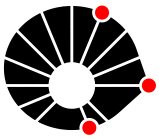

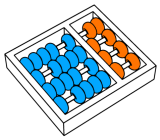

Renan H. Bastos

Relatório Técnico - IC-PFG-23-21
Projeto Final de Graduação
2023 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Geração de alternativas para testes e obtenção de contexto para questões

Renan Hiroki Bastos*

Resumo

Provas e testes são parte fundamental da aprendizagem. Neste trabalho foi utilizado o cojunto de dados EXAMS[2], contendo perguntas, respostas, contexto e matéria relativos ao conteúdo do ensino médio para realizar diversos treinamentos de modelos de processamento de linguagem natural, com o objetivo de gerar alternativas corretas e incorretas para testes. Os dados foram pré-processados para permitir diferentes estratégias de treinamento dos modelos. O modelo base utilizado para os treinamentos foi o PTT5[5], que se mostrou o mais robusto modelo pré-treinado para a tarefa de geração de texto na lingua portuguesa. Os melhores resultados foram obtidos fornecendo-se pergunta, matéria, resposta correta e contexto como entrada ao modelo, com métricas ROUGE1 = 77.52 e BLEU = 63.68. Além disso foi realizada pesquisa no âmbito de geração de contexto para perguntas para a criação de um conjunto de dados melhorado, a ser utilizado em treinamentos futuros.

1 Introdução

Provas são parte importante do processo de aprendizado. A utilização de testes repetidamente traz um benefício mnemônico, estudado em psicologia cognitiva no chamado "efeito de ser testado"[1] e auxilia na retenção de informação. Assim, as provas podem ser utilizadas tanto com seu caráter avaliativo tradicional, quanto como método ditádico. Em uma era digital, de dispersão quase instantânea da informação e de curtos períodos de atenção, a obtenção instantânea de quizzes pode se tornar uma maneira efetiva de melhorar o aprendizado.

Processamento de linguagem natural (NLP na sigla em inglês) é um ramo da Interligência Artificial (IA) que estuda a dar a computadores a capacidade de entender textos e falas da mesma maneira que os humanos. Modelos NLP já fazem parte integral da vida de muitas pessoas, sendo o chat GPT provavelmente a grande referência

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

conhecida por quase todos. Esses modelos comunmente utilizam transformers que seguem a seguinte arquitetura. Primeiramente o texto passa por um processo de embedding que transforma as palavras em vetores numéricos que preservam semântica e contexto[3], esses vetores são entregues a uma pilha de enconders e decoders, na qual cada um possui uma camada de "auto-atenção", e ao fim a saída desta pilha é entregue as camadas linear final e softmax, que transformam o vetor numérico novamente em palavras[4].

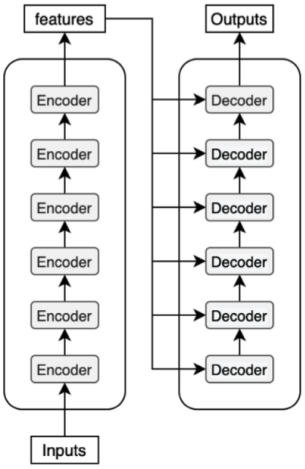

Figura 1: *Arquitetura de um transformer*

Nos dias atuais, existe uma grande quantidade de modelos de NLP com licença open source disponíveis na rede. Entre esses modelos há também grande variedade de tipo e de tarefas realizadas. Utilizando estes modelos como base é possível grande customização e especificação da tarefa final.

Este trabalho utiliza esses modelos para cumprir partes da tarefa maior de geração quizzes (questões de multipla escolha e suas alternativas) a partir de uma entrada de texto. Sendo estas partes, a geração de alternativas certas e erradas a partir de uma questão e um contexto, esta sendo uma tarefa-fim do trabalho. E geração de contexto a partir de uma pergunta, uma tarefa-meio, necessária para a realização da primeira.

