Grandes Modelos de Linguagem

Alunos: João Gabriel de Morais Bezerra e Daniel Henrique Peres Servejeira

Laboratório de Simulações Numéricas e Inteligência Artificial

27 de junho de 2026

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Modelos de linguagem

Lembre-se do modelo de linguagem simples n-grama:

- Atribui probabilidades a sequências de palavras;
- Gera texto amostrando possíveis próximas palavras;
- É treinado em contagens de frequência computadas a partir de corpora textuais massivos.

Grandes modelos de linguagem compartilham semelhanças, mas diferem fundamentalmente:

- Atribuem probabilidades a sequências de palavras;
- Geram texto amostrando possíveis próximas palavras;
- São treinados utilizando redes neurais profundas para aprender a adivinhar a próxima palavra.

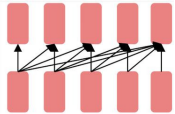
Grandes modelos de linguagem

Mesmo que esses modelos complexos sejam pré-treinados exclusivamente com o objetivo simples de prever a próxima palavra sequencial, eles implicitamente aprendem uma quantidade massiva de estrutura linguística útil, conhecimento factual e capacidades de raciocínio devido à escala astronômica dos textos que processam durante o treinamento.

Três arquiteturas para grandes modelos de linguagem

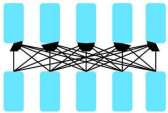
Decodificadores

- GPT Series
- Claude
- Llama
- Mixtral



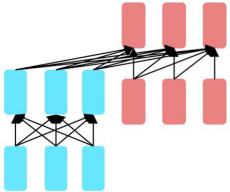
Codificadores

- BERT family
- HUBERT



Codificadores-Decodificadores

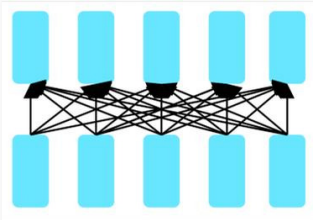
- Flan-T5
- Whisper



Codificadores

Muitas variedades existem no ecossistema:

- Variante mais popular: Modelos de Linguagem Mascarada (MLMs), como a família BERT.
- Treinados prevendo palavras que foram artificialmente mascaradas do contexto ao redor em ambos os lados, esquerdo e direito.
- Por serem bidirecionais, eles geralmente passam por um ajuste fino (treinados em dados supervisionados) exclusivamente para tarefas de classificação e extração em vez de geração.

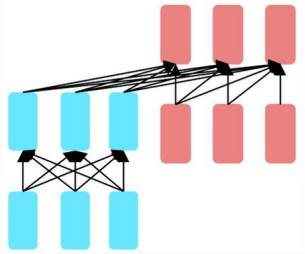


Codificadores-Decodificadores

Treinados fundamentalmente para mapear de uma sequência completa para outra.

Muito populares para tarefas que requerem compreensão total do contexto:

- Tradução automática (mapear uma frase inteiramente de um idioma para outro)
- Reconhecimento de fala (mapear acústica de áudio bruto em palavras escritas)



Grandes Modelos de Linguagem

Grandes Modelos de Linguagem

Quais tarefas eles podem realizar?

A grande ideia

Um avanço monumental no processamento de linguagem natural é a constatação de que muitas tarefas cognitivas complexas podem ser reformuladas matematicamente como tarefas simples de prever palavras!

Nesta aula: modelos apenas decodificadores

Modelos apenas decodificadores são comumente chamados de:

- LLMs Causais
- LLMs Autorregressivos
- LLMs da Esquerda para a Direita

Esses modelos funcionam estritamente prevendo palavras sequencialmente da esquerda para a direita, sendo totalmente incapazes de analisar o contexto futuro.

Geração Condicional

Gerando texto logicamente condicionado ao texto anterior!

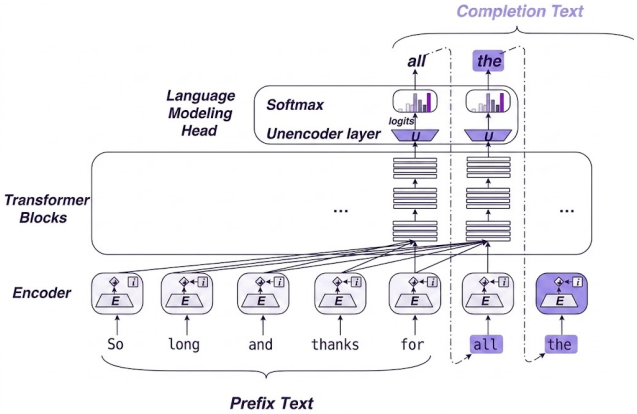


Figura: Esquema de previsão de texto.

