Nome: Fernando Buligon Antunes **Data:**31/03/2025

Large Language Models (Tópicos 3 e 4)

O Início é dado falando que produzir LLMs não é um trabalho fácil, principalmente por causa da demanda de recursos computacionais ser muito alta e também temos que considerar as possíveis dificuldades técnicas, uma das maneiras de facilitar o trabalho é fazendo uso de LLMs já criadas e de recursos disponíveis publicamente para estudos.

Dado o alto custo de pré treinamento, a criação de checkpoints é importante, e falando sobre pré treinamento, outro fator que temos que levar em consideração é o número de parâmetros, que é um dos pontos chaves de uma LLMs, sendo separado e dois níveis de escalas: dezenas de bilhares de parâmetros e centenas de bilhares de parâmetros. Também é possível rodar as tarefas em APIs públicas ao invés de rodar o modelo localmente.

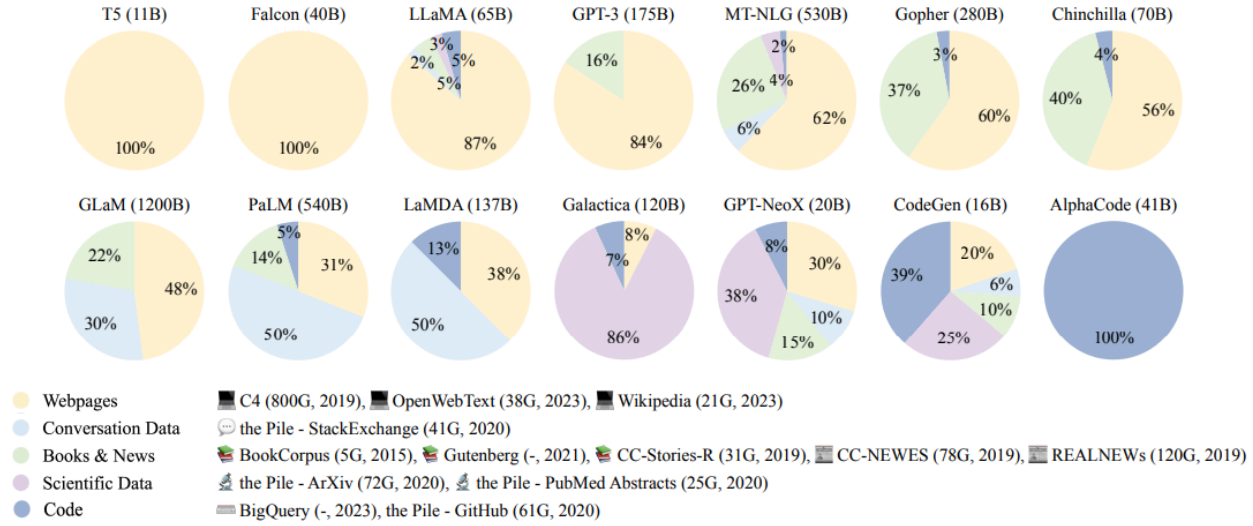
Modelos com dezenas de bilhares de parâmetros, geralmente os modelos que estão dentro dessa categoria estão na casa de 10B a 20B, exceto por modelos mais grandes como o LLaMA (70B), LLaMA2 (70B), NLLB (54.5B) e o Falcon (40B), outros modelos notáveis dentro do range padrão são mT5, PanGu-α, T0, GPT-NeoX-20B, CodeGen, UL2, Flan-T5, mT0.

Modelos com centenas de bilhares de parâmetros, apenas alguns modelos dessa escala foram liberados publicamente, como o OPT, OPT-IML, BLOOM, BLOOMZ, tendo praticamente a mesma quantia de parâmetros que o GPT-3 (175B), outros modelos são GLM (130B), Galactica (120B).

Família de modelos LLaMA, foram introduzidos em em fevereiro de 2023 pela Meta AI, possuindo quatro variações que se diferem pela quantidade total de parâmetros, existindo de 7B, 13B, 30B e 65B. Como seus resultados foram muito bons e são modelos open source, acabaram ganhando certa popularidade dentro da comunidade, sendo uns dos mais usados, outro ponto que também influenciou muito nesse resultado foi o fato de que seus custo computacionais são consideravelmente baixos em relação aos outros modelos.

APIs públicas para LLMs, ao invés de rodar cópias dos modelos localmente, é bem mais vantajoso fazer uso de APIs públicas.

Modelos com uma grande quantidade de parâmetros necessitam de uma vasta quantia de dados, as maiores bases de dados vem de livros, Reddit, Wikipedia, códigos, Common Craw e entre outros. Na imagem abaixo é possível ver alguns modelos famosos e quais foram suas principais fontes para o pré treinamento.



Depois do pré treinamento, ainda é preciso passar por duas etapas antes de chegar ao produto final, namely instruction tuning (fine-tuning supervisionado) e alignment tuning.

Existem várias bibliotecas disponíveis que auxiliam no processo de desenvolvimento de LLMs, algumas delas são: Transformers, DeepSpeed, Megatron-LM, JAX, Cpçpssaç-AI, BMTrain, FastMoE, vLLM, DeepSpeed-MII, DeepSpeed-Chat.

A fase de pré treinamento estabelece as principais capacidades de uma LLM, essa fase é composta por alguns passos como a coleta de dados e preparação, é importante que os dados selecionados sejam altos em qualidades e em números, como falado anteriormente e como podemos ver na imagem acima, existem diferentes fontes de dados, é importante saber de onde eles vão ser retirados pois isso afeta diretamente os resultados, por exemplo, se quer que o modelo se comporte de uma maneira mais formal, talvez redes sociais não sejam uma escolha tão boa, uma escolha mais sábia seria fazer uso de livros, artigos e noticiários. Também tem o pré Processamento de dados, depois de ter feito a coleta dos dados, é importante conferir como eles estão, removendo dados de baixa qualidade, dados duplicados, e informações pessoais que se enquadram em PII (Personally identifiable information).

Arquitetura, existem diferentes tipos de arquiteturas, mas atualmente a que mais é usada é a baseada em transformers, tem sua grande popularidade devido a sua alta performance.

Depois é falado sobre o treinamento, que possui algumas configurações de otimização como batch training, learning rate, optimizer e técnicas de training stability. Também há algumas técnicas de treinamento escaláveis como parallelism 3D, ZeRO (Zero Redundancy Optimizer) e mixed precision training. Depois de ter explicado cada uma, tem algumas recomendações como combinar as técnicas de treinamento escaláveis para aumentar a eficiência.