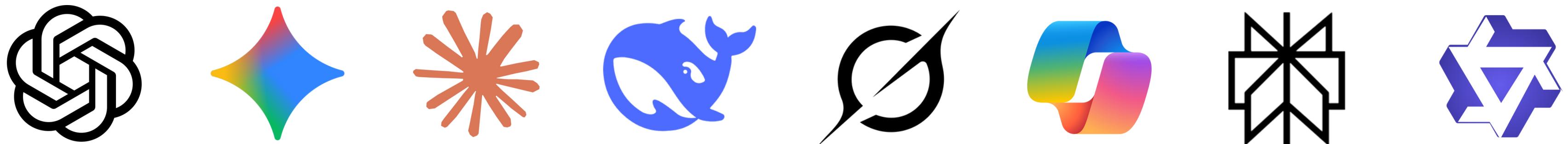


Large Language Models: A Survey

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu Richard Socher,
Xavier Amatriain, Jianfeng Gao



- Aluno: Maruan Biasi El Achkar
- Curso: Engenharia de Software
- Matéria: Aprendizagem de Máquina
- Professor: Claudinei Dias

O artigo

- Escrito por pesquisadores de IA.
- Feito para resumir o cenário atual de LLMs.

Large Language Models: A Survey

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu
Richard Socher, Xavier Amatriain, Jianfeng Gao

Abstract—Large Language Models (LLMs) have drawn a lot of attention due to their strong performance on a wide range of natural language tasks, since the release of ChatGPT in November 2022. LLMs' ability of general-purpose language understanding and generation is acquired by training billions of model's parameters on massive amounts of text data, as predicted by scaling laws [1], [2]. The research area of LLMs, while very recent, is evolving rapidly in many different ways. In this paper, we review some of the most prominent LLMs, including three popular LLM families (GPT, LLaMA, PaLM), and discuss their characteristics, contributions, and limitations. We also give an overview of techniques developed to build, and augment LLMs. We then survey popular datasets prepared for LLM training, fine-tuning, and evaluation, review widely used LLM evaluation metrics, and compare the performance of several popular LLMs on a set of representative benchmarks. Finally, we conclude the paper by discussing open challenges and future research directions.

I. INTRODUCTION

Language modeling is a long-standing research topic, dating back to the 1950s with Shannon's application of information theory to human language, where he measured how well simple n-gram language models predict or compress natural language text [3]. Since then, statistical language modeling became fundamental to many natural language understanding and generation tasks, ranging from speech recognition, machine translation, to information retrieval [4], [5], [6].

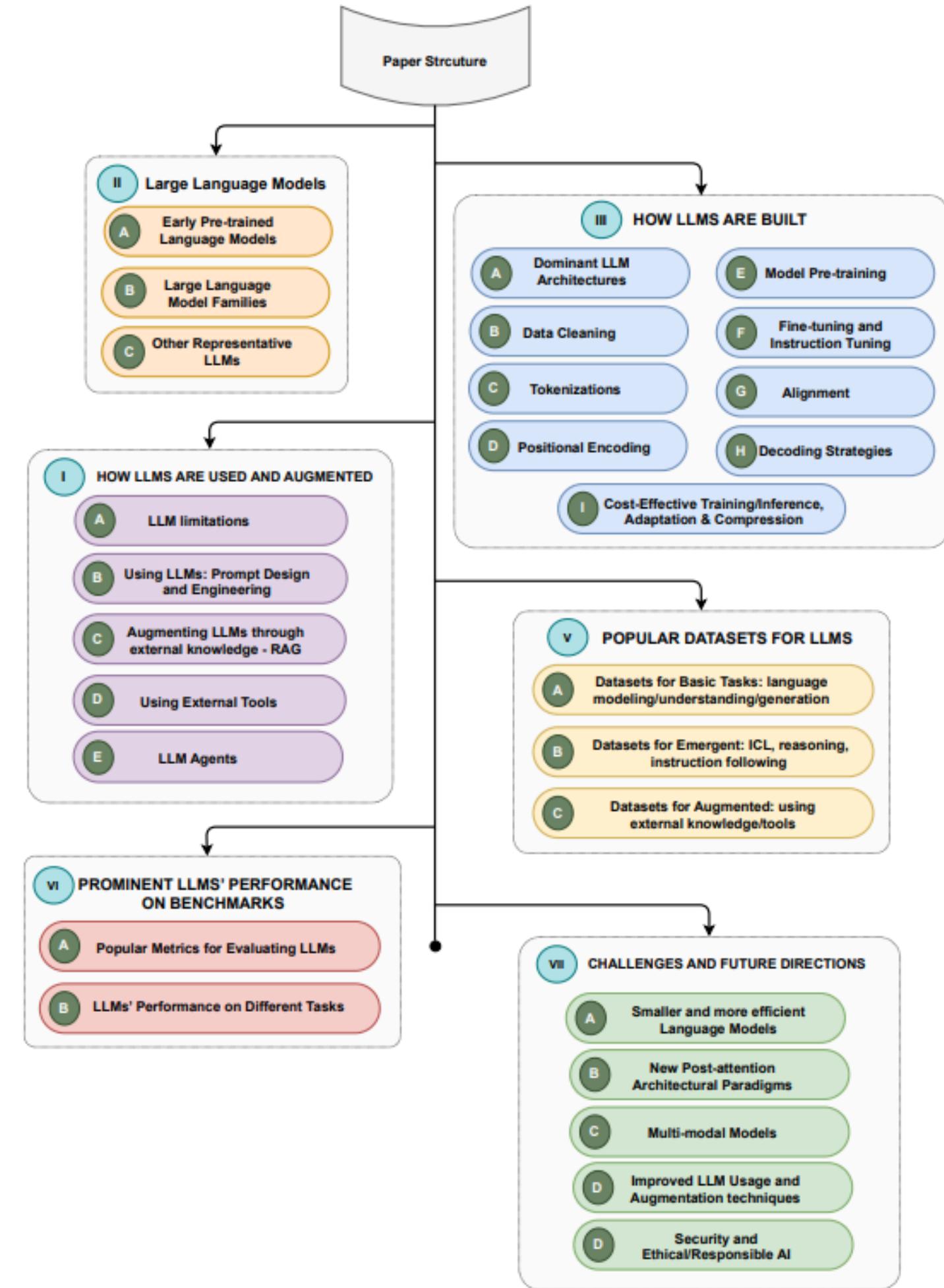
The recent advances on transformer-based large language models (LLMs), pretrained on Web-scale text corpora, significantly extended the capabilities of language models (LLMs). For example, OpenAI's ChatGPT and GPT-4 can be used not only for natural language processing, but also as general task solvers to power Microsoft's Co-Pilot systems, for instance, can follow human instructions of complex new tasks performing multi-step reasoning when needed. LLMs are thus becoming the basic building block for the development of general-purpose AI agents or artificial general intelligence (AGI).

that have different starting points and velocity: statistical language models, neural language models, pre-trained language models and LLMs.

Statistical language models (SLMs) view text as a sequence of words, and estimate the probability of text as the product of their word probabilities. The dominating form of SLMs are Markov chain models known as the n-gram models, which compute the probability of a word conditioned on its immediate proceeding $n - 1$ words. Since word probabilities are estimated using word and n-gram counts collected from text corpora, the model needs to deal with data sparsity (i.e., assigning zero probabilities to unseen words or n-grams) by using *smoothing*, where some probability mass of the model is reserved for unseen n-grams [12]. N-gram models are widely used in many NLP systems. However, these models are incomplete in that they cannot fully capture the diversity and variability of natural language due to data sparsity.

