Engenharia de Prompts para Ciência de Dados

Instituto InfNet

Institucional

Reitor

Prof. Eduardo Ramos
eduardo.ramos@infnet.edu.br

Coordenador do Curso

Prof. Fernando Ferreira
fernando.gferreira@prof.infnet.edu.br

Institucional

https://www.linkedin.com/in/tciodaro/

DSc. Thiago Ciodaro Xavier

Pós doutor em engenharia elétrica, ênfase em inteligência computacional pela COPPE/UFRJ. Mais de 10 anos de experiência em desenvolvimento de projetos de IA em pesquisa e consultorias (MJV, EY).

Líder de pesquisa em IA pelo NetLAB-UFRJ e professor de **advanced analytics** do Instituto Infnet.

Como vai ser o curso?

- Total de **9 etapas** e **18 aulas**
- Cada **aula** possui **1.5h**
- Total de **2 aulas** por **semana**
- Estudo extra por **semana**
  - Leitura de **capítulos da bibliografia** ou;
  - Implementação de **tutoriais**
- **Assessment**
  - Desenvolvimento de **trabalho prático** com os conhecimentos adquiridos no curso.

- **Competências**
  - Explicar o que é Inteligência Artificial Generativa e os Modelos Grandes de Linguagem (LLMs)
  - Gerar textos a partir de técnicas com LLMs usando Prompt Engineering
  - Utilizar técnicas avançadas de Prompt Engineering
  - Criar soluções a partir de Prompt Engineering
  - Utilizar técnicas Prompt Engineering para gerar imagens

Setup do Ambiente

- **Anaconda**: distribuição científica para Python
- **Git**: controle e versionamento de códigos
- **JupyterLab:** desenvolvimento de notebooks
- **Streamlit**: desenvolvimento de apps
- **LangChain**: desenvolvimento de aplicações com LLM
- **HuggingFace**: comunidade de LLMS

ETAPA 1

Introdução à inteligência artificial generativa e modelos de linguagem grande para Ciência de Dados

O que é Inteligência Artificial Generativa

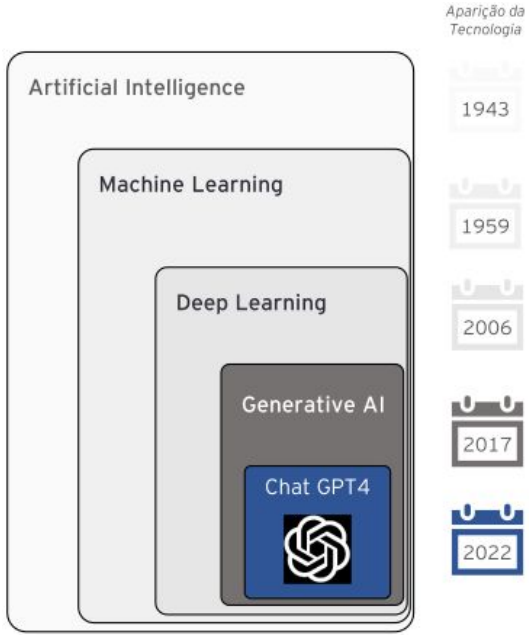
O que é Inteligência Artificial Generativa

Criatividade
Mensagens de textos criadas segundo as características especificadas pelo usuário, podendo resumir arquivos e artigos em tópicos. Flexibilidade para caracterizar o texto desejado para uso geral em diversas aplicações.

Imagens
Mensagens de textos criadas segundo as características especificadas pelo usuário, podendo resumir arquivos e artigos em tópicos. Flexibilidade para caracterizar o texto desejado para uso geral em diversas aplicações.

Contextos
Elaboração de textos baseados em bibliografia específica de textos e imagens, contextualizando, assim, perguntas, assuntos e tópicos que serão apresentados a diferentes públicos.

Moderação
Análise de sentimento sobre as respostas e textos gerados segundo as políticas de conteúdo da openAI, auxiliando na implantação de regras de governança e compliance.



1943: Warren McCulloch e Walter Pitts - "A Logical Calculus of the Ideas Immanent in Nervous Activity"

O artigo introduziu o conceito de neurônio artificial, um modelo computacional básico que lançou as bases para redes neurais. O modelo é conhecido por representar operações lógicas com circuitos neurais simples.

1958: Frank Rosenblatt - "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain"

Rosenblatt apresentou o Perceptron, um dos primeiros algoritmos a tentar aprendizado de máquina com base no modelo de neurônio de McCulloch-Pitts. O Perceptron demonstrou como as máquinas poderiam classificar entradas de forma linear, mas não conseguia resolver problemas não lineares, como o problema do XOR.

1969: Marvin Minsky e Seymour Papert - "Perceptrons"

O livro destacou as limitações do Perceptron em resolver problemas não lineares, como o problema do XOR, o que levou a uma diminuição do interesse por redes neurais, um período conhecido como "o inverno da IA".

1986: David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams - "Learning Representations by Back-propagating Errors"

Este artigo seminal introduziu o algoritmo de retropropagação (backpropagation), que permitiu o treinamento de redes neurais com múltiplas camadas, reativando o interesse pela IA e resolvendo o problema do XOR.

1998: Yann LeCun, Léon Bottou, Yoshua Bengio e Patrick Haffner - "Gradient-Based Learning Applied to Document Recognition"

Introduziu as Redes Neurais Convolucionais (CNNs) com a arquitetura LeNet, utilizada para reconhecimento de escrita manual, marcando um marco importante no aprendizado profundo para dados visuais.

2006: Geoffrey Hinton, Simon Osindero e Yee-Whye Teh - "A Fast Learning Algorithm for Deep Belief Nets"

Este artigo impulsionou a era moderna do aprendizado profundo ao demonstrar como o aprendizado não supervisionado poderia ser usado para treinar redes neurais profundas de forma eficiente.

2012: Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton - "ImageNet Classification with Deep Convolutional Neural Networks"

Introduziu a arquitetura AlexNet, que superou significativamente os métodos anteriores na competição ImageNet e demonstrou o poder das GPUs no treinamento de modelos de aprendizado profundo.

2014: Ian Goodfellow et al. - "Generative Adversarial Nets"

Propôs as Redes Adversárias Generativas (GANs), uma estrutura em que duas redes neurais (geradora e discriminadora) competem, levando à criação de dados sintéticos realistas.

2014: Dzmitry Bahdanau, Kyunghyun Cho e Yoshua Bengio - "Neural Machine Translation by Jointly Learning to Align and Translate"

Introduziu o mecanismo de atenção, permitindo que modelos se concentrem em partes relevantes da sequência de entrada, o que melhorou significativamente as tarefas de tradução automática.

2017: Ashish Vaswani et al. - "Attention is All You Need"

Introduziu a arquitetura Transformer, que usa mecanismos de auto-atenção e tornou-se a base para muitos dos modelos de última geração em PLN, incluindo o BERT e o GPT.

2018: Alec Radford et al. - "Improving Language Understanding by Generative Pre-Training (GPT-1)"

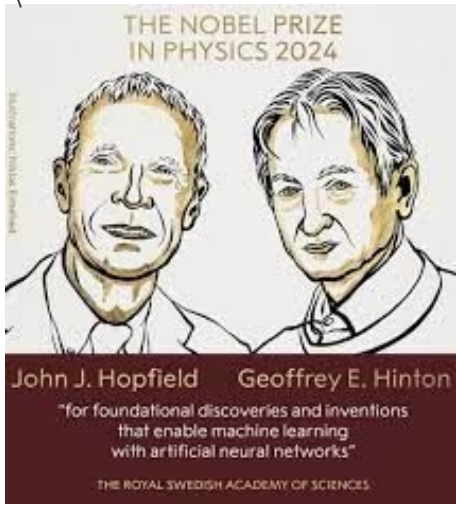
Apresentou a primeira versão do modelo Generative Pre-trained Transformer (GPT), estabelecendo uma nova abordagem para compreensão de linguagem não supervisionada.

