em: <<https://aclanthology.org/2024.naacl-long.179/>>

LONGPRE, S. et al. **The Responsible Foundation Model Development Cheatsheet: A Review of Tools & Resources**., 2025. Disponível em: <<https://arxiv.org/abs/2406.16746>>

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>>

LV, K. et al. **Full Parameter Fine-tuning for Large Language Models with Limited Resources**., 2024. Disponível em: <<https://arxiv.org/abs/2306.09782>>

MALEKI, N.; PADMANABHAN, B.; DUTTA, K. **AI Hallucinations: A Misnomer Worth Clarifying**. 2024 IEEE Conference on Artificial Intelligence (CAI). **Anais...**Singapore, Singapore: IEEE, jun. 2024. Disponível em: <<https://ieeexplore.ieee.org/document/10605268/>>

MARKOV, A. A. The theory of algorithms. **Trudy Matematicheskogo Instituta Imeni VA Steklova**, v. 42, p. 3–375, 1954.

MATHUR, N.; BALDWIN, T.; COHN, T. **Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics**. Proceedings of the Conference on Empirical Methods in Natural Language Processing. **Anais...**Association for Computational Linguistics, 2020. Disponível em: <<https://aclanthology.org/2020.acl-main.448/>>

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.

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.

MCMILLAN-MAJOR, A.; BENDER, E. M.; FRIEDMAN, B. **Data Statements: From Technical Concept to Community Practice**. **ACM Journal on Responsible Computing**, p. 3594737, 2023.

MCQUILLAN, D. **Resisting AI: an anti-fascist approach to artificial intelligence**. Bristol, UK: [s.n.].

MECKLENBURG, N. et al. **Injecting New Knowledge into Large Language Models via Supervised Fine-Tuning**., 2024. Disponível em: <<https://arxiv.org/abs/2404.00213>>

MEJIAS, U. A.; COULDRY, N. **Datafication**. **Internet Policy Review**, v. 8, n. 4, nov. 2019.



- MEVA, DR. D.; KUKADIYA, H. *Performance Evaluation of Large Language Models: A Comprehensive Review*. **International Research Journal of Computer Science**, v. 12, p. 109–114, mar. 2025.
- MICELI, M. et al. **Who Trains the Data for European Artificial Intelligence?** The Left, DiPLab, Weizenbaum Institute; DAIR Institute., 2024. Disponível em: <<https://hal.science/hal-04662589v1>>
- 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.
- MILLER, B. *Is Technology Value-Neutral?* **Science, Technology, & Human Values**, v. 46, n. 1, p. 53–80, jan. 2021.
- MINAEE, S. et al. Large language models: A survey. **arXiv preprint arXiv:2402.06196**, 2024.
- MINAEE, S. et al. **Large Language Models: A Survey.**, 2025. Disponível em: <<https://arxiv.org/abs/2402.06196>>
- MITCHELL, M. et al. **Model Cards for Model Reporting**. Proceedings of the Conference on Fairness, Accountability, and Transparency. **Anais...Atlanta GA USA: ACM**, jan. 2019. Disponível em: <<https://dl.acm.org/doi/10.1145/3287560.3287596>>
- MIZRAHI, D. et al. **Language Models Improve When Pretraining Data Matches Target Tasks.**, 2025. Disponível em: <<https://arxiv.org/abs/2507.12466>>
- MOHAMMAD, S. **Ethics Sheets for AI Tasks**. 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.573>>
- MOREIRA, J.; VIANNA, D. **Challenges of Generative AI in Legal Practice: A Jusbrasil Perspective**. Proceedings of the 40th Brazilian Symposium on Databases (SBBD 2025). **Anais...Brazil: Brazilian Computer Society (SBC)**, 2025. Disponível em: <<https://sbbd.org.br/2025/wp-content/uploads/2025/10/SBBD25-Jusbrasil.pdf>>
- MORESCHI, B.; PEREIRA, G.; COZMAN, F. G. *The Brazilian Workers in Amazon Mechanical Turk: Dreams and realities of ghost workers*. **Revista Contracampo**, v. 39, n. 1, abr. 2020.
- MOROZOV, E. **To save everything, click here : the folly of technological solutionism**. New York, NY: PublicAffairs, 2014.
- MUNGER, K. **Chatbots for Good and Evil**. EACL via Underline Science Inc., 2023. Disponível em: <<https://underline.io/lecture/72154-chatbots-for-good-and-evil>>
- NAIR, P. A. et al. **Matryoshka Quantization**. Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference. **Anais...2025**.
- NARAYAN, S.; COHEN, S. B.; LAPATA, M. **Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization**. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. **Anais...Brussels, Belgium: 2018**.
- NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **A Plan for Global Engagement on AI Standards**. [s.l.] U.S. Department of Commerce, National Institute of Standards; Technology, jul. 2024. Disponível em: <<https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-5.pdf>>.
- NEPOMUCENO, J. et al. **Agentic AI Meets Fundamental Financial Analysis: Limits and Possibilities**. Proceedings of the 14th Brazilian Conference on Intelligent Systems (BRACIS 2025). **Anais...2025**.
- NGO, R.; CHAN, L.; MINDERMAN, S. The Alignment Problem from a Deep Learning Perspective. **arXiv preprint arXiv:2303.16200**, 2023.