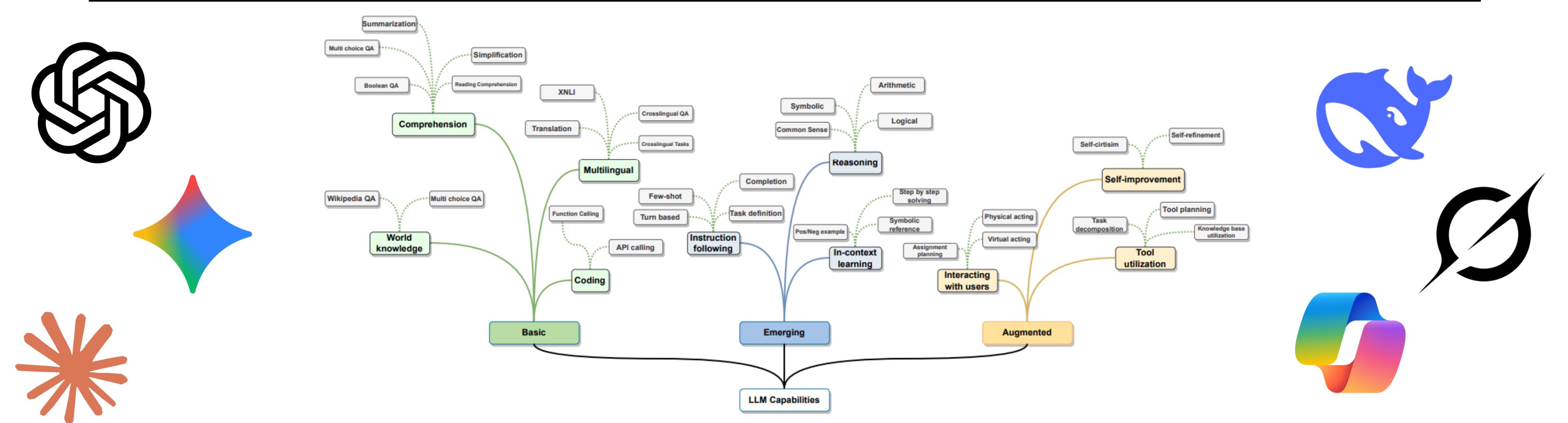
Early neural language models (NLMs) [13], [14], [15], [16] deal with data sparsity by mapping words to low-dimensional continuous vectors (embedding vectors) and predict the next word based on the aggregation of the embedding vectors of its proceeding words using neural networks. The embedding vectors learned by NLMs define a hidden space where the semantic similarity between vectors can be readily computed as their distance. This opens the door to computing semantic similarity of any two inputs regardless their forms (e.g., queries vs. documents in Web search [17], [18], sentences in different languages in machine translation [19], [20]) or modalities (e.g., image and text in image captioning [21], [22]). Early NLMs are task-specific models, in that they are trained on task-specific data and their learned hidden space is task-specific.

Pre-trained language models (PLMs), unlike early NLMs, are task-agnostic. This generality also extends to the learned hidden embedding space. The training and inference of PLMs follows the *pre-training and fine-tuning* paradigm, where language models with recurrent neural networks [23] or transformers [24], [25], [26] are pre-trained on Web-scale unlabeled



O que são LLMs

- LLM = Large Language Model
- São **modelos de IA gigantes** baseados em Transformers.
- Treinados em textos.
- Conseguem conversar, responder perguntas, escrever e até programar.
- Exemplos: ChatGPT, Claude, Gemini, etc.



História: Modelos de Linguagem Estatísticos (*SLMs*)

- Tratam texto como sequência de palavras
- **Calculam a probabilidade da próxima palavra**
- Sistema n-gramas
- Tem dificuldades com novas palavras.

- Surgiram na década de 1950
- Evoluíram muito nos anos 80-90

- Exemplos: Unigram, Bigram, Trigram
- Uso: Speech Recognition, Tradução, Corretor de Texto



História: Modelos de Linguagem Neurais (*NLMs*)

- Usam redes neurais para aprender representações contínuas de palavras.
- **Preveem a próxima palavra**
- Sistema de vetores
- Capturam similaridade semântica entre palavras no espaço vetorial.
- Surgiram no início dos anos 2000
- **Objetivo: Ser mais generalista que os SLMs**
- Exemplos: **Word2Vec**, RNNLM, LSTM
- Uso: Tradução
- Avanços: Entendem semântica, contexto mais longo, facilidade com palavras raras.



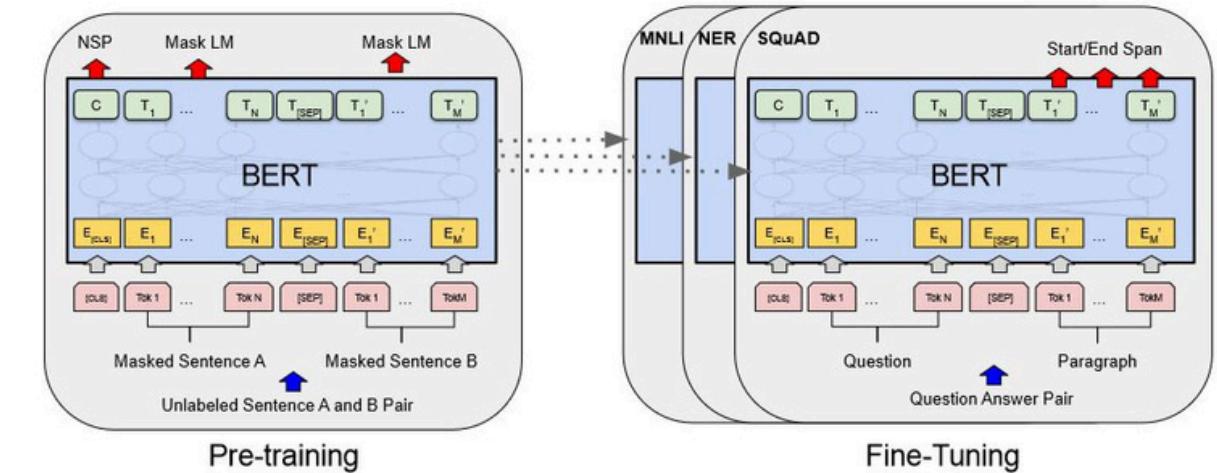
História: Modelos de Linguagem Pré-treinados (PLMs)

- Modelos grandes e generalistas
- **Treinados de forma geral e ajustado posteriormente**
- Baseados em transformers

- Surgiram em 2018–2019
- **Objetivo: Ter um modelo universal, super generalista.**