2020: Tom B. Brown et al. - "Language Models are Few-Shot Learners (GPT-3)"

O GPT-3, um modelo com 175 bilhões de parâmetros, demonstrou o poder do aprendizado com poucos exemplos (few-shot learning), mostrando que grandes modelos de linguagem podem generalizar bem em várias tarefas de PLN sem treinamento específico para cada tarefa.

2023: OpenAI - Lançamento do GPT-4

O modelo GPT-4 apresentou desempenho aprimorado na compreensão de instruções complexas, geração de texto coerente e suporte para entradas multimodais (texto e imagem).



O que é Inteligência Artificial Generativa

BENEFÍCIOS

Criação de Conteúdo: A IA generativa facilita a produção de textos, imagens, vídeos e músicas de maneira rápida e escalável.

Automação Criativa: Permite que empresas e indivíduos automatizem processos criativos, otimizando tempo e recursos.

Prototipagem Rápida: Acelera o desenvolvimento de ideias, como design de produtos e narrativas, antes de implementar versões finais.

Personalização: Gera conteúdo adaptado às necessidades e preferências dos usuários, oferecendo experiências únicas e mais engajantes.

Exploração de Novos Domínios: Auxilia na descoberta de novos padrões e tendências em áreas como arte, ciência e negócios.

O que é Inteligência Artificial Generativa

DESAFIOS

Bias e Preconceitos: Algoritmos podem reproduzir e amplificar vieses existentes, resultando em conteúdo tendencioso ou discriminatório.

Uso Indevido: Pode ser utilizado para criar deepfakes ou disseminar informações falsas e manipulação digital.

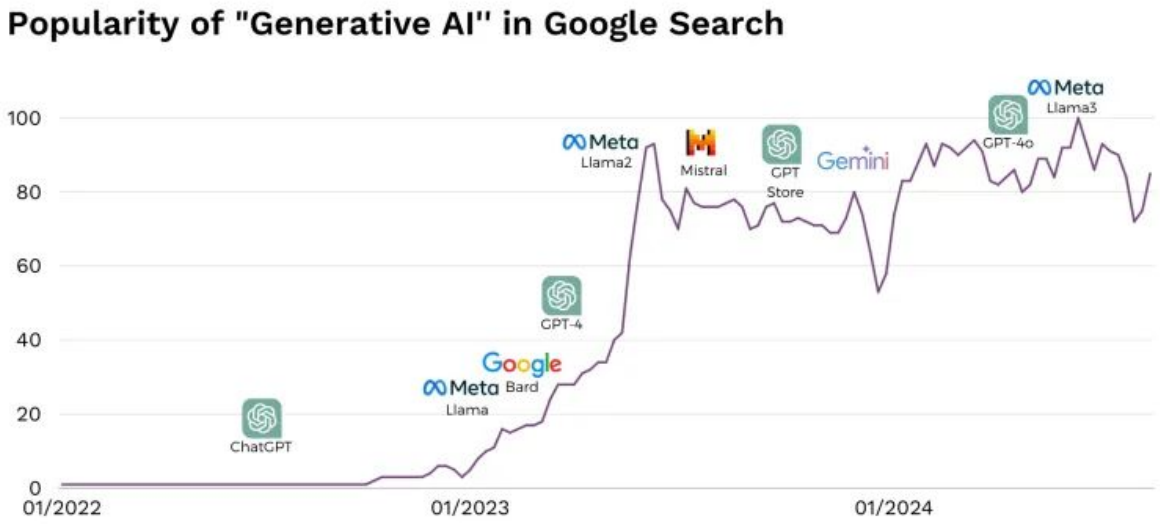
Impacto no Trabalho: Potencial substituição de certas funções e profissões criativas.

Controle e Transparência: Falta de clareza sobre como as IAs tomam decisões e geram conteúdo.

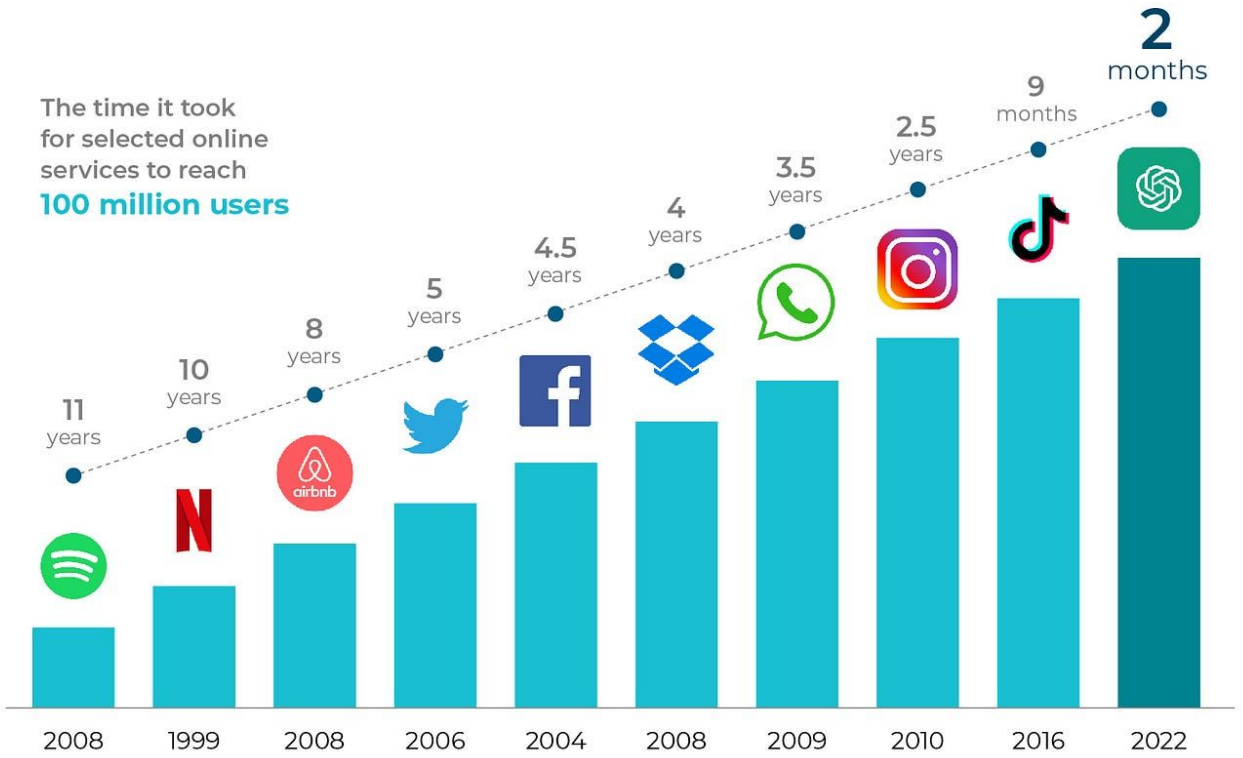
Privacidade: Risco de utilização inadequada de dados pessoais para treinamento de modelos.

O que é Inteligência Artificial Generativa

O interesse por IA generativa explodiu desde outubro de 2022, graças ao lançamento do ChatGPT. Além disso, a Gartner prevê que, até 2025, o percentual de dados gerados por IA generativa representará 10% de todos os dados gerados.



Chat-GPT sprints to 100 million users



Source: World of Statistics

Modelos grandes de linguagem (LLMs)

LFM (Large Foundation Models):

- Modelos de fundação que servem como base para várias tarefas de aprendizado de máquina e inteligência artificial.
- São treinados com grandes quantidades de dados de forma genérica e podem ser adaptados (finetuning) para resolver problemas específicos.
- **Exemplos**: modelos multimodais que entendem texto, imagem, e até áudio simultaneamente.

LLM (Large Language Models):

- Subconjunto dos LFMs especializado em Processamento de Linguagem Natural (NLP).
- Projetados para lidar com texto e gerar linguagem natural.
- Capazes de compreender, gerar e interagir em linguagem humana.
- **Exemplos**: GPT-3, GPT-4, BERT, T5.

