UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

O Poder dos Modelos de Linguagem

Avaliando LLMs de diferentes tamanhos na área da saúde

Luiz Guilherme Corrêa Figueredo

UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

LUIZ GUILHERME CORRÊA FIGUEREDO

O PODER DOS MODELOS DE LINGUAGEM

Avaliando LLMs de diferentes tamanhos na área da saúde

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): **LUIZ GUILHERME CORREA FIGUEREDO**

Título do trabalho:

O PODER DOS MODELOS DE LINGUAGEM

Avaliando LLMs de diferentes tamanhos na área da saúde

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [**X**] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Luiz Guilherme Correa Figueredo**, **Discente**, em 15/02/2024, às 19:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 12/09/2024, às 11:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4383415** e o código CRC **F376782D**.

Referência: Processo nº 23070.008393/2024-57 SEI nº 4383415

LUIZ GUILHERME CORRÊA FIGUEREDO

O PODER DOS MODELOS DE LINGUAGEM

Avaliando LLMs de diferentes tamanhos na área da saúde

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

FIGUEREDO, LUIZ GUILHERME CORREA
O PODER DOS MODELOS DE LINGUAGEM [manuscrito] : Avaliando LLMs de diferentes tamanhos na área da saúde / LUIZ GUILHERME CORREA FIGUEREDO. - 2024.
111 f.

Orientador: Prof. Dr. FERNANDO MARQUES FEDERSON.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2024.

1. inteligência artificial. 2. modelos grandes de linguagem. 3. saúde. I. FEDERSON, FERNANDO MARQUES, orient. II. Título.

CDU 004

LUIZ GUILHERME CORRÊA FIGUEREDO

O PODER DOS MODELOS DE LINGUAGEM

Avaliando LLMs de diferentes tamanhos na área da saúde

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 08 de fevereiro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Vinicius Sebba Patto
Coordenador do BIA (INF-UFG)

Prof. Dr. Arlindo Rodrigues Galvão Filho
(INF-UFG)

LUIZ GUILHERME CORRÊA FIGUEREDO

O PODER DOS MODELOS DE LINGUAGEM

Avaliando LLMs de diferentes tamanhos na área da saúde

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Large Language Models**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, saúde.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Large Language Models**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, large language models, health.

Goiânia

2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada

Luiz Guilherme Corrêa Figueredo

Especialista em:
Large Language Models

Slides feitos com Slidesgo e Freepik

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: Luiz Guilherme Corrêa Figueredo

Especialidade: Large Language Models

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Antes de iniciar o processo da Disciplina de Residência em IA, eu já tinha a convicção de que iria me especializar na área de **Large Language Models (LLMs)**, visto que era uma área que eu já estava pesquisando em outros projetos. No entanto, ainda não havia definido, para a disciplina, qual a área de aplicação eu exploraria, para que eu aprimorasse meus conhecimentos em LLMs. Sendo assim, nas **Semanas 1 e 2**, inicialmente optei pela área financeira, porém não tive muita afinidade com a área e encontrei dificuldades de encontrar dados para LLMs e, assim, acabei optando pela área da saúde. Por fim, fiz uma revisão geral sobre Processamento de Linguagem Natural (NLP) e LLMs, com base nos conteúdos vistos na disciplina de NLP, no 6º período do Bacharelado, com a intenção de fixar os conteúdos que seriam úteis posteriormente no processo da Residência. No **Apêndice 1**, estão, de forma mais detalhada, as pesquisas feitas inicialmente acerca da área financeira e a revisão geral de NLP e LLMs.

Sendo assim, destaco o meu aprendizado em relação a verificar a disponibilidade de dados antes de definir uma área de aplicação, além da percepção de que a afinidade por uma área pode impulsionar a dedicação e pesquisa durante o processo.

[1] Dez semanas, entre setembro de 2023 e janeiro de 2024.

A partir da definição da área de aplicação, nas **Semanas 3 e 4**, obtive dados da área da Saúde com perguntas e respostas de vestibulares e dúvidas de pessoas leigas, de uma forma geral, e que seriam necessários para treinar um LLM. Em seguida, lidei com o pré-processamento dos dados, separando em treino e teste e traduzindo-os, visto que estavam em inglês. No **Apêndice 2**, é possível encontrar documentos que detalham mais os conjuntos de dados obtidos, tanto para treino quanto para avaliação, e o pré-processamento aplicado.

Todo o esforço dedicado ao pré-processamento (tradução) dos dados, me fez perceber a utilidade de se utilizar LLMs para tarefas desgastantes, visto que, inicialmente, considerei analisar a tradução de forma manual, porém, ao perceber os erros mais comuns, optei por utilizar o ChatGPT para a tradução, destacando os erros mais comuns no prompt.

Com os dados em mãos, a **Semana 5** foi dedicada para a pesquisa de estratégias de treino de LLMs, além de frameworks em Python que poderiam ser utilizados para o treinamento, mas também para a avaliação. Ao fim, optei por realizar o treino de LLMs, através de Instruction Tuning, utilizando o método QLoRA com os frameworks transformers, peft e trl, que são todos do HuggingFace. Além disso, nesta semana, também foram selecionados 3 LLMs de diferentes tamanhos para serem treinados, que são: Zephyr (7 bilhões de parâmetros), GPT-2 (1.5 bilhões de parâmetros) e DistilGPT-2 (82 milhões de parâmetros). No **Apêndice 3**, estão detalhadas as estratégias de treino pesquisadas e os frameworks considerados.

Na **Semana 6**, o processo de treino dos LLMs foi iniciado, ou seja, os 3 LLMs, selecionados anteriormente, foram treinados em 19 mil dados e avaliados em perguntas de múltipla escolha da área da Saúde, de diversos conjuntos de dados. Os resultados foram o seguinte: o modelo Zephyr acertou de 30 a 40% (dentre os diversos conjuntos de dados utilizados) das perguntas de múltipla escolha, o modelo GPT-2 acertou de 20 a 40% (dentre os diversos conjuntos de dados utilizados) das perguntas de múltipla escolha e o modelo DistilGPT-2 acertou de 20 a 40% (dentre os diversos conjuntos de dados utilizados) das

perguntas de múltipla escolha. Todo o processo de treino, juntamente com os conjuntos de dados, os parâmetros utilizados e as curvas de loss, estão documentados no **Apêndice 4**.

Dessa forma, foi possível perceber que todos os LLMs apresentavam um desempenho semelhante, mesmo com uma quantidade diferente de parâmetros, e isso despertou minha curiosidade para procurar justificativas para esse comportamento, além de buscar alternativas para melhorar o desempenho desses modelos também.

Nas **Semanas 7 e 8**, com o objetivo de melhorar os resultados, a quantidade de dados de treino foi aumentada de 19 mil para 53 mil dados. Os mesmos modelos foram treinados novamente, mas o desempenho nas perguntas de múltipla escolha foi praticamente igual ao anterior. Além disso, buscando avaliar a qualidade da resposta gerada, e não só o acerto em perguntas de múltipla escolha, foi inserido uma nova forma de avaliação que analisa a semântica, organização e utilidade da resposta, através do GPT-4, e retorna uma nota que variando entre 0 e 15 pontos. Considerando que a v1 corresponde aos modelos treinados com 19 mil dados e a v2 aos modelos treinados com 53 mil dados, os resultados foram os seguintes:

- Zephyr-v1: 9.22 pontos
- Zephyr-v2: 3.7 pontos
- GPT-2-v1: 0.07 pontos
- GPT-2-v2: 0.03 pontos
- DistilGPT-2-v1: 0.12 pontos
- DistilGPT-2-v2: 0.08 pontos

Todo o processo de treino, juntamente com os conjuntos de dados e os parâmetros utilizados e as curvas de loss, das duas versões do conjunto de dados, o novo método de avaliação inserido e exemplos de respostas dadas pelos modelos treinados, estão documentados no **Apêndice 5**.

Sendo assim, mesmo com o desempenho semelhante em perguntas de múltiplas escolhas, percebi que os modelos com uma menor quantidade de parâmetros estavam gerando textos incoerentes (apenas repetindo palavras), enquanto o Zephyr, treinado com

menos dados, conseguiu gerar textos coerentes e consistentes. Tudo isso me fez pensar que avaliar as respostas em perguntas de múltipla escolha talvez não fosse a melhor escolha ou ainda que poderia haver algum erro no código de avaliação em perguntas de múltipla escolha. Apesar disso, fiquei satisfeito com os textos produzidos pelo Zephyr-v1.

Nas **Semanas 9 e 10**, meus esforços foram direcionados para investigar o baixo desempenho dos LLMs treinados nas perguntas de múltipla escolha, mas também na falha apresentada na geração textual dos LLMs menores. Dessa forma, pude elencar três possíveis razões:

1. Qualidade dos dados: a primeira hipótese é que, pelos dados terem sido traduzidos, talvez a qualidade dos dados tenha sido comprometida e, possivelmente, tenha afetado o desempenho dos LLMs nas perguntas de múltipla escolha.

2. Tamanho dos modelos: relacionada aos modelos menores: GPT-2 (1.5 bilhões de parâmetros) e DistilGPT-2 (82 milhões de parâmetros). A ideia é que esses modelos possuem uma menor capacidade de aprendizado, podendo levar a uma queda de desempenho ao serem avaliados em perguntas de múltipla escolha, afetando também a geração textual desses modelos de linguagem.

3. Complexidade linguística: também relacionada aos modelos de linguagem menores e sua falha na geração textual. A ideia é que a complexidade linguística, inerente ao idioma em questão (português), além da profundidade e diversidade do corpus, dificultam o aprendizado desses modelos e, consequentemente, afetam sua capacidade de gerar textos consistentes e coerentes.

No **Apêndice 6**, estão os documentos que detalham as pesquisas feitas e as hipóteses levantadas por mim como razões para o baixo desempenho dos LLMs treinados nas perguntas de múltipla escolha e a falha na geração textual dos LLMs menores.

Desse modo, considerei que essas três hipóteses poderiam ser as causas dos comportamentos percebidos: geração textual falha por parte dos LLMs menores e baixo desempenho nas perguntas de múltipla escolha.

De modo geral, considerei todo o processo da Disciplina Residência em IA enriquecedor, tanto para a obtenção de conhecimento técnico quanto para o meu autoconhecimento.

Em relação ao conhecimento técnico, destaco uma das maiores lições aprendidas: ao focar muito no aprendizado de LLMs, esqueci da importância dos dados, mas é extremamente essencial ter dados de alta qualidade e curados, para que o aprendizado dos LLMs seja melhor aproveitado.

A respeito do autoconhecimento, aprendi a lidar com questões de autoconfiança, pois tive dúvidas se o que eu estava produzindo seria suficiente para a Disciplina, mas, ao fim do processo, percebendo o quão mais experiente eu havia me tornado e o aumento no meu domínio no assunto de LLMs, me senti mais confiante e realizado. Além disso, aprendi a conhecer meus limites pessoais, porque a responsabilidade por definir pesquisas, atividades, resultados e entregáveis foi totalmente minha.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 19 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Levando em conta que o tema que escolhi abordar foi LLM's aplicadas ao mercado financeiro, o meu planejamento anterior foi buscar na literatura, em artigos, blogs, sites, etc, como os LLMs estão sendo utilizados no setor financeiro, para que assim fosse possível definir o objetivo final relacionado ao tema escolhido. Nesse sentido, fiz essa busca proposta e colhi informações a respeito do que já existe em relação ao meu tema, o que me auxiliou a definir de fato o rumo que irei seguir em relação ao meu tema, que é realizar o fine tuning de um LLM para a área financeira, em português, de modo que, a partir do resultado, seja possível criar aplicações financeiras com tal tecnologia, como: assistente de investimentos, otimizador de portfólio, analista de relatórios e educador financeiro. No link a seguir, disponibilizo um docs em que coloquei todos os links do que li e algumas referências desses links, além disso coloquei um resumo sobre o que se trata cada link: Revisão - Financial LLMs

A outra parte da entrega consistiu na classificação do meu tema em relação aos termos que aparecem no CSCI 2023. Dessa forma, os termos pertinentes ao tema são: Natural language processing, Intelligent information systems, Neural networks and applications, Applications (finance), Unsupervised and Supervised Learning, Aspects of natural language processing.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega, pretendo criar um cronograma de etapas a serem cumpridas até o final da residência, visando o objetivo final em relação ao meu tema. Além da criação desse cronograma, pretendo realizar uma revisão técnica acerca do tema NLP e LLMs, o que seria, portanto, a primeira etapa do cronograma a ser proposto.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

["Revisão - Financial LLMS" citado no Termo de Aceite de Entrega de 19 de outubro de 2023]

Links lidos

- Financial LLMs: FinGPT & BloombergGPT - https://medium.com/@ashkangolgoon/financial-llms-fingpt-bloomberggpt-82dda11a6c05

  Apresenta dois LLMs com conhecimento financeiro, obtido através de fine tuning de modelos base, BloombergGPT que é um modelo privado criado pela Bloomberg com dados financeiros públicos e privados, ambos acurados por sua equipe, e o FinGPT que é um modelo Open-Source.

- Beginner's Guide to FinGPT: Training with LoRA and ChatGLM2–6B - https://byfintech.medium.com/beginners-guide-to-fingpt-training-with-lora-chatglm2-6b-9eb5ace7fe99

  É mais um guia técnico, demonstrando como realizar, a nível de código, o fine tuning de modelos base, como ChatGLM2-6B, buscando o conhecimento financeiro, como o FinGPT.

- Bloomberg & JHU's BloombergGPT: 'A Best-in-Class LLM for Financial NLP' - https://medium.com/syncedreview/bloomberg-jhus-bloomberggpt-a-best-in-class-llm-for-financial-nlp-ede1ecedf0a6

  Explica com um pouco mais de detalhes o modelo BloombergGPT

- Use-Case # 2.1 : Financial Fundamental Analysis Using LLM + RAG (Part I , TSLA case study) -

https://medium.com/betaflow/use-case-2-1-financial-fundamental-analysis-using-llm-rag-part-i-tsla-case-study-53e86f12c338

Discorre sobre a realização de análise fundamental a nível financeiro, utilizando LLM e RAG. O interessante é que é demonstrado uma outra alternativa de aplicação de LLMs no mercado financeiro, mas sem realizar fine tuning.

- Have You Met FinGPT? A New Open-Source Financial Large Language Model - https://odsc.medium.com/have-you-met-fingpt-a-new-open-source-financial-large-language-model-83597c753a4a

  Explica com um pouco mais de detalhes o modelo FinGPT.

- An AI based stock analyzer using LLM and Langchain - https://wire.insiderfinance.io/an-ai-based-stock-analyzer-using-llm-and-langchain-7f8a62cbcaaa

  Um outro artigo que demonstra uma aplicação financeira com LLMs, sem realizar fine tuning. Nesse caso, a partir de dados de ações e uma LLM para interpretação desses dados, é feita uma análise da ação, se é um bom investimento, se é lucrativo, etc.

- FinGPT: Powering the Future of Finance with 20 Cutting-Edge Applications - https://medium.datadriveninvestor.com/fingpt-powering-the-future-of-finance-with-20-cutting-edge-applications-7c4d082ad3d8

  Demonstra algumas aplicações possibilitadas pelo FinGPT, que é um LLM com fine tuning em dados financeiros. Interessante que é possível criar aplicações tanto com quanto sem fine tuning, porém percebe-se que com fine tuning é possível atingir algo mais amplo.

Papers

- Paper BloombergGPT - https://arxiv.org/abs/2303.17564
- Paper FinGPT - https://arxiv.org/abs/2306.06031

Outros

- Repositório GitHub do FinGPT - https://github.com/AI4Finance-Foundation/FinGPT

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 26 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Para essa entrega, foi feito um cronograma de etapas a serem seguidas durante a residência, visando atingir meu objetivo final de treinar um LLM em dados financeiros. Esse cronograma pode ser obtido no link: Planejamento - Residência em IA

Além disso, juntamente ao cronograma, foi feita uma revisão geral do tema de NLP e LLM, mas é importante ressaltar que são temas amplos e que nem tudo foi revisado, apenas aquilo que considero importante para o meu tema na residência e que irá me fornecer uma boa base futura. Essa revisão geral pode ser acessada em: Revisão geral - NLP e LLM

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Buscando seguir o cronograma proposto por mim, para a próxima entrega, planejo realizar as seguintes atividades:

- Estudo do mercado financeiro, a partir de uma ótica de inovação e tecnologia.
- Busca por datasets financeiros.
- Busca por benchmarks financeiros para avaliação dos modelos.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

["Revisão geral - NLP e LLM" citado no Termo de Entrega do dia 26 de outubro]

1. Introdução

A seguir, será feita uma revisão geral acerca dos temas Natural Language Processing (NLP) e Large Language Models (LLM). Nesse sentido, destaca-se que NLP é um campo amplo e, portanto, os tópicos abordados são apenas os tópicos que considero relevantes para o meu objetivo final e que, assim, possam me dar uma base para os próximos passos.

2. Pré-processamento

Os dados textuais, que são trabalhados em NLP, são dados não estruturados, o que torna difícil analisar e extrair informações significantes desses dados. Dessa forma, o pré-processamento ajuda a estruturar tais dados, facilitando a análise e extração de informações, além de preparar tais dados para a entrada em modelos de Machine Learning e Deep Learning. Sendo assim, algumas técnicas de pré-processamento serão apresentadas a seguir, mas é importante lembrar que nem sempre todas são utilizadas e que a utilização de cada uma depende do problema.

- **Tokenização**: consiste em dividir o texto em partes menores, chamadas tokens. Existem diversos tipos de tokenizadores, cada um com sua estratégia de divisão, podendo ela ser por caracteres, subpalavras, bytes, etc.

- **Remoção de caracteres especiais**: esse pré-processamento visa remover pontuações, como "!", ".", ":", dentre outros, mas também remover caracteres especiais, como tags HTML, que podem surgir em dados crawleados.

- **Letra minúscula**: essa técnica converte todo o texto para letras minúsculas, de modo que palavras iguais não sejam consideradas diferentes apenas porque uma tem letra maiúscula e outra não.

- **Remoção de stopwords**: stopwords são palavras que não tem um sentido em si, mas ajudam na construção da semântica do texto. Alguns exemplos de stopwords são artigos e preposições, como: "a", "o", "de", "para", etc. Em algumas tarefas, como classificação, é comum remover stopwords, já que elas não agregam em

sentido por si só, porém em outras tarefas, como geração de texto, as stopwords são mantidas para que a geração textual seja a mais natural possível.

- **Stemming**: consiste em converter as palavras para a sua forma radical. Por exemplo, as palavras "estudei", "estudo" e "estudarei" seriam convertidas para "estud".

- **Lematização**: a ideia da lematização é a mesma do stemming, porém, enquanto o stemming apenas remove o sufixo das palavras, a lematização converte as palavras para sua forma radical com uma base gramatical por trás, de modo que os radicais obtidos realmente sejam os radicais reais.

3. Representação de palavras

Sabe-se que o computador, como uma máquina de processamento, processa os dados de forma binária, ou seja, a partir de dados numéricos. Por isso, ao trabalhar com dados textuais, é necessário converter o texto em uma representação numérica. Porém, essa é uma atividade complexa, visto que tal representação numérica precisa conter as características textuais de sintaxe, morfologia, semântica, contexto, dentre outros. Dessa forma, surgem três grandes formas de se representar o texto de forma numérica: Representações clássicas, Representações densas, Representações contextuais.

3.1. Representações clássicas

As representações clássicas são formas mais rudimentares de representar o conhecimento textual de forma numérica. A seguir, algumas formas de representação:

- One-Hot Encoding: consiste em representar a palavra a partir de um vetor do tamanho do vocabulário, em que onde a posição equivale a posição da palavra no vetor o elemento equivale a 1 e no resto das posições cada elemento vale 0.
    - Essa forma de representar é muito esparsa, ocupa muita memória e apenas indica a presença da palavra, não há informação de sintaxe, morfologia, semântica, contexto e similaridade.

- Bag of Words: a ideia é a mesma do One-Hot Encoding, mas além de indicar a presença da palavra, indica quantas vezes a palavra apareceu no valor dos elementos do vetor.
    - Essa forma de representar, também, é muito esparsa, ocupa muita memória e apenas indica a presença da palavra, não há informação de sintaxe, morfologia, semântica, contexto e similaridade.

- Feature Engineering: corresponde a criação de features que são extraídas a partir do texto, de modo que a palavra passa a ser representada pelo conjunto de features.
    - É um processo extremamente manual e a inserção de novas features gera a necessidade de reanotar cada palavra do conjunto de dados.

- TF-IDF: é uma forma de representar que envolve a razão da frequência da palavra dentro de um documento, pelo inverso da quantidade de documentos que contém a palavra. O cálculo é feito a partir da seguinte fórmula:

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

Figura 1: fórmula TF-IDF.

- Latent Semantic Analysis: basicamente aplica SVD em vetores TF-IDF, de modo que a dimensionalidade é reduzida juntamente com a esparsidade, conservando as características mais relevantes.

3.2. Representações densas

As formas de representações densas são caracterizadas por resolver o problema da esparsidade das representações clássicas, de modo que as características mais relevantes do texto são captadas em uma dimensionalidade menor. Além disso,

destaca-se que as formas de construir tais representações geram características latentes, ou seja, são características abstratas a respeito do texto e não tão convencionais ou intuitivas. Através de tais representações, torna-se possível analisar a similaridade semântica de duas palavras e fazer analogias.

- Word2Vec: é uma técnica que pode ser aplicada através de duas abordagens, Skip-Gram e Continuous Bag of Words, para a obtenção de representações densas de palavras. Essa técnica é uma técnica neural, ou seja, envolve a utilização de redes neurais para a obtenção das representações.
    - Skip-Gram: consiste em, a partir de uma frase, utilizar uma palavra alvo para prever as palavras do contexto. As representações de cada palavra são os pesos da matriz de entrada do vocabulário, após o processo de treino.