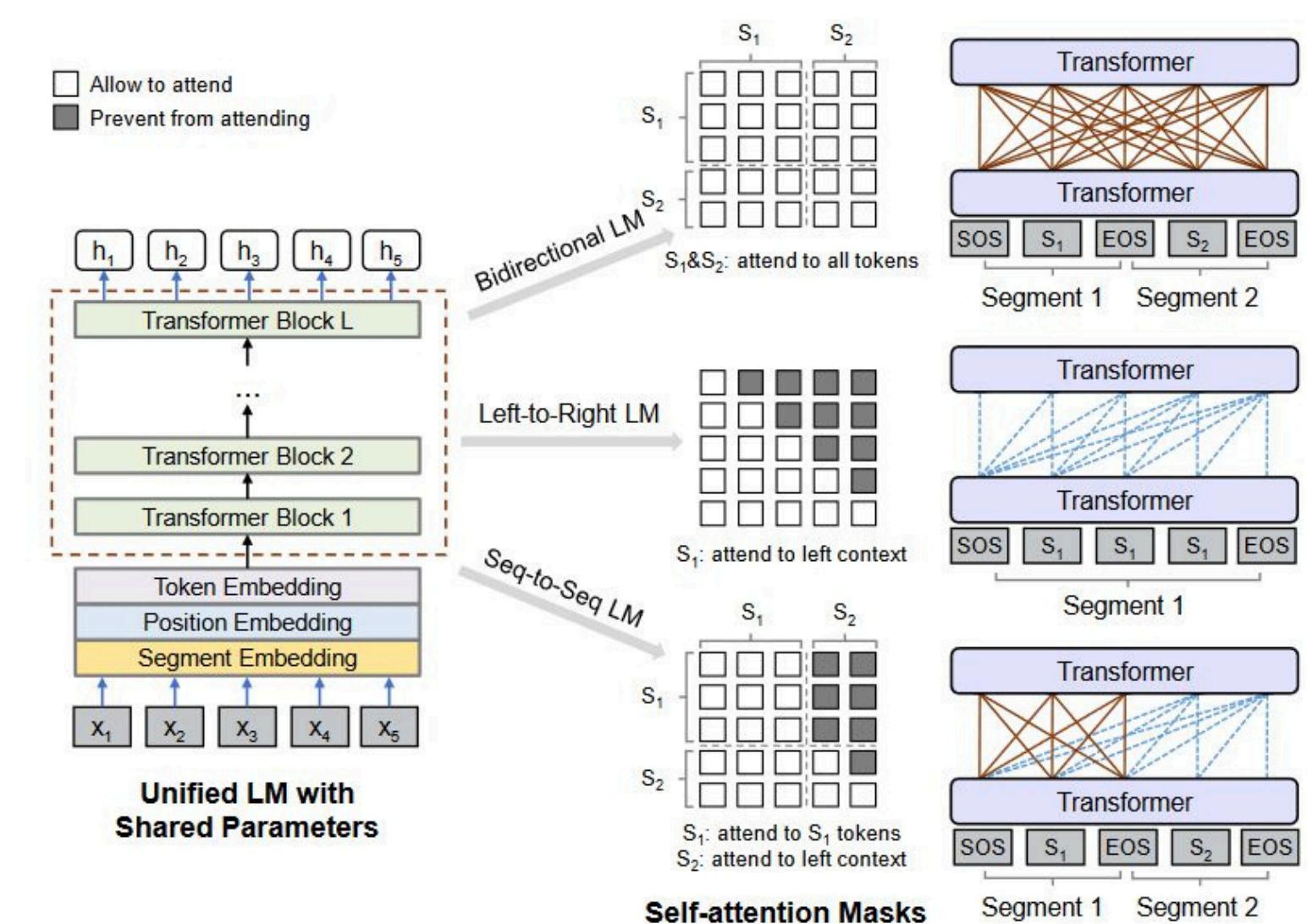
- Exemplos: BERT, RoBERTa, GPT-1, GPT-2, BART
- Diferencial: Um modelo faz várias tarefas.

- Avanços: Muito mais generalistas e inteligentes, treino mais simples.



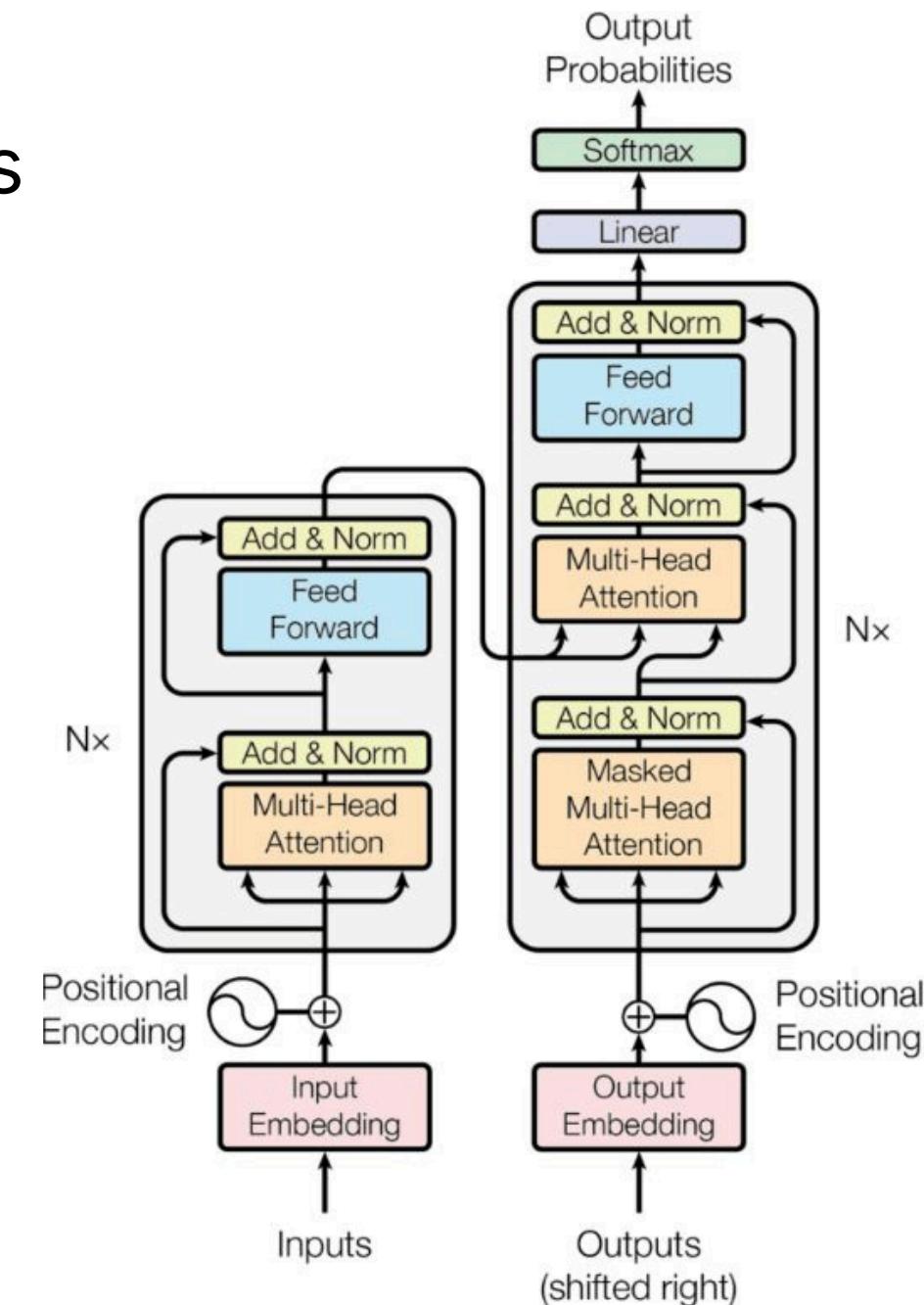
História: A invenção do *Transformer*

- Entendem relações entre todas as palavras ao mesmo tempo
- Usam **mecanismos de atenção**, dando pesos diferentes para cada palavra.
- Surgiram em 2017
- Base dos modelos: BERT, GPT, LLaMA, etc.
- Muito **mais rápido**, usa a GPU
- Melhor em **grandes contextos**
- **Escalabilidade** para bilhões de parâmetros



Arquiteturas PLM/LLM

- **Encoder-Only:** Lê tudo de uma vez, **entende bem, gera mal**
- Uso: Classificação, análise de sentimento, extração de informações
- Exemplos: BERT
- **Decoder-Only:** Só olha o que vem antes, **gera bem, entende mal**
- Uso: Geração de texto, chatbots, auto-completion
- Exemplos: ChatGPT
- **Encoder-Decoder:** Junta os dois!
- Uso: Tradução e resumos
- Exemplos: T5