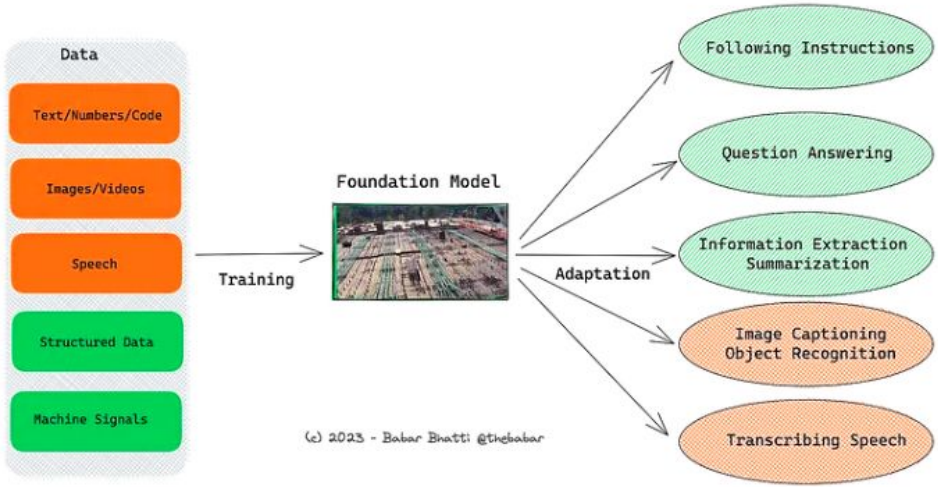
Modelos grandes de linguagem (LLMs)

LFM (Large Foundation Models):

- Treinado para múltiplas modalidades e propósitos.
- Capaz de ser utilizado como um ponto de partida para diversas aplicações (visão computacional, NLP, aprendizado multimodal).
- Flexível e genérico.

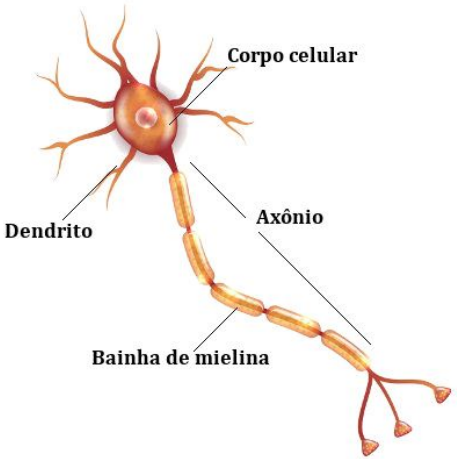
LLM (Large Language Models):

- Focado exclusivamente em texto.
- Excelente em compreensão e geração de linguagem natural.
- Pode ser utilizado como um componente de um LFM em tarefas específicas de linguagem.



Modelos grandes de linguagem (LLMs)

Os Modelos de Linguagem Grande (LLMs) são redes neurais profundas especializadas na compreensão e geração da linguagem humana, fundamentais em áreas como criação de conteúdo e Processamento de Linguagem Natural (PLN), onde o objetivo é desenvolver algoritmos capazes de entender e gerar texto em linguagem natural. Esses modelos são treinados principalmente através de aprendizado não supervisionado em grandes quantidades de dados textuais, permitindo-lhes aprender padrões e estruturas linguísticas de forma eficaz.



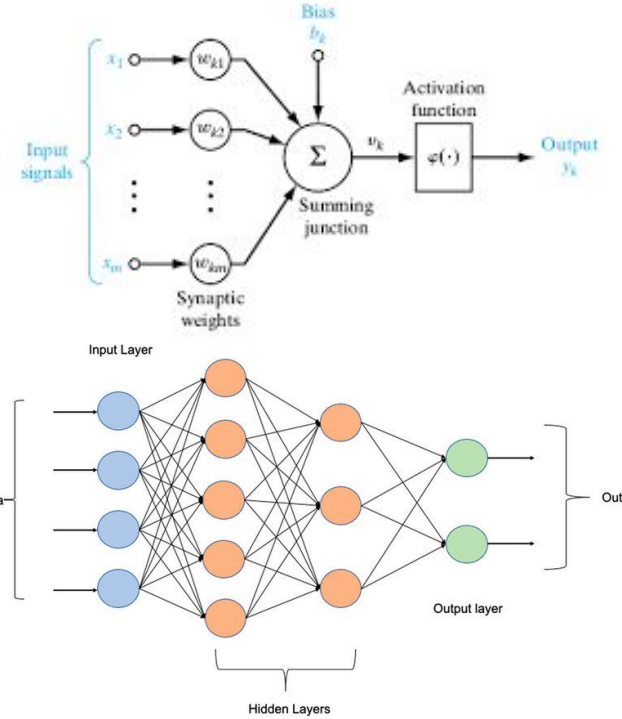
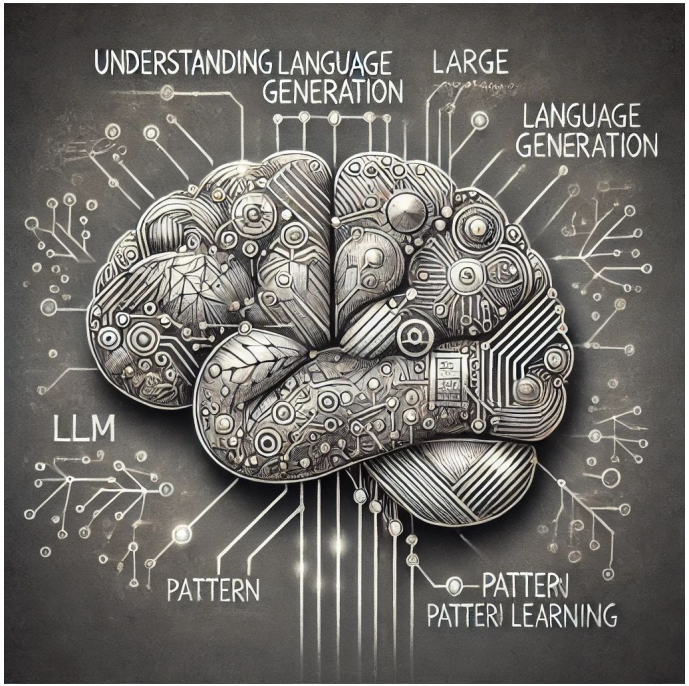


Ilustração: Brasil Escola. O que é um Neurônio.

Modelos grandes de linguagem (LLMs)

A atual geração de LLMs, como o GPT-4, utiliza arquiteturas de rede neural baseadas no modelo de transformadores. A força notável desses modelos reside na sua capacidade de funcionar como interfaces conversacionais, como chatbots, gerando respostas coerentes e contextualmente apropriadas em conversas abertas.