- NIJKAMP, E. et al. **ProGen2: Exploring the Boundaries of Protein Language Models**. **CoRR**, v. abs/2206.13517, 2022.
- NOVOBILSKÁ, L. **Free and Open Source Software Licensing Requirements and Copyright Infringement Involving Artificial Intelligence Technologies**. 2023.
- O'NEIL, C. **Weapons of math destruction : how big data increases inequality and threatens democracy**. First edition ed. New York: [s.n.].
- OLIVEIRA, L. L. **Inteligência artificial e desigualdade social: o impacto do colonialismo digital nas políticas públicas**. **Internet & Sociedade**, v. 5, n. 1, 2024.
- OUYANG, L. et al. **Training language models to follow instructions with human feedback**. (A. H. Oh et al., Eds.) **Advances in Neural Information Processing Systems**. **Anais...**2022. Disponível em: <<https://openreview.net/forum?id=TG8KACxEON>>
- 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>>
- PAIOLA, P. H. et al. **Adapting llms for the medical domain in portuguese: A study on fine-tuning and model evaluation**. **arXiv preprint arXiv:2410.00163**, 2024.
- PAN, S. et al. **Unifying large language models and knowledge graphs: A roadmap**. **IEEE Transactions on Knowledge and Data Engineering**, v. 36, n. 7, p. 3580–3599, 2024.
- PAN, Y. et al. **On the Risk of Misinformation Pollution with Large Language Models**. **Findings of the Association for Computational Linguistics: EMNLP 2023**. **Anais...**Singapore: Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.findings-emnlp.97>>
- PANICKSSERY, A.; BOWMAN, S. R.; FENG, S. **LLM Evaluators Recognize and Favor Their Own Generations**. **Proceedings of the 38th International Conference on Neural Information Processing Systems**, 2024.
- 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>>
- PARK, K. et al. **OffsetBias: Leveraging Debaised Data for Tuning Evaluators.**, 2024. Disponível em: <<https://arxiv.org/abs/2407.06551>>
- PARMAR, M. et al. **Don't Blame the Annotator: Bias Already Starts in the Annotation Instructions**. **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**. **Anais...**Dubrovnik, Croatia: Association for Computational Linguistics, 2023. Disponível em: <<https://aclanthology.org/2023.eacl-main.130>>
- PASSINATO, E.; RIOS, W.; FILHO, A. G. **Integração de Modelos de Linguagem e RAG na Criação de Chatbots Oftalmológicos**. **Anais do XXIV Simpósio Brasileiro de Computação Aplicada à Saúde**. **Anais...**Porto Alegre, RS, Brasil: SBC, 2024. Disponível em: <<https://sol.sbc.org.br/index.php/sbcas/article/view/28831>>
- PAULLADA, A. et al. **Data and its (dis)contents: A survey of dataset development and use in machine learning research**. **Patterns**, v. 2, n. 11, p. 100336, nov. 2021.
- 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>>

PEYRARD, M. **Studying Summarization Evaluation Metrics in the Appropriate Scoring Range**. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). **Anais...**Online: Association for Computational Linguistics, 2019. Disponível em: <<https://aclanthology.org/P19-1502/>>

PIĘKOS, 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>>

PIRES, R. et al. **Sabiá: Portuguese Large Language Models**. (M. C. Naldi, R. A. C. Bianchi, Eds.) Intelligent Systems. **Anais...**Cham: Springer Nature Switzerland, 2023.

PLOUG, T. **The Right Not to Be Subjected to AI Profiling Based on Publicly Available Data—Privacy and the Exceptionalism of AI Profiling**. *Philosophy & Technology*, v. 36, n. 1, p. 14, mar. 2023.

POLO, F. M. et al. **Efficient Multi-Prompt Evaluation of LLMs**. **Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)**, 2024.

POMBAL, J. et al. **M-Prometheus: A Suite of Open Multilingual LLM Judges**. *CoRR*, v. abs/2504.04953, 2025.

PROTOTYPEJAM. **Lake Merritt: AI Evaluation Workbench**. https://prototypejam.github.io/lake_merritt/, 2025.

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.

RACKAUCKAS, Z. Rag-fusion: a new take on retrieval-augmented generation. **arXiv preprint arXiv:2402.03367**, 2024.

RADFORD, A. et al. **Improving Language Understanding by Generative Pre-Training.**, 2018. Disponível em: <https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf>

RADFORD, A. et al. **Language Models are Unsupervised Multitask Learners**. a2019.

RADFORD, A. et al. Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, b2019.

RADFORD, A. et al. **Learning Transferable Visual Models From Natural Language Supervision.**, 2021. Disponível em: <<https://arxiv.org/abs/2103.00020>>

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.

RAE, J. W. et al. **Scaling Language Models: Methods, Analysis & Insights from Training Gopher.**, 2022. Disponível em: <<https://arxiv.org/abs/2112.11446>>

RAFAILOV, R. et al. **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**. **Advances in Neural Information Processing Systems (NeurIPS)**. **Anais...**2023.



RAFFEL, C. et al. **Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.** *Journal of Machine Learning Research*, v. 21, n. 140, p. 1–67, a2020.

RAFFEL, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, v. 21, n. 1, p. 140:5485–140:5551, jan. b2020.

RAFFEL, C. et al. **Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.**, 2023. Disponível em: <<https://arxiv.org/abs/1910.10683>>

RAJI, D. et al. **AI and the Everything in the Whole Wide World Benchmark.** (J. Vanschoren, S. Yeung, Eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. **Anais...**Curran, 2021. Disponível em: <https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/084b6fbb10729ed4da8c3d3f5a3ae7c9-Paper-round2.pdf>

RAJPURKAR, P. et al. **SQuAD: 100,000+ Questions for Machine Comprehension of Text.** (J. Su, K. Duh, X. Carreras, Eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. **Anais...**Austin, Texas: Association for Computational Linguistics, nov. 2016. Disponível em: <<https://aclanthology.org/D16-1264>>

REAL, L.; CARVALHO, A.; SILVA, A. DA. **Avaliação de Grandes Modelos de Linguagem: Fundamentos, Métodos Tradicionais e Desafios Atuais.** Em: CASELI, H. M.; NUNES, M. G. V. (Eds.). **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português.** 3. ed. [s.l.] BPLN, 2024.