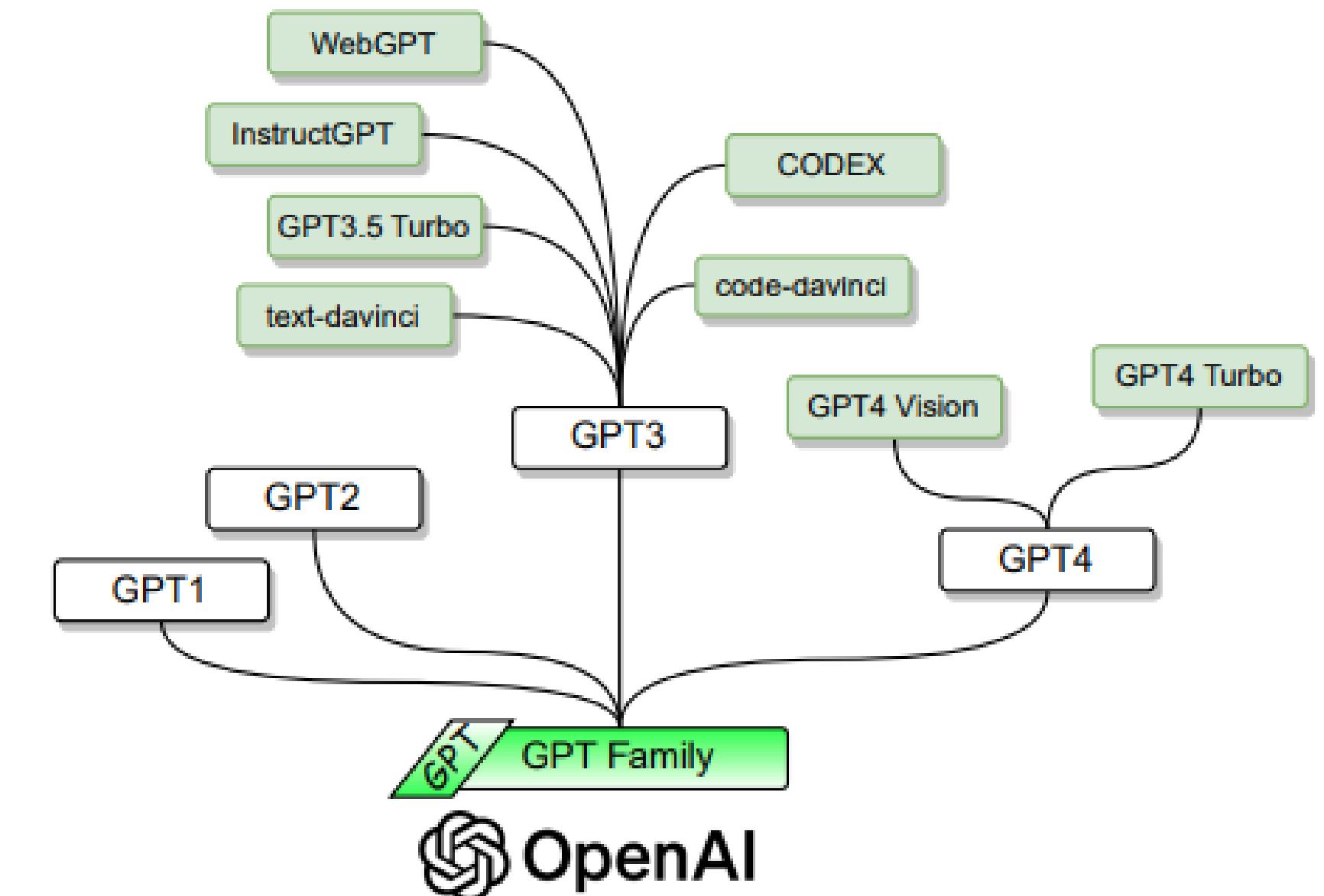
LLMs: Definição e Habilidades Emergentes

- Modelo muito grande, bilhões de parâmetros
 - Treinado em massas gigantescas de texto
 - Baseado em Transformers
 - Possui capacidades avançadas que não existiam em modelos menores
- 👉 • **In-Context Learning:** Aprende com prompts, sem mexer nos pesos.
- **Instruction Following:** Segue instruções diretas sem precisar de exemplos
- **Multi-Step Reasoning:** Quebra tarefas grandes em etapas pequenas



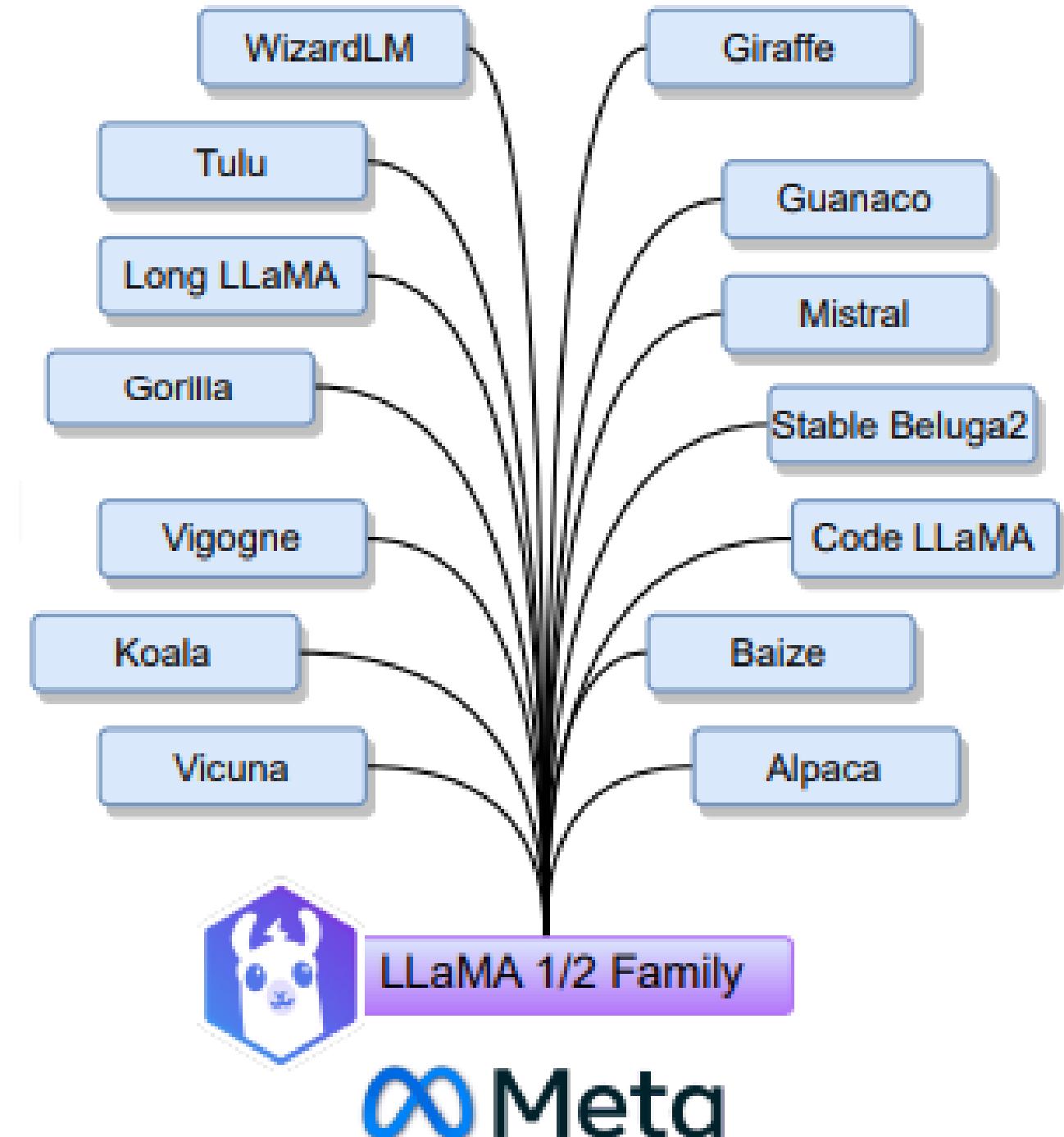
Principais Famílias: GPT Family (OpenAI)

