Universidade Federal da Paraíba

Centro de informática

Processamento de Linguagem Natural

Sabiá-7B Instruction Following Fine Tuning

Paraíba

2024

Equipe Responsável

- Antonio Isaac Araujo Firmino de Sousa - 20200020501
- Douglas da Silva Pereira Veras - 20190113674
- Jonas Gabriel Leite de Araújo - 20200007608

Sumário

1. Apresentação do problema

O Sabia-7B é um Modelo de Linguagem Grande (LLM) desenvolvido pela empresa brasileira MaritacaAI e lançado em novembro de 2023. O modelo do Sabia-7B foi desenvolvido a partir da arquitetura e pesos do LLaMA 7B, outra LLM que é, majoritariamente, treinada com dados de textos em inglês. Dessa forma, o objetivo do Sabia-7B é ser uma LLM refinada para lidar, com maior precisão, com textos em português. Então, o modelo foi treinado com 10.4 bilhões de tokens com o tamanho do vocabulário de 32000 tokens únicos.

Entretanto, o Sabia-7B disponibilizado para o público é apenas o modelo pré-treinado, isto é, o modelo é capaz apenas de prever próximas palavras e completar frases com exemplos pequenos, como pode ser visto nas figuras 1 e 2.

```
prompt = """Classifique a resenha de filme como "positiva" ou "negativa".

Resenha: Gostei muito do filme, é o melhor do ano!
Classe: positiva

Resenha: O filme deixa muito a desejar.
Classe: negativa

Resenha: Apesar de longo, valeu o ingresso.
Classe:"""
```

Figura 1: Requisição para o modelo Sabia-7B classificar uma resenha de filme como positiva ou negativa.

```
print(tokenizer.decode(output, skip_special_tokens=True))
```

```
positiva
```

Figura 2: Resposta do modelo para o prompt anterior

Sendo assim, percebe-se que o Sabia-7B possui pouca capacidade para seguir instruções humanas, como pode ser comprovado pela figura 3. Logo, o modelo precisa passar por um processo de refinamento (fine tuning) para poder lidar com instruções humanas.

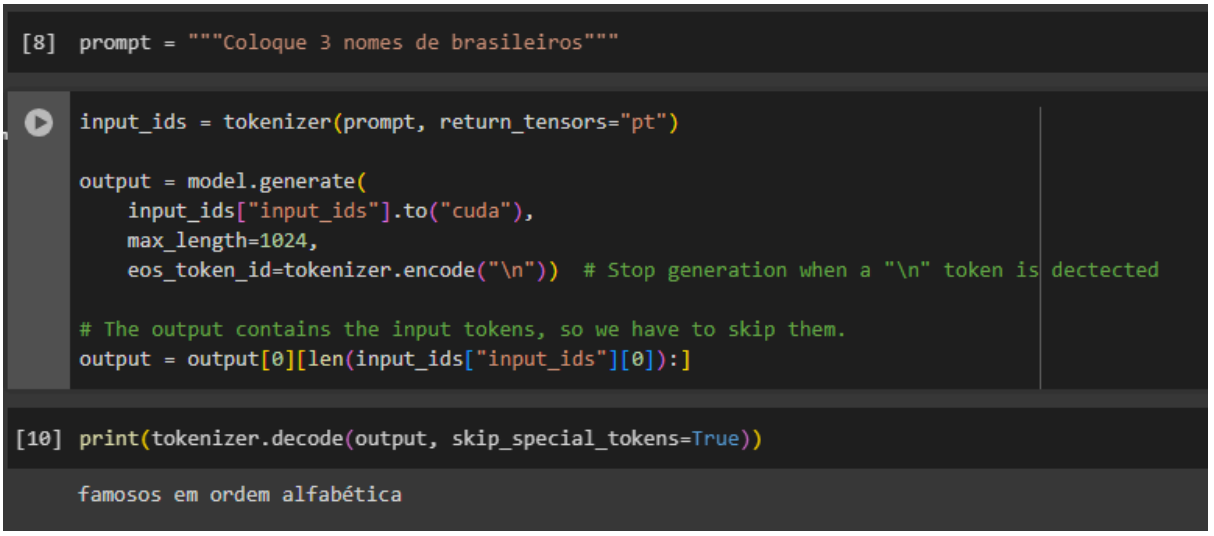

Figura 3: Prompt pede 3 nomes de brasileiros e recebe como retorno "famosos em ordem alfabética"

2. Objetivos

O objetivo deste projeto é avaliar a influência do refinamento do modelo base do Sabia-7B com dados de instruções. Busca-se determinar até que ponto a inclusão deste conjunto de dados pode melhorar a capacidade do modelo de processar e seguir instruções detalhadas, proporcionando uma interação eficaz do usuário com um modelo focado na língua portuguesa. Por fim, faz-se necessário fazer uma série de testes para comparar o desempenho do modelo base com o modelo após o treinamento com as instruções, focando em métricas que evidenciem avanços em precisão, relevância e naturalidade da linguagem gerada.