Para o treinamento de modelos para a tarefa-fim foi utilizado o conjunto de dados pronto EXAMS[2]. Para a tarefa-meio, cujo objetivo era expandir este conjunto de dados, foram utilizadas como fontes de contexto as apostilas pré-vestibular disponibilizadas pela Unesp e pela Cecierj.

Este relatório está organizado da seguinte forma: Na Seção 2 citamos alguns trabalhos relacionados ao nosso estudo. Na Seção 3 descrevemos a metodologia do trabalho em duas partes, na primeira explicando como foi criada a corrente de perguntas e respostas utilizada para a geração de contexto, e na segunda explicando as fontes de dados e modelo base utilizados e como estes foram tratadas para criar um

modelo degeração de alternativas. Apresentamos também como os experimentos foram organizados e as estratégias aplicadas. A Seção 4 apresenta os resultados obtidos; A Seção 5 desenvolve uma discussão sobre os resultados. Por fim, apresentamos uma conclusão na Seção 6.

Todos os Jupyter Notebooks utilizados nos experimentos de treinamento, bem como os resultados dos testes e os pdfs usados no gerador de contexto podem ser encontrados no repositório `https://github.com/hirokibastos/quizzer/tree/main`

2 Síntese da Literatura

Diversos estudos feitos nos últimos anos sobre interfaces cérebro-computador baseados em EEG podem ser encontrados na literatura. A Tabela 1 apresenta alguns deles, que serão descritos a seguir.

No trabalho de Wang *et al.*, técnicas de reconhecimento de emoções por ondas cerebrais foram consideradas e um método de análise de emoções foi proposta utilizando o domínio de *valence-arousal* (valência e alertamento). Um modelo multidimensional de reconhecimento de emoções foi utilizado para analisar e caracterizar os sinais de EEG como algoritmo SVM como método de aprendizado de máquina aplicado. O resultado mostrou que cada canal de 4 bandas de frequência pode reconhecer eficientemente 20 emoções diferentes.

Utilizando redes neurais profundas baseado em 1D-CNN (rede neural convolucional de uma dimensão), Jiang *et al.* propôs um método de classificação de EEG para melhorar a acurácia de reconhecimento de padrões de BCI (Interfaces cérebro-computador). O trabalho considera pacientes com Doença de Alzheimer por meio de cenas e imagens mostradas em um ambiente de realidade virtual. O dispositivo EEG recebe os sinais cerebrais do paciente e rotula a imagem de acordo com o tipo do sinal.

Gena *et al.* conduziu um experimento no qual usuários eram expostos a um conjunto de trabalhos artísticos que sugeriam emoções para identificar os níveis de atenção pelas ondas cerebrais. O dispositivo Emotiv Epoc foi utilizado para obter os dados. O trabalho analisou a existência de diferenças sobre as ondas cerebrais nos diferentes hemisférios cerebrais em relação a emoções. Um índice de atenção foi medido de maneira assimétrica utilizando a informação de eletrodos posicionados em ambos hemisférios cerebrais. O índice foi comparado com o engajamento dos participantes para testar a confiabilidade de detecção de atenção baseada em BCI.

Kujit e Almardani adotaram uma abordagem preditiva ao estudar processamento emocional humano baseado em atividade cerebral. Eles investigaram a predição de empatia humana por autoavaliação baseando-se na assimetria cortical em diferentes áreas do cérebro pela análise de EEG. Foram avaliados diferentes tipos de modelos preditivos, como análises de regressão linear e classificadores binários. Os resulta-

dos mostraram que a análise da lateralização das oscilações do cérebro em faixas de frequência específica é importante para predizer níveis de empatia por autoavaliação.

Assim como no trabalho de Wang *et al.*, o nosso presente trabalho utiliza um classificador SVM. Exploramos dados para treinamento do modelo que foram coletados de maneira diferente, sem a quantificação por meio do domínio valência-alertamento.