- **Foco em** qualidade de geração de texto e **conversação natural**
 - Modelos **decoder-only** extremamente otimizados
 - Forte capacidade de **seguir instruções**
 - Grande ecossistema
-
- **Prós:**
 - Bom raciocínio
 - Versáteis
 - Amigáveis
-
- **Cons:**
 - Closed Source
 - API é cara
 - Menor transparência em pesquisas



Principais Famílias: LLaMA Family (Meta)

- **Open-source**
 - Focados em alta performance
 - Ótima segurança e transparência
-
- **Prós:**
 - Bom custo benefício
 - Alta customabilidade
 - Facilidade em fine-tuning
-
- **Cons:**
 - Não tem métricas tão boas
 - Dependem muito do pós treino



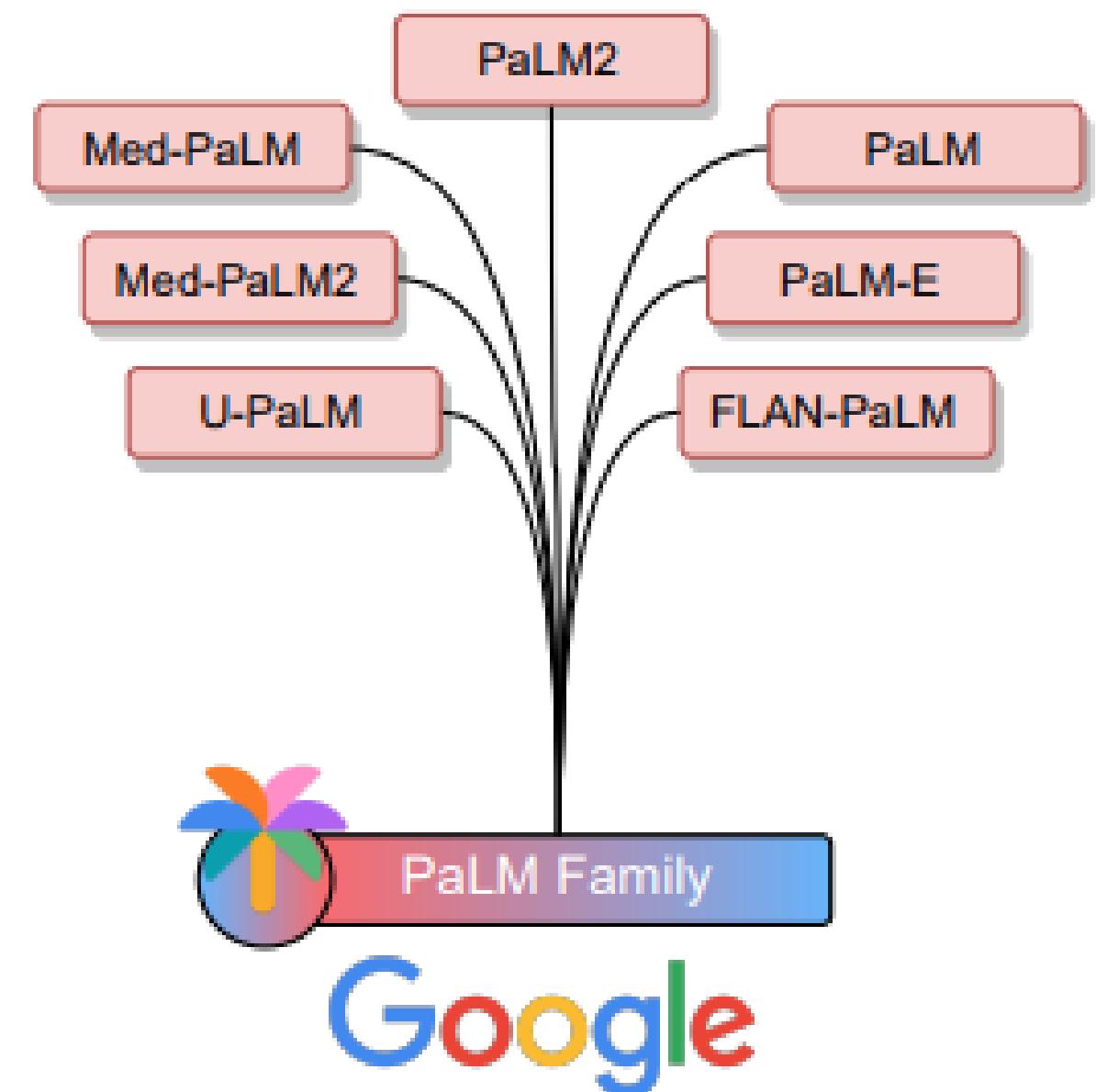


Principais Famílias: PaLM Family (Google)

- Forte foco em **multilinguismo** e **raciocínio lógico**
- Utiliza arquiteturas avançadas
- Treinamento massivo baseado em Pathways (vários treinos de uma vez)
- Versões especializadas