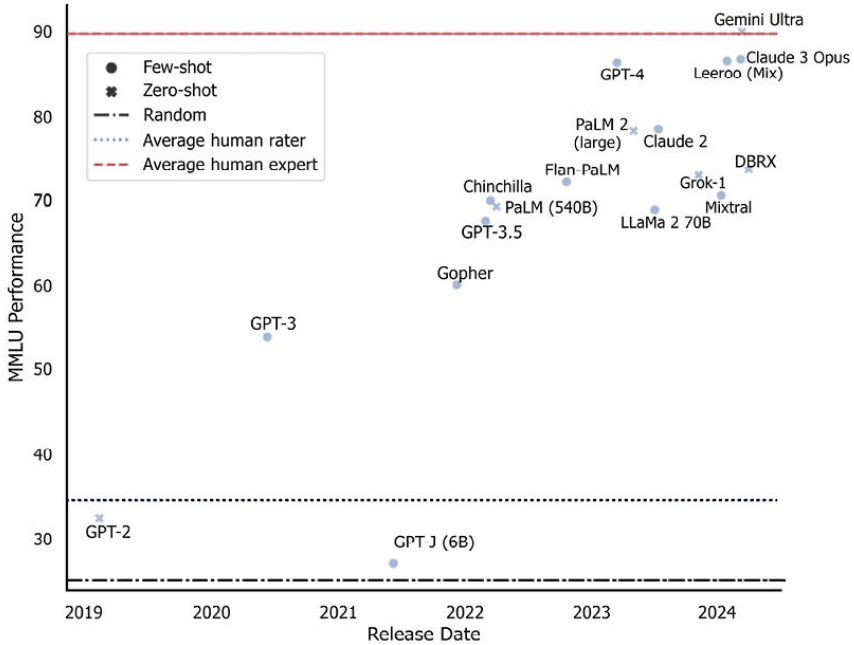
Este avanço na modelagem de linguagem e no PLN é amplamente dependente da qualidade do aprendizado de representações, onde o modelo codifica informações sobre os textos nos quais foi treinado e gera novos textos baseando-se no que aprendeu.



Modelos grandes de linguagem (LLMs)

Avaliação de LLMs

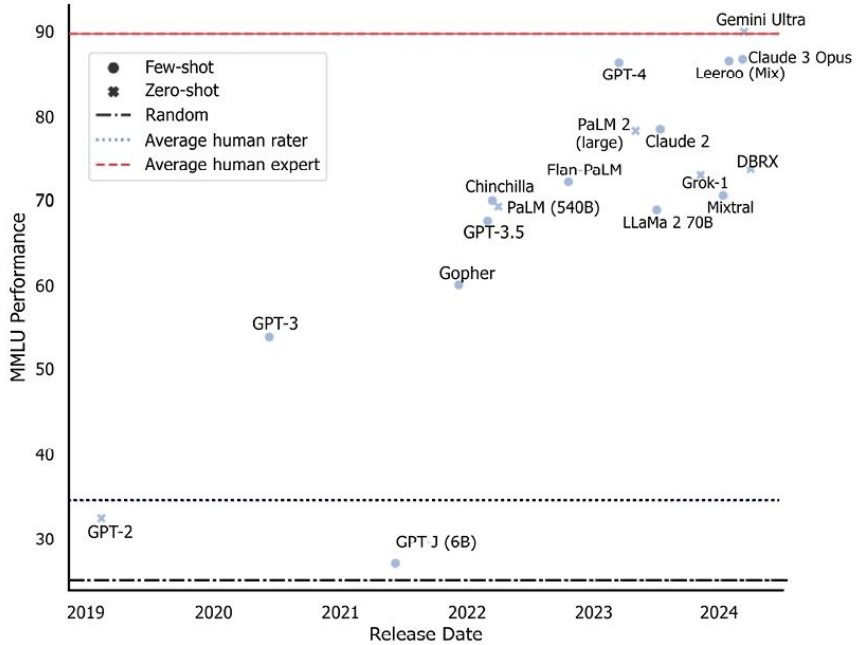
- **Importância dos Benchmarks:**
  - Os benchmarks que capturam o desempenho em tarefas de diferentes domínios são fundamentais para o desenvolvimento de LLMs.
  - Eles oferecem uma maneira padronizada de avaliar o desempenho multitarefa e as capacidades amplas dos LLMs.



Modelos grandes de linguagem (LLMs)

O que é MMLU?

- **Massive Multitask Language Understanding (MMLU):**
  - É um conjunto abrangente composto por 57 tarefas que abrangem diversos domínios, como matemática, história, ciência da computação e direito.
  - Avalia LLMs em configurações de zero-shot (sem exemplos prévios) e few-shot (com poucos exemplos).



Modelos grandes de linguagem (LLMs)

Exemplos de bases de dados públicas para MMLU

GLUE (General Language Understanding Evaluation)

- GLUE é um benchmark composto por uma coleção de recursos de dados que testam as capacidades de compreensão de linguagem de um modelo em múltiplas tarefas, como análise de sentimento e inferência textual.

SuperGLUE

- SuperGLUE foi projetado como um benchmark mais difícil e abrangente que o GLUE original, incluindo tarefas mais desafiadoras como o raciocínio de causa e efeito, compreensão de leitura mais complexa e raciocínio lógico.

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

CoLA

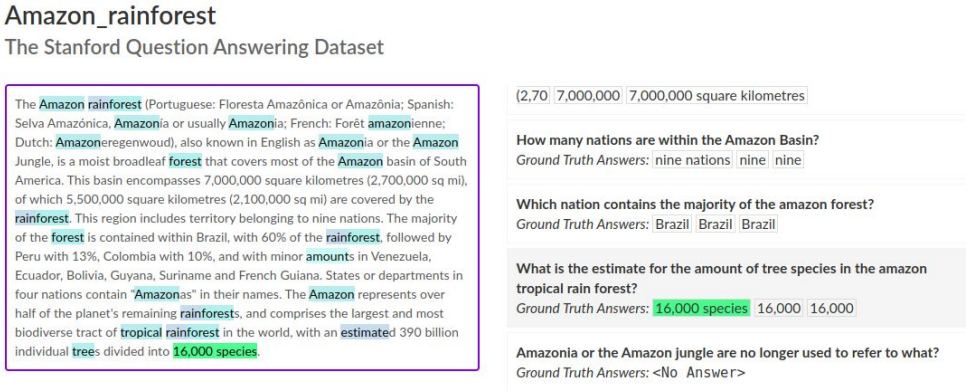
Label	Sentence
*	The more books I ask to whom he will give, the more he reads.
✓	I said that my father, he was tight as a hoot-owl.
✓	The jeweller inscribed the ring with the name.
*	many evidence was provided.
✓	They can sing.
✓	The men would have been all working.
*	Who do you think that will question Seamus first?
*	Usually, any lion is majestic.
✓	The gardener planted roses in the garden.
✓	I wrote Blair a letter, but I tore it up before I sent it.

Modelos grandes de linguagem (LLMs)

Exemplos de bases de dados públicas para MMLU

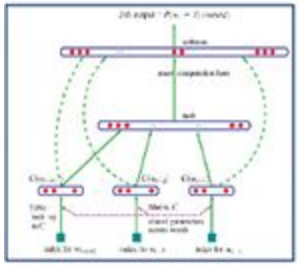
SQuAD (Stanford Question Answering Dataset)

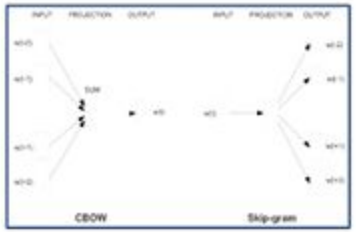
- SQuAD é um conjunto de dados popular para avaliar o desempenho de modelos de compreensão de leitura. Contém perguntas feitas por humanos baseadas em um conjunto de artigos da Wikipedia, onde o modelo deve fornecer a resposta correta extraída do texto.



https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Amazon_rainforest.html

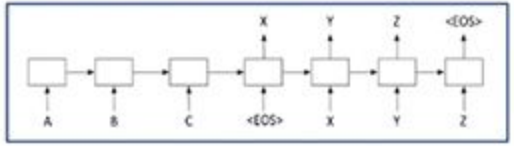
2001
Neural Language Models

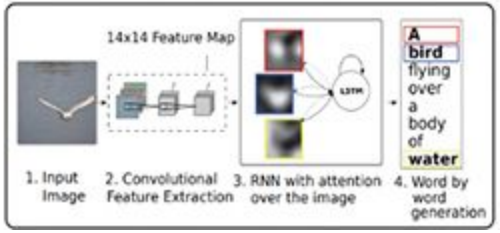




2013
Encoding Semantic Meaning with Word2vec

2014–2017
Seq2seq + Attention





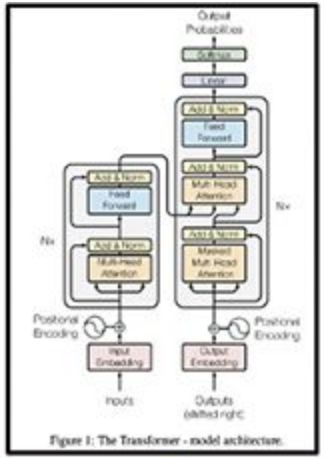


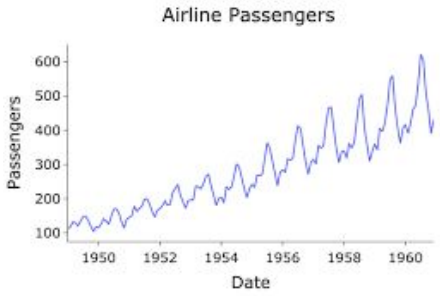
Figure 1: The Transformer - model architecture.