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>

REHAK, R. **AI Narrative Breakdown. A Critical Assessment of Power and Promise.** Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency. **Anais...** FAccT '25. New York, NY, USA: Association for Computing Machinery, 2025. Disponível em: <<https://doi.org/10.1145/3715275.3732083>>

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>>

RIBEIRO, M. T. et al. **Beyond Accuracy: Behavioral Testing of NLP Models with CheckList.** 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.442>>

RÍO, B. G. DEL; VAAHTIO, T. **Improving LLM systems with A/B testing.** <https://www.flow-ai.com/blog/improving-llm-systems-with-a-b-testing>, 2024.

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.



- ROGERS, A. **Changing the World by Changing the Data**. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). **Anais...Online**: Association for Computational Linguistics, 2021. Disponível em: <<https://aclanthology.org/2021.acl-long.170>>
- ROGERS, A.; BALDWIN, T.; LEINS, K. “Just What do You Think You’re Doing, Dave?” **A Checklist for Responsible Data Use in NLP**. Findings of the Association for Computational Linguistics: EMNLP 2021. **Anais...Punta Cana, Dominican Republic**: Association for Computational Linguistics, 2021. Disponível em: <<https://aclanthology.org/2021.findings-emnlp.414>>
- 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>>
- RUSSEL, S. **Human Compatible Artificial Intelligence and the Problem of Control**. [s.l.] Penguin Books, 2019.
- 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.
- SAID, H. **40 Large Language Model Benchmarks and The Future of LLMs**. Arize AI; <https://arize.com/blog/llm-benchmarks-mmlu-codexglue-gsm8k>, 2025.
- SAMPAIO, R. C.; SABBATINI, M.; LIMONGI, R. **Diretrizes para o uso ético e responsável da inteligência artificial generativa: um guia prático para pesquisadores**. **Boletim Técnico do PPEC**, v. 10, p. e025003–e025003, 2024.
- SAMUYLOVA, E. **LLM-as-a-Judge: A Complete Guide to Using LLMs for Evaluations**. <https://www.evidentlyai.com/llm-guide/llm-as-a-judge>, 2025.
- SÁNCHEZ, L. C. **Ethical Considerations and Best Practices in LLM Development**. <https://neptune.ai/blog/llm-ethical-considerations>, 2024.
- SANH, V. et al. **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**. **CoRR**, v. abs/1910.01108, 2019.
- 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, D. **O projecto Processamento Computacional do Português: Balanço e perspectivas**. (M. das Graças Volpe Nunes, Ed.) V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR 2000). **Anais...São Paulo: ICMC/USP**, 2000. Disponível em: <<https://www.linguateca.pt/Diana/download/SantosPROPOR2000.pdf>>
- SANTOS, H.; WOLOSZYN, V.; VIEIRA, R. **BlogSet-BR: A Brazilian Portuguese Blog Corpus**. (N. Calzolari et al., Eds.) 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/L18-1105/>>
- SARDINHA, T. B.; FILHO, J. L. M.; ALAMBERT, E. **Manual Corpus Brasileiro**. [s.l.: s.n.].
- SARDINHA, T. B.; KAUFFMANN, C.; ACUNZO, C. M. **A multi-dimensional analysis of register variation in Brazilian Portuguese**. **Corpora**, v. 9, n. 2, p. 239–271, 2014.
- SARTHI, P. et al. **RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval**. International Conference on Learning Representations (ICLR). **Anais...2024**.
- SAWCZYN, A. et al. **FactSelfCheck: Fact-Level Black-Box Hallucination Detection for LLMs**. **arXiv**,



2025.

SCAO, T. L. et al. **BLOOM: A 176B-Parameter Open-Access Multilingual Language Model**. *CoRR*, v. abs/2211.05100, 2022.

SCHAAKE, M. **The Tech Coup: How to Save Democracy from Silicon Valley**. [s.l.] Princeton University Press, 2024.

SCHUEERMAN, M. K.; HANNA, A.; DENTON, E. **Do Datasets Have Politics? Disciplinary Values in Computer Vision Dataset Development**. *Proceedings of the ACM on Human-Computer Interaction*, v. 5, n. CSCW2, p. 1–37, out. 2021.

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>>

SCHIRRU, L. **Direito autoral e Inteligência Artificial: autoria e titularidade nos produtos da IA**. [s.l.] Dialetica, 2023.

SCHLANGEN, D. **Targeting the Benchmark: On Methodology in Current Natural Language Processing Research**. *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.85>>

SCHMIDHUBER, J.; HEIL, S. **Sequential neural text compression**. *IEEE Transactions on Neural Networks*, v. 7, n. 1, p. 142–146, 1996.

SCHNABEL, T. et al. **Evaluation methods for unsupervised word embeddings**. (L. Màrquez, C. Callison-Burch, J. Su, Eds.) *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/D15-1036/>>

SCHULMAN, J. et al. **Proximal Policy Optimization Algorithms**. *arXiv preprint arXiv:1707.06347*. *Anais...* 2017.

SCHUSTER, M.; NAKAJIMA, K. **Japanese and Korean voice search**. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). *Anais...* 2012.