Tabela 1: Estudos relacionados a interfaces cérebro-computador baseados em EEG.

Ano	Título
2021	Multidimensional emotion recognition based on semantic analysis of biomedical EEG signals for knowledge discovery in psychological healthcare.
2020	An EEG emotion classification system based on one-dimension convolutional neural network and virtual reality.
2020	Prediction of Human Empathy based on EEG Cortical Asymmetry.
2019	Do BCIS detect user's engagement? The results of an empirical experiment with emotional artworks.

3 Metodologia

Este estudo está separado em duas partes principais, que são a obtenção de contexto a partir de perguntas e a geração de alternativas a partir de perguntas e contexto.

3.1 Obtenção de contexto

Durante o processo de treinamento do modelo gerador de alternativas, percebemos que seria necessário fornecer contexto na entrada deste, visto que na grande maioria dos casos o conteúdo da pergunta não seria o suficiente para obtenção das respostas. Como o dataset sendo preparado para uso neste treinamento em um estudo realizado em paralelo possuia apenas questões sem contexto, mostrou-se necessário desenvolver um método para obtenção de contexto a partir de perguntas.

3.1.1 Tecnologia utilizada

A principal biblioteca utilizada para esta tarefa foi a LangChain. Esta biblioteca fornece um framework para conectar "large language models"(LLMs) a outras fontes de dados, como páginas web e arquivos pdfs. Ela nos permite indexar arquivos através de diversos tipos de Vectore Stores (representações matemáticas de textos que permitem rápidas buscas por similaridade), e utilizar estes Vectore Stores para criar chains (correntes) de perguntas e respostas para fazer perguntas ao modelo utilizado.

A LangChain faz uso de diversas outras bibliotecas para chegar em seus reultados. A FAISS é uma biblioteca para rápidas buscas por similaridade em vetores

densos, contendo algorítmos que fazem buscas em vetores de qualquer tamanho (até aqueles que não cabem na RAM). Ela é escrita em C++ e possuí wrappers para python/numpy. Neste trabalho a FAISS é utilizada para criar vector stores dos documentos pdf.

3.1.2 Modelo Utilizado

A peça mais fundamental na qualidade das saídas geradas a partir de uma corrente do LangChain é o LLM utilizado para gerar as respostas. Sendo assim, foram testados diferentes modelos para a tarefa, entre eles alguns modelos multilinguais, e um modelo pré-treinado na língua portuguesa (unicamp-dl/ptt5-base-portuguese-vocab)[5].

Até o momento da escrita deste relatório, o ecossistema Langchain-HuggingFace permite apenas o uso de modelos dos tipos text-generation e text2text-generation,o que limita bastante a variedade de modelos pré-treinados em português disponíveis. Além disso modelos muito grandes constantemente resultavam em timeout da API da HuggingFace durante a geração de respostas para as queries.

Este modelo obteve o melhor resultado por alta marge, e for, portanto, escolhido para a tarefa.

3.1.3 Fontes de Contexto

São utilizadas como fontes de dados para a geração de contexto as apostilas pré-vestibular disponibilizadas pela Unesp (`https://www2.unesp.br/portal#!/servico_ses/materiais-didaticos/`) e pela Cecierj(`https://canal.cecierj.edu.br/conteudo/pre_vestibular_social`). Estas apostilas são arquivos pdfs, separados por grandes áreas do conhecimento abordadas no ensino médio (Matemática, Física, Química, Linguagens, Biologia e Ciências Humanas).

3.1.4 Criando a corrente e obtendo o contexto

Para a criação da corrente geradora de contexto primeiramente carregam-se os arquivos pdfs através de um loader, neste trabalho foram utilizados dois loaders diferentes (PyPDFDirectoryLoader e UnstructuredPDFLoader), pois cada um produzia um resultado melhor que o outro em etapas diferentes do processo. Em seguida é carregado um embedding consistente com o LLM a ser utilizado, neste caso, como o modelo será carregado do HuggingFace, é utilizado o HuggingFaceEmbeddings.