Formulando muitas tarefas como previsão de palavras: Análise de Sentimentos

Análise de sentimentos da frase: "Eu gosto do Jackie Chan"

1. Nós fornecemos ao modelo de linguagem autorregressivo esta string de prefixo específica:
   O sentimento da frase "Eu gosto do Jackie Chan"é:
2. Em seguida, avaliamos a probabilidade matemática da palavra que ele acha que vem a seguir:

$$P(\text{positivo} \mid \text{prefixo}) \quad \text{versus} \quad P(\text{negativo} \mid \text{prefixo})$$

O modelo avalia as probabilidades de pares semânticos como gostar/odiar, amar/não gostar e adorar/detestar para executar a classificação.

Formulando muitas tarefas como geração condicional: QA

Perguntas e Respostas (QA): "Quem escreveu A Origem das Espécies"

1. Damos ao modelo de linguagem esta string estrutural rigorosa:
   `P: Quem escreveu o livro "A Origem das Espécies"?  R:`
2. E avaliamos qual palavra a distribuição de probabilidade indica que virá a seguir.
3. Anexamos o resultado e iteramos o processo:
   `P: Quem escreveu o livro "A Origem das Espécies"?  R: Charles`

Resumo via Geração Condicional

Exemplo de Artigo Original:
A única coisa mais louca do que um cara na nevada Massachusetts empacotando aquela neve branquinha e oferecendo-a para venda online? As pessoas estão realmente comprando. Por US$ 89, o auto-intitulado empreendedor Kyle Waring enviará cerca de 3 quilos de neve da área de Boston em uma caixa de isopor isolada - o suficiente para 10 a 15 bolas de neve, diz ele. Mas não se você mora na Nova Inglaterra ou nos estados vizinhos.

Resumo Gerado (solicitado por "tl;dr"):
Kyle Waring enviará a você cerca de 3 quilos de neve da área de Boston em uma caixa isolada, o suficiente para 10 a 15 bolas de neve, diz ele. Mas não se você mora na Nova Inglaterra ou nos estados vizinhos. Ele brincou sobre enviar o material para amigos e familiares em estados mais quentes, e assim nasceu uma ideia.

Sumarização de texto

Executando compressão usando o delimitador "Too long; didn't read".

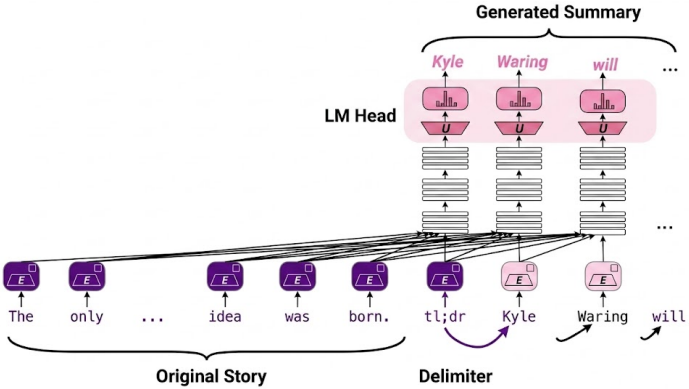


Figura: Esquema de resumo de texto.

Amostragem para Geração de LLMs

Amostragem para Geração de LLMs

Decodificação e Amostragem

- A tarefa computacional de escolher uma palavra específica para gerar com base nas probabilidades de saída do modelo é chamada de **decodificação**.
- O método mais comum para decodificação em LLMs é a **amostragem**.
- Amostrar a partir da distribuição de palavras de um modelo envolve escolher palavras aleatórias de acordo com a probabilidade matemática precisa atribuída pela rede.
- Após cada token ser gerado, amostramos palavras subsequentes de acordo com sua probabilidade estritamente condicionada a todas as nossas escolhas anteriores.
- Um modelo de linguagem Transformer produzirá continuamente essa distribuição de probabilidade sobre o vocabulário.

Amostragem aleatória

Intuição algorítmica para a amostragem aleatória pura:

```
i = 1
w_1 \leftarrow p(w)
while w_i != EOS:
    i = i + 1
    w_i \leftarrow p(w | w_1,..., w_{i-1})
```

- i e w_i: Rastreiam o índice posicional e a palavra escolhida.
- $\leftarrow p$: Indica a extração probabilística da palavra a partir da distribuição.
- **EOS**: O sinal crítico de parada (Fim da Sequência).
- $P(w \mid \ldots)$: A escolha da próxima palavra depende de todo o contexto histórico.