SCHWARTZ, L. **Primum Non Nocere: Before working with Indigenous data, the ACL must confront ongoing colonialism**. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. *Anais...* Dublin, Ireland: Association for Computational Linguistics, 2022. Disponível em: <<https://aclanthology.org/2022.acl-short.82>>

SELVAN, R. et al. **Carbon Footprint of Selecting and Training Deep Learning Models for Medical Image Analysis**. (L. Wang et al., Eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. *Anais...* Cham: Springer Nature Switzerland, 2022. Disponível em: <https://doi.org/10.1007/978-3-031-16443-9_49>

SERRAS, F. R. et al. **Exploring Computational Discernibility of Discourse Domains in Brazilian Portuguese within the Carolina Corpus**. (P. Gamallo et al., Eds.) *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*. *Anais...* Santiago de Compostela, Galicia/Spain: Association for Computational Linguistics, mar. 2024. Disponível em: <<https://aclanthology.org/2024.propor-1.26/>>

SHAH, D.; SCHWARTZ, H. A.; HOVY, D. **Predictive Biases in Natural Language Processing Models: A Conceptual Framework and Overview**. *Proceedings of the 58th Annual Meeting of the*



- Association for Computational Linguistics. **Anais...Online: Association for Computational Linguistics**, 2020. Disponível em: <<http://arxiv.org/abs/1912.11078>>
- SHAH, R. et al. Goal Misgeneralization in Deep Reinforcement Learning. **arXiv preprint arXiv:2210.01790**, 2022.
- SHANNON, C. E. Prediction and entropy of printed English. **Bell System Technical Journal**, v. 30, n. 1, p. 50–64, 1951.
- SHAQIRI, M. et al. Differences between the correlation coefficients Pearson, Kendall and Spearman. **5th International Conference of Natural Science and Mathematics**, nov. 2023.
- SHENG, E. et al. **The Woman Worked as a Babysitter: On Biases in Language Generation**. (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/D19-1339/>>
- SHI, H. et al. Continual learning of large language models: A comprehensive survey. **ACM Computing Surveys**, 2024.
- SHI, Z.; LIPANI, A. **Don't Stop Pretraining? Make Prompt-based Fine-tuning Powerful Learner.**, 2023. Disponível em: <<https://arxiv.org/abs/2305.01711>>
- SHMUELI, B. et al. **Beyond Fair Pay: Ethical Implications of NLP Crowdsourcing**. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. **Anais...Online: Association for Computational Linguistics**, 2021. Disponível em: <<https://aclanthology.org/2021.naacl-main.295>>
- SHUMAILOV, I. et al. **AI models collapse when trained on recursively generated data**. **Nature**, v. 631, n. 8022, p. 755–759, 2024.
- ŞİMŞEK, C. **AI resistance: Who says no to AI and why?** Zenodo, 2025. Disponível em: <<https://zenodo.org/doi/10.5281/zenodo.16893847>>
- ŞİMŞEK, C.; YASAR, A. G. **From Rejection to Regulation: Mapping the Landscape of AI Resistance**. 2025.
- SØGAARD, A.; PLANK, B.; HOVY, D. **Selection Bias, Label Bias, and Bias in Ground Truth**. (Q. Liu, F. Xia, Eds.) Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts. **Anais...Dublin, Ireland: Dublin City University; Association for Computational Linguistics**, ago. 2014. Disponível em: <<https://aclanthology.org/C14-3005/>>
- SOLAIMAN, I. et al. **Evaluating the Social Impact of Generative AI Systems**. Em: **The Oxford Handbook of the Foundations and Regulation of Generative AI**. [s.l.] Oxford University Press, 2025.
- 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.
- 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.**, a2020. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf>
- STIENNON, N. et al. Learning to Summarize with Human Feedback. **Advances in Neural Information Processing Systems (NeurIPS)**, b2020.
- STRUBELL, E.; GANESH, A.; MCCALLUM, A. **Energy and Policy Considerations for Deep**



Learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. **Anais...**Florence, Italy: Association for Computational Linguistics, 2019. Disponível em: <<https://www.aclweb.org/anthology/P19-1355>>

STURZENEEKER, M. et al. **Carolina's Methodology: building a large corpus with provenance and typology information.** (C. Trojahn et al., Eds.)Proceedings of the Second Workshop on Digital Humanities and Natural Language Processing (2nd DHandNLP 2022) co-located with International Conference on the Computational Processing of Portuguese (PROPOR 2022). **Anais...**Fortaleza, Brazil: 2022. Disponível em: <<https://ceur-ws.org/Vol-3128/paper4.pdf>>

SUN, K.; DREDZE, M. **Amuro & Char: Analyzing the Relationship between Pre-Training and Fine-Tuning of Large Language Models.** (V. Adlakha et al., Eds.)Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025). **Anais...**Albuquerque, NM: Association for Computational Linguistics, 2025. Disponível em: <<https://aclanthology.org/2025.repl4nlp-1.11/>>

SURESH, H.; GUTTAG, J. V. **A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. Equity and Access in Algorithms, Mechanisms, and Optimization**, p. 1–9, out. 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>>

TAMKIN, A. et al. **Evaluating and Mitigating Discrimination in Language Model Decisions.**, 2023. Disponível em: <<https://arxiv.org/abs/2312.03689>>

TANG, Y. et al. **Multilingual Translation with Extensible Multilingual Pretraining and Finetuning. CoRR**, v. abs/2008.00401, 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.

TEI CONSORTIUM. **TEI P5: Guidelines for Electronic Text Encoding and Interchange.** [s.l.] Retrieved May 20, 2021, 2021.

THAKUR, A. S. et al. **JUDGING THE JUDGES: EVALUATING ALIGNMENT AND VULNERABILITIES IN LLMs-AS-JUDGES. arXiv preprint arXiv:2406.12624v5**, 2025.

THOPPILAN, R. et al. **LaMDA: Language Models for Dialog Applications. CoRR**, v. abs/2201.08239, 2022.