Estes dois componentes são, então, utilizados para a criação de Vector Stores. Foram criados dois Vector Stores diferentes, um utilizando a FAISS, que será utilizado para fazer uma pré-busca por similaridade nos documentos para a geração do input da corrente. E um segundo, utilizando a classe VectorstoreIndexCreator, que é utilizado como retriever na corrente final. Para a criação deste segundo Vectore Store é necessário ainda definir o tamanho dos pedaços nos quais os documentos de

entrada serão divididos, valores mais altos podem trazer resultados melhores, mas requerem maior poder de computação, o valor 1000 mostrou bons resultadoscom boa performance.

Por fim, é criada a corrente utilizando a classe RetrievalQA. E é possível, então, fazer perguntas ao modelo.

3.2 Geração de Alternativas

3.2.1 Dados utilizados

Para os treinamentos realizados neste trabalho foi utilizado o conjunto de dados EXAMS[2], ele é um conjunto de dados multilingual de perguntas e respostas de questões de ensino médio, disponível no HuggingFace(`https://huggingface.co/datasets/exams`). Para este projeto foi utilizado apenas o subset de perguntas em português (que possui 924 questões, dois quais 80% fazem parte do split de treinamento e 20% do split de validação, o dataset nao possui split de teste).

No inicio deste trabalho este foi o único dataset de questões de multipla-escolha em português encontrado, sendo esta a razão para seu uso.

Os items do conjunto possuem a seguinte estrutura:

- id: ID da questão, único no dataset
- question: um sub-dicionário contendo:
  - stem: o texto da questão
  - choices: um sub-dicionário com 4 respostas candidatas, cada uma contendo:
    - text: texto da alternativa
    - label: um rótulo no conjunto ['A', 'B', 'C', 'D'] usado para corresponder com o answerKey
    - para: (opicional) um parágrafo de contexto do Wikipedia da mesma língua da questão
- answerKey: o rótulo da alternativa correta
- info: sub-dicionário contendo informação extra sobre a pergunta:
  - grade: o ano de escola do qual a pergunta foi retirada
  - subject: a matéria a qual a pergunta pertence
  - language: o nome em inglês da língua da pergunta

3.2.2 Pré-processamento dos dados

Para o treinamento de geração de alternativas foi necessário pré-processar este conjunto de dados. Inicialmente isso foi feito separando cada questão em quatro items diferentes, um para cada uma das alternativas, contendo os campos stem, choices/text, choices/para, info/subject e um novo campo correct, com a informação de se a alternativa esta correta ou não.

Para a maioria dos experimentos, este conjunto ainda foi dividido em duas partes, um contendo apenas as alternativas corretas e outra contendo as alternativas incorretas.

Para alguns experimentos o conjunto foi dividido em splits de treino, validação e teste.

3.2.3 Modelo base

O modelo base utilizado para os treinamentos dos modelos geradores de alternativa é o mesmo utilizado para o gerador de contexto, o PTT5. Ele é um modelo T5 pré-treinado no corpus BrWac, uma grande coleção de páginas web em português, e utiliza vocabulário próprio, treinado na Wikipédia em português.

Modelos pré-treinados diretamente na língua portuguesa se mostraram muito mais eficientes na geração de alternativas em português, quando comparados com modelos multilinguais. Dentre estes o PTT5 é o mais robusto e mais documentado, sendo assim selecionado para uso neste projeto.

3.3 Protocolo experimental

Realizamos experimentos para treinar um modelo que seja efetivo em gerar alternativas corretas ou incorretas. Visamos entender se os dados em mãos eram suficientes para treinar esse modelo. Realizamos cinco experimentos, que se diferenciam no pré-processamento dos dados e quantidade de modelos gerados. Em todos os experimentos faziam parte da entrada do modelo o texto da pergunta (sten) e sua matéria (subject), outras informações variam entre os experimentos.