3. Dados utilizados e pré-processamento dos dados

3.1. Dados utilizados

Para o treinamento focado em instruções, foi utilizado o Canarim-Instruct-PTBR-Dataset, uma base de dados que contém mais de 300.000 instruções em português, variando de solicitações simples a complexas, no desempenho do modelo em tarefas de compreensão e geração de linguagem

As instruções foram criadas a partir da tradução e adaptação de diversos outros datasets focados em instruções humanas. O dataset possui três colunas: instrução, entrada e saída. A instrução é o comando que será passada para a LLM, a entrada é

opcional, mas ela geralmente fornece mais contexto para a instrução e a saída é a resposta apropriada para a instrução.

3.2. Pré-processamento dos dados

3.3.1. Formatação dos dados

Primeiro aplicamos uma formatação de prompts no Dataset, a fim de que o modelo consiga identificar melhor o que é a instrução, a entrada e o que deve se seguir como saída. Dessa forma, para cada entrada do dataset, um prompt é criado de acordo com o seguinte algoritmo:

1. Concatenar, no início do prompt, o seguinte texto caso não haja uma entrada: "Abaixo está uma instrução que descreve uma tarefa. Escreva uma resposta que conclua adequadamente a solicitação."
    1.1. Caso haja entrada, o texto introdutório é: "Abaixo está uma instrução que descreve uma tarefa, emparelhada com uma entrada que fornece mais contexto. Escreva uma resposta que conclua adequadamente a solicitação."
2. Em sequência, coloca-se um identificador que precede a instrução do dataset. O identificador é: "### Instrução: ".
3. Se houver entrada, colocar o seguinte identificador para a entrada do dataset: "### Entrada: ".
4. Colocar identificador de resposta: "### Resposta: ".
5. Por fim, colocar o token especial de fim de sentença no fim do prompt

Para exemplificar, os seguintes textos se referem a prompts formatados do dataset Canarim sem entrada e com entrada, respectivamente:

- <s> Abaixo está uma instrução que descreve uma tarefa. Escreva uma resposta que conclua adequadamente a solicitação.
  ### Instrução: Dê três dicas para se manter saudável.
  ### Resposta: 1. Coma uma dieta equilibrada e certifique-se de incluir muitas frutas e vegetais. 2. Exercite-se regularmente para manter seu corpo ativo e forte. 3. Durma o suficiente e mantenha um horário de sono consistente. </s>

- <s> Abaixo está uma instrução que descreve uma tarefa, emparelhada com uma entrada que fornece mais contexto. Escreva uma resposta que conclua adequadamente a solicitação.
  ### Instrução: Nomeie as capitais dos três países seguintes
  ### Entrada: Índia, Canadá, Egito
  ### Resposta: Índia: Nova Deli. Canadá: Ottawa. Egito: Cairo. </s>

3.2.1. Tokenização

O tokenizador utilizado pelo modelo base do Sabia-7B foi construído a partir do LLaMATokenizer, este que é um modelo de tokenização feito com a técnica de Byte-Pair Enconding (BPE). Sendo assim, o tokenizer do Sabia-7B conta com um tamanho de vocabulário de 32000 tokens distintos, incluindo três tokens especiais: início de sentença (<s>); fim de sentença (</s>; palavra desconhecida (<unk>).

Para o treinamento, consideramos o token especial de fim de sentença como o token de *padding*, ou seja, para preencher e normalizar todas as entradas textuais para o mesmo tamanho, que é o tamanho de sequência máximo do modelo base, que é de 2048 tokens.

Dessa maneira, após formatação dos dados textuais, utilizamos o tokenizador para converter todos os textos em *input_ids* e *attention_mask*. Os *input_ids* são os tokens convertidos para identificadores únicos, de acordo com o treinamento feito do modelo base, e o *attention_mask* se refere a quais tokens deverão ser considerados pelo mecanismo de atenção do modelo ou não, isto é, ele irá desconsiderar processar os tokens de padding (preenchimento).