THUN, M. VON; HANLEY, D. A. **Stopping Big Tech from Becoming Big AI: A Roadmap for Using Competition Policy to Keep Artificial Intelligence Open for All.** Open Markets Institute, 2024. Disponível em: <<https://www.openmarketsinstitute.org/publications/report-stopping-big-tech-big-ai-roadmap>>

TOLLES, J.; MEURER, W. J. Logistic Regression: Relating Patient Characteristics to Outcomes. **JAMA**, v. 316, n. 5, p. 533–534, ago. 2016.

TONIAZZO, D.; BARBOSA, T.; RUARO, R. **O Direito à Explicação nas Decisões Automatizadas: uma Abordagem Comparativa Entre o Ordenamento Brasileiro e Europeu. Revista Internacional Consinter de Direito**, v. 13, p. 55–69, dez. 2021.

TOUVRON, H. et al. **LLaMA: Open and Efficient Foundation Language Models. CoRR**, v. abs/2302.13971, a2023.



TOUVRON, H. et al. **LLaMA: Open and Efficient Foundation Language Models.**, b2023. Disponível em: <<https://arxiv.org/abs/2302.13971>>

TRAUTMANN, D. et al. **Measuring the Groundedness of Legal Question-Answering Systems.** Proceedings of the Natural Legal Language Processing Workshop 2024. **Anais...**Singapore: Association for Computational Linguistics, 2024. Disponível em: <<https://aclanthology.org/2024.nllp-1.14>>

ULMER, D. et al. **Experimental Standards for Deep Learning in Natural Language Processing Research.** Findings of the Association for Computational Linguistics: EMNLP 2022. **Anais...**Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dez. 2022. Disponível em: <<https://aclanthology.org/2022.findings-emnlp.196>>

VAROQUAUX, G.; LUCCIONI, S.; WHITTAKER, M. **Hype, Sustainability, and the Price of the Bigger-is-Better Paradigm in AI.** Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency. **Anais...**Athens Greece: ACM, jun. 2025. Disponível em: <<https://dl.acm.org/doi/10.1145/3715275.3732006>>

VASWANI, A. et al. **Attention is All you Need.** (I. Guyon et al., Eds.)Advances in Neural Information Processing Systems. **Anais...**Curran Associates, Inc., a2017. Disponível em: <<https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>>

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, b2017.

VONGTHONGSRI, K. **G-Eval Simply Explained: LLM-as-a-Judge for LLM Evaluation.** Confident AI; <https://www.confident-ai.com/blog/g-eval-the-definitive-guide>, 2025.

WAGNER FILHO, J. A. et al. **The brWaC Corpus: A New Open Resource for Brazilian Portuguese.** (N. Calzolari et al., Eds.)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>>

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**, v. 32, p. 3261–3275, a2019.

WANG, A. et al. SuperGLUE: a stickier benchmark for general-purpose language understanding systems. Em: **Proceedings of the 33rd International Conference on Neural Information Processing Systems.** Red Hook, NY, USA: Curran Associates Inc., 2019b.

WANG, B.; KOMATSUZAKI, A. **GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.** <https://github.com/kingoflolz/mesh-transformer-jax>, 2021.

WANG, R. et al. **Can LLMs Replace Human Evaluators? An Empirical Study of LLM-as-a-Judge in Software Engineering.** **Proc. ACM Softw. Eng.**, v. 2, n. ISSTA, jun. 2025.

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, 2021. Disponível em: <<https://doi.org/10.18653/v1/2021.emnlp-main.685>>

WANG, Y. et al. **PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization.** The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. **Anais...**OpenReview.net, 2024. Disponível em: <<https://openreview.net/forum>>



?id=5Nn2BLV7SB>

WEBER, M. et al. **RedPajama: an Open Dataset for Training Large Language Models.**, 2024. Disponível em: <<https://arxiv.org/abs/2411.12372>>

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>

WEIDINGER, L. et al. **Ethical and social risks of harm from Language Models.** *arXiv:2112.04359* [cs], dez. 2021.

WEIGEND, A. **Data for the People: How to Make Our Post-Privacy Economy Work for You.** [s.l.] Basic Books, 2017.

WEIZENBAUM, J. **Computermacht und Gesellschaft : freie Reden / Joseph Weizenbaum. Hrsg. von Gunna Wendt ..** Original-Ausgabe, 1. Auflage ed. Frankfurt am Main: [s.n.].

WENDEHORST, C. Liability for Artificial Intelligence: The Need to Address Both Safety Risks and Fundamental Rights Risks. Em: VOENEKY, S. et al. (Eds.). **The Cambridge Handbook of Responsible Artificial Intelligence: Interdisciplinary Perspectives.** Cambridge Law Handbooks. [s.l.] Cambridge University Press, 2022. p. 187–209.

WENZEK, G. et al. **CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data.** (N. Calzolari et al., Eds.) Proceedings of the Twelfth Language Resources and Evaluation Conference. *Anais...* Marseille, France: European Language Resources Association, 2020. Disponível em: <<https://aclanthology.org/2020.lrec-1.494/>>

WERBOS, P. J. **Backpropagation through time: what it does and how to do it.** *Proc. IEEE*, v. 78, n. 10, p. 1550–1560, 1990.

WESTENBERGER, J.; SCHULER, K.; SCHLEGEL, D. **Failure of AI projects: understanding the critical factors.** *Procedia Computer Science*, v. 196, p. 69–76, 2022.

WHITTAKER, M. et al. **Disability, bias, and AI.** *AI Now Institute*, v. 8, n. 11, 2019.

WHITTAKER, M. **The steep cost of capture.** *Interactions*, v. 28, n. 6, p. 50–55, nov. 2021.