O treinamento do modelo foi conduzido da mesma forma em todos os experimentos, utilizando a classe Seq2SeqTrainer da biblioteca transformers em notebooks python no ambiente Google Colab. A utilização da versão gratuita desta ferramenta, que possuí limites baixos de memória e processamento de dados, limitou consideravelmente os treinamentos possíveis, por exemplo, não foi possível tornar o número máximo de tokens da entrada maior do que 512 em nenhum dos experimentos, devivo ao limite de memória.

Durante e ao final do processo foram utilizadas as métricas Rouge e Bleu para avaliar os resultados dos treinamentos. Em parte dos experimentos foi utilizada pla-

taforma de registro de modelos "Weights & Biases", para armazenar e comparar métricas e parâmetros dos diferentes treinamentos.

- **Experimento #1**

  - Modelo único / Sem contexto

  O primeiro experimento foi um treinamento de um único modelo gerador de alternativas certas e erradas. Neste experimento as entradas continham o texto da pergunta, a matéria e se a alternativa a ser gerada deveria ser correta ou incorreta, não foram incluídos cotexto e resposta correta na entrada no modelo. O principal objetivo deste experimento foi desenvolver o algorítimo que seria usado para os futuros treinamentos.

- **Experimento #2**

  - Modelo único / Com contexto

  O segundo experimento seguiu tentando realizar o treinamento de um único modelo. Nele foi incluído contexto na entrada do modelo, porém a resposta correta continuou ausente.

- **Experimento #3**

  - Modelos separados / Com contexto

  Após o experimento #2, verificamos que possivelmente, o treinamento de dois modelos separados, um gerador de respotas corretas e um gerador de respotas incorretas pudesse trazer melhores resultados. Assim no experimento #3 foram treinados dois modelos, um gerador de respostas corretas e outro gerador de respostas incorretas, ambos com contexto na entrada, mas ainda sem a resposta correta.

- **Experimento #4**

  - Modelos separados / Com contexto / Respota certa na entrada

  O quarto experimento foi realizado após análise de que a inserção da resposta correta na entrada poderia auxiliar na geração de alternativas incorretas[6]. Assim foi feito um novo treinamento do modelo de geração de alternativas incorretas, desta vez incluindo a alternativa correta na entrada.

  Além disso, a partir deste experimento o dataset passou a ser dividido entre splits de treino, validação e teste

- **Experimento #5**

  - Modelos especializados / Com contexto / Respota certa na entrada

O quinto experimento foi uma tentativa de dividir ainda mais os modelos, realizando o treinamento de um modelo para cada grande área do conhecimento (Biologia, Filosofia, Economia, etc). Neste experimento foram realizados o treinamento de quatro modelos geradores de alternativas incorretas. Vale ressaltar aqui que a quantidade de dados para algumas das áreas, principalmente para alternativas corretas, passou a ficar bastante reduzida neste experimento.

A entrada dos modelos continha tanto o contexto, como a resposta correta.

- **Experimento #6**

  - Geração de contexto

  No sexto experimento foi fornecido o material de Biologia das apostilas coletadas para a chain de perguntas e respostas criadas com o LangChain. Então foram feitas diversas perguntas sobre tópicos da grade de ensino da matéria para o ensino médio.