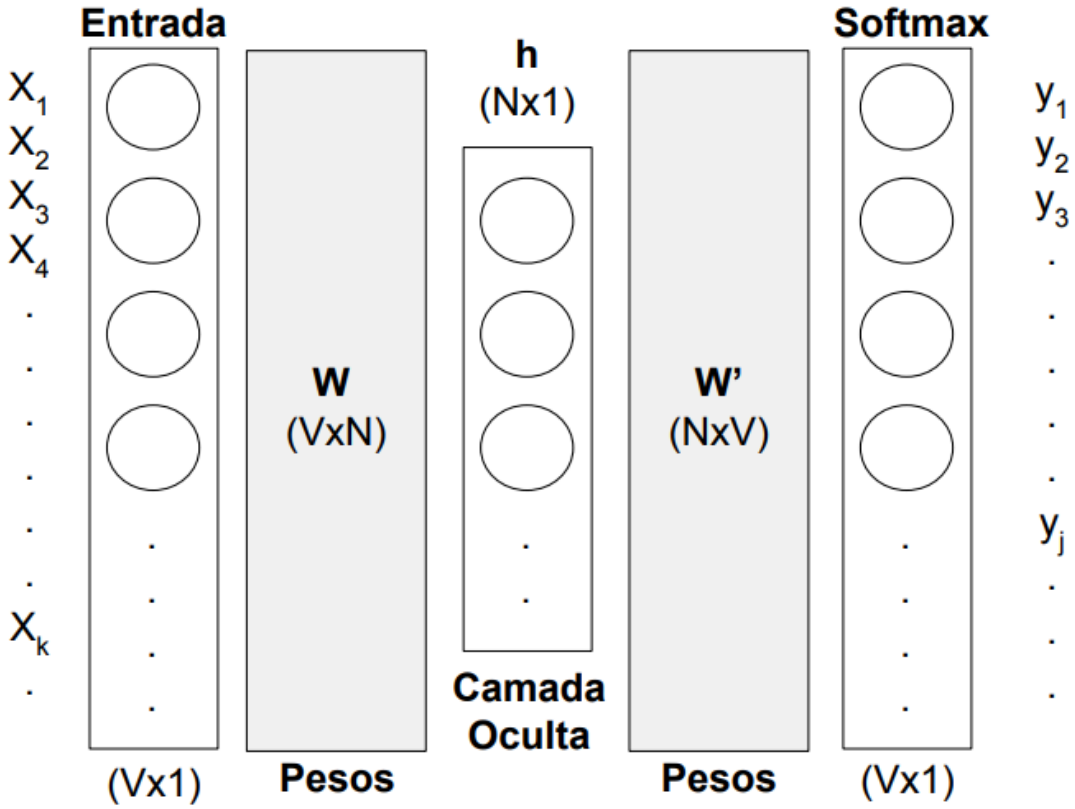


Figura 2: Arquitetura e esquema do Skip-Gram.

- Continuous Bag of Words: é uma abordagem contrária à Skip-Gram, de modo que, a partir do contexto, a palavra alvo é predita. A forma de obtenção da representação é a mesma do Skip-Gram.

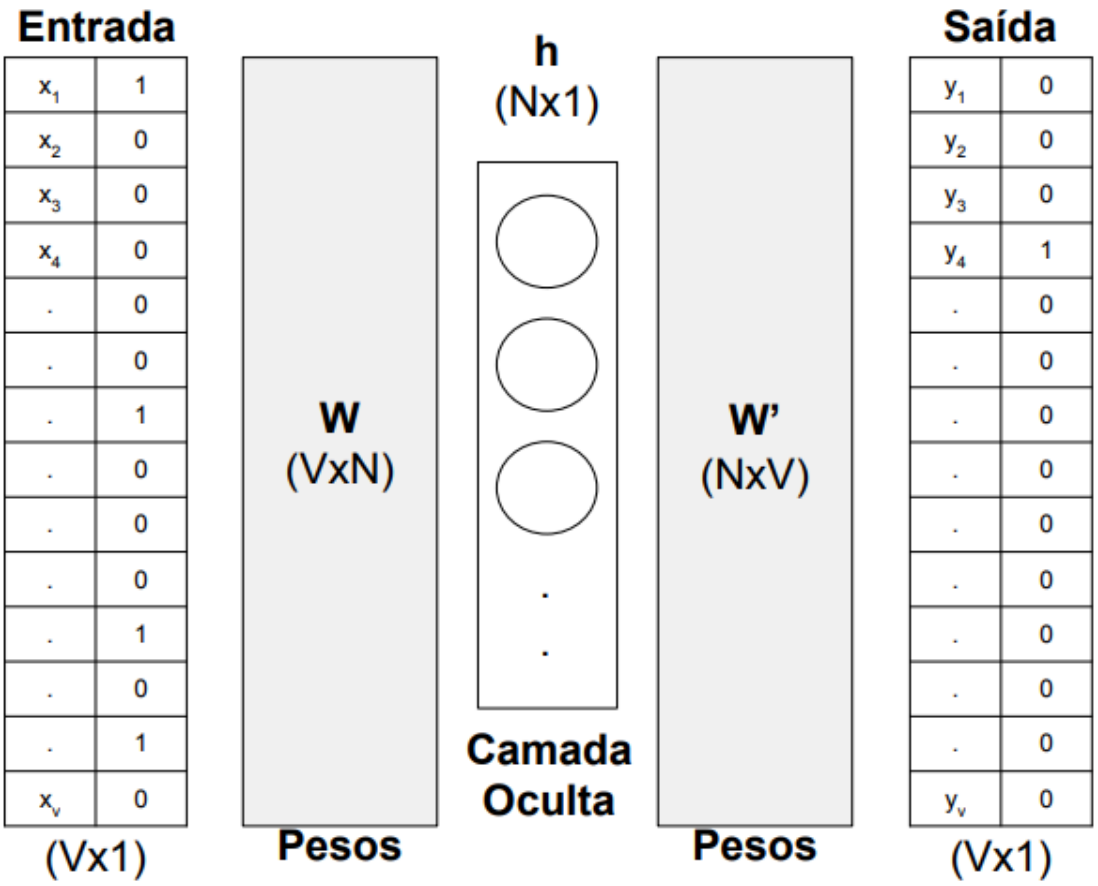


Figura 3: Arquitetura e esquema do Continuous Bag of Words.

- Wang2Vec: abordagem semelhante ao Word2Vec, mas com algumas adaptações para focar na ordem das palavras.

- GloVe: cria a representação a partir das taxas de probabilidades condicionais de co-ocorrências entre as palavras, de acordo com a fórmula:

$$GloVe = -\sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{i,j})(\log X_{i,j} - w_i^T w_j)^2$$

Todas essas formas de representar ainda possui limitações, são elas: polissemia, visto que as representações geradas são estáticas, as representações não cobrem todos os sentidos possíveis para uma palavra e, além disso, por serem estáticos, os vetores de representação não são adaptados para novos domínios.

OBS: aqui foi dito representação de palavras, mas existem algoritmos que trabalham na construção de representações de caracteres e subpalavras, de modo que a representação da palavra é construída a partir da junção de representações de caracteres e/ou subpalavras. Além disso, também é possível unir representações de palavras (seja concatenando, com média, ou alguma combinação linear) para representar frases e documentos.

3.3. Representações contextuais

As representações contextuais visam resolver os problemas remanescentes das representações densas, que são a falta de adaptação para outros domínios e ausência de cobertura da polissemia.

Antes de apresentar uma das técnicas utilizadas para a criação de representações contextuais, é necessário introduzir o que é um modelo de linguagem. Um modelo de linguagem é uma tarefa de NLP que, a partir de um texto de entrada, o modelo tenta realizar a predição da próxima palavra.

Nesse sentido, uma das técnicas responsáveis pela extração de representações contextuais é o ELMo (Embeddings from Language Models)

A ideia do ELMo é utilizar um modelo de linguagem bidirecional, formado por CNNs e biLSTMs. Dessa forma, a partir dos pesos resultantes das camadas de projeção, após o treino, as representações são extraídas.

$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big)$$

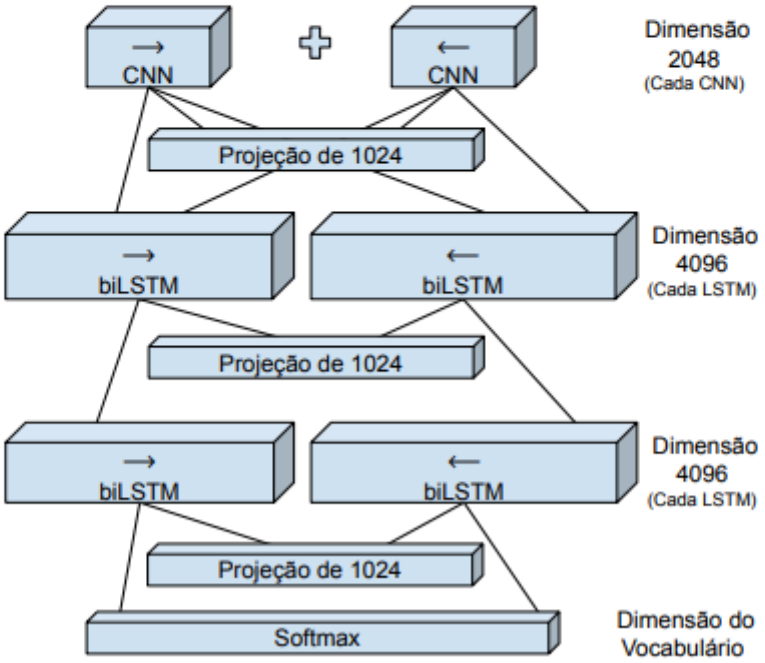


Figura 4: Arquitetura do ELMo.

OBS: essa é apenas uma forma de gerar representações contextuais, visto que, após o surgimento do mecanismo de atenção e dos Transformers, outras formas também são apresentadas.

4. Mecanismo de atenção

O mecanismo de atenção, como o nome sugere, é um forma de auxiliar algoritmos a focar naquilo que mais importa e, sendo assim, tal mecanismo surge como uma forma de melhorar a tarefa de Machine Translation, que é a tarefa de NLP responsável por realizar a tradução textual.

No início de tudo, essa tarefa era feita utilizando dicionários e correspondências de palavras, ao evoluir ela passa a ser feita de forma estatística e, em uma evolução posterior, passa a ser feita de forma neural, utilizando Redes Neurais Recorrentes no modelo seq2seq.

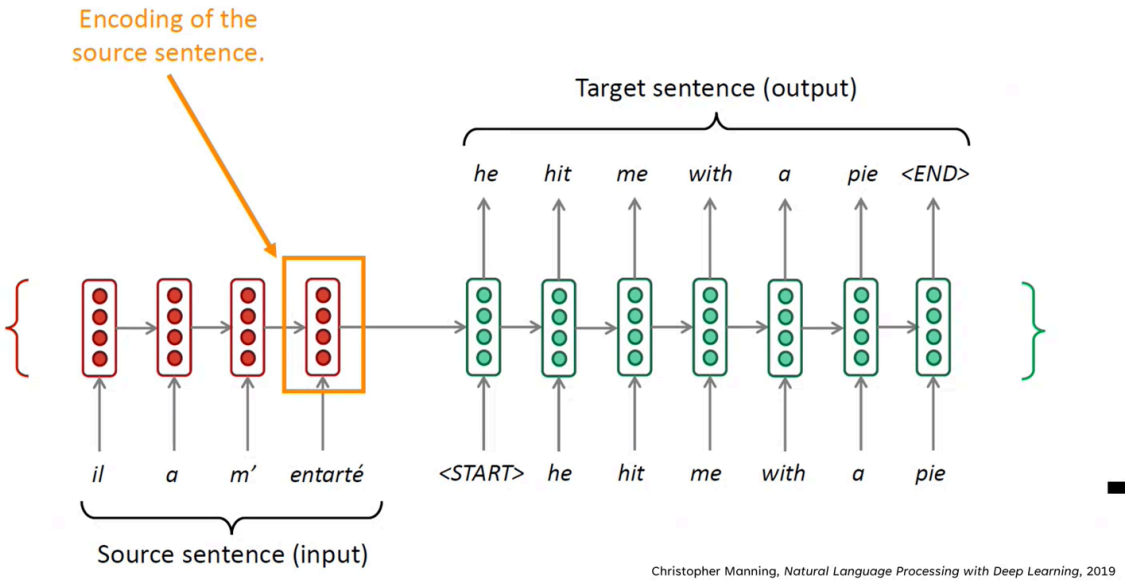


Figura 5: realização do Machine Translation com seq2seq.

No entanto, tais redes recorrentes, por serem sequenciais, possuíam dificuldades em armazenar contexto de grandes sequências textuais. Dessa forma, o mecanismo de atenção surge como uma forma de auxiliar o algoritmo a focar apenas na parte mais importante do texto para a realização da tarefa em questão. Então, para o cálculo do mecanismo de atenção no seq2seq, é calculado o produto interno de uma representação para todas as outras, os valores obtidos são normalizados e formam um vetor a ser concatenado com a representação em que o algoritmo está trabalhando.

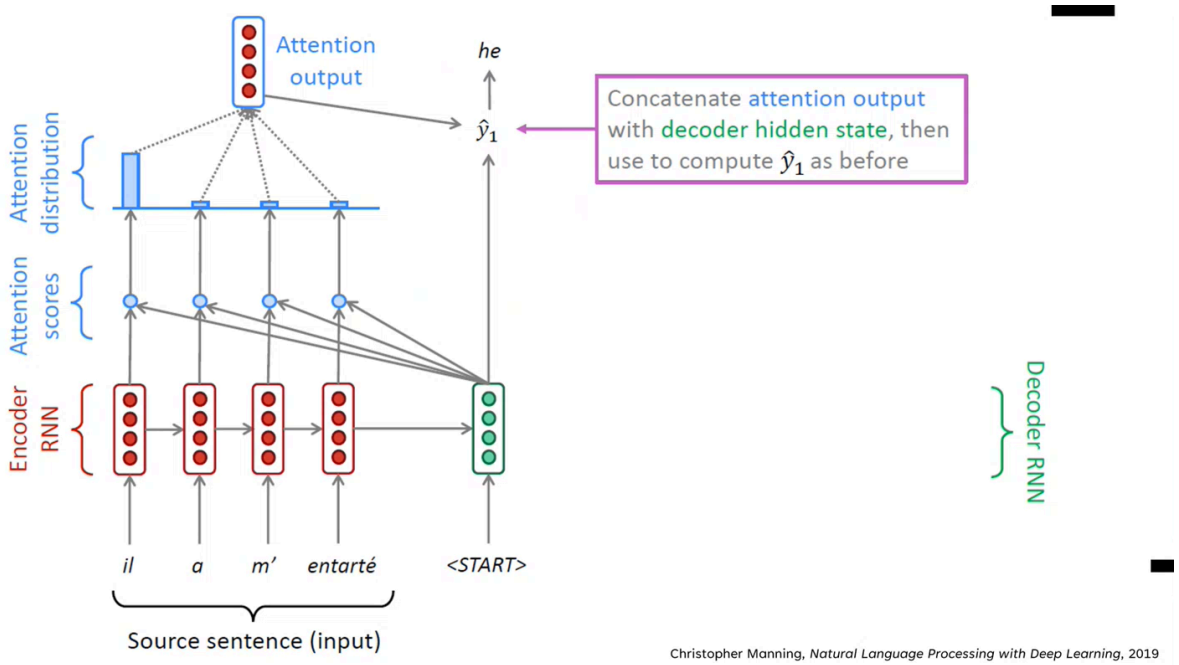


Figura 6: Aplicação do mecanismo de atenção no seq2seq.

Sendo assim, o mecanismo de atenção passa a ser introduzido em algoritmos neurais, mas as redes recorrentes ainda possuíam outros problemas: mesmo com atenção o contexto ainda não era captado totalmente, ocorria o Gradient Vanishing/Exploding e as redes recorrentes não são paralelizáveis; tudo isso leva ao surgimento dos Transformers.

5. Transformers

A arquitetura Transformer é uma arquitetura encoder-decoder que revoluciona a Inteligência Artificial ao criar uma forma de trabalhar com dados sequenciais (ou não), captando contextos de longas sequências, sem Gradient Vanishing/Exploding, com menos etapas de treino e de forma paralelizável, em uma arquitetura predominada pelo mecanismo de atenção.

A seguir, será introduzida a imagem da arquitetura e cada ponto dessa arquitetura será explicado.

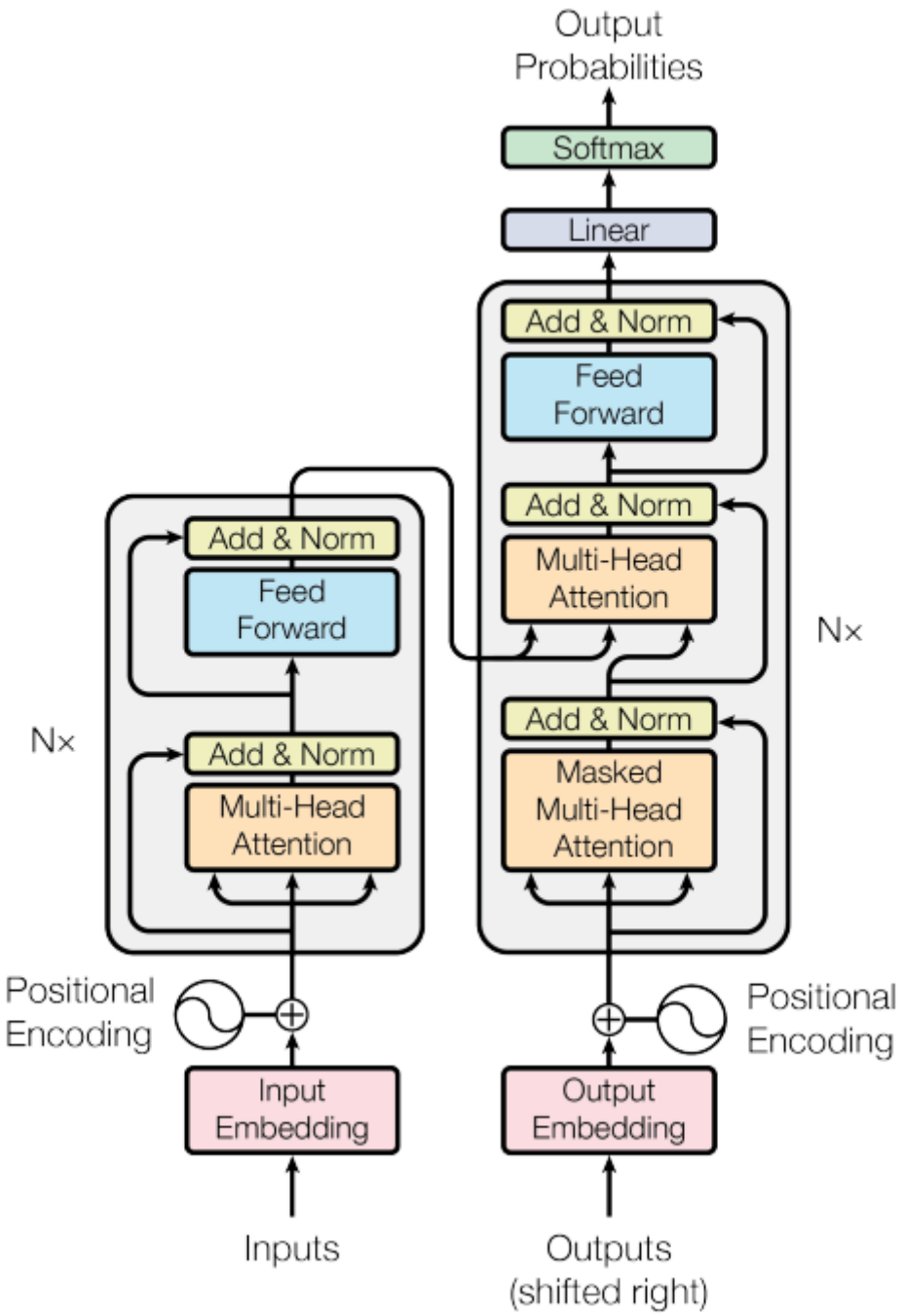


Figura 7 - Arquitetura Transformer.

A parte esquerda da arquitetura é a parte de Encoder, que irá codificar a entrada, retornando um vetor/embedding contextual da entrada. Dessa forma, irei detalhar cada parte do Encoder:

- **Input embedding**: nessa etapa, a entrada é tokenizada e os embeddings de cada token é calculado.

- **Positional encoding**: como a arquitetura Transformer trabalha com dados sequenciais em uma arquitetura não sequencial, de alguma forma é preciso inserir a ideia de sequência nos dados. Então, essa etapa faz isso ao adaptar o embedding da etapa anterior com a informação de posição e, assim, dar a ideia de sequência.

- Multi-Head Attention: a figura abaixo irá auxiliar na explicação. A partir do embedding da entrada, já com a informação de sequência, essa entrada será replicada em 3 matrizes: V, K, Q. Cada matriz dessa irá passar por uma camada linear, de modo a transformar levemente a entrada original. Então, as matrizes V, K e Q nada mais são que a entrada original transformadas levemente.