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>>

WIELING, M.; RAWEE, J.; VAN NOORD, G. **Reproducibility in Computational Linguistics: Are We Willing to Share?** *Computational Linguistics*, v. 44, n. 4, p. 641–649, dez. 2018.

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>>

WOLFRAM, W. **Variation and Language: Overview.** Em: BROWN, K. (Ed.). **Encyclopedia of Language & Linguistics (Second Edition).** Second Edition ed. Oxford: Elsevier, 2006. p. 333–341.

WRIGHT, B. **Manufacturing Reality: Slavoj Zizek and the Reality of the Virtual.** London Ben Wright Film Productions, 2004.



WU, T. **The attention merchants : from the daily newspaper to social media : how our time and attention is harvested and sold.** London: [s.n.].

WU, X.-K. et al. **LLM Fine-Tuning: Concepts, Opportunities, and Challenges.** *Big Data and Cognitive Computing*, v. 9, n. 4, p. 87, abr. 2025.

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>>

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, K. et al. **Com² : A Causal-Guided Benchmark for Exploring Complex Commonsense Reasoning in Large Language Models.** (W. Che et al., Eds.)Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). *Anais...Vienna, Austria: Association for Computational Linguistics*, jul. 2025. Disponível em: <<https://aclanthology.org/2025.acl-long.785/>>

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, A. et al. **Detoxifying Language Models Risks Marginalizing Minority Voices.** (K. Toutanova et al., Eds.)Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. *Anais...Online: Association for Computational Linguistics*, jun. 2021. Disponível em: <<https://aclanthology.org/2021.naacl-main.190/>>

XU, H.; ASHLEY, K. **A question-answering approach to evaluating legal summaries.**, 2023. Disponível em: <<https://arxiv.org/abs/2309.15016>>

XU, Q.; HE, X. **Security Challenges in Natural Language Processing Models.** Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts. *Anais...Singapore: Association for Computational Linguistics*, 2023. Disponível em: <<https://aclanthology.org/2023.emnlp-tutorial.2>>

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, S.-Q. et al. **Corrective Retrieval Augmented Generation.**, 2024. Disponível em: <<https://arxiv.org/abs/2401.15884>>



- YANG, T. et al. **Ethics of Data Work. Principles for Academic Data Work Requesters.** Weizenbaum Institute, 2025. Disponível em: <<https://www.weizenbaum-library.de/handle/id/920>>
- 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>>
- YAO, X. et al. **NLP From Scratch Without Large-Scale Pretraining: A Simple and Efficient Framework.**, 2022. Disponível em: <<https://arxiv.org/abs/2111.04130>>
- YE, Y. et al. **LIMO: Less is More for Reasoning.** arXiv, jul. 2025. Disponível em: <<http://arxiv.org/abs/2502.03387>>. Acesso em: 7 ago. 2025
- YEH, Y.-T.; ESKÉNAZI, M.; MEHRI, S. **A Comprehensive Assessment of Dialog Evaluation Metrics.** **ArXiv**, v. abs/2106.03706, 2021.
- YEO, W. et al. UniversalRAG: Retrieval-Augmented Generation over Corpora of Diverse Modalities and Granularities. **arXiv preprint arXiv:2504.20734**, 2025.
- 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>>
- ZHANG, D.; XU, Z.; ZHAO, W. **LLMs and Copyright Risks: Benchmarks and Mitigation Approaches.** (M. Lomeli, S. Swayamdipta, R. Zhang, Eds.)Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts). **Anais...**Albuquerque, New Mexico: Association for Computational Linguistics, a2025. Disponível em: <<https://aclanthology.org/2025.naacl-tutorial.7/>>
- ZHANG, K. et al. **CitaLaw: Enhancing LLM with Citations in Legal Domain.** (W. Che et al., Eds.)Findings of the Association for Computational Linguistics: ACL 2025. **Anais...**Vienna, Austria: Association for Computational Linguistics, jul. b2025. Disponível em: <<https://aclanthology.org/2025.findings-acl.583/>>
- ZHANG, S. et al. **Instruction Tuning for Large Language Models: A Survey.**, c2025. Disponível em: <<https://arxiv.org/abs/2308.10792>>
- 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>>
- ZHANG, Y. et al. **Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models.** **Computational Linguistics**, v. 51, n. 4, p. 1373–1418, dez. d2025.
- ZHAO, W. X. et al. **A Survey of Large Language Models.** **CoRR**, v. abs/2303.18223, 2023.
- ZHAO, Y. et al. **One Token to Fool LLM-as-a-Judge.**, 2025. Disponível em: <<https://arxiv.org/abs/2507.08794>>
- ZHOU, C. et al. **LIMA: Less Is More for Alignment.** **CoRR**, v. abs/2305.11206, a2023.
- ZHOU, H. et al. A survey of large language models in medicine: Progress, application, and challenge. **arXiv preprint arXiv:2311.05112**, b2023.
- ZIEGLER, D. M. et al. **Fine-Tuning Language Models from Human Preferences.** **CoRR**, v. abs/1909.08593, 2019.



Referências

ZUBOFF, S. **The age of surveillance capitalism : the fight for a human future at the new frontier of power.** First edition ed. New York: [s.n.].



<https://brasileiraspln.ufscar.br/livro-pln-4ed-vol2/>



Sobre as/os autoras/es

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 com os Capítulos 1 e 7.