A amostragem aleatória não funciona muito bem...

- Embora a amostragem aleatória pura geralmente gere palavras coerentes e de alta probabilidade, existem milhares de palavras estranhas e de baixa probabilidade residindo na cauda da distribuição.
- Cada uma dessas palavras individuais possui uma probabilidade mínima, mas quando somadas matematicamente, elas constituem uma parte grande e perigosa da massa de probabilidade total.
- Portanto, sob amostragem aleatória pura, elas são escolhidas com frequência suficiente para gerar frases completamente estranhas e sem sentido.

Fatores na amostragem de palavras: Qualidade vs. Diversidade

Enfatizar palavras com probabilidade média introduz diversidade:

- **Prós:** Resulta em saídas textuais muito mais criativas e diversas.
- **Contras:** Reduz drasticamente a qualidade, resultando em declarações menos factuais e altamente inconsistentes.

Enfatizar palavras exclusivamente com alta probabilidade prioriza a qualidade:

- **Prós:** Garante que as saídas sejam altamente precisas, logicamente coerentes e factuais.
- **Contras:** Erradica a diversidade, produzindo textos entediantes, altamente repetitivos e robóticos.

Algoritmo de Amostragem Top-k

Para truncar a perigosa cauda de probabilidade, utilizamos o Top-k:

1. Escolha um valor inteiro estático "k"representando o número de palavras.
2. Para cada palavra em todo o vocabulário V, use o modelo de linguagem para calcular a probabilidade dessa palavra dado o prefixo contextual $p(w_t \mid w_{<t})$.
3. Ordene as palavras sequencialmente por probabilidade, mantendo apenas as "k"palavras mais prováveis e descartando o resto.
4. Renormalize as pontuações das k palavras restantes para que somem 1, criando uma distribuição de probabilidade válida.
5. Selecione aleatoriamente uma palavra das k palavras restantes, estritamente de acordo com sua probabilidade renormalizada.

Amostragem Top-p (= amostragem de núcleo)

Problema Fundamental com o top-k: O parâmetro k é fixo, o que significa que pode abranger quantidades muito diferentes da massa de probabilidade total em situações contextuais diferentes (por exemplo, distribuições planas vs. acentuadas).

A Ideia do Núcleo: Em vez de uma contagem fixa, mantenha os p por cento superiores da massa de probabilidade.

Dada uma distribuição $P(w_t \mid w_{<t})$, o vocabulário top-p $V(p)$ é definido matematicamente como o menor conjunto de palavras tal que:

$$\sum_{w \in V(p)} P(w \mid w_{<t}) \geq p$$

(Holtzman et al., 2020)

Amostragem por temperatura (1/4)

Em vez de truncar o vocabulário, podemos remodelar matematicamente toda a distribuição. Esse conceito extrai intuição da termodinâmica:

- Um sistema físico existente a uma temperatura alta é altamente flexível, energético e pode explorar muitos estados variantes possíveis.
- Um sistema a uma temperatura mais baixa possui energia mais baixa e tem maior probabilidade matemática de explorar apenas um pequeno subconjunto de estados (melhores) estáveis.

Em configurações de amostragem de baixa temperatura ($\tau \leq 1$), forçamos suavemente o algoritmo a:

- Aumentar drasticamente a probabilidade das palavras mais prováveis.
- Diminuir simultaneamente a probabilidade das palavras raras de cauda longa.

Amostragem por temperatura (2/4)

- **t = 0.1 a 0.3 (Variância Muito Baixa):** Excepcional para executar código de programação, equações matemáticas, traduções estritas ou responder a consultas factuais onde se deseja precisão absoluta e nenhum desvio criativo. O modelo quase sempre escolherá de forma gananciosa a palavra mais lógica.
- **t = 0.7 a 1.0 (Variância Normal/Alta):** Ideal para escrever poesia, sessões de brainstorming ou gerar histórias narrativas, pois a distribuição matemática permite que o modelo arrisque selecionar palavras menos óbvias, gerando uma saída textual muito mais diversa e "criativa".

Amostragem por temperatura (3/4)

Matematicamente, dividimos o logit bruto u por um parâmetro de temperatura τ antes de passá-lo pela camada final de ativação softmax.

Em vez do cálculo padrão:

$$y = \text{softmax}(u)$$

Calculamos a variante escalonada:

$$y = \text{softmax}\left(\frac{u}{\tau}\right)$$

Amostragem por temperatura (4/4)

Por que esse ajuste estrutural funciona?