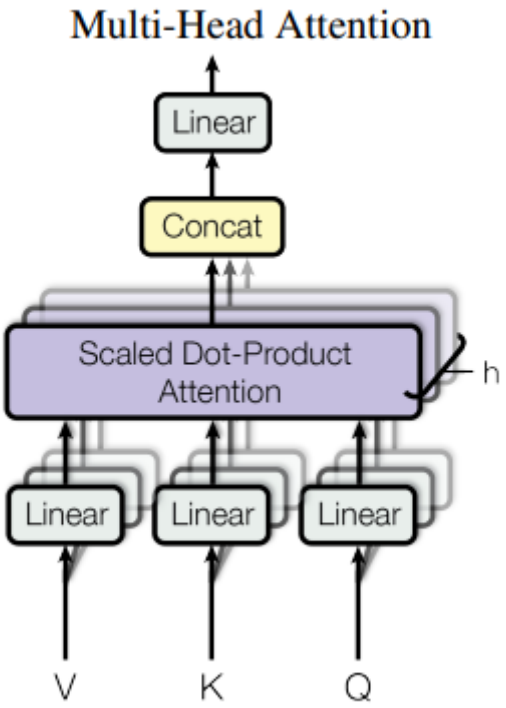


Figura 8 - Multi-Head Attention

  - Após isso, tais matrizes estarão envolvidas nos cálculos da figura abaixo. De forma teórica, esses cálculos basicamente permitem que cada token da entrada seja ressignificado e passe a ser representado

em relação aos outros tokens, ou seja, cada token passa a ter a informação de quanto os outros tokens são importantes para ele (aqui que acontece o mecanismo de atenção).

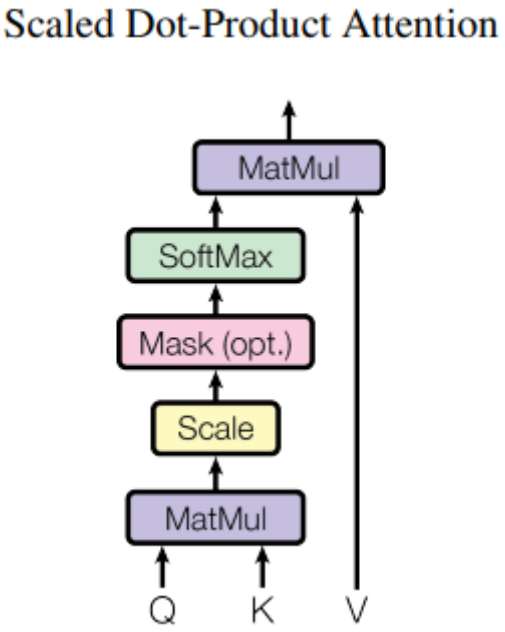


Figura 9 - Cálculo da atenção

- Tudo isso acontece em apenas uma "cabeça" de atenção, o Multi-Head Attention é todo esse processo repetido para diversas "cabeças" de atenção. Ao fim, os vetores resultantes de todas as "cabeças" são concatenados e passam por uma camada linear.

- **Add & Norm**: são conexões residuais, que auxiliam na solução do Gradient Vanishing/Exploding, e normalização, que é o que faz com que o treino exija menos etapas de treino.

- **Feed Forward**: é uma rede neural normal que insere mais parâmetros na arquitetura, permitindo que a representação aprendida se torne mais rica.

Dessa forma, após passar pelo Encoder, é retornada uma representação rica e contextual da entrada.

Agora, a parte direita, que é o Decoder, é responsável por decodificar e gerar o que vem após a entrada. As partes do Decoder são praticamente as mesmas que do Encoder, com apenas algumas diferenças:

- **Outputs (shifted right)**: aqui a entrada do decoder é deslocada para a direita para a inserção de tokens que indicam onde inicia a decodificação e onde termina, que são tokens <sos> (start of sentence) e <eos> (end of sentence).

- **Masked Multi-Head Attention**: a ideia é a mesma do Multi-Head Attention, com a diferença de que, como busca-se decodificar e gerar um novo texto, se o algoritmo "olhasse" o que vem após o texto de entrada ele estaria tendo vantagem e, portanto, não aprenderia nada. Desse modo, então, os tokens que vem após o token que está sendo decodificado são mascarados para que o algoritmo não "olhe" para eles.

- **O segundo Multi-Head Attention**: nesse Multi-Head Attention, ao invés das matrizes V, K e Q serem originárias da entrada do Decoder, as matrizes K e V vêm da representação codificada do Encoder e a matriz Q vem do Decoder.

- **Linear e Softmax**: ao fim do processo é inserida uma camada linear que trabalha em cima das representações retornadas pelo decoder e retorna uma saída que é normalizada pela camada Softmax. Essa saída irá depender do problema a ser trabalhado, mas para a geração textual pode ser as probabilidades relacionadas a próxima palavra a ser predita.

6. Large Language Models (LLM)

6.1. O que é?

Como foi abordado na seção 3.3, um modelo de linguagem é um modelo de NLP que busca realizar a predição do próximo token, a partir de um token de entrada. Ou seja, um modelo de linguagem é um gerador de textos.

Nesse sentido, nota-se uma evolução acerca dos modelos de linguagens, que foram de geradores de textos sem sentidos e mal escritos para geradores de

textos que escrevem perfeitamente e de forma humana. Essa evolução também caracterizou a transição de modelos de linguagem comuns para grandes modelos de linguagem e o que permitiu essa evolução foram dois fatores: tamanho do modelo e qualidade dos dados.

Dessa forma, os LLMs são "apenas" modelos de linguagem bem grandes, o que permite uma maior captação de padrões e relações textuais e, consequentemente, melhora o aprendizado, que foram treinados em dados de maior qualidade, que foram melhores tratados e limpos, o que também facilita o aprendizado.

6.2. Pré-treino e treino

Os LLMs costumam ser pré-treinados em uma grande quantidade de dados e o objetivo desse pré-treino é obter conhecimento de como um idioma funciona, a sintaxe, a morfologia, a semântica, de modo que a geração textual seja possibilitada.

Um LLM pré-treinado em uma língua pode ser reutilizado, através de ajuste fino, para outra língua. No entanto, caso a estrutura das línguas seja muito diferente, como por exemplo língua portuguesa e língua musical, é necessário que esse LLM seja treinado novamente na língua musical, para que ele possa aprender as estruturas dessa língua.

6.3. Instruction Finetuning

No entanto, apesar de tudo isso, no final das contas os LLMs acabam sendo "apenas" preditores do próximo token. Então, como fazer com que esses LLMs sejam capazes de conversar de uma forma humana, possam responder perguntas, dentre outras diversas coisas que eles são conhecidos por fazer?

Nesse sentido, é possível realizar o fine tuning do modelo em dados instrucionais. Ou seja, sabendo que o LLM já foi pré-treinado e, portanto,

conhece a estrutura da língua e é capaz de predizer o próximo token, agora ele vai ser ajustado em dados instrucionais que possuem o formato (Instrução (prompt), Resposta esperada). Sendo assim, o LLM além de gerar o próximo token, é capaz de gerar o próximo token a partir de uma instrução e, consequentemente, passa a ser capaz de conversar, responder perguntas, obedecer instruções, etc.

Uma forma de fazer um Instruction Fine Tuning é utilizar humanos para isso, através do Reinforcement Learning from Human Feedback (RLHF), como foi feito no ChatGPT (esquema apresentado na figura abaixo). Nessa técnica, um prompt (instrução) é passado para o LLM, juntamente com a resposta esperada e o ajuste fino é feito; após isso, um modelo de recompensa será treinado, de modo que um anotador humano irá anotar quais respostas são melhores e quais as piores e, então, o modelo de recompensa vai aprender a avaliar uma resposta a partir de uma instrução. Por fim, o LLM ajustado anteriormente será otimizado utilizando aprendizado por reforço, de modo que ele irá receber uma instrução, gerar uma resposta, o modelo de recompensa irá dar uma recompensa para aquela resposta e essa recompensa irá ajudar o LLM em seu aprendizado.

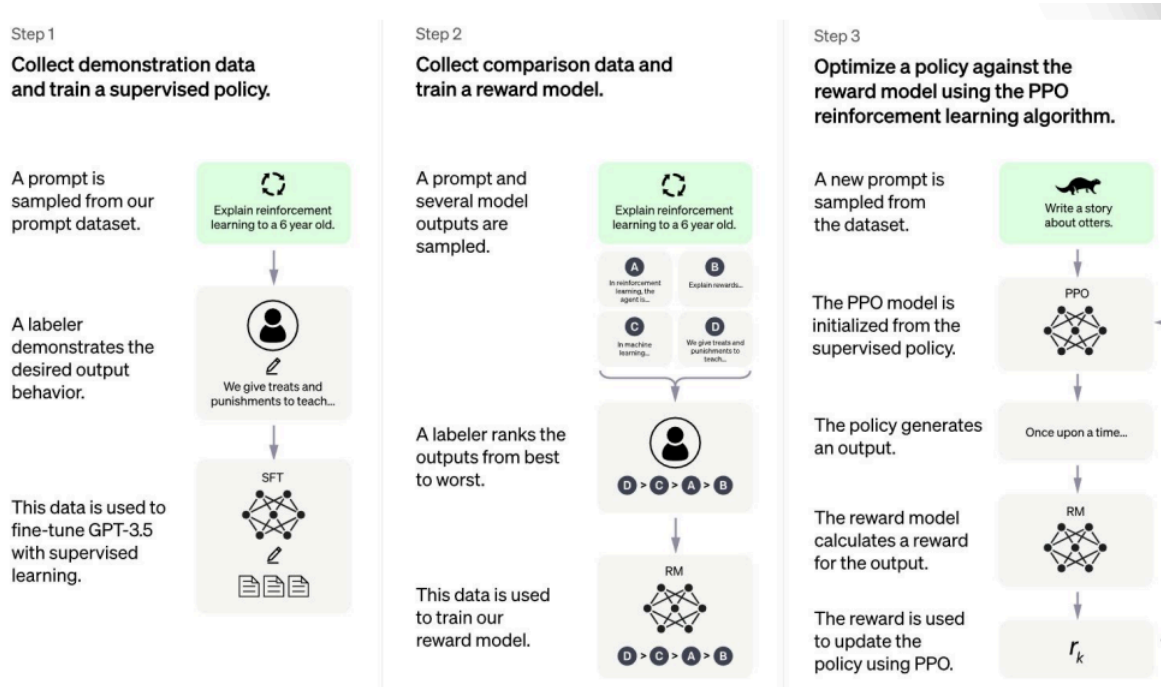


Figura 10 - Esquema de treinamento do ChatGPT.

7. Referências

LI, H. A Tutorial on LLM. Disponível em: <https://medium.com/@haifengl/a-tutorial-to-llm-f78dd4e82efc>. Acesso em: 25 out. 2023.

DROST, D. Different ways of training LLMs. Disponível em: <https://towardsdatascience.com/different-ways-of-training-llms-c57885f388ed>.

MATSUDA, A. NLP: Preprocessing text data (Part 1). Disponível em: <https://kazumatsuda.medium.com/nlp-preprocessing-text-data-part-1-b4641af2a5af>. Acesso em: 25 out. 2023.

TAUNK, D. NLP Preprocessing:- A useful and important step. Disponível em: <https://medium.com/analytics-vidhya/nlp-preprocessing-a-useful-and-important-step-e79895c65a89>. Acesso em: 25 out. 2023.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of Word2Vec for syntax problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1299–1304, Denver, Colorado. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. CoRR.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. J. Mach. Learn. Res., 3:1137–1155.

VASWANI, A. et al. Attention Is All You Need. [s.l: s.n.].

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Decidi mudar minha área de aplicação de LLMs e fui para a área da saúde (que já era algo que eu pensava antes da residência). Em seguida, iniciei uma busca por datasets de saúde em português e benchmarks, mas, novamente, encontrei apenas em inglês, porém, em compensação, os dados são mais consistentes e os benchmarks encontrados mais consolidados e padronizados. Os datasets e benchmarks selecionados, com maiores explicações, estão no documento: Datasets e benchmarks.
- Os datasets selecionados são os mesmos que foram utilizados no treinamento do Med-PaLM 2 (**SINGHAL, K. et al. Towards Expert-Level Medical Question Answering with Large Language Models. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.09617.pdf>.**), que é um LLM ajustado em dados da saúde que atingiu o estado da arte em diversos benchmarks. Além disso, os benchmarks selecionados também são os mesmos utilizados pelo Med-PaLM 2, que faz parte do MultiMedQA (**SINGHAL, K. et al. Large Language Models Encode Clinical Knowledge. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2212.13138.pdf>.**), que é um benchmark que inclui datasets de resposta a perguntas de múltipla escolha, datasets que exigem respostas mais longas para perguntas de profissionais médicos e datasets que exigem respostas de formulário mais longas para perguntas que podem ser feitas por não profissionais.
- Como fruto da alteração citada, o meu planejamento de residência também foi atualizado e pode ser encontrado em: Planejamento - Residência em IA.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Buscando seguir o cronograma proposto por mim, para a próxima entrega, planejo realizar as seguintes atividades:

- Preparação dos dados de saúde (tradução) captados para o treino/fine tuning de LLM.

- Curadoria dos dados (verificar qualidade pós preparação).
- Preparação do benchmark (tradução e código para calcular as métricas do benchmark de forma automatizada).

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

- No último gate, meus objetivos eram: estudar o mercado financeiro, procurar datasets financeiros em português e benchmarks financeiros. Comecei procurando datasets e benchmarks, visto que se não achasse, não faria sentido me aprofundar no mercado financeiro.
- Como resultado da minha busca, encontrei apenas datasets em inglês e quase nenhum benchmark, sendo que os que encontrei, no paper (**WU, S. et al. BloombergGPT: A Large Language Model for Finance. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2303.17564.pdf>.**), reveleram-se insuficientes, visto que não há uma forma padrão de testar tais benchmarks, já que a aplicação de LLMs na área financeira não foi muito reportada até o momento. Como prova disso, segue um trecho do mesmo paper: "Our public financial benchmarks include four tasks from the FLUE benchmark (Shah et al., 2022) and the ConvFinQA dataset (Chen et al., 2022). As LLM performance on most of these financial tasks have not been broadly reported, there is no standard testing framework. Thus, we adapt them to a few-shot setting.".
- Após a seleção dos datasets e benchmarks de saúde, ainda estava em dúvida sobre o que fazer com o idioma, já que queria algo em português e aí tive a ideia de traduzir os dados, mas fiquei inseguro pois não sei se faria sentido. Porém, após o Conecta CEIA, em que o Adalberto fez uma apresentação que citou que estava fazendo o finetuning de um LLM com dados traduzidos, eu fui falar com ele e obtive apoio e conselhos em relação às minhas ideias. Sendo assim, decidi seguir nesse rumo: mudar a aplicação para a área da saúde, utilizar os datasets e benchmarks escolhidos e traduzi-los.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Datasets e benchmarks” citado no Termo de Aceite de Entrega do dia 9 de novembro de 2023]

Datasets escolhidos

GBaker/MedQA-USMLE-4-options · Datasets at Hugging Face - Dataset MedQA (USMLE), que contém conhecimentos médicos gerais do exame de licenciamento médico dos EUA.

medmcqa · Datasets at Hugging Face - Dataset MedMCQA, que contém conhecimentos médicos gerais de vestibulares de medicina indianos.

truehealth/liveqa · Datasets at Hugging Face - Dataset LiveQA, que contém dúvidas de conhecimentos médicos gerais, provenientes de pessoas que não são da área.

truehealth/medicationqa · Datasets at Hugging Face - Dataset MedicationQA, que contém dúvidas frequentes sobre medicamentos, provenientes de pessoas que não são da área.

Benchmarks

GBaker/MedQA-USMLE-4-options · Datasets at Hugging Face - Divisão do dataset MedQA (USMLE), que foi explicado acima.

medmcqa · Datasets at Hugging Face - Divisão do dataset MedMCQA, que foi explicado acima.

hippocrates/pubmedqa_test · Datasets at Hugging Face - Divisão de teste do dataset PubMedQA, que contém dados da literatura científica de biomedicina.

cais/mmlu · Datasets at Hugging Face - Divisão de teste do dataset MMLU, que cobre questões de múltipla escolha acerca de conhecimento médico, cobrindo os seguintes temas: anatomia, conhecimento clínico, questões de faculdade de medicina, genética médica, questões medicina profissional e biologia universitária.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 16 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Na semana anterior, meu planejamento era preparar (traduzir) os datasets, tanto de treino, quanto dos datasets utilizados no benchmark, além de criar um algoritmo para calcular as métricas do benchmark com base nos dados passados, mas também realizar a curadoria dos dados preparados.
- Sendo assim, conduzi a preparação (utilizando o ChatGPT como tradutor) de tais dados e os disponibilizei no Hugging Face: luizlzg/drbyte_dataset · Datasets at Hugging Face. Além disso, criei um repositório no GitHub para armazenar todos os algoritmos utilizados na preparação dos dados, além dos algoritmos também gerados para a entrega (algoritmo para calcular as métricas do benchmark com base nos dados passados). O repositório se encontra no link: luizlzg/ResidenciaIA (github.com).
- Em relação à curadoria, ela não foi realizada em todo dataset; o que aconteceu foi uma inspeção manual inicial, em 10% da primeira versão do dataset, que foi criada com o Google Tradutor. A partir dessa inspeção manual, pude coletar os erros mais comuns e, assim, utilizei da capacidade instrutiva do ChatGPT para solicitar as traduções, mas pedindo um cuidado maior aos erros mais comuns encontrados.
- Por fim, detalhei um pouco mais do que foi produzido no seguinte documento: Entrega - Gate 16/11/23.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Buscando seguir o cronograma proposto por mim, para a próxima entrega, planejo realizar as seguintes atividades:

- Estudo de frameworks para fine tuning de LLMs.

- Estudo de estratégias possíveis para o fine tuning de um LLM.
- Avaliação e definição da estratégia a ser seguida.
- Busca e definição de quais LLMs serão utilizadas para o fine tuning (2-3).
- Criação do pipeline de fine tuning, com as devidas formatações de dados.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Entrega - Gate 16/11/23” citado no Termo de Aceite de Entrega do dia 16 de novembro de 2023]

Conjunto de dados

Como planejado, o conjunto de dados, formado pelos conjuntos de dados citados na entrega anterior, foi criado, através da tradução dos dados do inglês para o português com o ChatGPT.

Os dados podem ser encontrados em: luizlzg/drbyte_dataset · Datasets at Hugging Face.

Uma breve descrição a respeito dos dados gerados, que também pode ser encontrada no link acima:

- Divisão de treino:
    - MedQA (USMLE), que contém conhecimentos médicos gerais do exame de licenciamento médico dos EUA. (10082 dados);
    - MedMCQA, que contém conhecimentos médicos gerais de vestibulares de medicina indianos. (9736 dados);
    - LiveQA, que contém dúvidas de conhecimentos médicos gerais, provenientes de pessoas que não são da área. (622 dados);
    - MedicationQA, que contém dúvidas frequentes sobre medicamentos, provenientes de pessoas que não são da área. (687 dados).
    - Total de dados de treino: 21127 dados.
- Divisão de teste (corresponde aos conjuntos de dados do benchmark):
    - MedMCQA (SPLIT DE VALIDAÇÃO), que contém conhecimentos médicos gerais de vestibulares de medicina indianos. (4183 dados);

- MedQA (USMLE) (SPLIT DE TESTE), que contém conhecimentos médicos gerais do exame de licenciamento médico dos EUA. (1273 dados);
- PubMedQA (SPLIT DE TESTE), que contém dados da literatura científica de biomedicina. (500 dados);
- MMLU (SPLIT DE TESTE), que cobre questões de múltipla escolha acerca de conhecimento médico, cobrindo os seguintes temas: anatomia, conhecimento clínico, questões de faculdade de medicina, genética médica, questões medicina profissional e biologia universitária. (1089 dados).
- Total de dados de teste: 7045 dados.

Curadoria de dados

A respeito da checagem da qualidade da transcrição, pela falta de conhecimento de uma forma automatizada, foi conduzida uma inspeção manual de forma inicial, em um rascunho do conjunto de dados, obtido através da tradução com o Google Tradutor. Através dessa inspeção manual, cerca de 10% do conjunto de dados foi observado e, consequentemente, alguns erros comuns foram sendo mapeados.

Sendo assim, tendo conhecimento dos erros mais comuns e sabendo da capacidade de seguir instruções do ChatGPT, decidi utilizá-lo para as traduções e passar em suas instruções os cuidados para não cometer os erros mais comuns que foram mapeados na inspeção manual inicial. Com isso, a versão final e atual do conjunto de dados foi gerada, aparentando ser melhor que a versão com o Google Tradutor, mas afirmo isso apenas por observar algumas amostras, visto que nenhum teste mais amplo foi conduzido.

Algoritmos e repositório

Os algoritmos utilizados para:

- Realizar a tradução e organização dos dados;
- Prompts utilizados para contornar erros possíveis de tradução com o ChatGPT;
- Código para calcular as métricas do benchmark utilizado;

estão no seguinte repositório: luizlzg/ResidencialA (github.com). No mesmo repositório está explicado como ele está organizado.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 23 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Na semana anterior, meu planejamento era:
  - estudo e seleção de frameworks para fine tuning de LLMs;
  - estudo e seleção de estratégias possíveis para o fine tuning de um LLM;
  - busca e definição de quais LLMs serão utilizadas para o fine tuning;
  - criação do pipeline de fine tuning.
- Sendo assim,
  - a estratégia de fine tuning a ser seguida será o Instruction Learning, utilizando o QLoRA;
  - os LLMs a serem utilizados são: Zephyr (7 bi), GPT-2 (1.5 bi) e DistilGPT2 (82 mi);
  - o pipeline de fine tuning foi criado e se encontra no arquivo main.py do repositório luizlzg/ResidenciaIA (github.com).
- Por fim, a seleção de frameworks e o processo de forma mais detalhado estão explicados no documento: Entrega - Gate 23/11/2023 .

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Buscando seguir o cronograma proposto por mim, para a próxima entrega, planejo realizar as seguintes atividades:

- Realização do fine tuning, em dados de saúde, dos LLMs selecionados.