4. Detalhes da arquitetura e treinamento

4.1. Arquitetura do modelo

O modelo base do Sabia-7B usa a mesma arquitetura do LLaMA-1-7B, que é fundamentada a partir da arquitetura de *transformers*. O transformers é composto por várias camadas de codificadores e decodificadores, em que cada camada possui mecanismos de atenção que permitem ao modelo ponderar a importância de diferentes partes da entrada durante o processamento. Isso permite que o modelo leve em consideração o contexto global da sequência ao realizar tarefas como tradução, sumarização e geração de texto.

Para o fine tuning de instrução, foi utilizado a técnica de Parameter Efficient Fine Tuning (PEFT). O PEFT é uma técnica de fine tuning que é mais eficiente do que um fine tuning completo (treina todos os pesos do modelo base, que, nesse caso, são sete bilhões). Treinar um modelo de linguagem, especialmente para o refinamento completo de LLMs, exige recursos computacionais significativos. A alocação de memória não é necessária apenas para armazenar o modelo, mas também para os pesos durante o treinamento, apresentando um desafio para hardware simples. O PEFT resolve isso atualizando apenas um subconjunto de parâmetros e "congelando" o restante. Isso reduz o número de parâmetros treináveis, tornando os requisitos de memória mais gerenciáveis. Ao contrário do ajuste fino completo, o PEFT mantém os pesos originais do LLM, evitando a perda de informações previamente aprendidas.

Além do PEFT, a fim de melhorar a eficiência computacional, foi utilizado uma técnica de Local Rank Adaptation (LoRA) e, sua extensão, Quantized Local Rank Adaptation (QLoRA). O LoRA é um adaptador que, ao invés de modificar os pesos originais das camadas, ela aprende como "decompor" a matriz original em matrizes menores que se aproximam com os pesos originais, otimizando ainda mais os cálculos do treino. Ademais, a sua extensão, QLoRA funciona quantizando os pesos do modelo, isto é, reduzindo o tamanho dos parâmetros em 8 bytes, reduzindo o uso de memória do modelo.

A configuração do PEFT e do QLoRA são feitos utilizando as bibliotecas *peft* e *bitsandbytes* e ambas são suportadas automaticamente pela biblioteca *transformers* da plataforma *HuggingFace*. Na configuração do PEFT, definimos que o número de parâmetros treináveis do modelo será igual a 25.165.824, o que equivale a 0,72% dos parâmetros do Sabia-7B.

4.2. Treinamento

Para o treinamento do modelo, foi configurado um ambiente no Google Colab Pro, onde foi utilizado uma A100 GPU com 40GB de VRAM. Por limitação de tempo e recursos, foram utilizados apenas 27.000 dos dados para o treino e 1500 para a validação. Os hiperparâmetros de treino são os seguintes:

- **Batch size** = 8;
- **Epochs** = 3;
- **Learning rate** = 0,0002 (2e-4);

- **Optimizer** = AdamW;
- **Learning rate scheduler** = Cosine.

5. Resultados

Ao fim do processo de treinamento, obteve-se um resultado razoável para o modelo, com ele respondendo a pergunta, mas não conseguindo terminar a sentença da maneira correta, ficando repetindo uma parte em específico até chegar ao limite estabelecido como tamanho para o modelo. Usou-se, também, o streamlit para criar uma interface para o nosso modelo, como pode ser visto nas figuras abaixo.