4 Resultados

4.1 Experimento #1: Modelo único / Sem contexto

Nesse cenário, obtivemos as seguintes métricas ao fim do treinamento:

Rouge1	Rouge2	Rougel	Rougelsum
43.276400	27.196000	41.446500	41.451200

Tabela 2: Métricas ao fim do experimento #1

4.2 Experimento #2: Modelo único / Com contexto

Nesse cenário, obtivemos as seguintes métricas ao fim do treinamento:

Rouge1	Rouge2	Rougel	Rougelsum
78.650600	70.007700	77.109700	77.125400

Tabela 3: Métricas ao fim do experimento #2

4.3 Experimento #3: Modelos separados / Com contexto

Nesse cenário, obtivemos as seguintes métricas ao fim do treinamento:

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
69.450000	58.440000	67.470000	67.480000	53.020000

Tabela 4: Métricas ao fim do experimento #3: Gerador de alternativas incorretas

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
52.310000	39.570000	50.670000	50.630000	33.150000

Tabela 5: Métricas ao fim do experimento #3: Gerador de alternativas corretas

4.4 Experimento #4: Modelos separados / Com contexto / Respota certa na entrada

Nesse cenário, obtivemos as seguintes métricas ao fim do treinamento:

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
77.520000	67.780000	75.700000	75.680000	63.680000

Tabela 6: Métricas ao fim do experimento #4

4.5 Experimento #5: Modelos separados / Com contexto / Respota certa na entrada/ Especializado

Nesse cenário, obtivemos as seguintes métricas ao fim do treinamento:

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
82.670000	75.470000	81.820000	81.800000	76.520000

Tabela 7: Métricas ao fim do experimento #5: Gerador de alternativas incorretas especializado em Biologia

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
71.900000	61.650000	70.780000	70.800000	62.480000

Tabela 8: Métricas ao fim do experimento #5: Gerador de alternativas incorretas especializado em Economia

4.6 Comparação entre os experimentos:

Para alguns dos treinamentos de modelos geradores de alternativas incorretas foi utilizada a plataforma "Weights & Biases"para fazer um comparação entre os treinamentos. Eles podem ser visualizados na seguinte imagem.

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
65.710000	53.340000	62.640000	62.710000	48.470000

Tabela 9: Métricas ao fim do experimento #5: Gerador de alternativas incorretas especializado em Geologia

Rouge1	Rouge2	Rougel	Rougelsum	Bleu
53.200000	41.070000	50.230000	50.370000	37.200000

Tabela 10: Métricas ao fim do experimento #5: Gerador de alternativas incorretas especializado em Filosofia

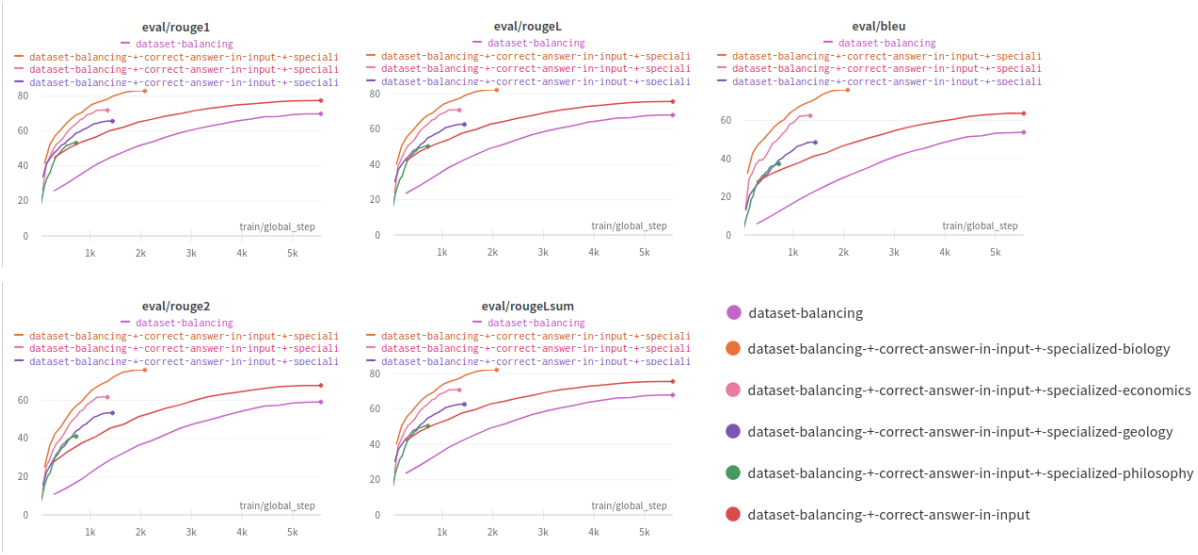

Figura 2: *Comparação entre treinamentos*

4.7 Experimento #6: Geração de contexto

Neste experimento foram obtidas as seguintes respostas para as queries realizadas.