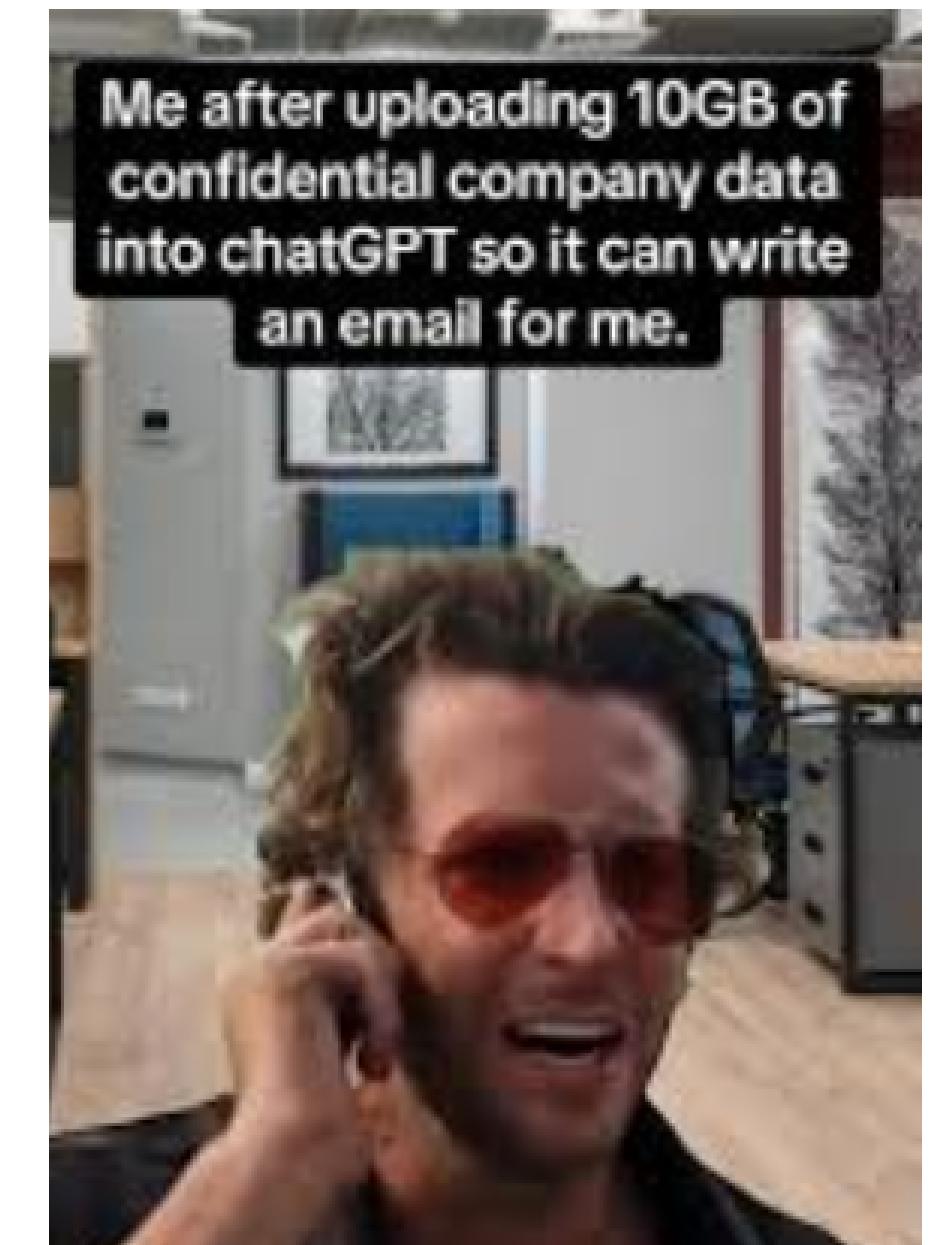
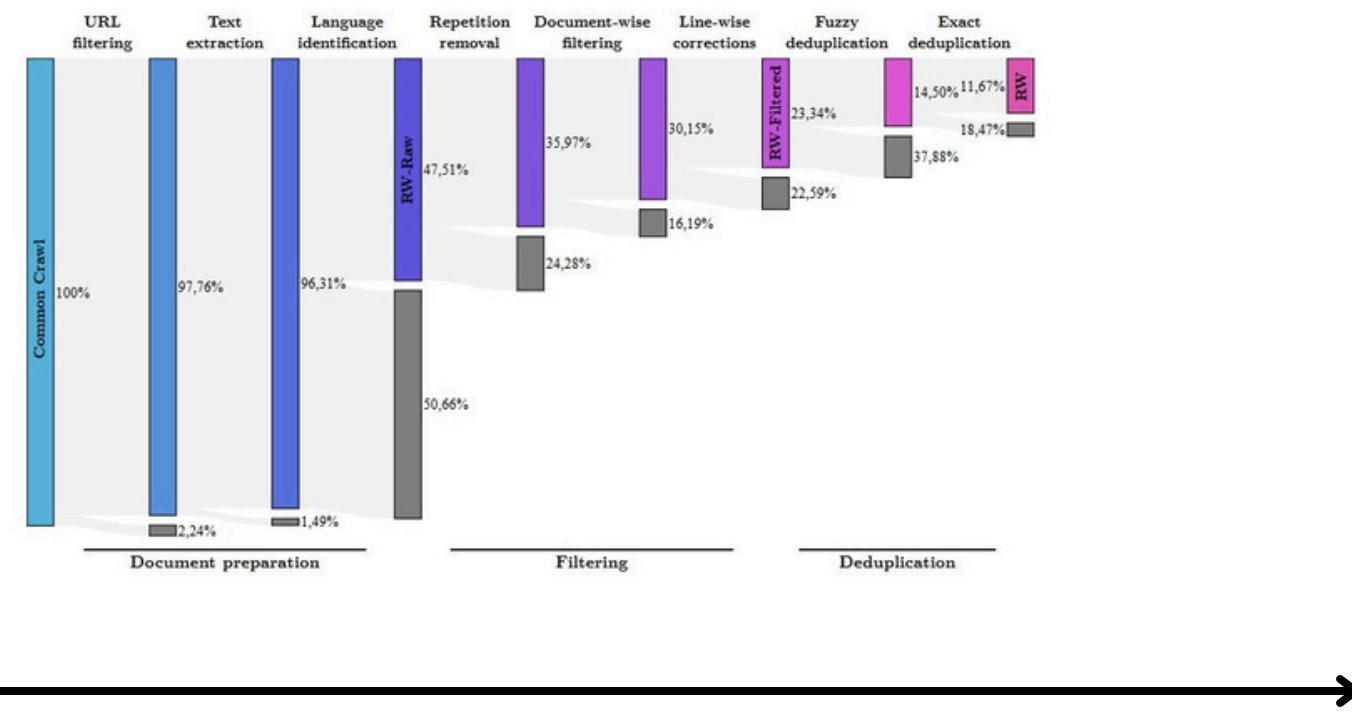
- **Prós:**
 - Ótimo para matemática
 - Versões médicas com métricas impressionantes
 - Fácil integração com stack Google

- **Cons:**
 - Closed Source
 - Cara e difícil de treinar
 - Conversação bem fraca

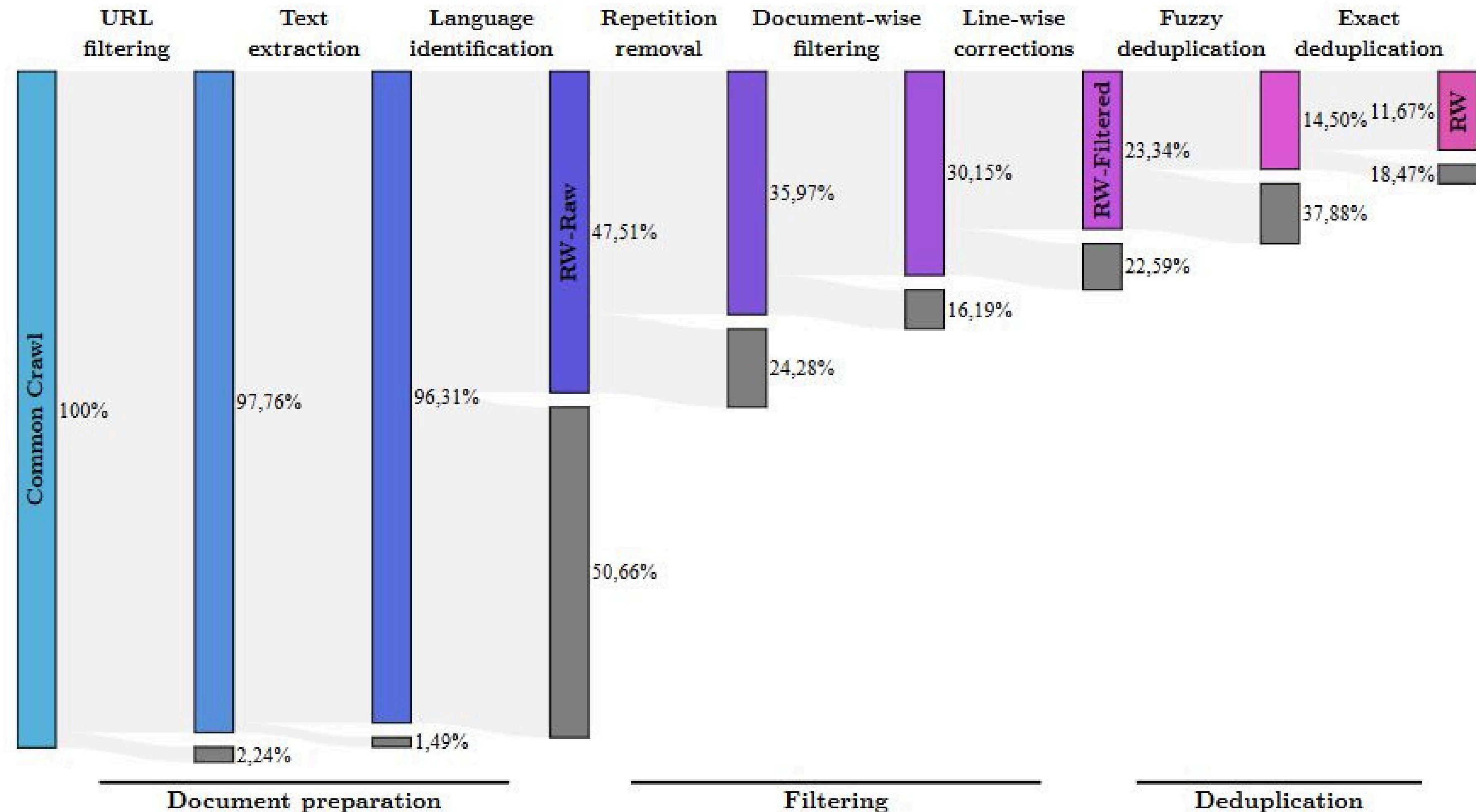


A Construção de uma LLM: Preparação de dados

- **Dados Usados:**
 - Internet
 - Livros
 - Artigos Científicos
 - Conversas
 - Código fonte
 - Dados privados