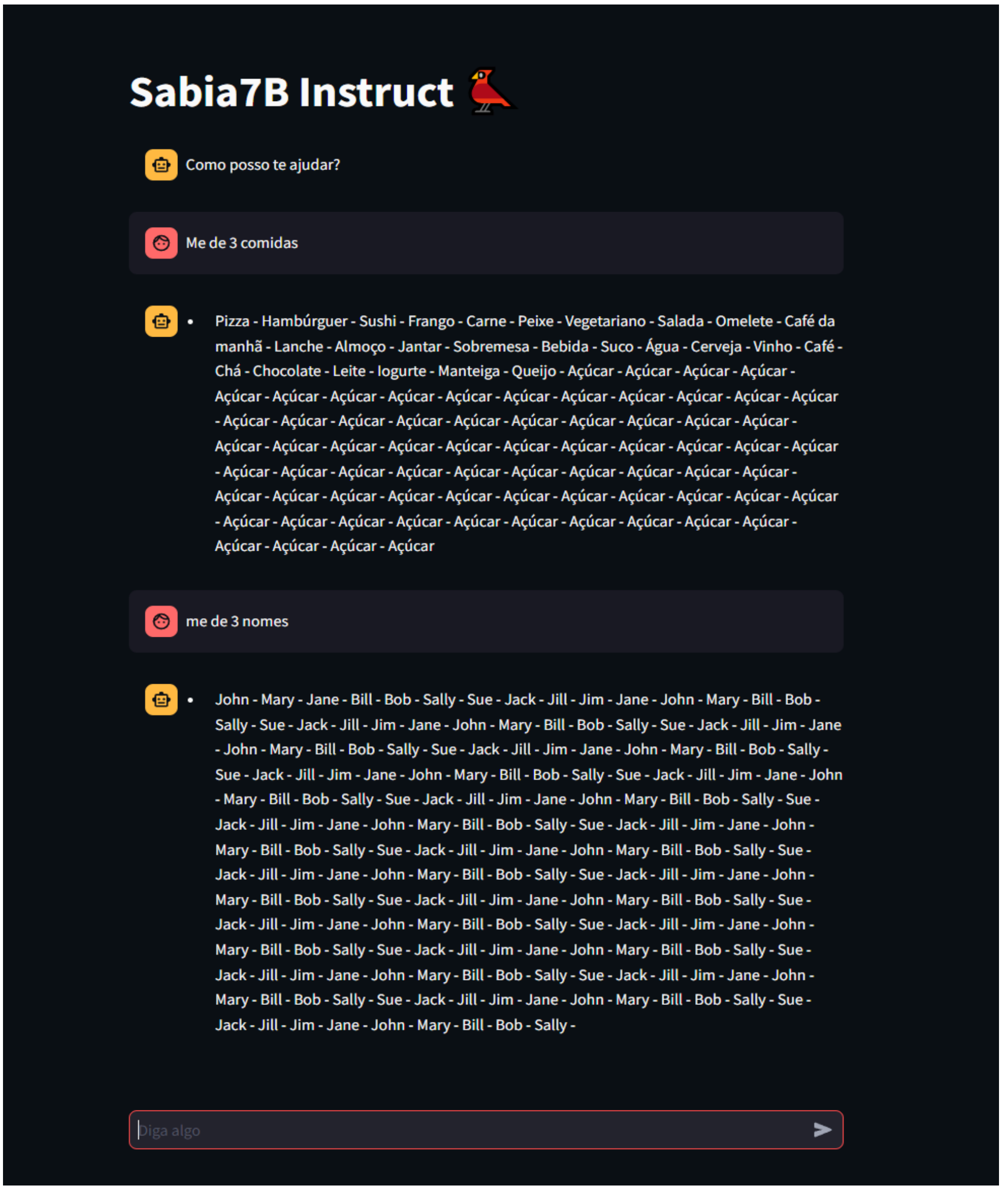

Figura 4: Prompts com resposta sendo repetida.

Sendo assim, uma boa maneira de melhorar o resultado do projeto seria fazer o modelo possuir mais tempo de treinamento, uma vez que, ele foi treinado por apenas 3 épocas com 10% do dataset, além de possuir a sua precisão reduzida para melhorar o custo computacional, devido à dificuldade de um treinamento adequado nos computadores dos integrantes do grupo.

Referências

- Pires, R.; Abonizio, H.; Almeida, T. S.; Nogueira, R. Sabia: Portuguese Large Language Models. In: Naldi, M. C.; Bianchi, R. A. C. (Eds.). Intelligent Systems. Cham: Springer Nature Switzerland, 2023. p. 226-240. Disponível em: <https://arxiv.org/pdf/2304.07880>
- DAS, S. Fine Tune Large Language Model (LLM) on a Custom Dataset with QLoRA. Disponível em: <https://dassum.medium.com/fine-tune-large-language-model-llm-on-a-custom-dataset-with-qlora-fb60abdeba07>
- CAPELLE, T. How to Fine-tune an LLM Part 3: The HuggingFace Trainer. Disponível em: <https://wandb.ai/capecape/alpaca_ft/reports/How-to-Fine-tune-an-LLM-Part-3-The-HuggingFace-Trainer--Vmlldzo1OTEyNjMy>
- Domingues, M. Canarim-Instruct-PTBR-Dataset (Revision c2de751). 2023. Disponível em: <https://huggingface.co/datasets/dominguesm/Canarim-Instruct-PTBR-Dataset>