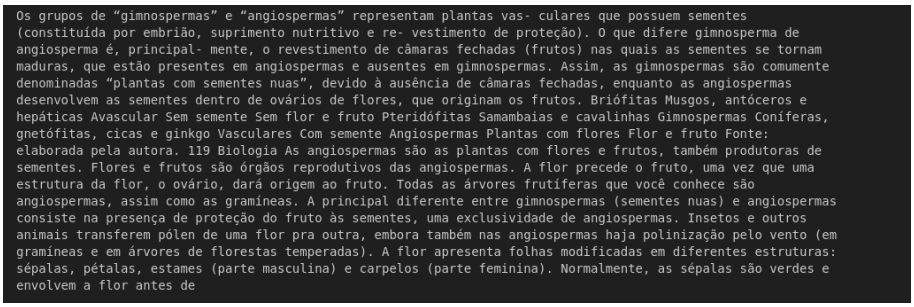

Figura 3: *Geração de contexto: O que é angiosperma?*

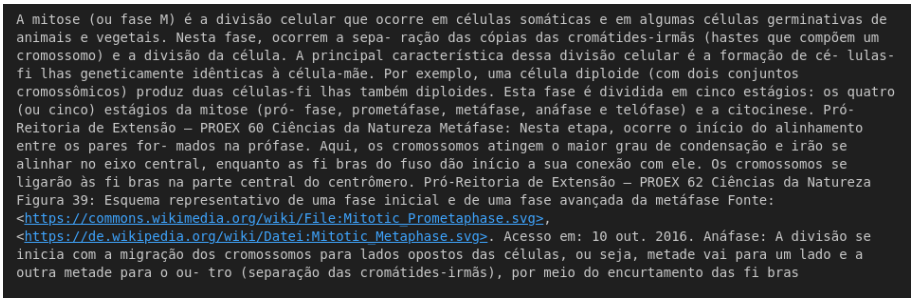

Figura 4: *Geração de contexto: Quais as fases da mitose?*

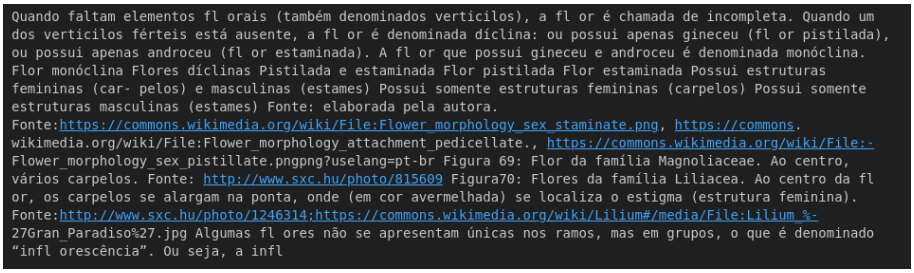

Figura 5: *Geração de contexto: Qual a diferença de floema e xilema??*

5 Discussão

Nos experimentos conduzidos, o segundo atingiu o melhor resultado nas métricas, apesar disso o modelo não se mostrou efetivo, pois gerava alternativas corretas e incorretas sempre idênticas, fato que não é percebido através das métricas.

Ao separar os modelos, houve queda no valor nas métricas, e as saídas obtidas para alternativas corretas e incorretas passaram a se tornar diferentes. O modelo

gerador de alternativas corretas raramente gerava alternativas incorretas, apesar de apresentar métricas mais baixas, o mesmo não pode ser dito sobre o modelo gerador de alternativas incorretas, este frequentemente gerava respostas verdadeiras.