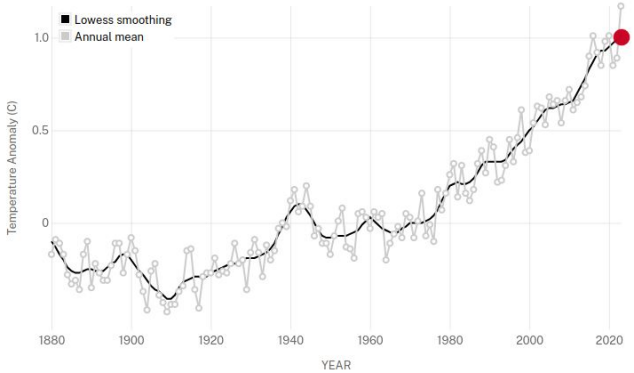
2017–Present
Transformers + Large Language Models

Processo de pré-treinamento de LLMs.

Modelos Auto-regressivos (AR): são usados para modelar e prever sequências de dados onde valores futuros são assumidos como funções lineares dos valores anteriores na série. Exemplos comuns incluem previsões de demanda de produtos, meteorológicas e preços de ações.

Aplicação em Linguagem: Em linguagem, modelos auto-regressivos preveem a próxima palavra ou token com base nas palavras anteriores na sequência de texto.



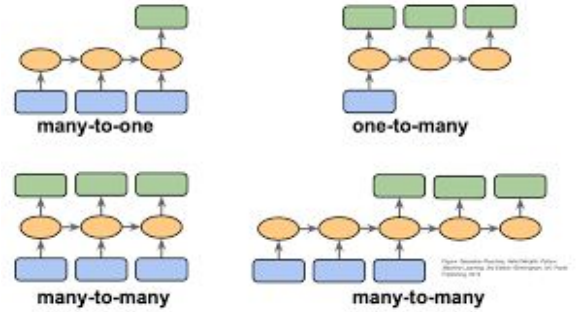


$$y_t = c + \sum_{n=1}^{p} \alpha_n y_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{P} \phi_n y_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

Processo de pré-treinamento de LLMs.

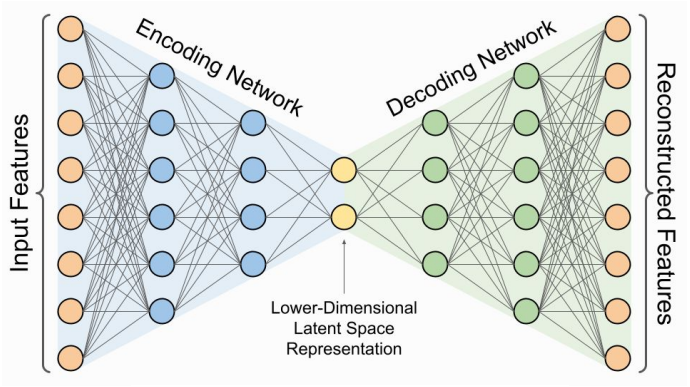
Modelos Sequenciais: processam e geram dados considerando a ordem em que os dados aparecem. São amplamente usados em tarefas que envolvem dependências temporais ou sequenciais como reconhecimento de fala, tradução automática e geração de texto.

Aplicações em IA Generativa: Modelos sequenciais são fundamentais para IA generativa, onde a criação de conteúdo novo depende das informações precedentes.

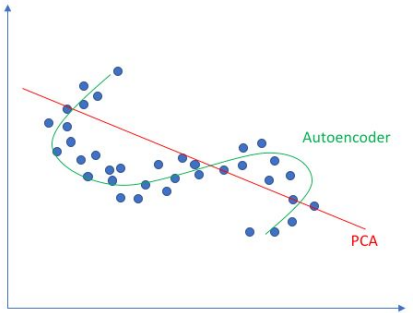


Processo de pré-treinamento de LLMs.

Modelos Autoencoders: são uma classe de redes neurais usadas para aprendizado não supervisionado. Eles aprendem a codificar os dados de entrada em uma representação compacta e depois decodificar essa representação de volta ao formato original. O **Encoder** comprime os dados de entrada em um espaço latente menor (codificado), cuja função é capturar as características essenciais dos dados, enquanto o **Decoder** reconstrói os dados a partir da representação comprimida.



Linear vs nonlinear dimensionality reduction



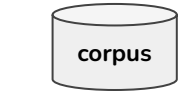
https://medium.com/@etorezone/differences-between-autoencoders-and-principal-component-analysis-pca-in-dimensionality-reduction-ca5f24364054
https://www.researchgate.net/figure/Comparison-between-PCA-and-Autoencoder-15_fig1_340049776

Processo de pré-treinamento de LLMs.

Codificando texto

Corpus: um grande conjunto de textos que será utilizado como fonte de dados para análise ou treinamento de modelos de linguagem. Um corpus pode ser formado por milhares de artigos de notícias, livros ou conversas de redes sociais. Serve como base para a construção do vocabulário e a compreensão das relações entre palavras.

Vocabulário: coleção de todas as palavras distintas presentes no corpus. Define quais palavras serão reconhecidas e codificadas pelo modelo.



Machado de Assis

```
['Memórias',
 'Póstumas',
 'Brás',
 'Cubas',
 'Ao',
 'verme',
 'primeiro',
 'roeu',
 'frias',
```

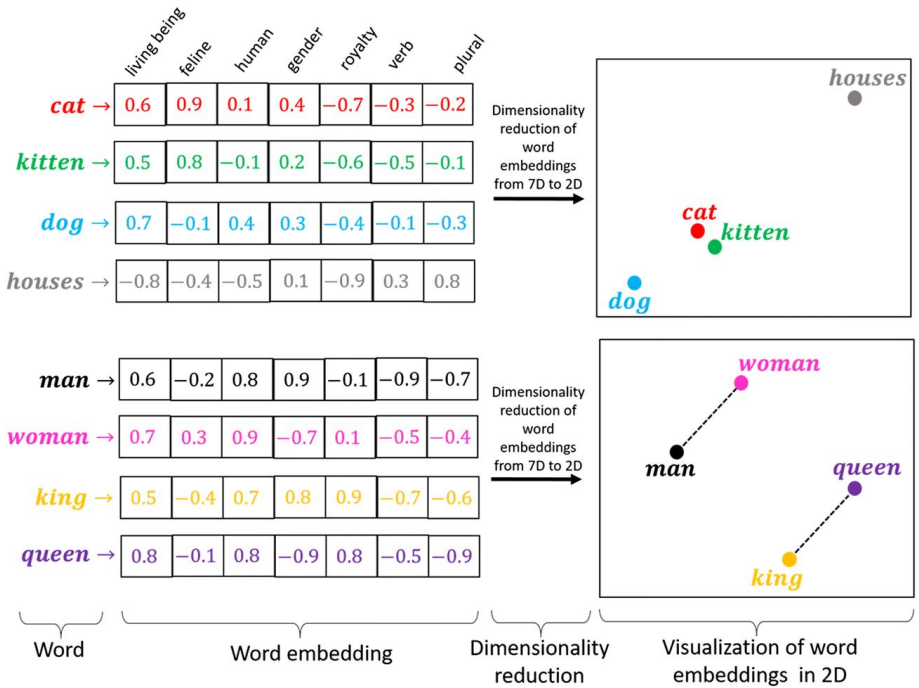
```
model.wv['amigo'].tolist()

[-1.9068045616149902,
 -3.3655452728271484,
 -3.0124053955078125,
 -2.5305261611938477,
 1.531314492225647,
 2.3493525981903076,
 -1.007277250289917,
 -2.291379690170288,
```

Processo de pré-treinamento de LLMs.