- Quando τ é exatamente 1, a distribuição matemática não muda.
- Quanto menor o τ, maiores serão as pontuações diferenciais passadas para o softmax. O softmax empurra inerentemente valores altos para 1 e valores baixos para 0.
- Conforme $\tau \to 0$, a probabilidade da palavra absolutamente mais provável se aproxima assintoticamente de 1.

Pré-treinamento de Grandes Modelos de Linguagem

Algoritmos e Arquiteturas

Paradigma de Pré-treinamento

A ideia monumental que impulsiona as incríveis métricas de desempenho dos modelos de linguagem modernos:

- Primeiro, pré-treinar um modelo Transformer massivo em quantidades astronomicamente grandes de texto bruto usando clusters de computação massivos.
- Em seguida, aplicar esse conhecimento generalizado a novas tarefas altamente específicas.

Algoritmo de treinamento autossupervisionado

Nós fundamentalmente apenas os treinamos para prever a próxima palavra!
Pegue um corpus massivo de texto. A cada passo de tempo computacional t...

1. Peça ao modelo neural para prever matematicamente a próxima palavra na sequência.
2. Treine o modelo utilizando a retropropagação (backpropagation) do erro e a descida do gradiente para minimizar o erro matemático nesta previsão.

Esse paradigma é classificado como aprendizado **"Autossupervisionado"** porque não requer anotadores humanos; ele simplesmente utiliza a próxima palavra natural no texto como o rótulo definitivo de verdade de base (ground-truth)!

Intuição do treinamento do modelo de linguagem: Perda

- **Utilidade da Função de Perda:** A métrica padrão é a perda de entropia cruzada.
- Inerentemente, queremos que o modelo neural atribua uma alta probabilidade à palavra verdadeira e empírica w.
- Consequentemente, queremos que a perda calculada seja excepcionalmente alta se o modelo atribuir uma probabilidade muito baixa a w.
- **Definição da Perda CE:** O logaritmo negativo da probabilidade que o modelo atribui exclusivamente à próxima palavra verdadeira w.
- Se o modelo atribui uma probabilidade muito baixa a w, o otimizador move forçosamente os bilhões de pesos do modelo na direção multidimensional que garante uma atribuição de probabilidade maior a w no futuro.

Perda de entropia cruzada para modelagem de linguagem

A perda CE define a divergência entre a distribuição de probabilidade correta e empírica e a distribuição prevista pelo modelo:

$$L_{CE} = -\sum_{w \in V} y_t[w] \log \hat{y}_t[w]$$

A distribuição verdadeira (ground-truth) correta y_t sabe a próxima palavra definitivamente, então ela funciona como um vetor one-hot: é exatamente 1 para a próxima palavra real e 0 para todos os outros tokens.

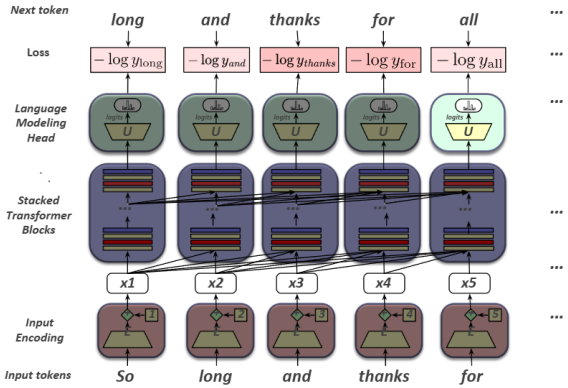
Portanto, nesta soma, todos os termos são multiplicados por zero e desaparecem, exceto um: a probabilidade logarítmica que o modelo atribui à próxima palavra correta. Assim:

$$L_{CE}(\hat{y}_t, y_t) = -\log \hat{y}_t[w_{t+1}]$$

Técnica de Teacher Forcing

- Em cada posição de token t, o modelo processa os tokens históricos corretos $w_{1:t}$.
- Ele calcula a perda (o logaritmo negativo da probabilidade) para sua previsão do próximo token w_{t+1}.
- Criticamente, na posição do token subsequente $t + 1$, ignoramos estritamente qualquer token errôneo que o modelo possa ter previsto para w_{t+1}.
- Em vez disso, pegamos a palavra verdadeira correta w_{t+1} do conjunto de dados, adicionamos de forma segura ao contexto e seguimos em frente. Isso evita o acúmulo de erros durante o treinamento.

Treinando um modelo de linguagem transformer



$$\text{Perda Total do Lote} = \frac{1}{T}\sum_{t=1}^{T} L_{CE}$$

Figura: Esquema de treinamento de um modelo de linguagem Transformer.