- Comparação dos resultados com o benchmark escolhido.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Entrega - Gate 23/11/2023” citado no Termo de Aceite de Entrega do dia 23 de novembro de 2023]

Métodos de fine tuning

As formas de realizar fine tuning de um LLM se dividem em 3:

- In-Context Learning: basicamente consiste em fornecer informações externas de um certo domínio no contexto do modelo, de modo que ele irá gerar sua resposta com base no que foi passado no contexto. Porém, tal método não altera os pesos do modelo e ele não aprende novas informações.

- Reinforcement Learning from Human Feedback: nessa estratégia, um modelo de recompensa é treinado para avaliar a qualidade das respostas de um LLM, a partir de uma opinião humana. Desse modo, utilizando esse modelo de recompensa, o LLM é treinado utilizando algoritmos de reforço para maximizar a recompensa, ou seja, para gerar respostas que mais agradam humanos.

- Instruction Learning: consiste em ensinar o LLM a gerar respostas com base em instruções, pedidos, perguntas, etc. É através dessa estratégia que é possível criar um algoritmo de perguntas e respostas, com uma conversa fluida e natural.

Para o fine tuning em questão, de um LLM no domínio da saúde, a estratégia selecionada é o Instruction Learning, visando que o modelo aprenda a seguir instruções/perguntas médicas e retorne respostas apropriadas.

Estratégias de Instruction Learning

Essencialmente, a maior mudança no fine tuning com instruções ocorre no formato dos dados. Porém, ao optar pelo Instruction Learning, ainda há algumas estratégias possíveis para se realizar o fine tuning do LLM seguindo essa abordagem. Essas estratégias são:

- Fine tuning total: consiste em realizar o fine tuning em todas as camadas do modelo, de forma completa. Pode levar a um melhor desempenho, porém os custos computacionais são extremamente maiores.

- LoRA: faz parte de um campo maior chamado Parameter Efficient Fine-Tuning (PEFT). Consiste em, ao invés de atualizar todos os pesos do modelo, atualizar apenas um subconjunto de pesos. Para isso, o LoRA (Low-Rank Adapters) substitui a matriz de atualização de pesos por duas outras matrizes menores, obtidas através de uma decomposição de baixo nível. Além disso, todas as outras camadas do modelo são congeladas, sendo treinadas apenas as selecionadas pelo LoRA. Desse modo, é possível obter um desempenho próximo ao obtido quando todo o modelo é treinado, mas com um custo computacional muito menor.

- QLoRA: Em essência, o QLoRA possui a mesma forma de atuação que o LoRA, a mudança está na representação dos dados; o “Q” em “QLoRA” é de “Quantization”, ou seja, a mudança está na quantização dos dados. Desse modo, o QLoRA reduz o nível de precisão na representação dos parâmetros das camadas do modelo. Dessa forma, menos bits são utilizados para representar tais informações e, consequentemente, menos memória é utilizada e o desempenho computacional é maior que no LoRA puro.

Para o fine tuning em questão, que será realizado na residência, visando um fine tuning mais ágil, mas com desempenho semelhante ao treino de um modelo completo, a estratégia escolhida para o Instruction Learning é o QLoRA.

Frameworks para a realização do fine tuning

Para a realização do fine tuning, irei dividir os frameworks em categorias que são necessárias para realizar o fine tuning de acordo com as estratégias escolhidas.

- Obtenção dos LLMs: para a obtenção e instanciação dos LLMs que serão treinados, o framework escolhido foi o transformers do Hugging Face. Não

houve uma busca por outros frameworks que cuidassem da obtenção e instanciação de LLMs, visto que tal framework é o padrão há muito tempo e conta com vários recursos que facilitam a utilização.

- Obtenção dos dados: para a obtenção dos dados, foi utilizado o framework datasets do Hugging Face. Esse framework permite a captação, de forma simples e fácil, dos dados hospedados na plataforma do Hugging Face. A opção por tal framework foi devido a simplicidade de uso, mas também porque meus dados já estão na plataforma.

- Quantização: para a realização da quantização dos parâmetros dos LLMs, visando a redução de memória e um maior desempenho computacional, foi escolhido o framework bitsandbytes, que é utilizado em conjunto com o framework transformers, para que a quantização seja aplicada aos modelos. Para a quantização, também foi verificada a possibilidade de uso do framework AutoGPTQ, porém, além de ser mais lento, ele possui uma quantização pior que o bitsandbytes.
- LoRA: para a aplicação da técnica LoRA, o framework escolhido foi o peft, que permite a aplicação de tal técnica em modelos hospedados no Hugging Face.

- Treino: para a realização do treino, foi utilizado o framework trl, também do Hugging Face, que é próprio e otimizado para o treino/fine tuning de LLMs.

- Acesso a GPU: para a utilização de recursos de GPU, foi utilizado o framework PyTorch, que permite a alocação de modelos e dados em GPU.

LLMs escolhidos

A realização do fine tuning na residência busca comparar o desempenho de modelos de diversos tamanhos. Sendo assim, três LLMs de diversos tamanhos foram selecionados:

- Zephyr: é um modelo que conta com 7 bilhões de parâmetros e ficou extremamente conhecido por superar o LLama 2, com 70 bilhões de parâmetros, no benchmark MT, que mede a eficiência de chatbots.

- OpenAI GPT2: é o modelo GPT-2 da OpenAI, que antecede o GPT-3 que foi a base do famoso ChatGPT. Conta com 1.5 bilhões de parâmetros.

- DistilGPT2: é uma versão menor do GPT-2 da OpenAI, obtida através da técnica knowledge distillation (em que um modelo maior passa conhecimento para um modelo menor). Conta com 82 milhões de parâmetros.

Pipeline de fine tuning

Por fim, o repositório atualizado, contando com os LLMs escolhidos, os frameworks selecionados e o pipeline para a realização do fine tuning de acordo com as técnicas e estratégias escolhidas, podem ser encontrados em: luizlzg/ResidenciaIA (github.com)

Referências

Quantize 🤗 Transformers models. Disponível em: <https://huggingface.co/docs/transformers/main_classes/quantization>.

Making LLMs even more accessible with bitsandbytes, 4-bit quantization and QLoRA. Disponível em: <https://huggingface.co/blog/4bit-transformers-bitsandbytes>.

A Gentle Introduction to 8-bit Matrix Multiplication for transformers at scale using transformers, accelerate and bitsandbytes. Disponível em: <https://huggingface.co/blog/hf-bitsandbytes-integration>.

LoRA. Disponível em: <https://huggingface.co/docs/peft/conceptual_guides/lora>.

MANYI. More about LoraConfig from PEFT. Disponível em: <https://medium.com/@manyi.yim/more-about-loraconfig-from-peft-581cf54643db>. Acesso em: 22 nov. 2023.

Supervised Fine-tuning Trainer. Disponível em: <https://huggingface.co/docs/trl/v0.7.2/en/sft_trainer>. Acesso em: 22 nov. 2023.

MARIE, B. Mistral 7B: Recipes for Fine-tuning and Quantization on Your Computer. Disponível em: <https://towardsdatascience.com/mistral-7b-recipes-for-fine-tuning-and-quantization-on-your-computer-631401583f77>. Acesso em: 22 nov. 2023.

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 30 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Como planejado na última semana, esta entrega está relacionada ao treino de 3 modelos (Zephyr, GPT-2 e DistilGPT-2) em dados da saúde, além da comparação desses com o benchmark escolhido. Sendo assim, todo o processo realizado está descrito no documento Entrega - Gate 30/11/2023.

- Nesse documento, tudo está explicado na seguinte estrutura:
  - Benchmark: descrição geral do benchmark e os datasets que o compõe;
  - Treino: são apresentados os pipelines de forma gráfica, os parâmetros utilizados para treino e as curvas de loss resultantes;
  - Resultado: são apresentados os resultados obtidos no benchmark, em comparação com outros modelos, e, ao final, são feitas algumas observações de como os resultados podem melhorar e como isso vai ser conduzido nos próximos passos.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Alterando um pouco o que estava planejado no cronograma, os próximos passos consistem em:
  - Gerar mais dados;
  - Realizar o fine tuning do Zephyr alterando os parâmetros e com uma maior quantidade de dados;
  - Buscar formas de avaliar a qualidade e completude de resposta dos modelos;

- Buscar formas de avaliar o fine tuning em português para aplicar no GPT-2 e DistilGPT-2;
- Atualização do planejamento da residência.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

[Documento “Entrega - Gate 30/11/2023” citado no Termo de Aceite de Entrega do dia 30 de novembro de 2023]

Benchmark

O benchmark utilizado foi o benchmark MultiMedQA [1], que é um benchmark para perguntas e respostas médicas. A seguir, na tabela 1, estão os datasets que compõem esse benchmark, o formato do dataset, ou seja, o tipo de dado que há nesses datasets, a quantidade de dados utilizada em cada dataset e a explicação do que é o dataset.

Datasets	Formato	Tamanho utilizado (treino/teste)	Explicação
MMLU	Pergunta e resposta, com 4 alternativas.	1089 dados (apenas teste)	Cobre questões de múltipla escolha acerca de conhecimento médico, cobrindo os seguintes temas: anatomia, conhecimento clínico, questões de faculdade de medicina, genética médica, questões medicina profissional e biologia universitária.
MedicationQA	Pergunta + resposta longa	687 dados (apenas treino)	Contém dúvidas frequentes sobre medicamentos, provenientes de pessoas que não são da área.
LiveQA	Pergunta + resposta longa	622 dados (apenas treino)	Contém dúvidas de conhecimentos médicos gerais, provenientes de pessoas que não são da área.
MedMCQA	Pergunta e resposta, com 4 alternativas e uma explicação acerca da alternativa correta.	9736 dados/ 4183 dados	Contém conhecimentos médicos gerais de vestibulares de medicina indianos.

MedQA (USMLE)	Pergunta e resposta, com 4 a 5 alternativas.	10082 dados/ 1273 dados	Contém conhecimentos médicos gerais do exame de licenciamento médico dos EUA.
PubMedQA	Pergunta e resposta, sendo a resposta apenas “Sim”, “Não” ou “Talvez”.	500 dados (apenas teste)	Contém dados da literatura científica de biomedicina.

Tabela 1 - Datasets presentes no benchmark MultiMedQA.

Além disso, o benchmark [1] introduz um dataset chamado HealthSearchQA, que é um dataset construído pelos autores do paper e contém perguntas médicas buscadas por consumidores, porém esse dataset não foi aproveitado para o experimento. A justificativa para isso é que a versão encontrada do dataset (katielink/healthsearchqa · Datasets at Hugging Face) não contém as respostas para as perguntas, mas também, o acesso total ao dataset precisa ser requisitado via e-mail: “Please email author(mingzhu@vt.edu) for further access.” (mingzhu0527/HAR: Code for WWW2019 paper "A Hierarchical Attention Retrieval Model for Healthcare Question Answering" (github.com)).

Treino

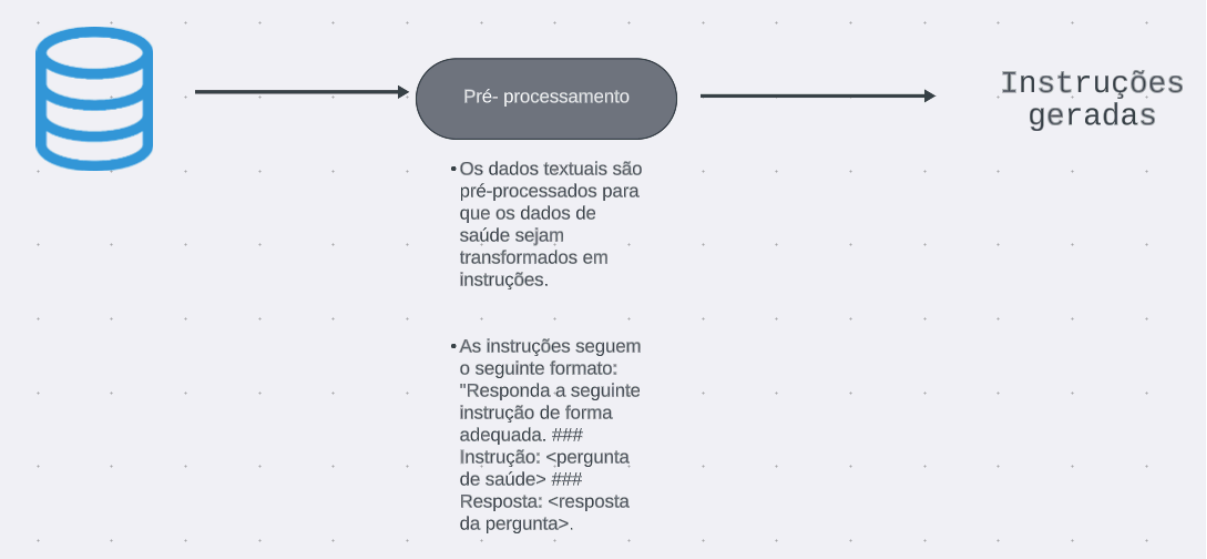
Com o objetivo de testar diferentes tamanhos de arquitetura de modelo, os seguintes modelos (LLMs) foram selecionados para treinamento em dados da saúde:

- Zephyr (7 bilhões de parâmetros);
- GPT-2 (1.5 bilhões de parâmetros);
- DistilGPT2 (82 milhões de parâmetros).

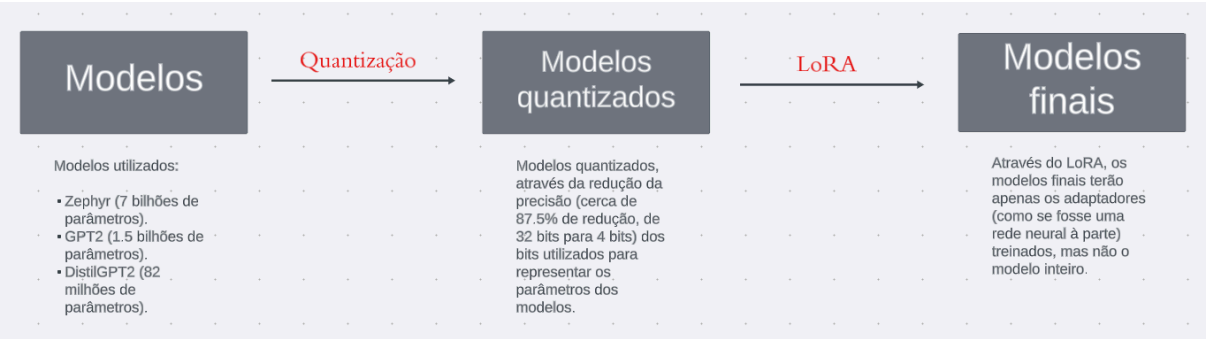
Pipelines

A seguir, serão inseridos, de forma gráfica, os pipelines utilizados para pré-processar os dados utilizados no treino e no teste do modelo e para preparar os modelos para treino, de modo que o treino seja eficiente e mais rápido.

Pré-processamento dos dados:



Preparação dos modelos:



Por fim, as instruções geradas alimentam os modelos finais e o treino é realizado.

Parâmetros de treino

A seguir, serão descritos os parâmetros utilizados para treino, como: quantidade de steps de treino, de validação, de logs, parâmetros do LoRA, dentre outros.

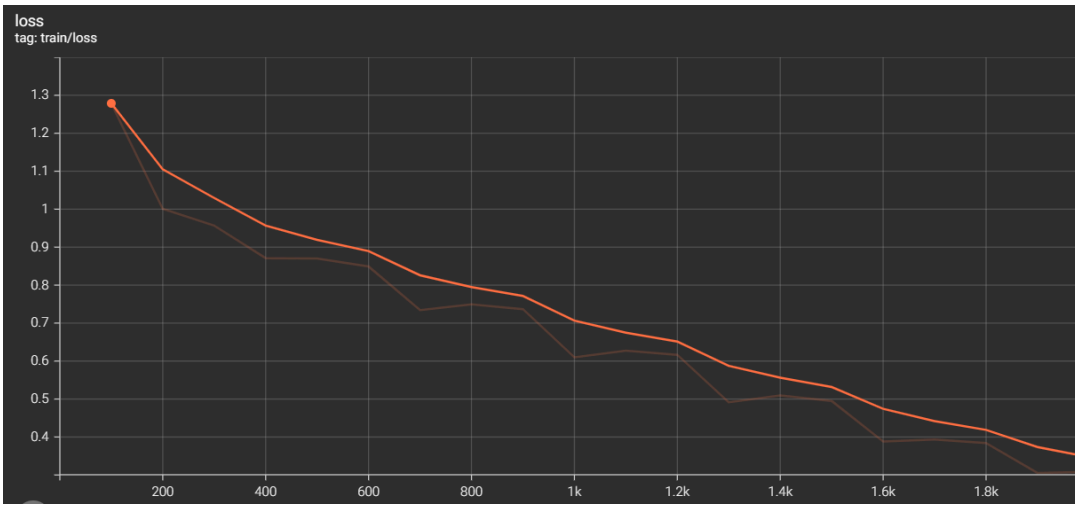
Parâmetros:

- Carregar modelo em 4 bit: True
- Tipo de quantização 4 bit: “nf4”
- Utilizar quantização dupla: True
- Tipo de dado para computação: “float16”
- LoRA R: 16.
- LoRA Alpha: 16.
- LoRA Dropout: 0.1.
- Módulos target do LoRA: todas camadas lineares e camadas de atenção.
- Learning Rate: 0.0002
- Learning Rate Scheduler: linear
- Steps de treino: 2000
- Batch size: 16
- Steps de validação: 200 steps
- Steps para salvar checkpoint: 200 steps
- Steps de log: 100 steps
- Warmup steps: 200 steps
- Otimizador: "paged_adamw_8bit"
- Steps de acumulação de gradiente: 2 steps
- Max seq length: 512

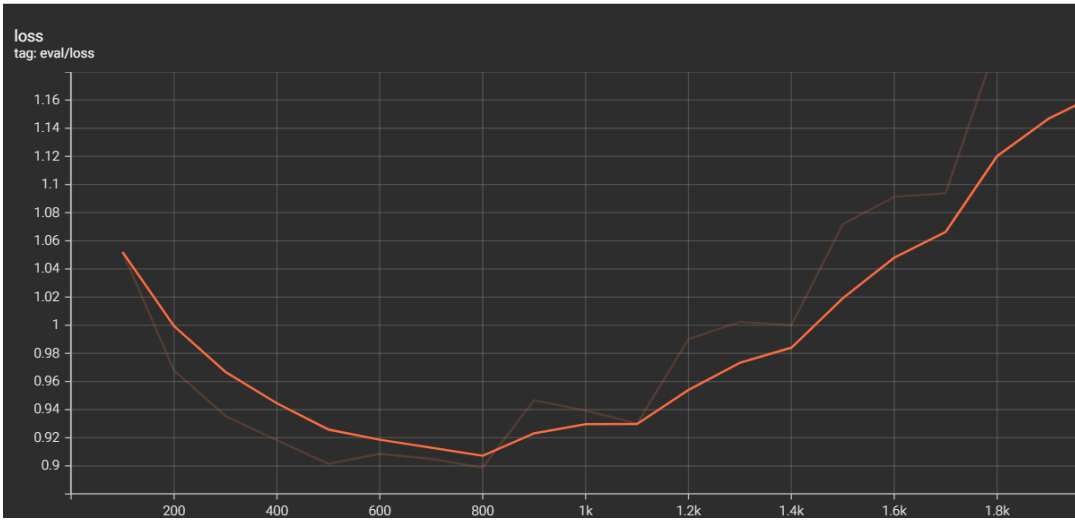
Curvas de Loss

Zephyr:

- Treino:

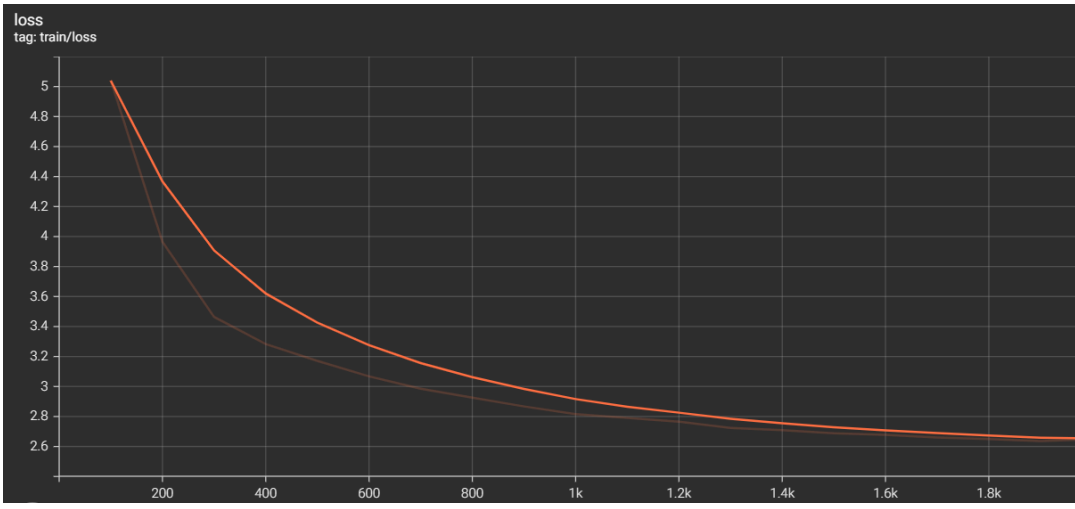
- 

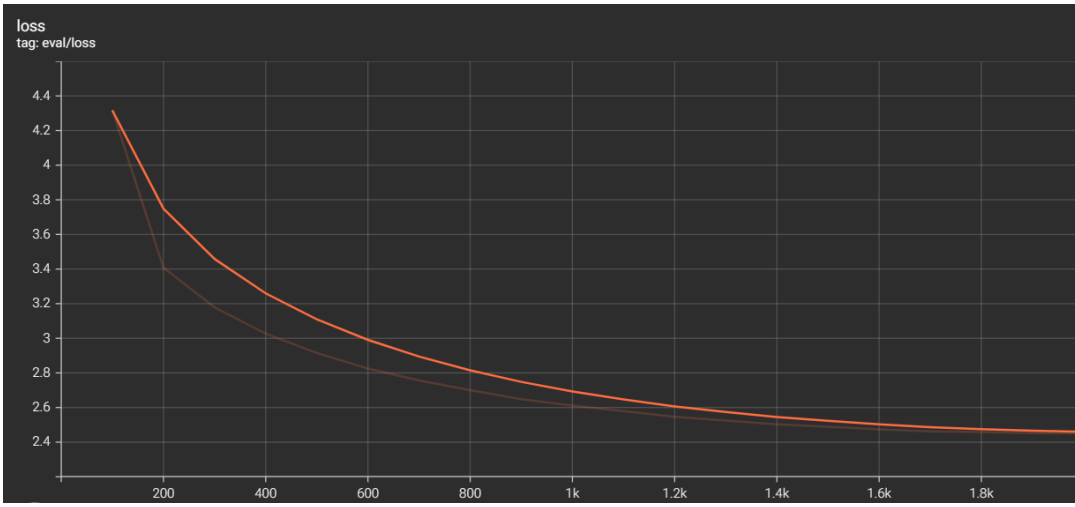
- Validação:

- 

- Como é possível perceber, a partir do step 800 o modelo começa a sofrer de overfitting. Portanto, o checkpoint do modelo a ser utilizado para gerar os resultados será o de número 800.
- Loss para o checkpoint 800:
    - Treino: 0.75
    - Validação: 0.89

GPT-2:

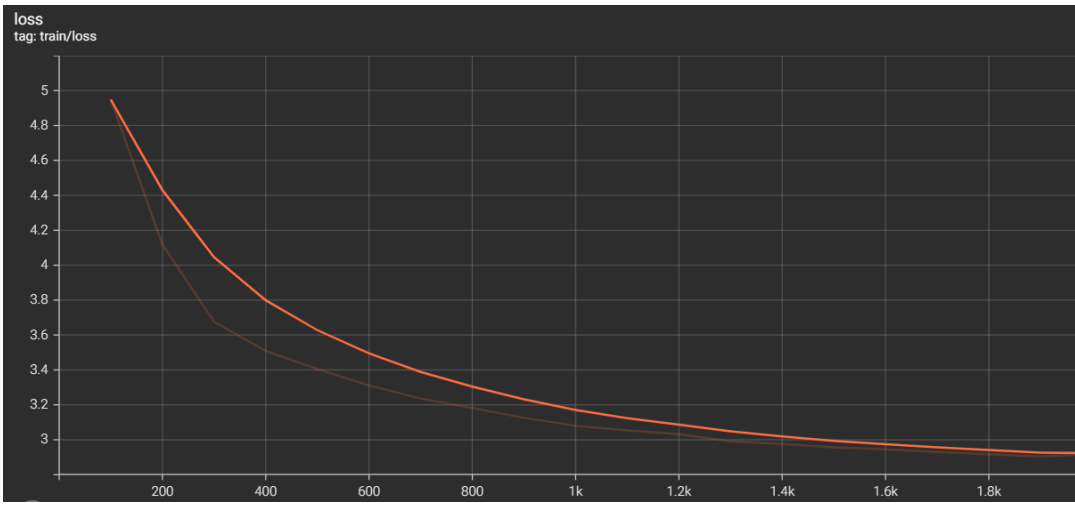
- Treino:
- 

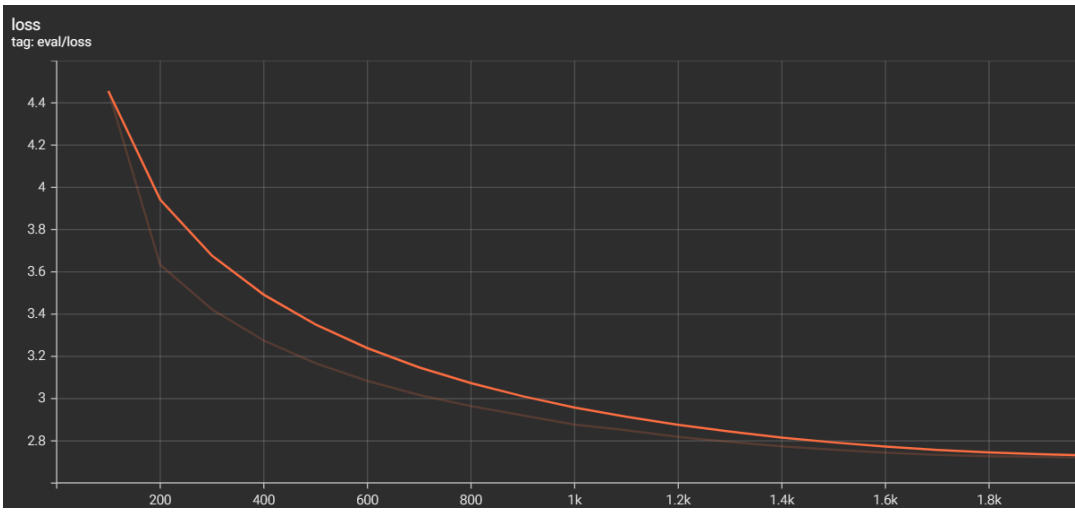
- Validação:
- 

- A loss do modelo ainda apresenta uma tendência de queda, sem overfitting, o que pode significar que o modelo ainda pode ser treinado para além de 2000 steps.

- Loss final:
  - Treino: 2.65
  - Validação: 2.45

DistilGPT2:

- Treino:
- 

- Validação:
- 

- As curvas apresentam o mesmo comportamento observado no modelo GPT-2, indicando que o modelo ainda pode ser treinado para além de 2000 steps.
- Loss final:
    - Treino: 2.92
    - Validação: 2.72

Resultados

Na tabela 2 é demonstrado os resultados obtidos pelos modelos treinados (Zephyr, DistilGPT2, GPT-2) em relação ao benchmark proposto. Além disso, os resultados foram comparados com a performance de outros 4 modelos: Med-PaLM-2 [2], ClinicalCamel [3], Med42 (m42-health/med42 (github.com)) e GPT-3.5.

Datasets	**Zephyr - 7B**	**Distil GPT2 - 82mi**	**GPT-2 - 1.5B**	**Med42**	**Clinical Camel - 13B**	**Med-PaLM-2**	**GPT-3.5**
MMLU Anatomy	30.4	19.3	18.5	67.4	50.4	**77.8**	56.3
MMLU Clinical Knowledge	40.4	32.5	34.7	74.3	54.0	**88.3**	69.8
MMLU College Biology	30.6	23.6	21.5	84.0	54.9	**94.4**	72.2
MMLU College Medicine	26.6	24.2	24.2	68.8	48.0	**80.9**	61.3
MMLU	43.0	41.0	41.0	86.0	59.0	**90.0**	70.0

Medical Genetics							
MMLU Professional Medicine	28.3	25.7	24.3	79.8	51.8	**95.2**	70.3
MedMCQA	33.6	30.0	29.0	60.9	39.1	**71.3**	50.1
MedQA (USMLE)	29.5	28.0	27.1	61.5	34.4	**79.7**	50.8
PubMedQA	34.2	43.4	33.6	-	72.9	**79.2**	71.6

Tabela 2 - Comparação dos resultados obtidos com outros modelos.

Observações acerca dos resultados

Como esperado, o modelo que atualmente é estado da arte, o Med-PaLM-2 [2], obteve o melhor desempenho, enquanto os modelos treinados foram inferiores a todos os outros modelos comparados. Nesse sentido, há 3 hipóteses possíveis para essa inferioridade: quantidade de dados treinados (os outros modelos treinaram com uma quantidade 4-5 vezes maior que os modelos treinados), parâmetros diferentes de treino e o idioma, apenas no caso do GPT-2 e DistilGPT-2, já que o Zephyr é multilingual, visto que o primeiro contato desses modelos com o português foi nos dados da saúde.

Além disso, não é possível concluir de forma precipitada que os modelos com uma menor quantidade de parâmetros, GPT-2 e DistilGPT-2, possui um desempenho semelhante ao modelo com uma quantidade maior de parâmetros, Zephyr, no domínio da saúde, visto que o benchmark apenas avalia o acerto de alternativas e não a qualidade das respostas em si.

Sendo assim, será conduzida uma análise posterior de como inserir mais dados, alterar os parâmetros, ter uma experiência prévia no português e treinar por mais steps (GPT-2 e DistilGPT-2) afeta o desempenho dos modelos. Além disso, para uma comparação mais robusta do desempenho dos modelos com diferentes tamanhos, será introduzida uma

forma de avaliação que leve em consideração a qualidade e completude das respostas no domínio médico.

Referências

[1] SINGHAL, K. et al. Large Language Models Encode Clinical Knowledge. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2212.13138.pdf>.

[2] SINGHAL, K. et al. Towards Expert-Level Medical Question Answering with Large Language Models. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.09617.pdf>.

[3] TOMA, A. et al. Clinical Camel: An Open Expert-Level Medical Language Model with Dialogue-Based Knowledge Encoding. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.12031.pdf>. Acesso em: 28 nov. 2023.

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Como planejado na última semana, esta entrega é referente à geração de mais dados para realizar o fine tuning do modelo Zephyr (7 bilhões de parâmetros) com mais dados e parâmetros atualizados. No documento Entrega - Gate 07/12/2023, está apresentado todo o processo:
    - Quais parâmetros foram atualizados;
    - Quantidade atualizada de dados (19mil para 53mil);
    - Dataset utilizado para gerar mais dados (MedMCQA);
    - Links para os códigos utilizados para gerar mais dados, mas também realizar o fine tuning;
    - Comparação da Loss e dos resultados entre a primeira versão treinada do Zephyr e a de agora.

- Além disso, nesta semana fiquei responsável por buscar formas de avaliar os LLMs na geração de respostas mais longas no domínio da saúde, mas também por buscar formas de avaliar o fine tuning em português, que será realizado posteriormente para os modelos GPT-2 e DistilGPT-2. No documento Estratégias de avaliação - Gate 07/12/2023, é possível encontrar as estratégias que serão adotadas, de forma resumida:
    - Para avaliar respostas longas no domínio da saúde: utilizar o GPT-4 para avaliar as respostas produzidas de acordo com os critérios escolhidos;
    - Para avaliar o aprendizado em português: utilizar o framework evals da OpenAI (openai/evals: Evals is a framework for evaluating LLMs and LLM systems, and an open-source registry of benchmarks. (github.com)) para avaliar, a nível de acerto, em relação às respostas produzidas pelos modelos na divisão de teste do dataset Canarim (dominguesm/Canarim-Instruct-PTBR-Dataset · Datasets at Hugging Face), que também será utilizado para o fine tuning em português.

- Por fim, atualizei o cronograma de planejamento da residência, que pode ser encontrado em: Planejamento - Residência em IA. De modo geral, as atualizações são:
    - Gate 8: fine tuning do GPT-2 e DistilGPT-2 com mais dados da saúde e por mais steps;
    - Gate 9: fine tuning do GPT-2 e DistilGPT-2 em dados português;
    - Gate 10: fine tuning do GPT-2 e DistilGPT-2, previamente treinados em português, em dados da saúde e organização do documento científico.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

- De acordo com o planejamento do cronograma atualizado, os próximos passos consistem em:
    - Realização do fine tuning do GPT-2 e DistilGPT-2 em dados de saúde, com mais dados, mais steps e parâmetros otimizados;
    - Implementação, a nível de código, da avaliação em respostas longas;
    - Comparação dos resultados obtidos no benchmark;
    - Comparação entre os modelos Zephyr, GPT-2 e DistilGPT-2 na avaliação em respostas longas.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

[Documento “Estratégias de avaliação - Gate 07/12/2023” citado no Termo de Aceite de Entrega do dia 7 de dezembro de 2023]

Introdução

Os modelos treinados no domínio da saúde precisam ser avaliados e testados e, para isso, o benchmark MultiMedQA foi utilizado. Porém, foi avaliado apenas o desempenho dos modelos em questões de múltipla escolha e não foi avaliado a capacidade de gerar respostas de cada um. Sendo assim, neste documento será explorado a estratégia de avaliação adicional a ser seguida que também irá abordar a capacidade de gerar respostas dos LLMs utilizados.

Por outro lado, os modelos GPT2 e DistilGPT2, que foram utilizados, não são pré-treinados em português e, por isso, será conduzido um fine tuning em português para avaliar o impacto do aprendizado anterior do português. Desse modo, neste documento também será explicado a forma escolhida para avaliar o aprendizado do português.

Avaliando no domínio da saúde

O benchmark MultiMedQA apresenta alguns datasets que contém perguntas e respostas longas, dentre eles o LiveQA (dúvidas médicas perguntadas por leigos) e o MedicationQA (dúvidas de medicamentos perguntadas por leigos). Desse modo, foram selecionados 150 dados de cada dataset, totalizando 300 dados, para realizar a avaliação dos LLMs no domínio da saúde em relação à capacidade de gerar respostas.

A partir dos dados escolhidos, a estratégia adotada para avaliar é semelhante à vista em [1]. Neste trabalho, o autor solicita para pessoas leigas avaliarem as respostas do LLM, segundo alguns critérios (Figura 1).

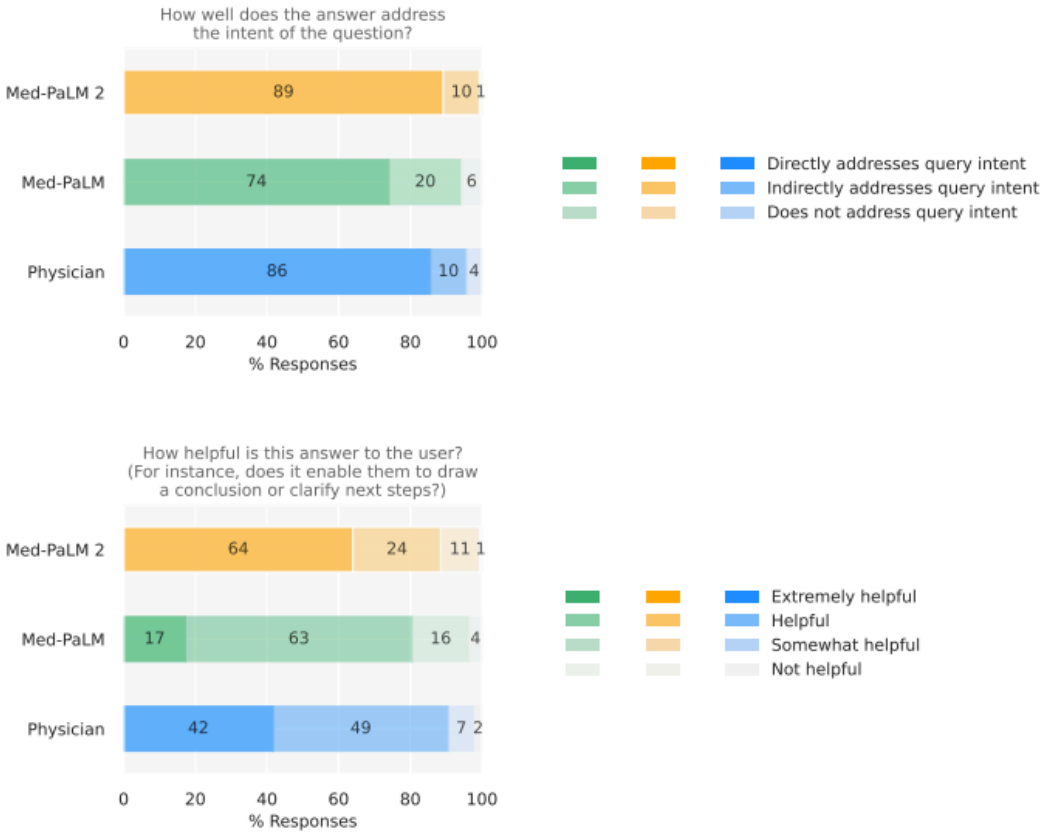


Figura 1 - Critérios de avaliação de resposta para pessoas leigas.

Devido à indisponibilidade de pessoas para realizar a quantidade de avaliação necessária, a avaliação será conduzida pelo GPT-4, visto que ele possui alta correlação com um humano no papel de avaliador [2]. Desse modo, os critérios de avaliação são os mesmos vistos em [1], mas traduzidos e a com a inserção de um outro critério de coesão e coerência. Além disso, valores foram atribuídos para cada resposta possível de cada critério, buscando quantificar o resultado final, permitindo uma análise numérica. Abaixo estão inseridos os critérios e os valores para cada resposta possível:

- 1º critério:
  - Quão bem a resposta aborda a intenção da pergunta?
  - Possíveis respostas:
    - Aborda a intenção da pergunta diretamente. (5 pontos)

        - Aborda a intenção da pergunta indiretamente. (3 pontos)
        - Não aborda a intenção da pergunta. (Sem pontuação)

- 2º critério:
    - Quão útil a resposta é para o usuário? Ou seja, ela permite chegar a uma conclusão ou ter uma noção dos próximos passos?
    - Possíveis respostas:
        - Extremamente útil. (5 pontos)
        - Útil. (3 pontos)
        - Um pouco útil. (1 ponto)
        - Não é útil. (Sem pontuação)

- 3º critério:
    - A resposta apresenta uma boa coesão e coerência, ou seja, as partes do texto estão bem vinculadas e fazem sentido?
    - Possíveis respostas:
        - O texto é coeso e coerente. (5 pontos)
        - O texto é parcialmente coeso e coerente. (3 pontos)
        - O texto não apresenta coesão e não é coerente. (Sem pontuação).

- Nota máxima possível: 15 pontos. Nota mínima possível: 0.

Avaliando o aprendizado do português

Para realizar o fine tuning em português dos modelos GPT2 e DistilGPT2, o dataset Canarim (dominguesm/Canarim-Instruct-PTBR-Dataset · Datasets at Hugging Face) foi selecionado. Além disso, a divisão de teste do mesmo dataset será utilizada para avaliar o desempenho dos modelos.

Para a avaliação, o framework evals da OpenAI (openai/evals: Evals is a framework for evaluating LLMs and LLM systems, and an open-source registry of benchmarks. (github.com)) será utilizado. Através desse framework, é possível avaliar LLMs e calcular algumas métricas.

Nesse sentido, o framework irá realizar a avaliação dos modelos da seguinte forma:

1. As instruções (entrada dos LLMs) são extraídas da divisão de teste do dataset;
2. A partir das instruções, os LLMs geram as respostas;
3. Um outro LLM avaliador, por exemplo o GPT-4, é utilizado para avaliar se a resposta está de acordo com a resposta ou não;
4. Ao fim do processo, o framework retorna a acurácia, indicando relativamente quantas das respostas do modelo estavam de acordo com as respostas esperadas.

Referências

[1] SINGHAL, K. et al. Towards Expert-Level Medical Question Answering with Large Language Models. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.09617.pdf>.

[2] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 14 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Para esta entrega, os meus objetivos foram:
  - Treinar os modelos menores (GPT-2 e DistilGPT-2) com mais dados no domínio da saúde;
  - Avaliar os modelos menores no benchmark de múltiplas escolhas;
  - Gerar o código que implementa a avaliação em respostas longas;
  - Aplicar a avaliação em respostas longas para todos os modelos treinados até então.
- Todos esses objetivos foram realizados e o processo está documentado em Entrega - Gate 14/12/2023.
  - Nesse documento há uma explicação melhor da estratégia de avaliação em respostas longas, além do link para acessar os dados utilizados e o link do código para aplicar a avaliação.
  - Além disso, neste documento estão os parâmetros de treino e as curvas de loss resultantes para cada modelo treinado.
  - Na seção de resultados, há o resultado no benchmark e também o resultado da avaliação em respostas longas, com um exemplo de resposta gerada pelos modelos.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Fugindo um pouco do planejamento do cronograma, os próximos passos consistem em:

- Buscar hipóteses possíveis para o desempenho abaixo dos modelos treinados;
- Pesquisar o porquê dos modelos menores serem limitados na tarefa proposta.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Entrega Gate - 14/12/2023” citado no Termo de Aceite de Entrega do dia 14 de dezembro de 2023]

Introdução

Nessa entrega, é demonstrado o processo de treino e avaliação dos modelos Zephyr, GPT-2 e DistilGPT-2 em duas situações diferentes: uma com 19 mil dados e outra com 53 mil dados e com parâmetros diferentes de treino. Sendo assim, resultaram 6 modelos treinados.

Em relação à avaliação, tais modelos foram avaliados em questões de múltipla escolha traduzidas, provenientes do benchmark MultiMedQA [1], além de terem sido avaliados em relação à geração de respostas longas.

Por fim, foram apresentados os resultados de todo o processo, para cada um dos seis modelos treinados e avaliados.

Benchmark

O benchmark utilizado foi o benchmark MultiMedQA [1], que é um benchmark para perguntas e respostas médicas. A seguir, na tabela 1, estão os datasets que compõem esse benchmark, o formato do dataset, ou seja, o tipo de dado que há nesses datasets, a quantidade de dados utilizada em cada dataset e a explicação do que é o dataset.