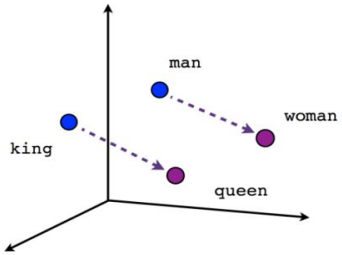
Word2Vec é um modelo de aprendizado de palavras que transforma cada palavra em um vetor de números em um espaço vetorial contínuo. Ele utiliza redes neurais para aprender representações distribuídas de palavras com base em seu contexto.

Ideia Principal: Palavras com significados semelhantes estão mais próximas no espaço vetorial. Por exemplo, "gato" e "cachorro" terão representações vetoriais mais próximas do que "gato" e "carro".

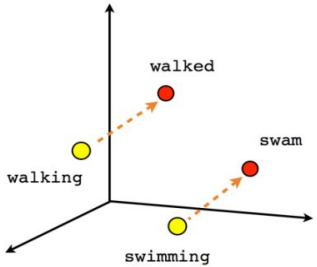


https://swatimeena989.medium.com/training-word2vec-using-gensim-14433890e8e4

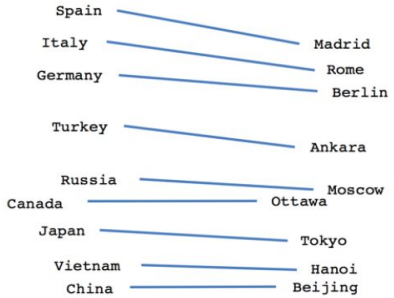
Processo de pré-treinamento de LLMs.



Male-Female



Verb tense



Country-Capital

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

MIKOLOV et al. **Distributed Representations of Words and Phrases and their Compositionality**.

Processo de pré-treinamento de LLMs.

Principais Parâmetros de Treinamento:

Vector Size: Determina o número de dimensões do vetor de palavras (embedding).

Window Size: Define quantas palavras à esquerda e à direita da palavra-alvo o modelo deve considerar para o contexto.

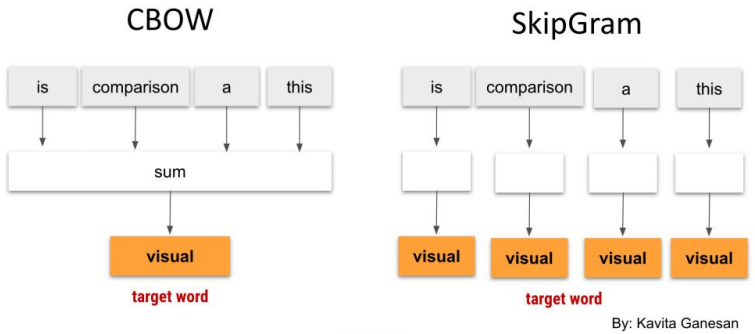
Min Count: Filtra palavras raras que aparecem menos vezes do que o limite definido.

SG (Skip-Gram ou CBOW):Escolhe o método de treinamento:

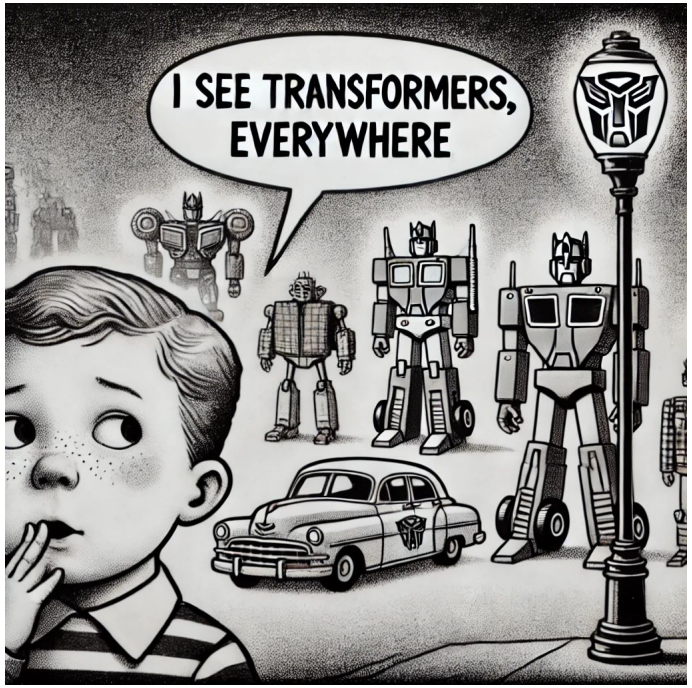
- Skip-Gram (prever contexto a partir da palavra-alvo).
- CBOW (Continuous Bag of Words, prever a palavra-alvo a partir do contexto).

Skip-gram é eficaz para capturar informações sobre palavras menos frequentes, pois ele treina diretamente as relações entre a palavra-alvo e cada palavra do contexto.

CBOW tende a ser mais eficiente em termos de treinamento e funciona bem para palavras mais frequentes, já que ele faz uma predição a partir de um conjunto de palavras em vez de fazer predições separadas para cada palavra de contexto.



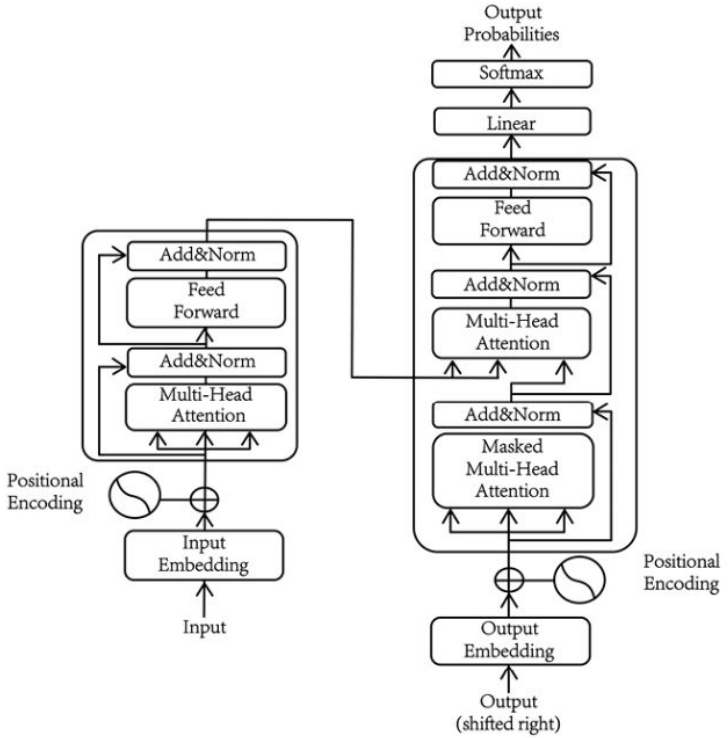
Processo de pré-treinamento de LLMs.



Processo de pré-treinamento de LLMs.

Transformers: são modelos neurais de arquitetura complexa, estruturados em um codificador e um decodificador capazes de identificar palavras segundo o seu contexto. Se diferenciam em relação a memória e entendimento de contexto devido a como cada arquitetura lida com o contexto sequencial e as dependências a longo prazo.