Os LLMs são treinados principalmente na web

- **Common Crawl:** Instantâneos massivos de toda a web pública produzidos recursivamente pela organização sem fins lucrativos Common Crawl, contendo bilhões de páginas HTML brutas.
- **Colossal Clean Crawled Corpus (C4):** Criado por Raffel et al. (2020), este conjunto de dados contém 156 bilhões de tokens de texto em inglês que foram rigorosamente filtrados por qualidade.
- **O que constitui esses dados?** A maior parte dos tokens de alta qualidade deriva de documentos de patentes, bancos de dados da Wikipedia e sites de notícias internacionais.

The Pile: um corpus de pré-treinamento projetado

Um conjunto de dados de modelagem de linguagem de código aberto de 825 GiB diversificado e meticulosamente balanceado.

Acadêmico	**Web & Geral**	**Livros & Diálogos**
PubMed Central	Pile-CC	Bibliotik
ArXiv	OpenWebText2	PG-19
FreeLaw	StackExchange	BookCorpus2
PhilPapers	Wikipedia	Ubuntu IRC / GitHub

O que um modelo aprende no pré-treinamento?

O texto bruto ingerido contém uma enorme quantidade de conhecimento humano. O pré-treinamento com bilhões de textos contendo todo esse conhecimento é o que fundamentalmente dá aos modelos de linguagem sua capacidade de executar tanta lógica.

- Lógica Contextual: "Há caninos por toda parte! Um cachorro na sala da frente e dois cachorros."
- Escalonamento Semântico: "Não era apenas grande, era enorme."
- Recuperação Factual: "A autora de 'Um Teto Todo Seu' é Virginia Woolf."
- Verdades Matemáticas: "A raiz quadrada de 4 é 2."

Ajuste fino como "pré-treinamento contínuo"

- A abordagem mais direta é treinar ainda mais todos os bilhões de parâmetros do modelo nos novos dados específicos.
- Empregamos exatamente o mesmo método (previsão da próxima palavra) e função de perda matemática (perda de entropia cruzada) utilizados para o pré-treinamento original.
- Tratamos a operação como se os novos dados especializados fossem simplesmente anexados no final da sequência massiva de dados de pré-treinamento.
- Portanto, esse processo é frequentemente classificado na literatura como **pré-treinamento contínuo**.

Ajuste fino

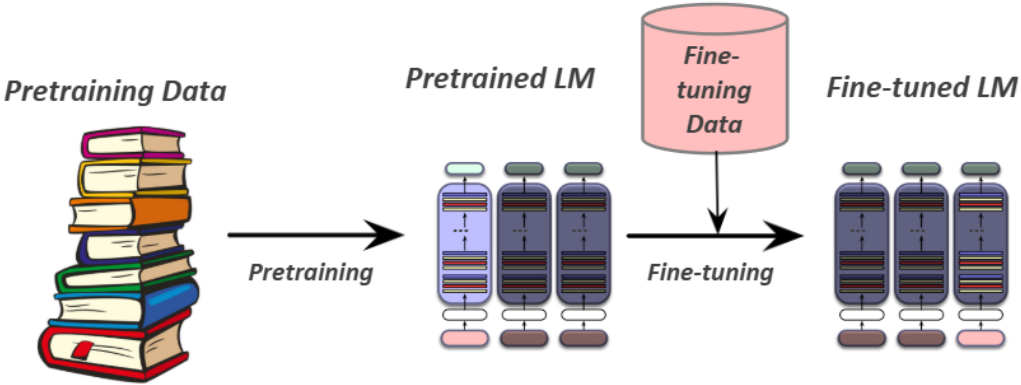


Figura: Finetuning flow

Avaliando Grandes Modelos de Linguagem

Avaliando Grandes Modelos de Linguagem

Métricas e Infraestrutura de Escala.

Métrica de Perplexidade

A indústria usa a Perplexidade para medir matematicamente quão bem o LM prevê distribuições de texto não vistas.

A perplexidade de um modelo em um conjunto de teste não visto é definida como o inverso da probabilidade que o modelo atribui ao conjunto de teste, normalizada geometricamente pelo comprimento do conjunto de teste.

Para uma sequência de teste de n tokens $w_{1:n}$, a perplexidade é calculada como:

$$\text{Perplexidade}(w_{1:n}) = P(w_{1:n})^{-\frac{1}{n}}$$

Por que perplexidade em vez de probabilidade bruta?