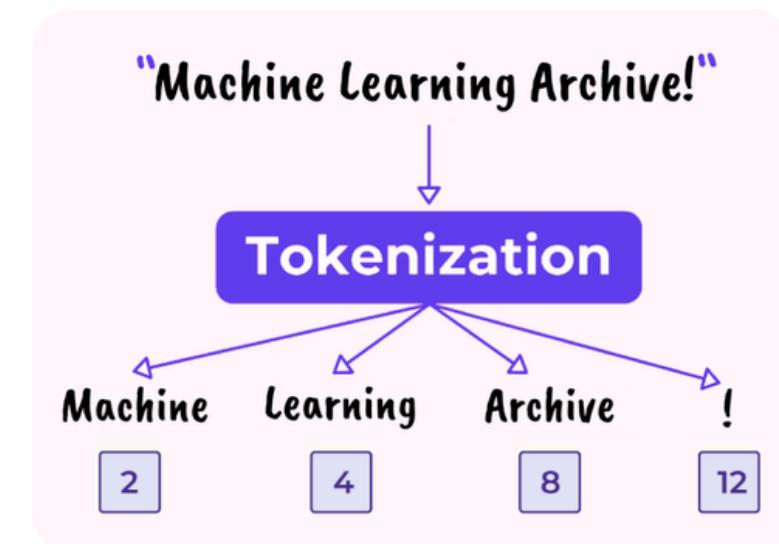


- **Etapas:**
 - **Data Cleaning:** Remover spam e arquivos inúteis
 - **Filtering:** Remover linguagem ofensiva e baixa qualidade
 - **Deduplication:** Remover textos repetidos



A Construção de uma LLM: Tokenização

- **Quebra o texto em unidades menores**
- Pode ser uma palavra inteira, um pedaço dela ou até um caractere
- Cada Token vira um número (ID numérico)
- Permite que o modelo entenda as palavras

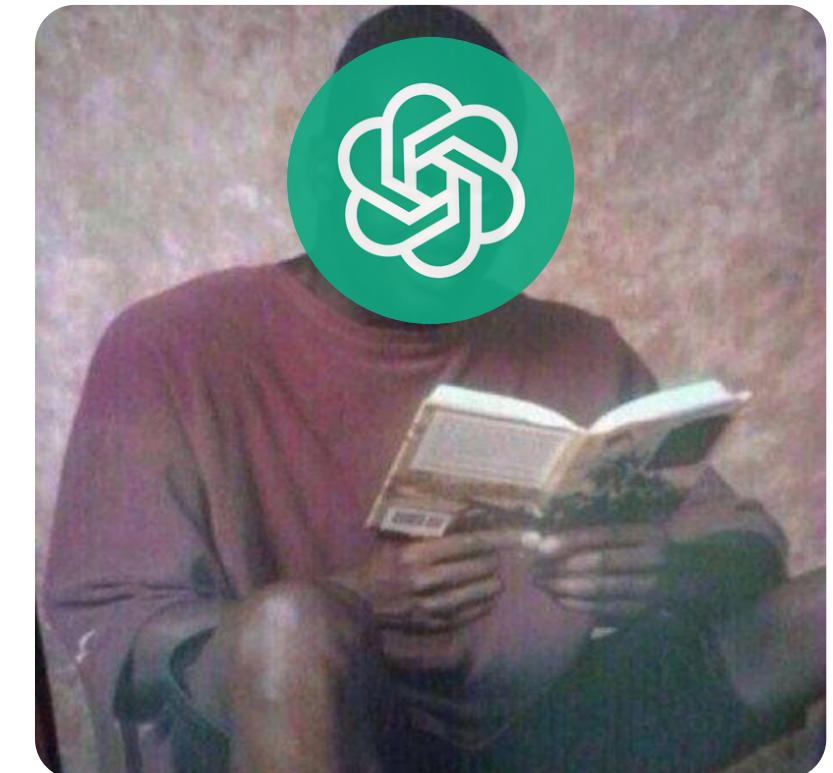


Large Language Models (LLMs), such as GPT-3 and GPT-4, utilize a process called tokenization. Tokenization involves breaking down text into smaller units, known as tokens, which the model can process and understand. These tokens can range from individual characters to entire words or even larger chunks, depending on the model. For GPT-3 and GPT-4, a Byte Pair Encoding (BPE) tokenizer is used. BPE is a subword tokenization technique that allows the model to dynamically build a vocabulary during training, efficiently representing common words and word fragments. Although the core tokenization process remains similar across different versions of these models, the specific implementation can vary based on the model's architecture and training objectives.

GPT-4	LLAMA 3
3.141592653589793238462643383 279502884197169399375105820 974944592307816406286208998 628034825342117067982148086 513282306647093844609550582 231725359408128481117450284 102701938521105559644622948	3.141592653589793238462643383 279502884197169399375105820 974944592307816406286208998 628034825342117067982148086 513282306647093844609550582 231725359408128481117450284 102701938521105559644622948
CLAUDE	GEMMA
3.14159265358979323846264338 3279502884197169399375105820 74944592307816406286208998 280348253421170679821480865 1328230664709384460955058223 17253594081284811174502841027 01938521105559644622948954930	<bos>3.1415926535897932384626 43383279502884197169399375105 82097494459230781640628620899 86280348253421170679821480865 13282306647093844609550582231 72535940812848111745028410270 1938521105559644622948954930

A Construção de uma LLM: Pré-treinamento

- Aprende lendo textos
- LLM recebe o texto tokenizado
- Executa tarefas auto-supervisionadas
- **Causal LM:** Prever o próximo token
- **Masked LM:** adivinhar partes “escondidas” do texto
- Ajusta os bilhões de pesos para minimizar erros
- Aprende



A Construção de uma LLM: Fine/Instruction-tuning

- **Fine-tuning:** Ajustar a LLM para uma tarefa específica
 - É tipo um pós-treinamento com dados focados.
-
- **Instruction-tuning:** Ensinar a LLM a seguir instruções.
 - Feito usando vários exemplos de prompt e como a resposta deve ser.



Prompt
engineering

Fine-tuning

A Construção de uma LLM: Alinhamento

- É a “higienização” da LLM.
- Evita respostas ofensivas ou ilegais.
- **Supervised Fine-Tuning:** Humanos ensinam como responder na prática
- **Reinforcement Learning from Human Feedback:** Humanos avaliam respostas
- **Direct Preference Optimization:** LLM recebe pares bom vs ruim
- 👉 • **Kahneman-Tversky Optimization:** LLM recebe exemplos únicos bons e ruins



Limitações Críticas

- **Limitações Inerentes**
- Modelo reproduz viés, erros ou vácuos dos dados de treino
- **Informações Desatualizadas**
- O modelo só sabe o que aconteceu antes da data do treino
- **Estocasticidade**
- Duas respostas diferentes para a mesma pergunta
- **Alucinação Intrínseca**
- Quando a LLM inventa fatos ou preenche vácuos com confiança
- **Alucinação Extrínseca**
- Quando a LLLM erra por causa de dados incorretos no treinamento



Engenharia de Prompt

- São as técnicas que guiam a LLM a produzir respostas melhores
- **Chain of Thought (CoT)**
- Pede para o modelo explicar os passos do raciocínio
- **Tree of Thought (ToT)**
- Pede para o modelo seguir diferentes raciocínios e escolher o melhor
- **Self-Consistency**
- Pede para o modelo gerar várias resposta e escolher a mais frequente
- **Reflection**
- Pede para o modelo ler a própria resposta, refletir e melhorar.