Ao adicionar a resposta correta na entrada do modelo gerador de alternativas incorretas foi obtido o melhor resultado do trabalho, este foi o experimento que melhor combinou as métricas com frequência de geração de respostas com a corretude desejada.

Para os modelos especializados podemos ver na imagem de comparação que todos apresentam taxa de crescimento de métricas muito mais acelerados do que modeloso não especializados, e que o valor final da métrica é proporcional a quantidade de dados para cada matéria. Porém a quantidade limitada de de dados de treinamento não nos permite concluir que a separação dos modelos é um caminho viável a ser seguido.

Na geração de contexto, podemos perceber que os resultados são aceitaveis para queries simples (do tipo "O que é X", por exemplo), mas ruins para queries complexas. Estes resultados podem ser aprimorados através do uso de um modelo pré-treinado especificamente para a tarefa, com melhoria na escolha das queries, ou com a inclusão de mais fontes de dados.

Em trabalhos futuros visa-se aprimorar o conjunto de dados utilizado nos treinamentos, para isto pode ser utilizado o gerador de contexto desenvolvido neste projeto, em conjunto com a coleta de dados e o modelo gerador de tópicos criados em paralelo. É importante coletar mais dados que correspondam mais fortemente com a realidade do projeto, para que o modelo final consiga aprender mais efetivamente. É igualmente importante que mais estratégias de treinamento sejam testadas, podendo aumentar a qualidade das respostas geradas. É deixada, ainda, a recomendação do uso de um ambiente de treinamento mais poderoso que o Google Colab, visto que este foi fator limitante na condução deste projeto.

6 Conclusão

Neste trabalho realizamos o treinamento de modelos de processamento de linguagem natural para construir um modelo de geração de alternativas para testes. Foi utilizado um conjunto de dados pronto disponivel no hub da HuggingFace. Após pré-processamento dos dados, foram aplicadas diversas estratégias para o treinamento do modelo PTT5. O melhor treinamento teve métricas ROUGE1 = 77.52 e BLEU = 63.68. Além disso foi desenvolvido um método para obtenção de contexto para perguntas a partir de pdfs, utilizando a biblioteca LangChain.

Agradecimentos

Este trabalho teve apoio da aluna de Doutorado, Geovanna Evelyn Espinoza Taype, da aluna de graduação, Luma Oliveira Lombello, do laboratório de pesquisa InterHAD do IC/UNICAMP e do projeto temático FAPESP Socienativos (#2015/165280).

Referências

[1] Pastura, P. A., & Santoro-Lopes, G. (2013). O aprendizado melhorado por provas. Revista Brasileira De Educação Médica, 37(3), 429–433. https://doi.org/10.1590/s0100-55022013000300015.

[2] Hardalov, Momchil and Mihaylov, Todor and Dimitrina Zlatkova and Yoan Dinkov and Ivan Koychev and Preslav Nvakov(2020). EXAMS: A Multi-subject High School Examinations Dataset for Cross-lingual and Multilingual Question Answering. arXiv preprint arXiv:2011.03080.

[3] Peters, M. J., Neumann, M. E., Iyyer, M., Gardner, M., Clark, C. M., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. https://doi.org/10.18653/v1/n18-1202

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention Is All You Need. arXiv:1706.03762

[5] Carmo, Diedre and Piau, Marcos and Campiotti, Israel and Nogueira, Rodrigo and Lotufo, Roberto (2020). PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data. arXiv preprint arXiv:2008.09144

[6] Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic Distractor Generation for Multiple Choice Questions in Standard Tests. In Proceedings of the 28th International Conference on Computational Linguistics, pages 2096–2106, Barcelona, Spain (Online). International Committee on Computational Linguistics.