- A probabilidade bruta depende inteiramente do tamanho do conjunto de teste; a probabilidade se torna assintoticamente menor quanto mais longa for a sequência de texto.
- A perplexidade calcula a probabilidade inversa normalizada pelo número total de palavras (derivada da taxa de entropia cruzada).
- Enquanto a probabilidade opera em uma faixa estreita de $[0, 1]$, a faixa de perplexidade abrange $[1, \infty]$.
- Quanto maior a probabilidade atribuída à sequência de palavras, menor a perplexidade.

Leis de Escala

O desempenho empírico do LLM é estritamente dependente de:

- **Tamanho do modelo:** Número absoluto de parâmetros treináveis.
- **Tamanho do conjunto de dados:** A quantidade volumétrica bruta de dados.
- **Computação:** Quantidade absoluta de computação utilizada.

O desempenho de um grande modelo de linguagem (medido pela perda) escala matematicamente como uma lei de potência:

$$L(N) = (N_c/N)^{\alpha_N} \quad L(D) = (D_c/D)^{\alpha_D} \quad L(C) = (C_c/C)^{\alpha_C}$$

Número de parâmetros não-embedding N

Podemos estimar matematicamente o total de parâmetros via:

$$N \approx 2d_{model} n_{layers}(2d_{attn} + d_{ff})$$

$$\approx 12 n_{layers} d_{model}^2$$

Assumindo proporções dimensionais padrão de $d_{attn} = d_{ff}/4 = d_{model}$.

Assim, calculando para a arquitetura do GPT-3, possuindo $n = 96$ camadas e uma dimensionalidade massiva de $d = 12288$, determinamos:

$$12 \times 96 \times 12288^2 \approx 175 \text{ bilhões de parâmetros.}$$

O Gargalo do Cache KV

Durante a fase de pré-treinamento, calculamos a atenção matemática com eficiência em lotes paralelos massivos. Porém, isso falha na inferência, pois geramos tokens sequencialmente.

- Para cada novo token x, multiplicamos pelas matrizes de peso W_Q, W_K, W_V.
- O cálculo exige vetores chave e valor para todos os tokens anteriores $x_{<i}$, gerando um custo computacional $O(N^2)$.
- **A Solução de Engenharia:** Armazene os vetores na memória VRAM da GPU no cache KV, para uso com velocidade de inferência $O(N)$.

Arquitetura do Cache KV

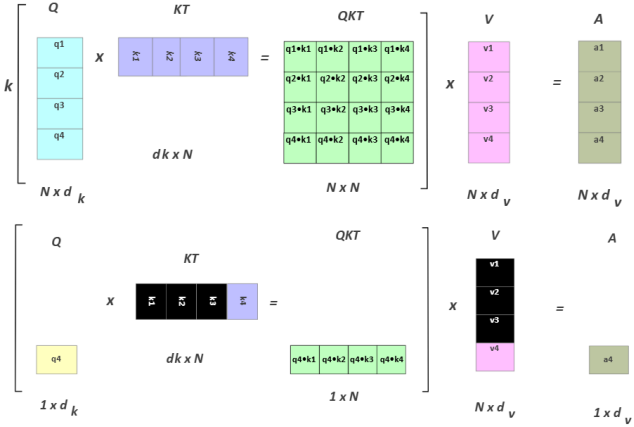


Figura: Representação do cache KV.

Teoria LORA (Adaptação de Baixo Posto)

Considere uma matriz de pesos base W ($N \times d$) que teoricamente precisa ser atualizada durante o ajuste fino. No algoritmo LORA, congelamos W e atualizamos uma decomposição de baixo posto de W:

- Matriz A com formato $N \times r$
- Matriz B com formato $r \times d$
- Onde r (posto) é muito pequeno (ex: 1, 2 ou 8)

Substituímos computacionalmente $W + \Delta W$ por $W + BA$.
Passagem direta (Forward pass):

$$h = xW + xAB$$

Danos Sociais dos Grandes Modelos de Linguagem

- **Alucinação**
- **Litígio de Direitos Autorais**
- **Violações de Privacidade**
- **Toxicidade e Exploração**
- **Desinformação Política**

Alucinação

Chatbots rotineiramente ’alucinam’ políticas corporativas falsas ou fatos inexistentes.

Chatbots May ‘Hallucinate’ More Often Than Many Realize

What Can You Do When A.I. Lies About You?

People have little protection or recourse when the technology creates and spreads falsehoods about them.

Air Canada loses court case after its chatbot hallucinated fake policies to a customer

The airline argued that the chatbot itself was liable. The court disagreed.