Email: alinepaes@ic.uff.br

Altigran Soares Silva

Altigran Soares da Silva concluiu doutorado em Ciência da Computação pela UFMG (2002) e é professor titular do Instituto de Computação da UFAM. Seus interesses incluem: Aprendizagem de Máquina, Modelos de Linguagem, Engenharia de Dados e Recuperação de Informação. Neste livro, colaborou nos capítulos Avaliação de Large Language Models: Fundamentos, Métricas Tradicionais e Desafios Atuais e Fundamentos da Geração Aumentada por Recuperação. Neste livro, colaborou com os Capítulos 4 e 5.

Email: alti@icomp.ufam.edu.br

André Carvalho

André Carvalho é doutor em Informática pela Universidade Federal do Amazonas, onde, desde 2013 é professor associado, sendo também também é coordenador do bacharelado em Inteligência Artificial. Também é coordenador do projeto de pesquisa IATS - Inteligência Artificial aplicada em Teste de Software, que tem como foco aplicação de técnicas de processamento de linguagem natural em tarefas de qualidade de software. Sua pesquisa tem como foco áreas como Inteligência Artificial, Aprendizagem de Máquina, Recuperação de Informação e Ciência de Dados. Neste livro, colaborou com os Capítulos 4 e 5.

Email: andre@icomp.ufam.edu.br

Brielen Madureira

Brielen Madureira é formada em Matemática Aplicada e Computacional pela Universidade de São Paulo (USP). Tem mestrado em Ciência e Tecnologia da Linguagem pela Universidade de Saarland (Alemanha) e doutorado em Linguística Computacional pela Universidade de Potsdam (Alemanha), onde participou do grupo de pesquisa em Diálogo. Mais amplamente, tem interesse na avaliação de tecnologias de linguagem e nas urgentes considerações éticas que elas demandam. Atualmente é pesquisadora em fase de pós-doutorado na Universidade de Leipzig (Alemanha), no grupo Climate Discourse. Neste livro, colaborou com os Capítulos *Avaliação de tecnologias de linguagem*, *Diálogo e Interatividade* e 6.

Email: brielen.madureira@uni-leipzig.de

Cláudia Freitas

Cláudia Freitas é formada em Letras pela PUC-Rio, onde concluiu o doutorado em Estudos da Linguagem (2007) sobre a construção de ontologias a partir de *corpus*. De 2012 a 2022 foi docente do Programa de Pós-Graduação em Estudos da Linguagem da PUC-Rio, onde fundou o grupo ComCorHD (Linguística Computacional, *Corpus* e Humanidades Digitais). De 2007 a 2022 foi também pesquisadora da Linguateca. Ao longo de 2023 e 2024 foi pesquisadora do ICMC/USP, vinculada ao C4AI, e atualmente é pesquisadora independente. Suas áreas de interesse incluem construção e avaliação de *datasets/corpora*, semântica/significado, e articulação entre abordagens simbólicas e neurais. Neste livro, colaborou com os Capítulos *E o significado?*, *Conjunto de dados, dataset e corpus*, *Avaliação conjunta em português* e 7.



Email: claudiafreitas@usp.br

Cristiane Namiuti

Cristiane Namiuti é Bacharel em Linguística pela UNICAMP (2002) onde também obteve o título de Doutora em Linguística (2008). Fez parte da equipe do Corpus Anotado do Português Histórico Tycho Brahe (1998-2010). Desde 2008, atua como docente da Universidade Estadual do Sudoeste da Bahia (UESB) e no PPGLin/UESB desde sua fundação em 2010. Atualmente, integra o grupo de pesquisadores e co-fundadores dos Laboratórios LAPELINC/UESB e LaViHD/USP e, desde 2020, faz parte de uma das equipes do C4AI - Centro de Inteligência Artificial da USP (LaViHD-C4AI). Suas áreas de interesse incluem metodologias automáticas de anotação e busca de dados em textos escritos, mudança linguística, história do português, linguística de corpus e humanidades digitais. Neste livro, colaborou com o Capítulo 2.

Email: cristianenamiuti@gmail.com

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 1.

Email: dvianna@gmail.com

Danilo Samuel Jodas

Danilo Jodas é formado em Ciência da Computação pelo Centro Universitário do Norte Paulista (UNORP), com pós-graduação em processamento de imagens e aprendizado de máquina pela UNESP e FEUP (Universidade do Porto). É pesquisador de pós-doutorado na UNESP de Bauru desde 2019, atuando em pesquisas de aprendizado de máquina e PLN. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: danilo.jodas@unesp.br

Douglas Rodrigues

Douglas Rodrigues é formado em Informática para Gestão de Negócios pela FATEC/Botucatu, Mestrado (UNESP/Bauru) e Doutorado (UFSCar/São Carlos) em Ciências da Computação. Atualmente, atua como pós-doutorando e pesquisador no Recogna/UNESP. Suas áreas de interesse incluem aprendizado de máquina e otimização matemática. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: d.rodrigues@unesp.br

Gabriel Lino Garcia

Gabriel Lino Garcia é graduado em Análise e Desenvolvimento de Sistemas pela FATEC e mestre em Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN) pela UNESP, onde faz doutorado focado na aplicação de modelos de linguagem na área médica. Seus interesses incluem aprendizado de máquina, grandes modelos de linguagem, aprendizado multimodal e aplicações médicas. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: linaoglino@gmail.com

Gabriela Alves Lachi

Gabriela Alves Lachi é graduanda em Letras pela Universidade de São Paulo (USP). Suas áreas de interesse em Processamento de Linguagem Natural envolvem a construção e o desenvolvimento de corpus, bem como o treinamento e a avaliação de grandes modelos de linguagem. Neste livro, colaborou com o Capítulo 2.