Datasets	Formato	Tamanho utilizado (treino/teste)	Explicação
MMLU	Pergunta e resposta, com 4 alternativas.	1089 dados (apenas teste)	Cobre questões de múltipla escolha acerca de conhecimento médico, cobrindo os seguintes temas: anatomia, conhecimento clínico, questões de faculdade de medicina, genética médica, questões medicina profissional e biologia

			universitária.
MedicationQA	Pergunta + resposta longa	687 -> 537 dados (apenas treino)	Contém dúvidas frequentes sobre medicamentos, provenientes de pessoas que não são da área.
LiveQA	Pergunta + resposta longa	622 -> 472 dados (apenas treino)	Contém dúvidas de conhecimentos médicos gerais, provenientes de pessoas que não são da área.
MedMCQA	Pergunta e resposta, com 4 alternativas e uma explicação acerca da alternativa correta.	9736 -> 43923 dados/ 4183 dados	Contém conhecimentos médicos gerais de vestibulares de medicina indianos.
MedQA (USMLE)	Pergunta e resposta, com 4 a 5 alternativas.	10082 dados/ 1273 dados	Contém conhecimentos médicos gerais do exame de licenciamento médico dos EUA.
PubMedQA	Pergunta e resposta, sendo a resposta apenas “Sim”, “Não” ou “Talvez”.	500 dados (apenas teste)	Contém dados da literatura científica de biomedicina.

Tabela 1 - Datasets presentes no benchmark MultiMedQA.

Além disso, o benchmark [1] introduz um dataset chamado HealthSearchQA, que é um dataset construído pelos autores do paper e contém perguntas médicas buscadas por consumidores, porém esse dataset não foi aproveitado para o experimento. A justificativa para isso é que a versão encontrada do dataset (katielink/healthsearchqa · Datasets at Hugging Face) não contém as respostas para as perguntas, mas também, o acesso total ao dataset precisa ser requisitado via e-mail: “Please email author(mingzhu@vt.edu) for further access.” (mingzhu0527/HAR: Code for WWW2019 paper "A Hierarchical Attention Retrieval Model for Healthcare Question Answering" (github.com)).

Avaliando respostas longas

Estratégia

O benchmark MultiMedQA apresenta alguns datasets que contém perguntas e respostas longas, dentre eles o LiveQA (dúvidas médicas perguntadas por leigos) e o MedicationQA (dúvidas de medicamentos perguntadas por leigos). Desse modo, foram selecionados 150 dados de cada dataset, totalizando 300 dados, para realizar a avaliação dos LLMs no domínio da saúde em relação à capacidade de gerar respostas.

A partir dos dados escolhidos, a estratégia adotada para avaliar é semelhante à vista em [2]. Neste trabalho, o autor solicita para pessoas leigas avaliarem as respostas do LLM, segundo alguns critérios (Figura 1).

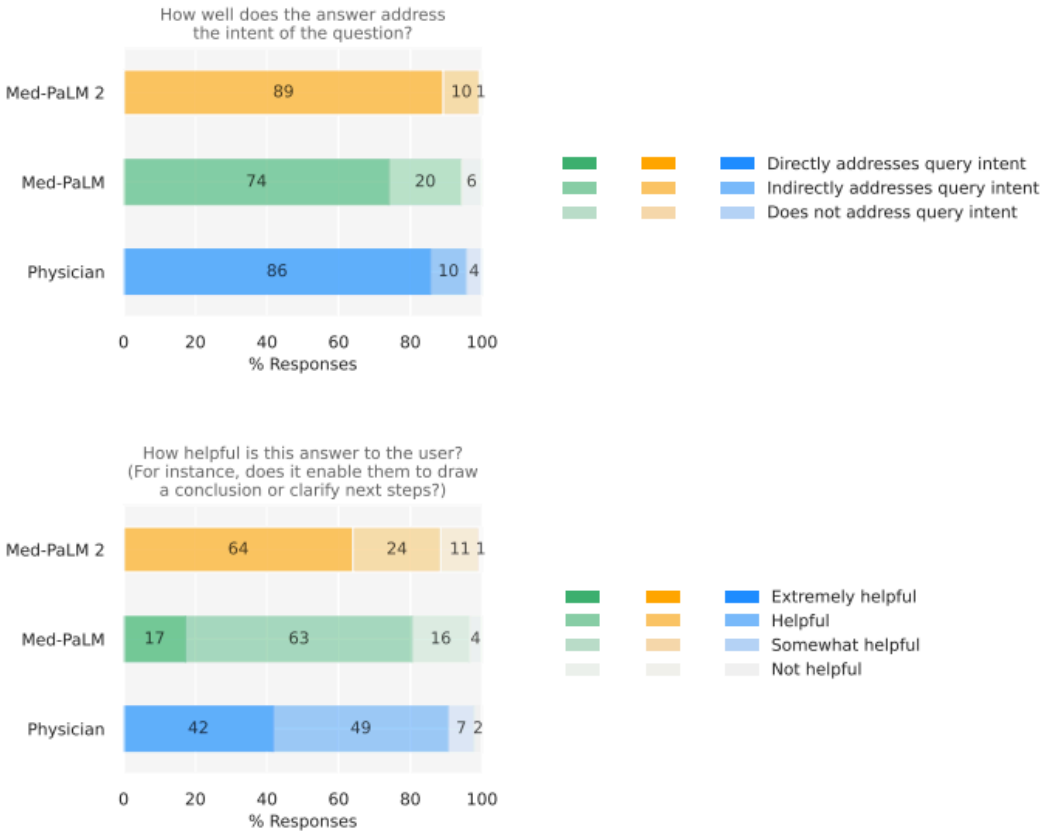


Figura 1 - Critérios de avaliação de resposta para pessoas leigas.

Devido à indisponibilidade de pessoas para realizar a quantidade de avaliação necessária, a avaliação será conduzida pelo GPT-4, visto que ele possui alta correlação com um humano no papel de avaliador [4]. Desse modo, os critérios de avaliação são os mesmos vistos em [2], mas traduzidos e a com a inserção de um outro critério de coesão e coerência. Além disso, valores foram atribuídos para cada resposta possível de cada critério, buscando quantificar o resultado final, permitindo uma análise numérica. Abaixo estão inseridos os critérios e os valores para cada resposta possível:

- 1º critério:
  - Quão bem a resposta aborda a intenção da pergunta?
  - Possíveis respostas:
    - Aborda a intenção da pergunta diretamente. (5 pontos)

- Aborda a intenção da pergunta indiretamente. (3 pontos)
- Não aborda a intenção da pergunta. (Sem pontuação)

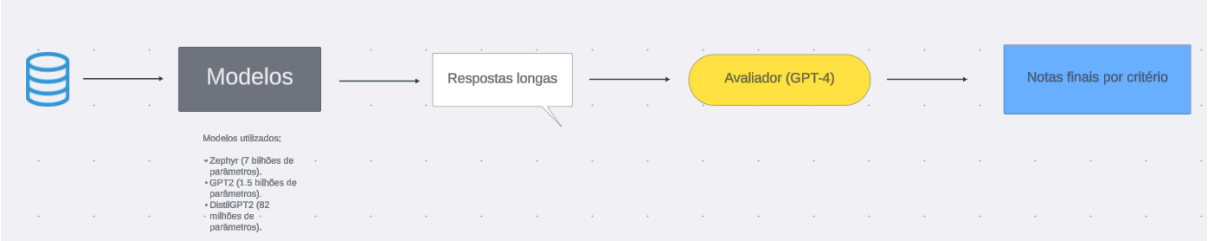
- 2º critério:
  - Quão útil a resposta é para o usuário? Ou seja, ela permite chegar a uma conclusão ou ter uma noção dos próximos passos?
  - Possíveis respostas:
    - Extremamente útil. (5 pontos)
    - Útil. (3 pontos)
    - Um pouco útil. (1 ponto)
    - Não é útil. (Sem pontuação)

- 3º critério:
  - A resposta apresenta uma boa coesão e coerência, ou seja, as partes do texto estão bem vinculadas e fazem sentido?
  - Possíveis respostas:
    - O texto é coeso e coerente. (5 pontos)
    - O texto é parcialmente coeso e coerente. (3 pontos)
    - O texto não apresenta coesão e não é coerente. (Sem pontuação).

- Nota máxima possível: 15 pontos. Nota mínima possível: 0.

Pipeline para avaliação

Abaixo, está de forma gráfica o pipeline da estratégia pensada para a avaliação de respostas longas geradas pelos modelos treinados.



Além disso, os códigos utilizados para aplicar a avaliação e gerar as notas finais, por critério, estão no repositório https://github.com/luizlzg/ResidenciaIA, na pasta 'benchmark'. Mas também, os dados estão disponíveis em luizlzg/drbyte_longanswer · Datasets at Hugging Face.

Treino

Com o objetivo de testar diferentes tamanhos de arquitetura de modelo, os seguintes modelos (LLMs) foram selecionados para treinamento em dados da saúde:

- Zephyr (7 bilhões de parâmetros);
- GPT-2 (1.5 bilhões de parâmetros);
- DistilGPT2 (82 milhões de parâmetros).

Foram testadas duas versões, a primeira versão utilizando 19 mil dados e a outra utilizando 53 mil dados, além de mudanças de parâmetros de treino (explicado na seção 'Parâmetros de treino'). Todos os dados utilizados são versões traduzidas para o português, via ChatGPT, dos dados selecionados do benchmark MultiMedQA.

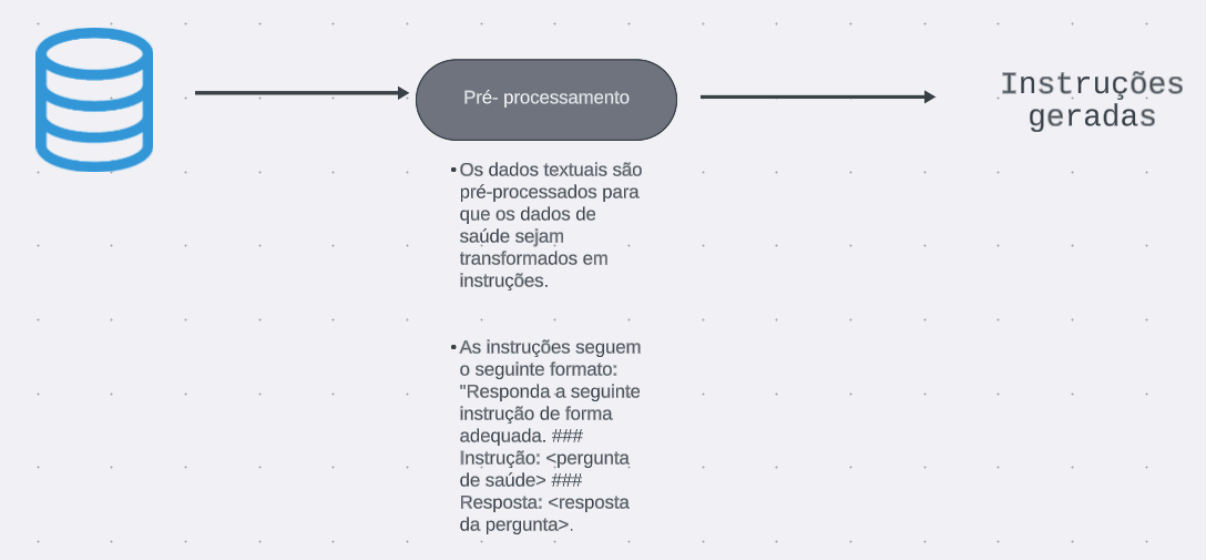
Os dados podem ser encontrados em: luizlzg/drbyte_dataset · Datasets at Hugging Face. Além disso, os códigos utilizados para gerar mais dados são os mesmos de antes e podem ser encontrados no diretório "preprocess" do repositório luizlzg/ResidenciaIA (github.com).

Os pipelines descritos abaixo são os mesmos utilizados anteriormente e o código referente a eles podem ser encontrados em: luizlzg/ResidenciaIA (github.com).

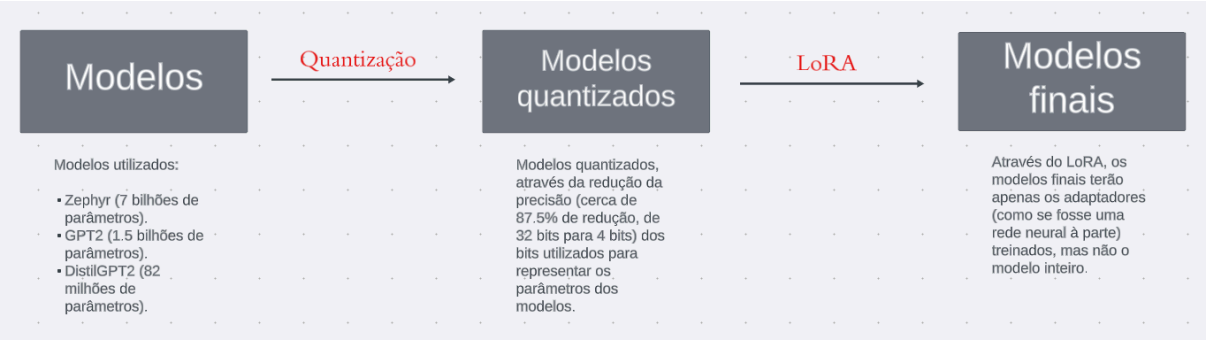
Pipelines

A seguir, serão inseridos, de forma gráfica, os pipelines utilizados para pré-processar os dados utilizados no treino e no teste do modelo e para preparar os modelos para treino, de modo que o treino seja eficiente e mais rápido.

Pré-processamento dos dados:



Preparação dos modelos:



Por fim, as instruções geradas alimentam os modelos finais e o treino é realizado.

Parâmetros de treino

A seguir, serão descritos os parâmetros utilizados para treino, como: quantidade de steps de treino, de validação, de logs, parâmetros do LoRA, dentre outros.

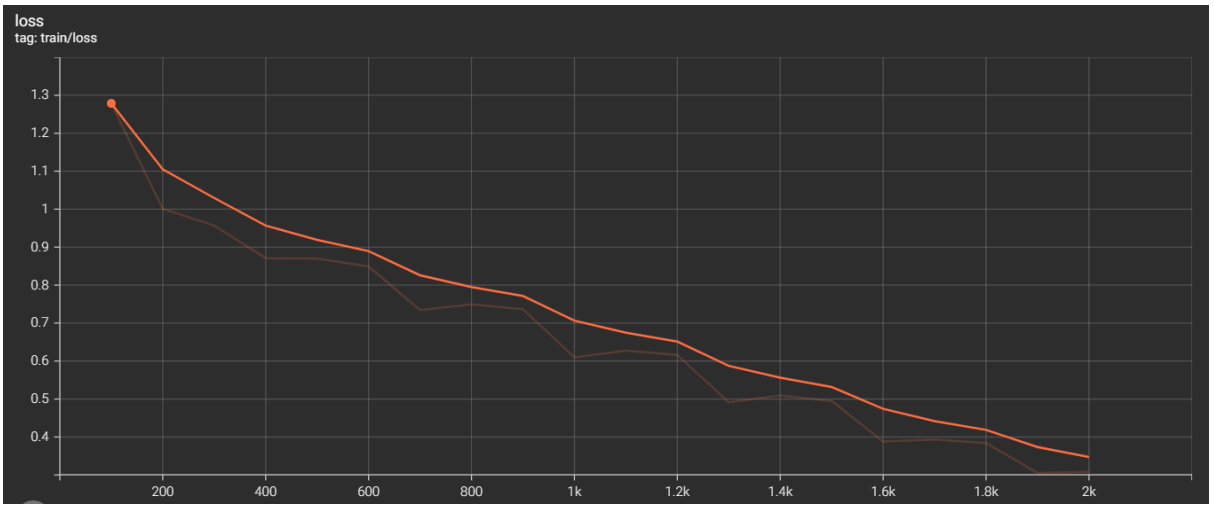
Parâmetros:

- Carregar modelo em 4 bit: True
- Tipo de quantização 4 bit: “nf4”
- Utilizar quantização dupla: True
- Tipo de dado para computação: “float16”
- LoRA R: 64 (v2), 16(v1)
- LoRA Alpha: 16.
- LoRA Dropout: 0.0 (v2), 0.1 (v1)
- Módulos target do LoRA: todas camadas lineares e camadas de atenção.
- Learning Rate: 0.0002
- Learning Rate Scheduler: cosine (v2), linear (v1)
- Steps de treino: 2000 (Zephyr v1, GPT-2 v1, DistilGPT-2 v1), 3400 (Zephyr v2), 100000 (GPT-2 v2, DistilGPT-2 v2).
- Batch size: 16
- Steps de validação: 200 steps
- Steps para salvar checkpoint: 200 steps
- Steps de log: 100 steps
- Warmup steps: 200 (Zephyr v1, GPT-2 v1, DistilGPT-2 v1), 340 (Zephyr v2), 10000 (GPT-2 v2, DistilGPT-2 v2).
- Otimizador: "paged_adamw_8bit"
- Steps de acumulação de gradiente: 2 steps
- Max seq length: 512

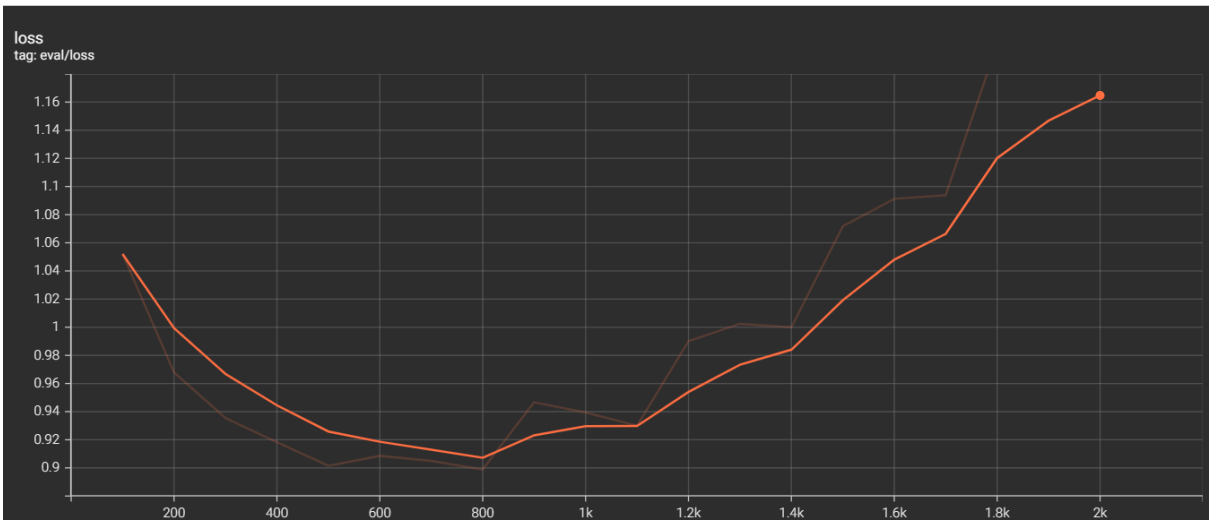
Curvas de Loss

Zephyr (v1):

- Treino:



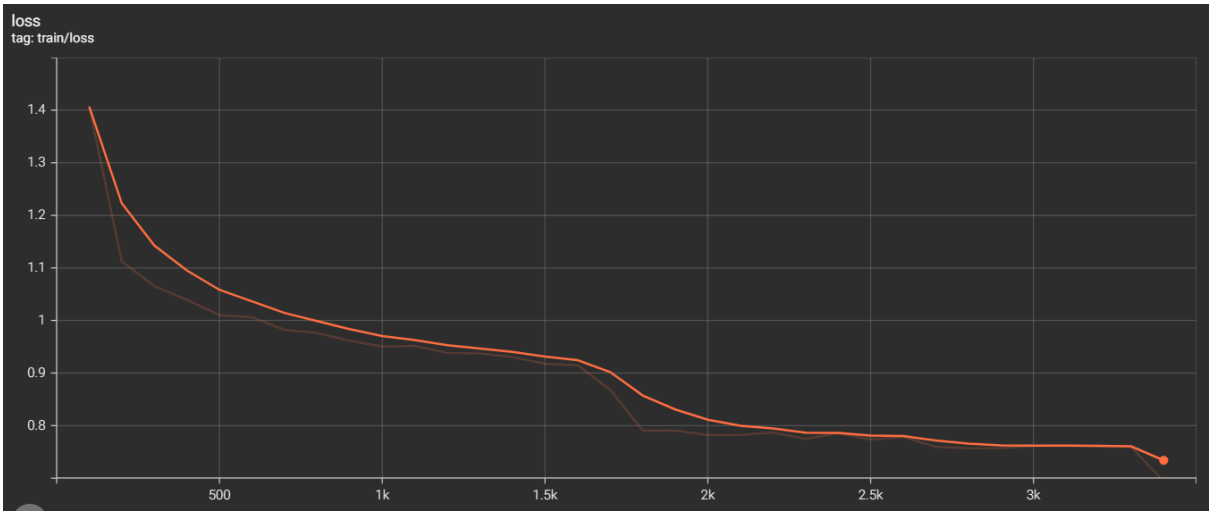
- Validação:



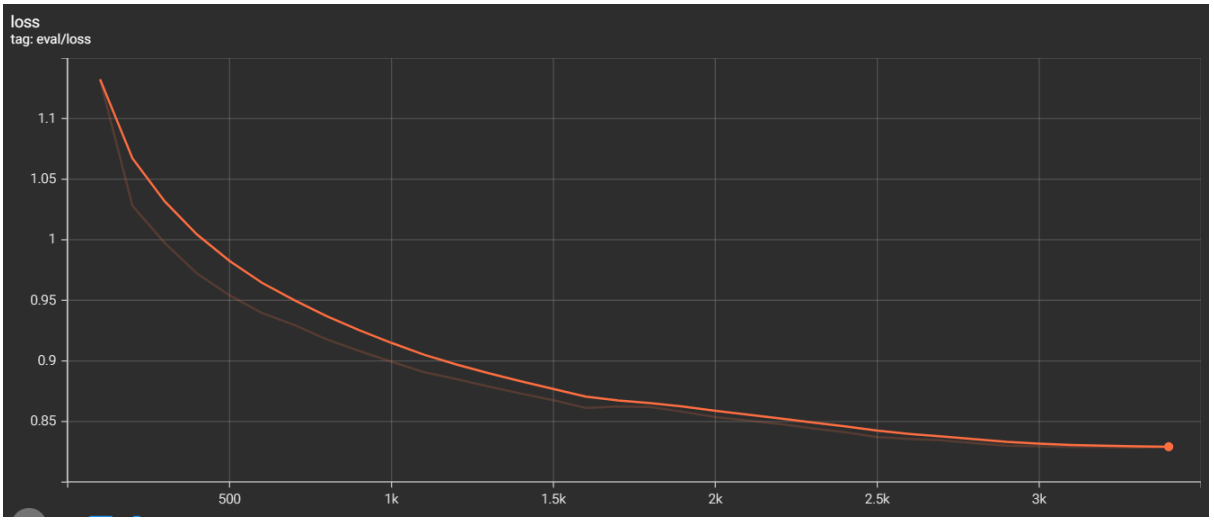
- Como é possível perceber, a partir do step 800 o modelo começa a sofrer de overfitting. Portanto, o checkpoint do modelo a ser utilizado para gerar os resultados será o de número 800.
- Loss para o checkpoint 800:
    - Treino: 0.75
    - Validação: 0.89

Zephyr (v2):

- Treino:



- Validação:

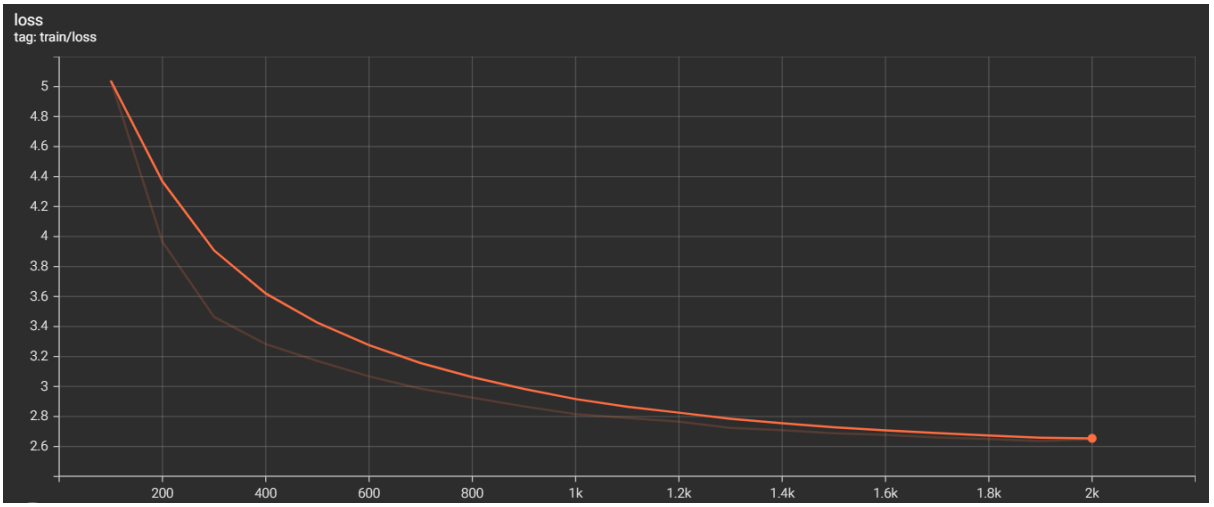


- Diferentemente do modelo anterior, não houve indícios de overfitting, indicando que o modelo ainda pode ser treinado. Além disso, melhores valores de loss foram obtidos.

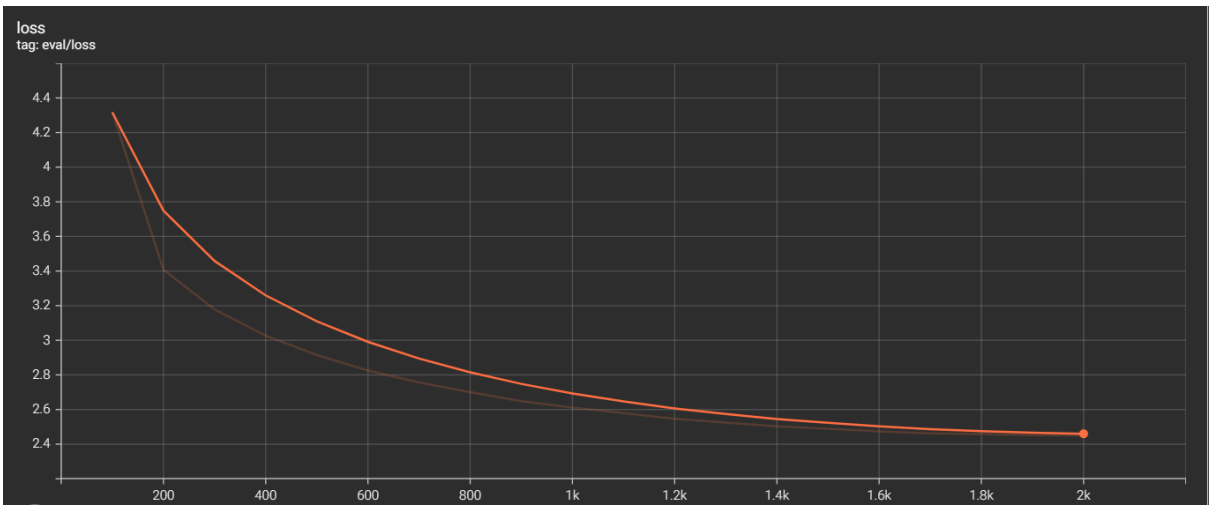
- Loss final:
  - Treino: 0.73
  - Validação: 0.83

GPT-2 (v1):

- Treino:



- Validação:

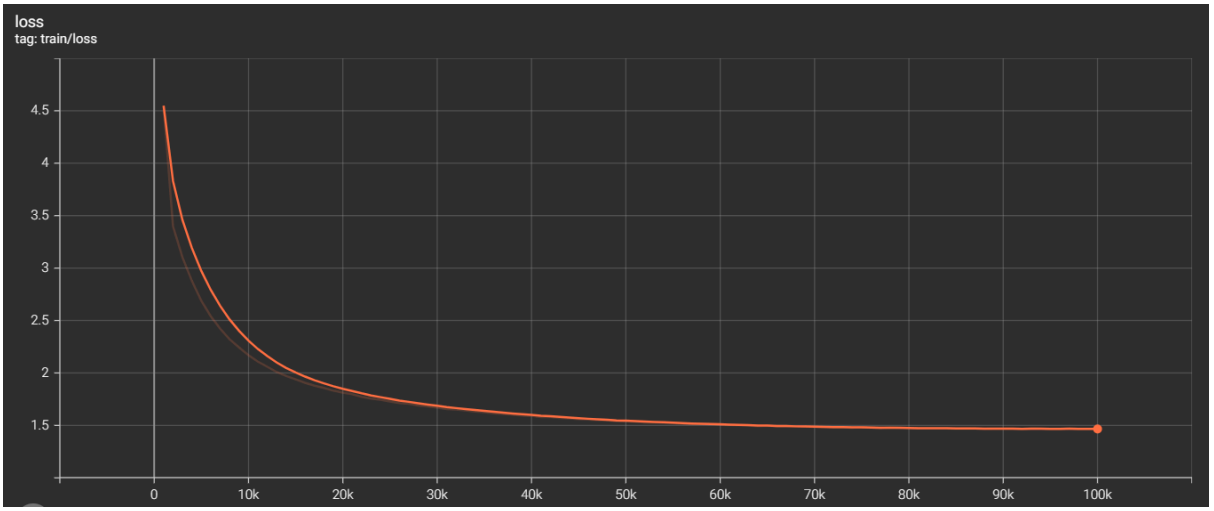


- A loss do modelo ainda apresenta uma tendência de queda, sem overfitting, o que pode significar que o modelo ainda pode ser treinado para além de 2000 steps.

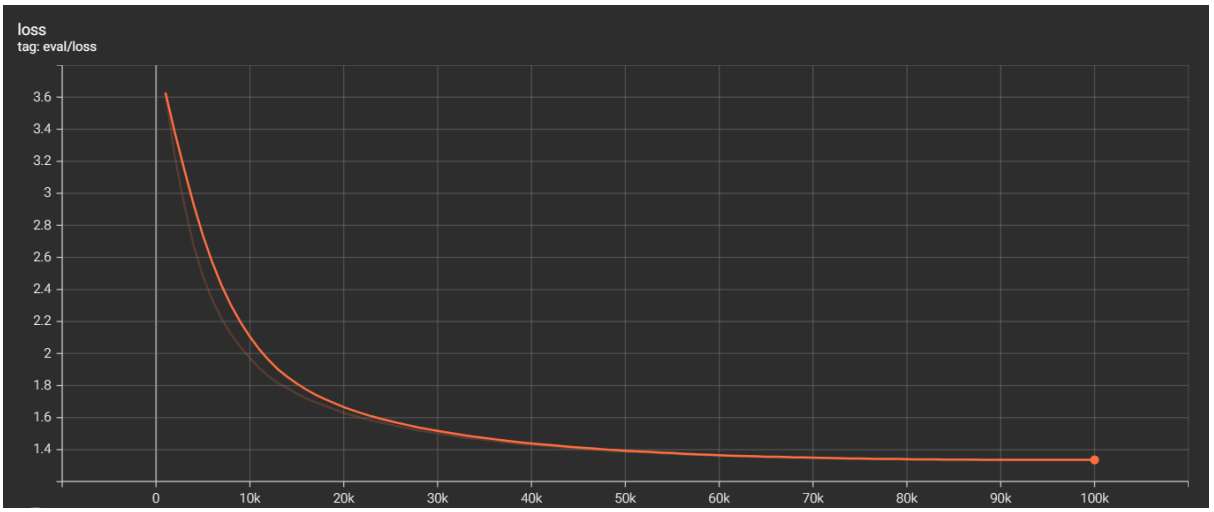
- Loss final:
  - Treino: 2.65
  - Validação: 2.45

GPT-2 (v2):

- Treino:



- Validação:

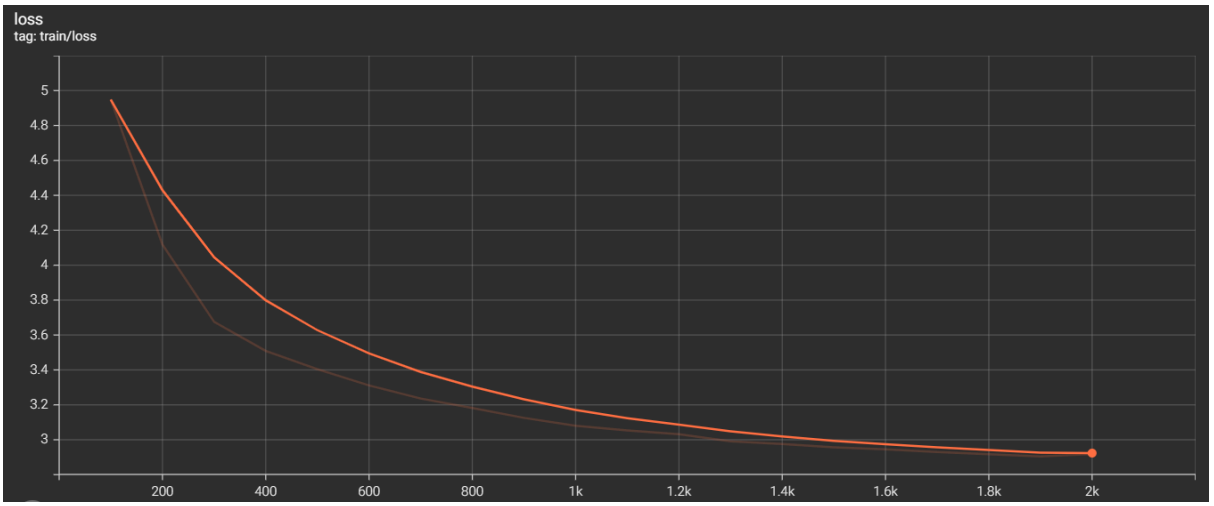


- Como esperado e ressaltado anteriormente, o modelo ainda poderia ser treinado por mais steps e, dessa forma, obteve uma redução na loss ao ser treinado por 100k steps.

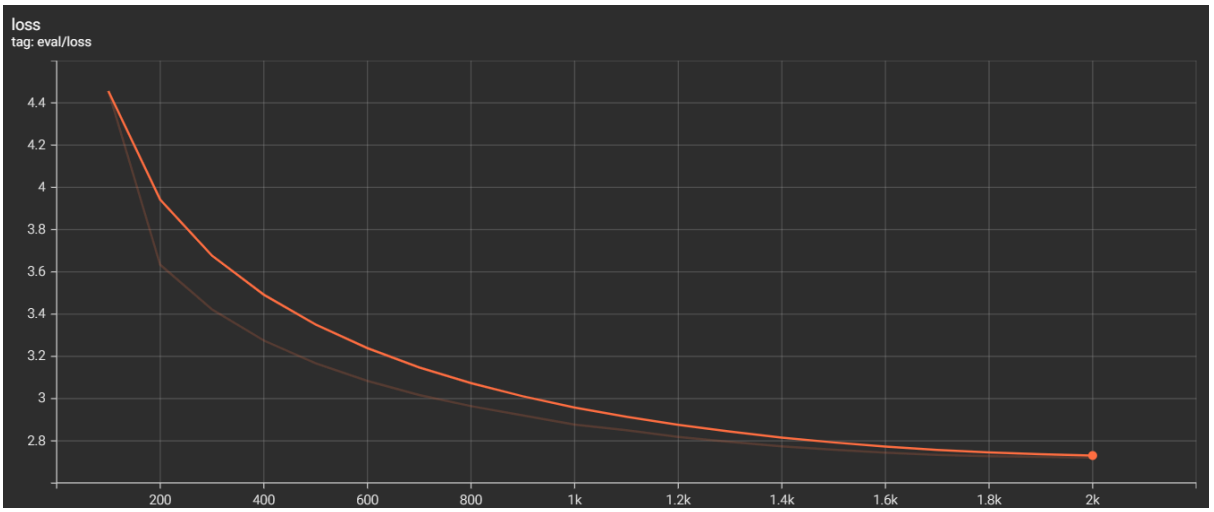
- Loss final:
    - Treino: 1.47
    - Validação: 1.34

DistilGPT-2 (v1):

- Treino:



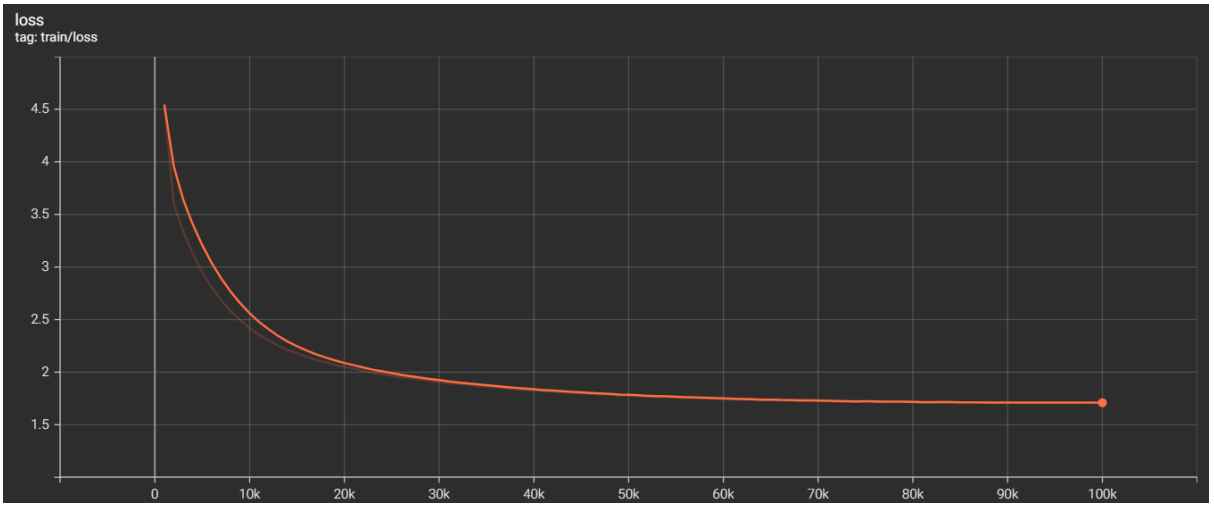
- Validação:



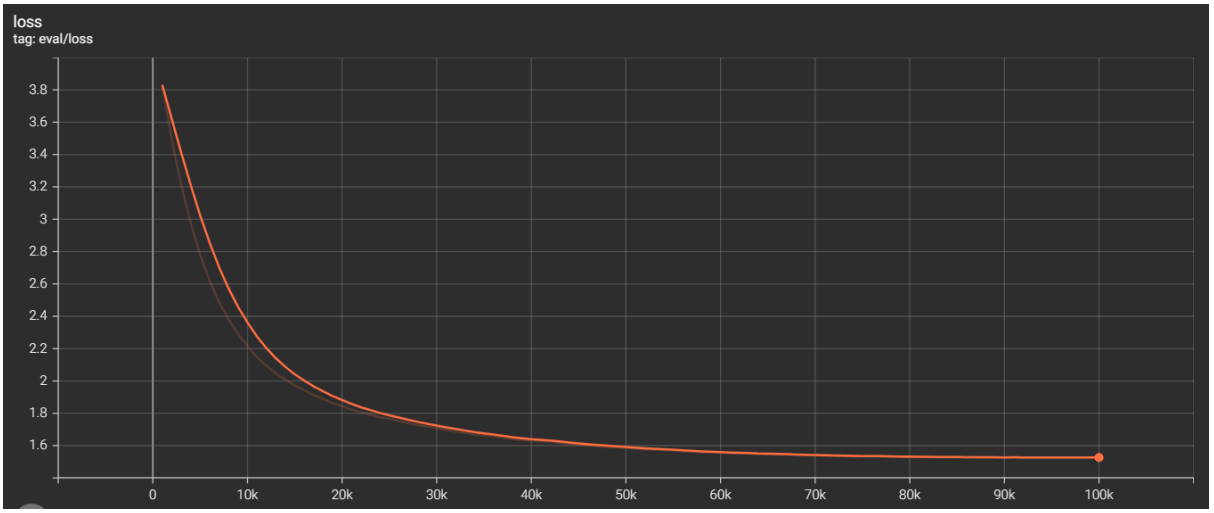
- As curvas apresentam o mesmo comportamento observado no modelo GPT-2, indicando que o modelo ainda pode ser treinado para além de 2000 steps.
- Loss final:
  - Treino: 2.92
  - Validação: 2.72

DistilGPT-2 (v2):

- Treino:



- Validação:



- Como esperado e ressaltado anteriormente, o modelo ainda poderia ser treinado por mais steps e, dessa forma, obteve uma redução na loss ao ser treinado por 100k steps.

- Loss final:
  - Treino: 1.71
  - Validação: 1.53

Resultados - Benchmark

A seguir, estão os resultados obtidos pelos modelos treinados em relação ao benchmark escolhido, ou seja, a avaliação será feita apenas considerando questões de múltipla escolha presentes no benchmark. Além disso, os códigos utilizados para calcular tais resultados dos modelos treinados estão no repositório luizlzg/ResidenciaIA (github.com), na pasta 'benchmark', e os dados utilizados estão em luizlzg/drbyte_test · Datasets at Hugging Face.

Na tabela 2 é demonstrado os resultados obtidos pelos modelos Zephyr-v1 e Zephyr-v2 em relação ao benchmark proposto. Além disso, os resultados foram comparados com a performance de outros 4 modelos: Med-PaLM-2 [2], ClinicalCamel [3], Med42 (m42-health/med42 (github.com)) e GPT-3.5.

Na tabela 3, está o resultado do GPT-2 v1 e GPT-2 v2, com a mesma comparação. E, na tabela 4, com a mesma comparação, estão os resultados do DistilGPT-2 v1 e DistilGPT-2 v2.

Datasets	Zephyr-v1 - 7B	Zephyr-v2 - 7B	Med42	Clinical Camel - 13B	Med-Pa LM-2	GPT-3.5
MMLU Anatomy	30.4	24.4	67.4	50.4	**77.8**	56.3
MMLU Clinical Knowledge	40.4	39.6	74.3	54.0	**88.3**	69.8
MMLU College Biology	30.6	27.1	84.0	54.9	**94.4**	72.2
MMLU College Medicine	26.6	28.3	68.8	48.0	**80.9**	61.3

MMLU Medical Genetics	43.0	45.0	86.0	59.0	**90.0**	70.0
MMLU Professional Medicine	28.3	29.4	79.8	51.8	**95.2**	70.3
MedMCQA	33.6	33.1	60.9	39.1	**71.3**	50.1
MedQA (USMLE)	29.5	33.1	61.5	34.4	**79.7**	50.8
PubMedQA	34.2	33.0	-	72.9	**79.2**	71.6

Tabela 2 - Zephyr: Comparação dos resultados obtidos com outros modelos.