Prompt
Engineering



Fine-tune
existing LLM

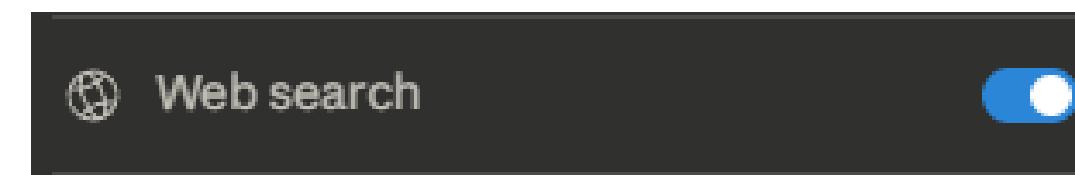
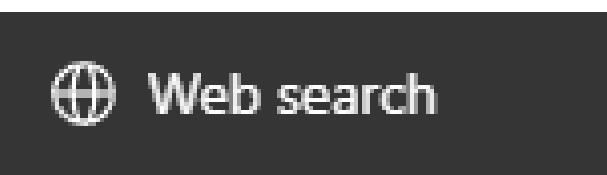


Build an LLM
from scratch



Aumentando LLMs: RAG

- Retrieval-Augmented Generation
- Combina LLM com **busca em bases externas**
- Permite que o modelo use informações atualizadas e específicas
- Evita *alucinações*
- Evita *Limitações Inerentes*
- Muito útil para dados privados (empresas, hospitais)  **LangChain**
- Pode ser conectado à internet (igual os modelos atuais)



Aumentando LLMs: Ferramentas

- Extensões que **permitem que uma LLM execute ações no mundo real**
- Geralmente são APIs, calculadoras, navegadores, etc.
- Criam novas capacidades para a LLM
- Evitam *alucinações*, principalmente na matemática



 Expedia Bring your trip plans to life—get there, stay there, find things to see and do.	 FiscalNote Provides and enables access to select market-leading, real-time data sets for legal, political, and regulatory data and information.	 Instacart Order from your favorite local grocery stores.	 KAYAK Search for flights, stays and rental cars. Get recommendations for all the places you can go within your budget.
--	--	---	---

Aumentando LLMs: Agentes de IA

- LLM que pode agir autonomamente, usando ferramentas, memória e objetivos
- Tipo um robô LLM
- **Planeja, executa ações e monitora resultados**

- Recebe um objetivo
- Planeja os passos necessários
- Executa ações usando ferramentas
- Avalia o resultado
- Reajusta o plano
- Repete até atingir o objetivo



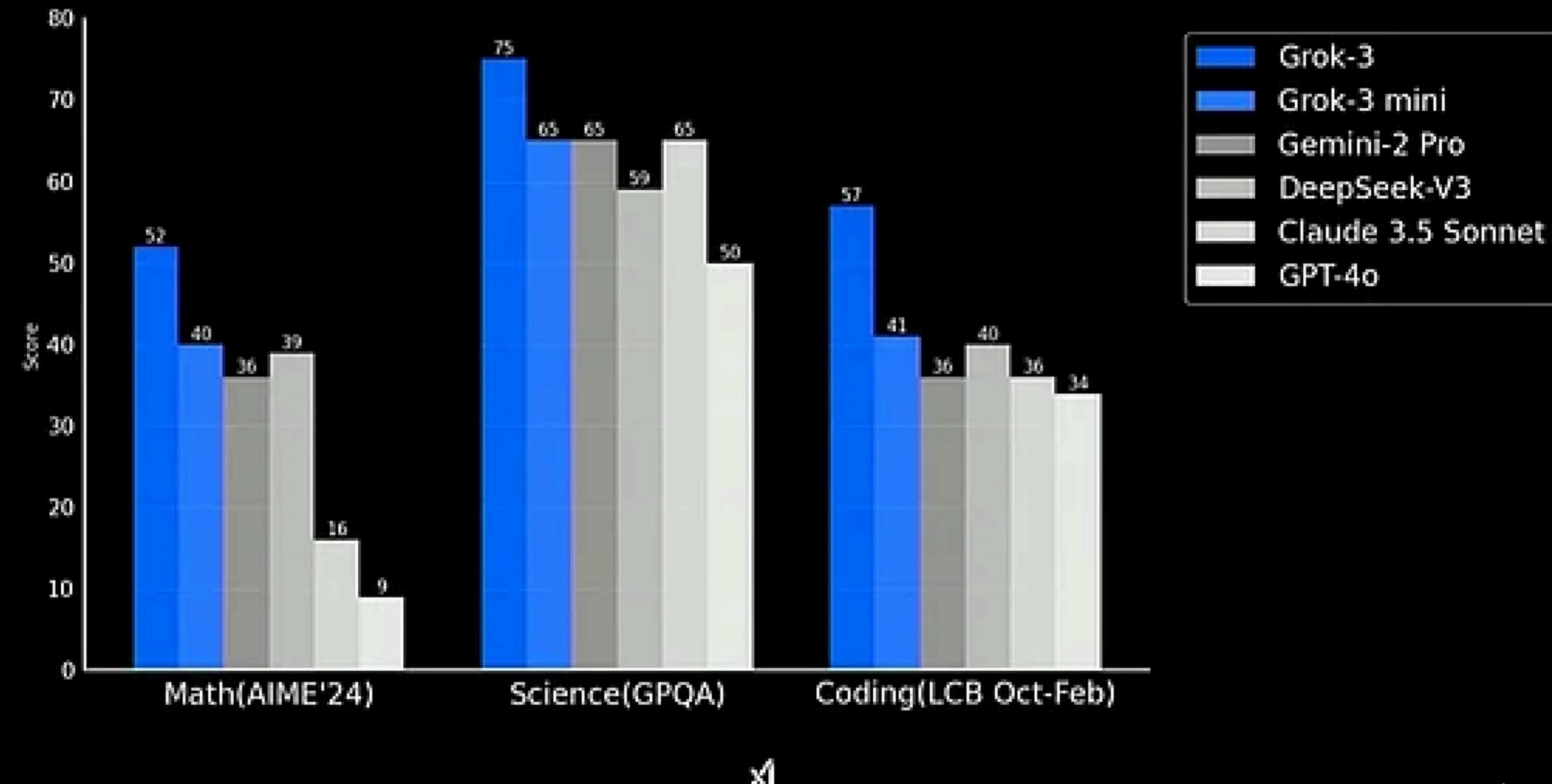
- Muito útil para automatizar tarefas complexas

Avaliação: Métricas

- É a forma de avaliar o desempenho de uma LLM
- **Accuracy:** Porcentagem de respostas corretas
- **BLEU / ROUGE:** Avaliam tradução e resumo comparando texto gerado com texto humano
- **Exact Match:** Exige que a resposta seja igual à correta
- **Perplexity:** Mede quão “surpreso” o modelo fica ao prever o próximo token
- São medidos usando **Benchmarks**
- Geralmente com um gabarito de respostas corretas



Benchmarks



O futuro das LLMs

- **Modelos Menores e Mais Eficientes (SLMs)**
 - Foco em modelos compactos que entregam desempenho igual de LLMs maiores
 - Menor consumo de energia, custo e hardware.
- **Novas Arquiteturas Pós-Atenção**
 - Busca substituir o self-attention por mecanismos mais rápidos
- **Mixture of Experts (MoE)**
 - Modelos com vários especialistas internos, ativando os necessários em cada pergunta
- **Modelos Multimodais**
 - LLMs que compreendem e geram texto, imagem, áudio, vídeo, código

Large Language Models: A Survey

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu Richard Socher,
Xavier Amatriain, Jianfeng Gao



Dúvidas?

Obrigado!

- Aluno: Maruan Biasi El Achkar
- Curso: Engenharia de Software
- Matéria: Aprendizagem de Máquina
- Professor: Claudinei Dias