Litígio de Direitos Autorais

Ações judiciais por infração massiva no uso de obras protegidas durante o treinamento sem permissão.

Authors Sue OpenAI Claiming Mass Copyright Infringement of Hundreds of Thousands of Novels

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

Violações de Privacidade

Vazamento acidental de e-mails e números de telefone privados extraídos dos dados de treinamento.

Toxicidade e Exploração

O processo de limpeza dos dados muitas vezes exige que humanos avaliem conteúdos tóxicos.

The New AI-Powered Bing Is Threatening Users.

Cleaning Up ChatGPT Takes Heavy Toll on Human Workers

Contractors in Kenya say they were traumatized by effort to screen out descriptions of violence and sexual abuse during run-up to OpenAI's hit chatbot

Desinformação Política

Geração de falsidades altamente persuasivas que podem afetar cenários globais e eleições.

Chatbots are generating false and misleading information about U.S. elections

Referências Bibliográficas I

- Anthropic. 2025. Release notes: System prompts. https://docs.anthropic.com/en/release-notes/system-prompts.
- Bellegarda, J. R. 1997. A latent semantic analysis framework for large-span language modeling. EUROSPEECH.
- Bengio, Y., R. Ducharme, and P. Vincent. 2000. A neural probabilistic language model. NeurIPS.
- Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. JMLR, 3:1137–1155.
- Bengio, Y., H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. 2006. Neural probabilistic language models. In Innovations in Machine Learning, 137–186. Springer.
- Bickmore, T. W., et al. 2018. Patient and consumer safety risks when using conversational assistants for medical information. Journal of Medical Internet Research, 20(9):e11510.
- Blodgett, S. L., S. Barocas, H. Daumé III, and H. Wallach. 2020. Language (technology) is power: A critical survey of “bias” in NLP. ACL.
- Bommasani, R., et al. 2021. On the opportunities and risks of foundation models. ArXiv.
- Brown, T., et al. 2020. Language models are few-shot learners. NeurIPS, 33.
- Carlini, N., et al. 2021. Extracting training data from large language models. USENIX Security 21.

Referências Bibliográficas II

- Cheng, M., E. Durmus, and D. Jurafsky. 2023. Marked personas: Using natural language prompts to measure stereotypes in language models. ACL.
- Cheng, M., et al. 2025. Sycophantic ai decreases prosocial intentions and promotes dependence. ArXiv preprint.
- Coccaro, N. and D. Jurafsky. 1998. Towards better integration of semantic predictors in statistical language modeling. ICSLP.
- Dai, A. M. and Q. V. Le. 2015. Semi-supervised sequence learning. NeurIPS.
- Deerwester, S. C., et al. 1988. Computer information retrieval using latent semantic structure: US Patent 4,839,853.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT.
- Dinan, E., et al. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. EMNLP.
- Dinan, E., et al. 2021. Anticipating safety issues in e2e conversational ai: Framework and tooling. ArXiv.
- Dodge, J., et al. 2019. Show your work: Improved reporting of experimental results. EMNLP.

Referências Bibliográficas III

- Dodge, J., et al. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. EMNLP.
- Ethayarajh, K. and D. Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. EMNLP.
- Friedman, B., D. G. Hendry, and A. Borning. 2017. A survey of value sensitive design methods. Foundations and Trends in Human-Computer Interaction.
- Friedman, B. and D. G. Hendry. 2019. Value Sensitive Design: Shaping Technology with Moral Imagination. MIT Press.
- Gao, L., et al. 2020. The Pile: An 800GB dataset of diverse text for language modeling. ArXiv.
- Gehman, S., S. Gururangan, M. Sap, Y. Choi, and N. A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. Findings of EMNLP.
- Gururangan, S., et al. 2020. Don't stop pretraining: Adapt language models to domains and tasks. ACL.
- Hashimoto, T., M. Srivastava, H. Namkoong, and P. Liang. 2018. Fairness without demographics in repeated loss minimization. ICML.
- Heafield, K. 2011. KenLM: Faster and smaller language model queries. Workshop on Statistical Machine Translation.