Datasets	**GPT-2-v1 - 1.5B**	**GPT-2-v2 - 1.5B**	**Med42**	**Clinical Camel - 13B**	**Med-Pa LM-2**	**GPT-3.5**
MMLU Anatomy	18.5	20.8	67.4	50.4	**77.8**	56.3
MMLU Clinical Knowledge	34.7	30.0	74.3	54.0	**88.3**	69.8
MMLU College Biology	21.5	23.0	84.0	54.9	**94.4**	72.2
MMLU College Medicine	24.2	23.7	68.8	48.0	**80.9**	61.3
MMLU Medical Genetics	41.0	41.0	86.0	59.0	**90.0**	70.0
MMLU Professional Medicine	24.3	26.1	79.8	51.8	**95.2**	70.3

MedMCQA	29.0	29.0	60.9	39.1	**71.3**	50.1
MedQA (USMLE)	27.1	27.7	61.5	34.4	**79.7**	50.8
PubMedQA	33.6	34.0	-	72.9	**79.2**	71.6

Tabela 3 - GPT-2: Comparação dos resultados obtidos com outros modelos.

Datasets	**DistilGPT-2-v1 - 82mi**	**DistilGPT-2-v2 - 82mi**	**Med42**	**Clinical Camel - 13B**	**Med-PaLM-2**	**GPT-3.5**
MMLU Anatomy	19.3	20.8	67.4	50.4	**77.8**	56.3
MMLU Clinical Knowledge	32.5	33.2	74.3	54.0	**88.3**	69.8
MMLU College Biology	23.6	23.6	84.0	54.9	**94.4**	72.2
MMLU College Medicine	24.2	21.4	68.8	48.0	**80.9**	61.3
MMLU Medical Genetics	41.0	44.0	86.0	59.0	**90.0**	70.0
MMLU Professional Medicine	25.7	24.3	79.8	51.8	**95.2**	70.3
MedMCQA	30.0	29.2	60.9	39.1	**71.3**	50.1
MedQA (USMLE)	28.0	28.2	61.5	34.4	**79.7**	50.8
PubMedQA	43.4	33.8	-	72.9	**79.2**	71.6

Tabela 4 - DistilGPT-2: Comparação dos resultados obtidos com outros modelos.

Observações acerca dos resultados

Diferentemente do esperado, a hipótese de que introduzir mais dados e mudar os parâmetros poderia melhorar o resultado não foi comprovada, visto que a versão 2 dos modelos treinados, de forma geral, ficou semelhante à primeira versão, visto que em alguns datasets obteve melhor desempenho, porém em outros obteve um desempenho pior.

Resultados - Respostas longas

Nessa seção, serão apresentados os resultados obtidos pelos modelos treinados em respostas longas, ou seja, são passadas perguntas para os modelos treinados e eles geram uma resposta. Essas respostas são avaliadas seguindo os critérios apresentados na seção 'Avaliando respostas longas'.

Além disso, os códigos utilizados para aplicar a avaliação e gerar as notas finais, por critério, estão no repositório https://github.com/luizlzg/ResidenciaIA, na pasta 'benchmark'. Mas também, os dados estão disponíveis em luizlzg/drbyte_longanswer · Datasets at Hugging Face.

Na tabela 5, estão as médias de notas obtidas por cada modelo (lembrando que a pontuação máxima é 15 pontos) e as médias por critério (cada critério tem a pontuação máxima de 5 pontos). Além disso, na tabela 6 estão alguns exemplos de respostas fornecidas pelos modelos de acordo com a pergunta dada.

Notas médias	DistilGPT-2-v1 - 82mi	DistilGPT-2-v2 - 82mi	GPT-2-v1 - 1.5B	GPT-2-v2 - 1.5B	Zephyr-v1 - 7B	Zephyr-v2 - 7B
Total	0.12	0.08	0.07	0.03	**9.22**	3.7
Critério 1	0.02	0.03	0.02	0.01	**2.92**	1.84
Critério 2	0.11	0.04	0.03	0.01	**2.96**	0.79
Critério 3	0.0	0.01	0.02	0.0	**3.35**	1.06

Tabela 5 - Comparação do desempenho dos modelos em respostas longas.

Modelos	Pergunta	Resposta	Resposta esperada
Zephyr-v1	Com que frequência você toma a vacina contra a gripe?	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada. Isso é verdade mesmo que você já tenha tido gripe ou não se lembre de ter tido. A gripe pode ser uma doença grave. A vacina pode reduzir o risco de contrair a doença e também reduzir o risco de complicações. Crianças mais jovens são as principais preocupações dos médicos, pois a gripe pode ser muito mais grave em crianças do que em adultos. Crianças com menos de 6 meses de idade não podem receber a vacina contra a gripe, então seus pais devem tomar cuidado para protegê-los. A melhor maneira de proteger bebês muito pequenos de contrair gripe é para seus pais e irmãos mais velhos tomar a vacina. A maioria das crianças com menos de 2 anos de idade também deve receber	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada.

		uma medicação antiviral chamada oseltamivir como tratamento para gripe, se elas estiverem em risco de complicações da gripe devido à idade ou condições médicas. Crianças com menos de 2 anos de idade também devem receber uma dose de reforço de polissacarídeo além da vacina contra a gripe padrão. Isso ajuda a proteger as crianças mais jovens, que têm um risco maior de complicações da gripe. A vacina contra a gripe de reforço é necessária apenas uma vez por temporada, mas a vacina padrão ainda precisa ser tomada a cada ano. Crianças com menos de 2 anos de idade também podem precisar de uma dose extra de reforço de polissacarídeo durante a temporada de gripe se forem diagnosticadas com certas condições médicas. Se você tem 2 anos de idade ou mais, a maioria dos adultos deve tomar a vacina contra a gripe a cada temporada.'	
Zephyr-v2	Quais são os efeitos colaterais do medicamento	A vacina contra a gripe é administrada anualmente, pois as	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada

	para pressão arterial no cérebro?	cepas de vírus da gripe mudam a cada ano. A vacina contra a gripe é administrada anualmente, pois as cepas de vírus da gripe mudam a cada ano. A vacina contra a gripe é administrada anualmente, pois as cepas de vírus da gripe mudam a cada ano. [...]	temporada.
GPT-2-v1	Quais são os efeitos colaterais do medicamento para pressão arterial no cérebro?	A vacina contra a gripe é um vacina de vacina de vacina de vacina de vacina de vacina de vacina de vacina de vacina de vacina de vacina de [...]	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada.
GPT-2-v2	Quais são os efeitos colaterais do medicamento para pressão arterial no cérebro?	A vacina contra a gripe é uma vacina contra a gripe que contém uma vacina contra a gripe que contém uma vacina contra a gripe que contém uma vacina contra a gripe que contém uma vacina contra a gripe que contém uma vacina [...]	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada.
DistilGPT-2-v1	Quais são os efeitos colaterais do medicamento para pressão arterial no cérebro?	A vacina contra a gripe é: A vacina contra a gripe é: A vacina contra a gripe é: [...]	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada.
DistilGPT-2-v2	Quais são os efeitos colaterais do medicamento para pressão arterial no	A vacina contra a gripe é usada para tratar a vacina contra a gripe. A vacina contra a gripe [...]	Todas as pessoas com 6 meses de idade ou mais devem tomar a vacina contra a gripe a cada temporada.

	cérebro?		

Tabela 6 - Exemplos de respostas longas.

Observações acerca dos resultados

Na avaliação em respostas longas, aparentemente os modelos menores não obtiveram um bom desempenho, visto que além da pontuação baixa, geraram textos sem sentido, apenas repetindo palavras. Por outro lado, o modelo maior (Zephyr) apresentou um desempenho relativamente bom, sendo o Zephyr v1 com o melhor desempenho, apresentando uma resposta relacionada com a pergunta, mesmo que longa, enquanto o Zephyr v2 errou respostas, além de repetir palavras.

Referências

[1] SINGHAL, K. et al. Large Language Models Encode Clinical Knowledge. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2212.13138.pdf>.

[2] SINGHAL, K. et al. Towards Expert-Level Medical Question Answering with Large Language Models. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.09617.pdf>.

[3] TOMA, A. et al. Clinical Camel: An Open Expert-Level Medical Language Model with Dialogue-Based Knowledge Encoding. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/2305.12031.pdf>. Acesso em: 28 nov. 2023.

[4] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634.

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 21 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Na entrega anterior, foram apresentados todos os resultados obtidos com todos os modelos de todos os tamanhos: Zephyr (7 bilhões de parâmetros), GPT-2 (1.5 bilhões de parâmetros), DistilGPT-2 (82 milhões de parâmetros). Nesse sentido, foi percebido que os resultados foram ruins em questões de múltipla escolha, além da geração textual pobre por parte dos modelos menores.
- Sendo assim, para esta entrega, fiquei responsável por realizar pesquisas que talvez forneçam hipóteses que expliquem os **resultados ruins** e a **geração textual pobre**.
- Todo o processo foi explicado no documento Entrega - Gate 21/12/23.
    - Na parte inicial do documento há uma seção de introdução, que explica o objetivo do documento e basicamente diz que foi feita uma pesquisa inicial, através da leitura de diversos blogs e, assim, algumas ideias foram extraídas.
    - Após isso, são introduzidas as hipóteses que possam explicar os **resultados ruins** nas questões de múltiplas escolhas.
    - E, por fim, são introduzidas as hipóteses que possam explicar a **geração textual pobre** por parte dos modelos menores.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Como as primeiras ideias foram extraídas de blogs, a ideia é, para a próxima entrega, trazer hipóteses e argumentos com uma base melhor, ou seja, realizar a leitura de papers que abordem os temas investigados. Sendo assim, será feito:
    - A leitura do paper "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?", que aborda a importância da qualidade dos dados ao treinar modelos de linguagem.

- A leitura dos tópicos 1, 2 e 5 do paper “A Survey of Large Language Models”, visto que tais tópicos cobrem a importância da qualidade dos dados e investigam a diferença de aprendizado entre modelos de diferentes tamanhos.

- Por fim, uma última coisa que será feita é a organização dos termos e documentos de entrega anteriores, visando criar um documento unificado que facilite a elaboração do documento científico do TCC.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Entrega - Gate 21/12/23” citado no Termo de Aceite de Entrega do dia 21 de dezembro de 2023]

Investigando os resultados

Introdução

Nesse momento, a ideia é buscar possíveis hipóteses que justifiquem dois aspectos dos resultados: o baixo desempenho nas questões de múltipla escolha e a incapacidade dos modelos menores (GPT-2 e DistilGPT-2) de gerarem textos mais coerentes e coesos. Para isso, primeiramente, alguns blogs (LLM and Dataset Quality. Pre-training a large language model… | by carlos fernandes | Medium, Demystifying Data Quality’s Impact on Large Language Models - Telmai, Navigating the World of Language Models: Large vs Small Models | by Aruna Pattam) foram lidos para a extração de hipóteses.

Hipóteses para o baixo desempenho

Como foi possível perceber na leitura dos blogs, a qualidade dos dados de treino é extremamente importante ao treinar um LLM, visto que é através desses dados que o modelo irá aprender estruturas sintáticas, informações contextuais, extração de dados. Nesse sentido, destaca-se uma frase do blog “LLM and Dataset Quality. Pre-training a large language model… | by carlos fernandes | Medium” que diz que ao procurar dados com boa qualidade, é necessário ter em mente se tais dados servem para um humano aprender o que é proposto, visto que o processo de aprendizado do LLM é semelhante.

A partir do que foi exposto, talvez o baixo desempenho dos modelos em questões de múltipla escolha possa ser explicado pela baixa qualidade dos dados de treinamento, uma vez que tais dados podem conter erros de tradução, além de nem todos possuírem um bom nível informacional porque muitas vezes nos dados de treino é colocado apenas a alternativa correta, sem explicação alguma, como é possível perceber na figura abaixo.

A retinoscopia em uma criança de 5 anos é melhor feita com: A)Atropina B)Homatropina C)Ciclopentolato D)Tropicamida	Alternativa A. Nenhuma.

Figura 1 - Exemplo de uma linha do dataset de treino. À esquerda a pergunta e à direita a resposta.

Hipóteses para a geração textual pobre

Além disso, outra linha investigativa é a busca por razões que explicam a geração textual pobre por parte dos modelos menores (GPT-2 e DistilGPT-2).

Nesse sentido, pelo que foi possível extrair das leituras dos blogs, além do fator da qualidade dos dados, citado anteriormente, como se sabe os modelos menores possuem menos parâmetros e, consequentemente, uma menor capacidade de aprendizado, o que dificulta o entendimento de estruturas gramaticais complexas, a percepção de contextos subentendidos e a identificação de intenções. Ademais, em seu pré-treino, tais modelos são expostos menos a linguagem em questão, visto que são menores, e, por isso, seu aprendizado da língua é destinado a tarefas mais específicas, como: Identificação de Entidades Nomeadas, Classificação de Texto, dentre outros.

Sendo assim, uma possível justificativa para a geração textual pobre dos modelos menores é que, além da baixa qualidade dos dados de treino, esses modelos possuem uma menor capacidade de aprendizado, sendo destinados a tarefas mais específicas, enquanto os modelos maiores são destinados a tarefas mais complexas e generalistas.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 11 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Guilherme Corrêa Figueredo

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Na última entrega, foram planejadas as seguintes entregas para este termo:
    - A leitura do paper "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?";
    - A leitura dos tópicos 1, 2 e 5 do paper "A Survey of Large Language Models".
- OBS: essas leituras tinham por objetivo buscar possíveis razões para o baixo desempenho nos benchmarks escolhidos e para a geração textual pobre, por parte do modelo de linguagem de pequeno porte escolhido (DistilGPT-2, com 82 milhões de parâmetros).
- A leitura dos tópicos do segundo paper foi feita, porém, não encontrei nenhuma ideia além do que já tinha visto na leitura de blogs e sites.
- A leitura do primeiro paper foi enriquecedora e, portanto, trouxe as observações no documento Entrega Gate - 11/01/2024 .
    - De forma resumida, esse paper destaca a importância da complexidade linguística do corpus para o aprendizado de modelos de linguagem de pequeno porte, de modo que, ao serem treinados em um corpus mais simples, eles podem gerar texto de forma consistente, coerente e até mesmo comparável a modelos 50 vezes maiores, a nível de parâmetro.
- Além disso, para esta entrega foi planejada a organização de todas as entregas anteriores. Sendo assim, no drive IA Residência - Entregas e Termos estão separados, por pastas de cada gate, as entregas e termos anteriores.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Elaboração do TCC.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

[Documento “Entrega Gate - 11/01/2024” citado no Termo de Aceite de Entrega do dia 11 de janeiro de 2024]

Introdução

No trabalho pesquisado [1], os autores buscam investigar se modelos de linguagem de pequeno porte conseguem gerar textos coerentes e consistentes. Nesse sentido, duas hipóteses são levantadas: a primeira, sendo a ideia de que apenas modelos de linguagem de grande porte podem gerar textos coerentes e consistentes, enquanto a segunda partia do princípio que a complexidade linguística, a profundidade e a diversidade do corpus utilizado para treino poderiam dificultar o aprendizado de modelos de linguagem de pequeno porte.

Dessa forma, o trabalho aborda a segunda hipótese, ao focar no desenvolvimento de um dataset mais simples, em relação à complexidade linguística, e realiza o treino de modelos de linguagem de pequeno porte neste dataset, buscando avaliar a capacidade desses de gerar textos coerentes e consistentes.

Dataset criado

O dataset criado, chamado de TinyStories, é um dataset sintético de pequenas histórias, que contém de 2 a 3 parágrafos, com um desfecho simples e um tema consistente. Além disso, com o objetivo de diminuir a complexidade linguística, o dataset foi construído com o vocabulário que uma criança de 3 anos poderia entender.

Mesmo que o vocabulário seja simples, todo o dataset foi criado com o auxílio do ChatGPT (nas versões 3.5 e 4), para que fosse possível criar um corpus que combinasse todos os elementos qualitativos encontrados na linguagem natural, como gramática, vocabulário, fatos, raciocínio, porém pequeno e menos diverso.

Modelos utilizados e resultados

Para a realização do treino neste dataset criado, foram utilizados modelos que variam de 1 milhão de parâmetros até 35 milhões de parâmetros, mas também com o número de camadas variando de 1 a 8.

Ademais, para a comparação dos resultados, foi utilizado um modelo relativamente grande, o GPT-2, com 1.5 bilhões de parâmetros, como base de comparação.

A seguir, está uma tabela de resultados, que contém o prompt, que é uma história que o modelo deve continuar, à esquerda estão a quantidade de parâmetros e camadas do modelo, ao meio a história gerada pelos modelos e à extrema direita estão as notas atribuídas pelo GPT-4 para cada história criada, sendo que foram considerados 3 critérios, que variam de 0 a 10: gramática, que analisa o nível gramatical do texto, criatividade, que analisa o quão criativo foi o desfecho da história criada pelo modelo e consistência, que analisa o quanto o modelo conseguiu se manter fiel à história original.

Model	Generated text	Scores
Prompt	Once upon a time there was a little girl named Lucy. She was very adventurous. She loved to explore the world around her, especially when it was bright and sunny outside. One day, while exploring the nearby park, Lucy came across a ladder leaning on a wall. She was curious to see what's on top, so she climbed the ladder, but when she reached the top, the ladder fell and she was stuck. A nearby park ranger noticed her and shouted out, "	
1M 8 layers	What are you doing here?" Lucy smiled and said, "I'm going to help!" The ranger smiled and said, "That's a great idea! Let's go see if we can find out."Lucy was so excited and said, "Yes, let's go!" So, Lucy and the ranger went to the park. They found lots of fun things and had lots of fun.	Grammar: 6/10 Creativity: 3/10 Consistency: 2/10
2.5M 8 layers	What are you doing, Lucy?" Lucy replied, "I'm trying to get out of the ladder, but I can't get out!" The ranger smiled and said, "Don't worry, I'll help you get out of the ladder. I'll help you get out." So, Lucy and the ranger got out of the ladder and started to climb up the ladder. When they reached the bottom, Lucy was so happy. She thanked the ranger for helping her and said, "I'm glad I could help." The ranger smiled and said, "You're welcome, Lucy. I'm glad I could help."	Grammar: 5/10 Creativity: 6/10 Consistency: 3/10
8.3M 8 layers	Lucy! What are you doing up there?" Lucy replied, "I'm trying to reach the top of the ladder, but I'm too small." The park ranger smiled and said, "Don't worry, I can help you. I can help you." The park ranger was very kind and said, "I can help you reach the top of the ladder if you want." Lucy was so excited and said, "Yes please!" The park ranger helped Lucy reach the top of the ladder and Lucy was so happy. She thanked the park ranger and said, "Thank you for helping me!" The park ranger smiled and said, "You're welcome, Lucy. I'm glad I could help."	Grammar: 7/10 Creativity: 5/10 Consistency: 5/10
28M 8 layers	What are you doing up there, little one?" Lucy replied, "I wanted to see what was on top of the ladder, but it fell and I'm stuck!" The park ranger smiled and said, "Don't worry, I'll help you down." He carefully lifted Lucy off the ladder and put her safely back on the ground. Lucy thanked the park ranger and said, "I'm glad you were here to help me." The park ranger smiled and said, "It was my pleasure. I'm glad I could help." Lucy smiled and ran off to explore the rest of the park.	Grammar: 9/10 Creativity: 6/10 Consistency: 9/10
21M 1 layer	Don't worry, Lucy! I'm here to rescue you!" Lucy quickly climbed up the ladder and rescued Lucy. She was so happy to be safe and thanked the park ranger for rescuing her. The park ranger smiled and said, "You're welcome, Lucy. I'm glad I could help." The park ranger smiled back and said, "You're welcome, Lucy. I'm glad I could help."	Grammar: 8/10 Creativity: 3/10 Consistency: 7/10
33M 2 layers	What are you doing up there, Lucy?" Lucy replied, "I wanted to see what was on top of the ladder." The park ranger smiled and said, "Well, it looks like you have a lot of fun. But remember, it's important to be careful when exploring new places." Lucy thanked the park ranger and carefully climbed back down the ladder. From then on, she always remembered to be careful when exploring new places.	Grammar: 7/10 Creativity: 6/10 Consistency: 4/10
GPT2-XL	Lucy, if you want to come up, I'll take you up and there's a ladder that'll help you get down." It was a very kind and simple act, and after trying many doors and ladders, she finally managed to climb up the ladder. However, since the ladder was very heavy and couldn't be used to jump down off, she climbed up by making the ladder lean against the wall. Her climbing was slow. First, she had to climb to a nearby rock, then she had to crawl to the cliff and then herassador and back. Finally, she managed to reach the top.	

Tabela 1 - comparação de resultados entre os diferentes modelos.

Conclusão do trabalho

A partir da leitura e análise deste trabalho, fica perceptível que uma das principais causas para a dificuldade de geração textual, coerente e consistente, por parte de modelos de linguagem de pequeno porte, é a complexidade linguística inerente ao idioma em questão, além da profundidade e diversidade do corpus.

No entanto, nota-se, também, que a criação de um corpus, mais simples, mas ainda mantendo as características gramaticais, sintáticas e semânticas, é possível fazer um

modelo de linguagem de pequeno porte gerar texto de forma coerente, consistente e até mesmo equiparável a um modelo de linguagem 50 vezes maior.

Conclusão geral

Sendo assim, a partir das pesquisas conduzidas nesta entrega e na entrega anterior (Entrega - Gate 21/12/23), destacam-se três razões para o baixo desempenho e geração textual pobre, por parte dos modelos de linguagem de pequeno porte:

- a quantidade menor de parâmetros desses modelos o que, consequentemente, reduz sua capacidade de aprendizado;
- a baixa qualidade dos dados, já que foram dados traduzidos, sem uma curadoria avançada, então tais dados podem conter erros de tradução, além de nem todos possuírem um bom nível informacional porque muitas vezes nos dados de treino é colocado apenas a alternativa correta, sem explicação alguma;
- os dados apresentam um nível de complexidade de linguagem alto, já que estão atrelados ao domínio específico da saúde e isso também dificulta o aprendizado de modelos menores.

Referências

[1] ELDAN, R.; LI, Y. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? Disponível em: <https://arxiv.org/abs/2305.07759>.