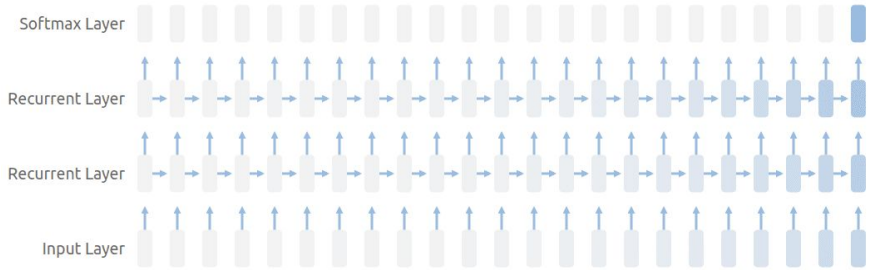
Enquanto o codificador aprende aspectos da linguagem e dos contextos das palavras, considerando os termos antes e depois, o decodificador aplica uma máscara nos termos para garantir que a amostra prevista dependa somente das amostras anteriores.



Processo de pré-treinamento de LLMs.

LSTMs têm uma dependência sequencial que, embora a informação de estados anteriores seja considerada para a previsão futura, longas sequências sofrem com problemas como *gradient vanishing*.

Diferente do LSTM, que processa a sequência passo a passo, o Transformer processa todas as palavras da sequência simultaneamente (paralelamente), o que melhora significativamente a eficiência e permite capturar melhor as relações entre palavras distantes.



Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the vanilla RNN unit.

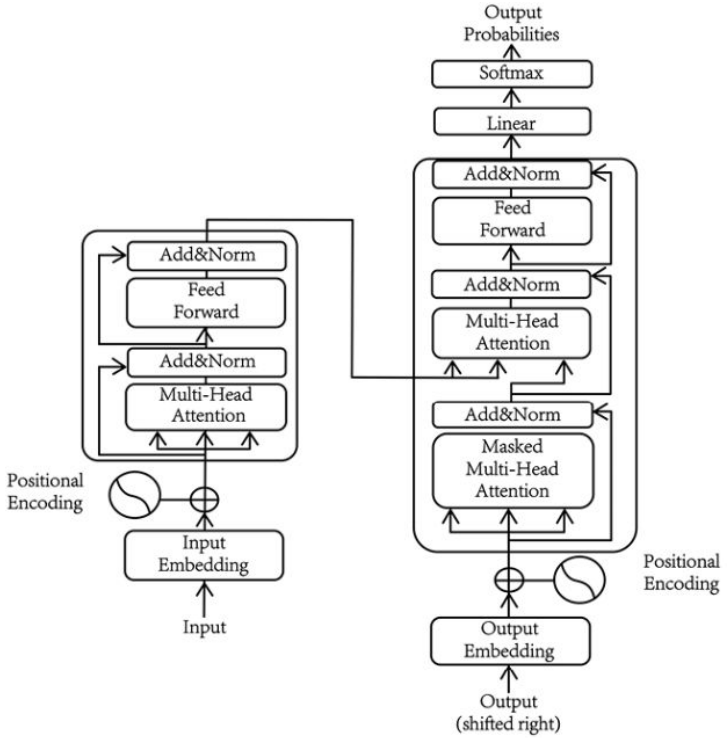
Processo de pré-treinamento de LLMs.

Codificação Posicional: como o transformer não processa a informação de forma sequencial, a posição dos termos na sequência é codificada no próprio embeddings.

Normalização por Camada: Estabiliza o aprendizado normalizando as entradas ao longo das características, melhorando a velocidade e a estabilidade do processo de otimização.

Atenção Multi-Head (MHA): Aplica o mecanismo de atenção várias vezes em paralelo, capturando diferentes tipos de informações para uma representação mais rica.

Mecanismo de Atenção: Calcula uma soma ponderada (vetor de contexto) dos valores nas posições de entrada, com base na similaridade, permitindo o foco seletivo em partes relevantes da entrada.



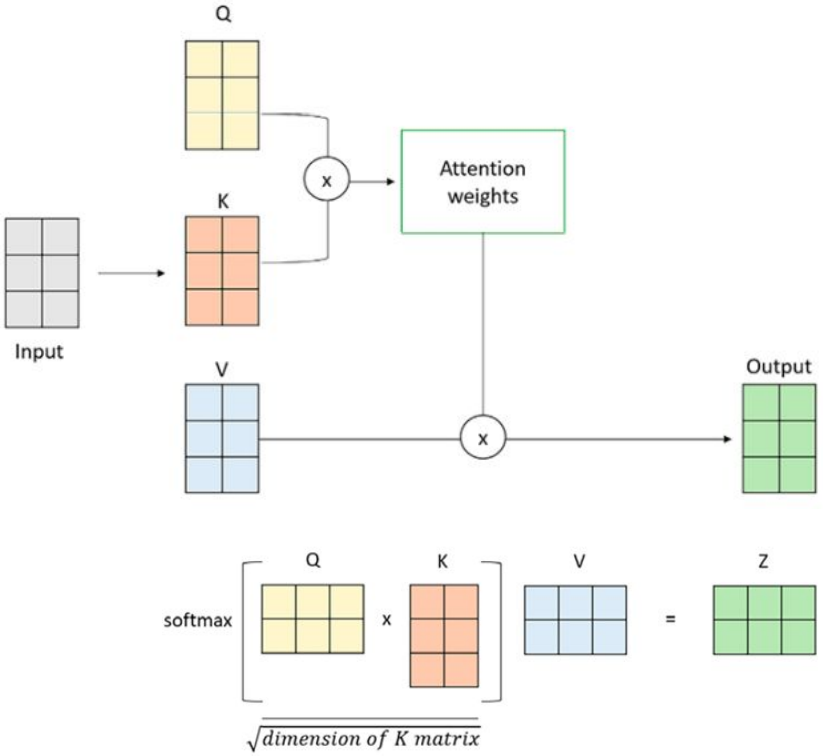
Processo de pré-treinamento de LLMs.

Mecanismo de atenção: permite que o modelo "preste atenção" em diferentes palavras (ou tokens) da sequência, dando mais ou menos peso a elas, dependendo de sua relevância para a tarefa em questão. Isso significa que o modelo pode identificar quais palavras ou partes do texto são mais importantes para entender o significado global ou para prever a próxima palavra.

Query (Q): Representa a palavra/tokens para a qual estamos calculando a atenção.

Key (K): Representa todas as palavras/tokens na sequência de entrada, que servirão como "chaves" para comparação com a query.

Value (V): Representa os valores das palavras/tokens que serão usados para calcular o "content vector", ou seja, as informações que o modelo vai usar de fato.

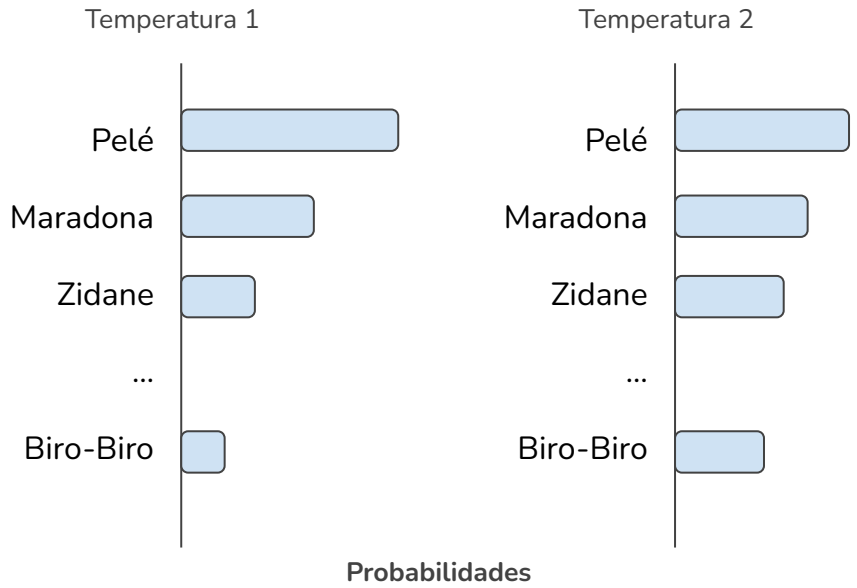


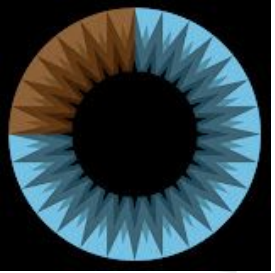
Processo de pré-treinamento de LLMs.