Email: gabriela.lachi@usp.br



Felipe Ribas Serras

Felipe Ribas Serras é Bacharel em Física pelo Instituto de Física da Universidade de São Paulo (USP) e Mestre em Ciência da Computação pelo Instituto de Matemática e Estatística da USP. Suas áreas de interesse incluem complexidade de linguagem e abordagens computacionais à variação linguística e tipologia. Neste livro, colaborou com os capítulos [Complexidade Textual e suas Tarefas Relacionadas](#), 2.

Email: frserras@ime.usp.br

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 (UFSCar). 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 o Capítulo 1.

Email: jessica.rodrigues@oii.ox.ac.uk

João Paulo Papa

João Paulo Papa é formado em Sistemas de Informação pela UNESP, com pós-graduação em processamento de imagens e aprendizado de máquina pela UFSCar e Unicamp, respectivamente. Possui pós-doutorados na Unicamp, Harvard e MIT. É professor titular da Unesp onde atua desde 2009 e suas áreas de interesse incluem visão computacional, processamento de imagens, aprendizado de máquina e aprendizado multimodal. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: joao.papa@unesp.br

João Renato Ribeiro Manesco

João Renato Ribeiro Manesco é doutorando em Ciência da Computação pela UNESP, onde também concluiu mestrado e graduação. Durante o mestrado, realizou estágio de pesquisa no Media Integration and Communication Center da Universidade de Florença, Itália. Seus interesses de pesquisa incluem adaptação de domínio, modelos de linguagem generativos e multimodais, e aplicações em contexto médico. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: joao.r.manesco@unesp.br

João Vitor Mariano Correia

João Vitor Mariano Correia é bacharel em Ciência da Computação pela UNESP, onde atualmente cursa o mestrado no Programa de Pós-Graduação em Ciência da Computação. Seus principais interesses de pesquisa estão na área de Processamento de Linguagem Natural (PLN), com foco em representações de conhecimento. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: mariano.correia@unesp.br

Livy Real

Livy Real é formada em Letras, com doutorado em linguística (UFPR) e atuação na interseção entre linguagem e inteligência artificial. Realizou estágios pós-doutorais na IBM Research e na Universidade Federal do Amazonas (UFAM). Atualmente, atua como cientista aplicada em inteligência artificial no Instituto Kunumi. Seus interesses incluem avaliação de modelos de IA, NLP aplicado, linguística computacional e questões éticas no uso de tecnologias emergentes. Neste livro, colaborou com os Capítulos 4 e 5.

Email: livyreal@gmail.com



Lucas Lasota

Lucas Lasota é mestre e doutor pela Universidade RUDN de Moscou com o doutorado reconhecido pela USP. É pesquisador e docente do Just Transition Center da Universidade de Halle-Wittenberg na Alemanha. É pesquisador associado do Weizenbaum Institute for the Networked Society de Berlin. Advogado inscrito na OAB-SP/432126. Sua pesquisa se concentra em medidas regulatórias de tecnologias digitais e seu impacto sobre direitos individuais e coletivos, bem como governança da internet, telecomunicações e direito contratual internacional. Neste livro, colaborou com o Capítulo 6.

Email: lucas.lasota@jtc.uni-halle.de

Marcelo Finger

Marcelo Finger é professor titular de Ciência da Computação no IME-USP, Pesquisador Principal do Centro de Inteligência Artificial USP-IBM-FAPESP C4AI, com interesses em Geração de Recursos de Processamento de Linguagem Natural (PLN), uso de PLN para a área de Saúde, estudo e interpretabilidade de redes neurais. Neste livro, colaborou com os Capítulos [Complexidade Textual e suas Tarefas Relacionadas](#), [Classificação de Áudio aplicada à Saúde](#) e 2.

Email: mfinger@ime.usp.br

Mariana Lourenço Sturzeneker

Mariana Lourenço Sturzeneker é graduada em Letras pela Universidade de São Paulo. Faz parte da equipe de desenvolvimento do Córpus Carolina no C4AI (Center for Artificial Intelligence - IBM/USP) desde 2020. Seu interesse principal é na área de construção de corpú. Neste livro, colaborou com o Capítulo 2.

Email: mariana.sturzeneker@usp.br

Mayara Feliciano Palma

Mayara Feliciano Palma é Bacharel em Letras (português e espanhol) pela Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo (USP). Entre suas áreas de interesse estão a construção de *corpora* e agentes. Neste livro, colaborou com o Capítulo 2.

Email: mayara.palma@usp.br

Miguel de Mello Carpi

Miguel de Mello Carpi é Bacharel em Ciência da Computação pelo Instituto de Matemática e Estatística da Universidade de São Paulo. Suas áreas de interesse incluem inteligência artificial e processamento de linguagem natural. Neste livro, colaborou com o Capítulo 2.

Email: miguel@ime.usp.br

Pedro Henrique Paiola

Pedro Henrique Paiola é mestre e doutorando em Ciência da Computação pela UNESP, onde também atua como professor substituto desde 2022. É pesquisador no projeto RADIAR, desenvolvido em parceria com a Petrobras. Suas áreas de interesse incluem sumarização automática, modelos de linguagem e avaliação automática de texto. Neste livro, colaborou com os Capítulos [Sumarização Automática](#) e 3.

Email: pedro.paiola@unesp.br

Vanessa Martins do Monte

Vanessa Martins do Monte é mestre e doutora em Filologia e Língua Portuguesa pela FFLCH-USP, onde atua como docente desde 2014. Coordena o Projeto M.A.P. – Mulheres na América Portuguesa e integra o C4AI – Centro de Inteligência Artificial da USP, atuando na constituição de *corpora* para PLN. Suas pesquisas concentram-se em Filologia e Humanidades Digitais. Neste livro, colaborou com o Capítulo 2.

Email: vmonte@usp.br



Uma iniciativa



brasileiraspln.com

Apoio