Referências Bibliográficas IV

- Henderson, P., et al. 2017. Ethical challenges in data-driven dialogue systems. AAAI/ACM AI Ethics and Society Conference.
- Henderson, P., et al. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. JMLR.
- Henderson, P., et al. 2023. Foundation models and fair use. JMLR, 24(400):1–79.
- Hofmann, V., P. R. Kalluri, D. Jurafsky, and S. King. 2024. Ai generates covertly racist decisions about people based on their dialect. Nature, 633(8028):147–154.
- Ischen, C., et al. 2019. Privacy concerns in chatbot interactions. International Workshop on Chatbot Research and Design.
- Jelinek, F., R. L. Mercer, and L. R. Bahl. 1975. Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE Transactions on Information Theory.
- Khattab, O., et al. 2024. DSPy: Compiling declarative language model calls into self-improving pipelines. ICLR.
- Kiela, D., et al. 2021. Dynabench: Rethinking benchmarking in NLP. NAACL HLT.
- Landauer, T. K., D. Laham, B. Rehder, and M. E. Schreiner. 1997. How well can passage meaning be derived without using word order? COGSCI.

Referências Bibliográficas V

- Liang, P., et al. 2023. Holistic evaluation of language models. Transactions on Machine Learning Research.
- Liu, H., et al. 2020. Does gender matter? Towards fairness in dialogue systems. COLING.
- Llama Team. 2024. The llama 3 herd of models.
- Longpre, S., et al. 2024a. Consent in crisis: The rapid decline of the ai data commons. ArXiv preprint.
- Longpre, S., et al. 2024b. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. NAACL HLT.
- McGuffie, K. and A. Newhouse. 2020. The radicalization risks of GPT-3 and advanced neural language models. ArXiv.
- Mikolov, T., et al. 2010. Recurrent neural network based language model. INTERSPEECH.
- Mikolov, T., et al. 2011. Extensions of recurrent neural network language model. ICASSP.
- Mikolov, T., K. Chen, G. S. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. ICLR.
- Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. NeurIPS.

Referências Bibliográficas VI

- Miller, G. A. and J. A. Selfridge. 1950. Verbal context and the recall of meaningful material. American Journal of Psychology, 63:176–185.
- Min, S., et al. 2022. Rethinking the role of demonstrations: What makes in-context learning work? EMNLP.
- Nadeem, M., A. Bethke, and S. Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. ACL.
- Neff, G. and P. Nagy. 2016. Talking to bots: Symbiotic agency and the case of Tay. International Journal of Communication.
- Parrish, A., et al. 2022. BBQ: A hand-built bias benchmark for question answering. Findings of ACL 2022.
- Peters, M., et al. 2018. Deep contextualized word representations. NAACL HLT.
- Radford, A., et al. 2019. Language models are unsupervised multitask learners. OpenAI tech report.
- Raffel, C., et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR, 21(140):1–67.
- Rawls, J. 2001. Justice as fairness: A restatement. Harvard University Press.
- Reeves, B. and C. Nass. 1996. The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places. Cambridge University Press.

Referências Bibliográficas VII

- Rosenfeld, R. 1992. Adaptive Statistical Language Modeling: A Maximum Entropy Approach. Ph.D. thesis, Carnegie Mellon University.
- Rosenfeld, R. 1996. A maximum entropy approach to adaptive statistical language modeling. Computer Speech and Language, 10:187–228.
- Sagawa, S., P. W. Koh, T. B. Hashimoto, and P. Liang. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. ICLR.
- Shannon, C. E. 1948. A mathematical theory of communication. Bell System Technical Journal, 27(3):379–423.
- Sheng, E., K.-W. Chang, P. Natarajan, and N. Peng. 2019. The woman worked as a babysitter: On biases in language generation. EMNLP.
- Soldaini, L., et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. ArXiv.
- Stolcke, A. 2002. SRILM – an extensible language modeling toolkit. ICSLP.
- Strubell, E., A. Ganesh, and A. McCallum. 2019. Energy and policy considerations for deep learning in NLP. ACL.
- Vaswani, A., et al. 2017. Attention is all you need. NeurIPS.

Referências Bibliográficas VIII

- Webson, A. and E. Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? NAACL HLT.
- Weizenbaum, J. 1966. ELIZA – A computer program for the study of natural language communication between man and machine. CACM.
- Wolf, M. J., K. W. Miller, and F. S. Grodzinsky. 2017. Why we should have seen that coming: Comments on Microsoft's Tay "experiment," and wider implications. The ORBIT Journal.
- Xu, J., et al. 2020. Recipes for safety in open-domain chatbots. ArXiv.
- Xu, A., et al. 2021. Detoxifying language models risks marginalizing minority voices. NAACL HLT.
- Zao-Sanders, M. 2025. How People Are Really Using Gen AI in 2025 — hbr.org. `https://hbr.org/2025/04/how-people-are-really-using-gen-ai-in-2025`.
- Zhang, A. K., et al. 2025. Language model developers should report train-test overlap. ICML.

OBRIGADO