Saída do LLM: o modelo transformer estima a probabilidade de cada token dado uma sequência de tokens de entrada. A decisão de qual token utilizar é estocástica, impedindo que somente os tokens mais prováveis sejam selecionados.

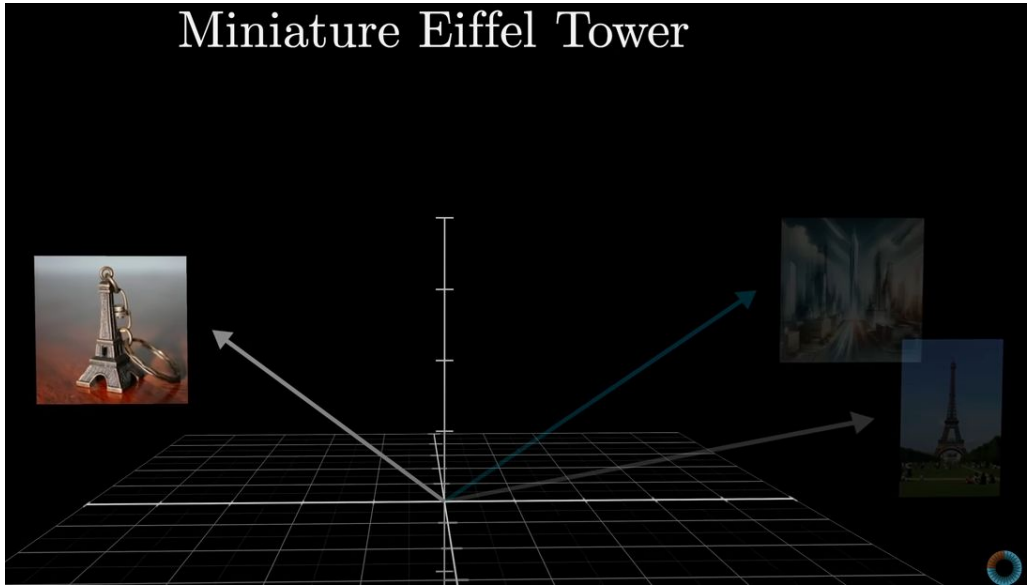
Temperatura: parâmetro que controla a estocasticidade da resposta. Quanto menor for a temperatura, mais determinística é a resposta, enquanto valores elevados auxiliam na criatividade do modelo, aumentando a chance de selecionarmos termos menos prováveis.

“O rei do futebol se chama:“

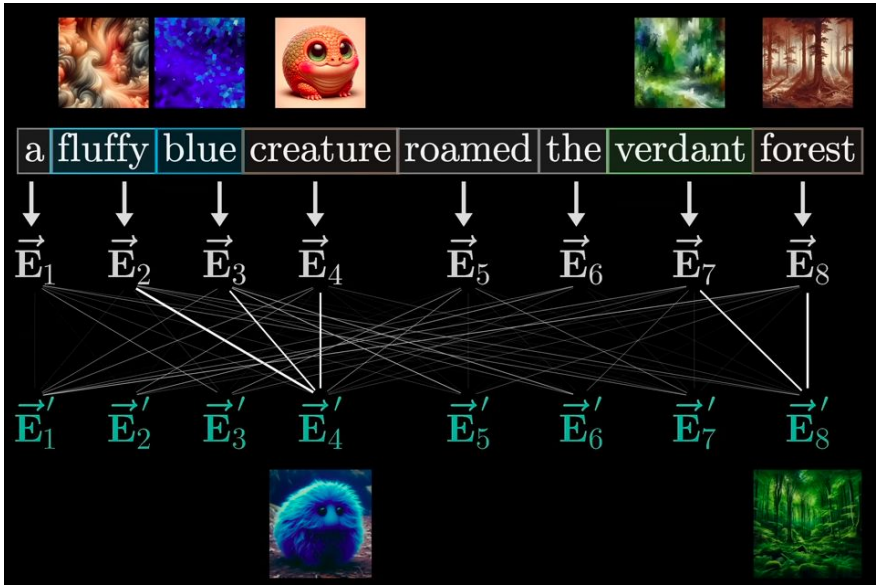




3Blue1Brown



https://www.youtube.com/watch?v=eMlx5fFNoYc



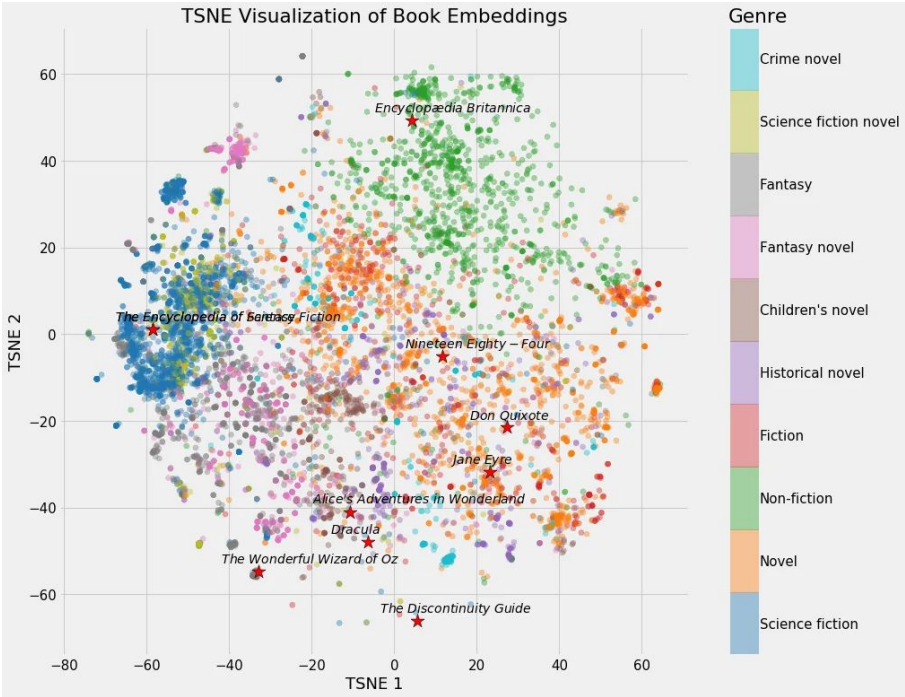
https://www.youtube.com/watch?v=wjZofJX0v4M

Embeddings em LLMs

Embeddings são representações vetoriais densas de palavras ou itens, usadas para capturar relações semânticas. Ao contrário de representações esparsas (one-hot), embeddings mapeiam palavras para um espaço de menor dimensão, preservando a similaridade semântica.

Palavras que aparecem em contextos semelhantes terão embeddings semelhantes, mesmo que não sejam sinônimos exatos. Esses modelos são capazes de identificar:

- Relações semânticas amplas (ex. “rei” e “rainha”),
- Hierarquias conceituais (ex. “fruta” e “maçã”),
- Dependências contextuais (ex. “corre” e “rápido”).



https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Embeddings em LLMs

Transferência de Conhecimento: Embeddings treinados em tarefas gerais podem ser ajustados para aplicações mais específicas através de fine-tuning. Um modelo genérico de linguagem pode ser adaptado para entender termos médicos ou jurídicos.

Contextos Específicos: Ao adaptar embeddings para contextos particulares, como análises financeiras ou diagnósticos médicos, as representações genéricas podem ser especializadas para focar em padrões e termos específicos dessas áreas.