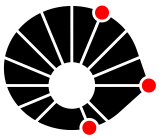

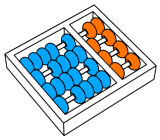

Treinamento de Modelos Generativos Multilíngues: Aplicação em Sistemas de Perguntas e Respostas

Arthur Cemin Baia *Julio Cesar dos Reis*

Relatório Técnico - IC-PFG-24-53
Projeto Final de Graduação
2024 - Agosto

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Treinamento de Modelos Generativos Multilíngues: Aplicação em Sistemas de Perguntas e Respostas

Arthur Cemin Baia Julio Cesar dos Reis *

Resumo

A literatura recente explora a necessidade crescente de sistemas de Inteligência Artificial (IA) capazes de compreender e responder a perguntas em várias línguas. Esta pesquisa se concentra no treinamento de modelos generativos multilíngues, com foco na geração de respostas às perguntas. Os modelos de linguagem desempenham um papel crucial na evolução dos sistemas de Perguntas e Respostas (QA), com modelos como o GPT demonstrando uma notável capacidade de compreensão contextual. Nosso estudo destaca a importância de modelos de linguagem de código aberto, que oferecem maior flexibilidade e controle, além de abordar questões de privacidade e segurança. O objetivo do trabalho é aprimorar a competência desses modelos na tarefa de responder perguntas em linguagem natural contextualizadas, utilizando modelos multilíngues para melhorar a acessibilidade e a eficácia da comunicação baseada em IA.

1 Introdução

Em um mundo cada vez mais globalizado e digital, a necessidade de sistemas de Inteligência Artificial (IA) capazes de compreender e responder às perguntas em várias línguas nunca foi tão grande. Essa necessidade é a motivação central desta investigação, que se concentra no treinamento de modelos generativos multilíngues com foco na geração de respostas a perguntas. Modelos de Linguagem têm desempenhado um papel crucial na evolução dos sistemas de Perguntas e Respostas (QA). Como evidenciado por modelos de última geração, como o GPT [1], esses modelos possuem uma capacidade notável de compreensão contextual. Essa habilidade é extremamente relevante para responder perguntas que estão intrinsecamente ligadas a um trecho específico de texto.

Esses modelos, alimentados por vastos conjuntos de dados e treinados através de técnicas avançadas de aprendizado de máquina, são meticulosamente projetados

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

para compreender a estrutura, a semântica e o contexto da linguagem. Isso lhes permite gerar respostas que são não apenas gramaticalmente corretas, mas igualmente relevantes e significativas em relação à pergunta de entrada.

Um exemplo notável disso é o *ChatGPT* [1], um sistema baseado em modelo de linguagem. Essa aplicação, que utiliza uma arquitetura de rede neural chamada *Transformer* [3], tem ganhado destaque no cenário tecnológico atual, devido a sua alta capacidade de lidar com textos. Sua relevância é evidenciada pelo fato de que foi o produto que alcançou mais rapidamente a marca de 1 milhão de usuários entre todos os produtos disponibilizados na internet[2].

Esse marco demonstra o potencial e a demanda por sistemas de linguagem avançados, efetivos e eficazes. Dessa forma, aproveitar esses modelos de linguagem sofisticados é permitir uma comunicação potencialmente mais efetiva e inclusiva em um cenário global. Ao desenvolver e treinar modelos de linguagem multilíngues, podemos esperar melhorar a acessibilidade e a eficácia da comunicação baseada em IA, abrindo novas possibilidades para a Interação Humano-Computador.

Os modelos de linguagem possuem uma alta capacidade linguística, sendo capazes de realizar múltiplas tarefas como análise de sentimento, classificação de texto e, como foco deste estudo, perguntas e respostas. No entanto, essa capacidade encontra um obstáculo significativo: muitos desses modelos possuem código fechado. Isso significa que seu código-fonte e os detalhes de implementação não estão disponíveis ao público.

Esta restrição gera várias desvantagens para os usuários. Primeiramente, não há garantias de privacidade dos dados, além dos termos de serviço propostos pelas empresas proprietárias dos modelos. Isso resulta em uma falta de transparência sobre como os dados gerados pelas interações com o modelo serão utilizados, levantando questões relacionadas à transparência e à auditabilidade. Adicionalmente, esses modelos de código fechado dificultam o processo de personalização. Como os pesos dos modelos não são públicos, não é possível realizar processos de treinamento via ajuste fino, o que poderia melhorar o desempenho em tarefas específicas. Portanto, apesar de sua alta capacidade linguística, a natureza de código fechado desses modelos acaba limitando o uso deles em cenários específicos.

O objetivo deste trabalho é aprimorar a competência desses modelos na tarefa de responder perguntas em linguagem natural contextualizadas. Em outras palavras, buscamos melhorar a efetividade desses modelos em identificar e fornecer respostas precisas dentro de um determinado trecho de texto. Esta investigação se justifica pela capacidade inerente e o contexto em que os modelos de linguagem são aplicados, pois eles possuem um potencial significativo para resolver uma ampla variedade de problemas.

Optamos por utilizar especificamente modelos de linguagem de código aberto neste estudo. A razão para isso é que esses modelos oferecem uma série de vantagens em

[1] `https://openai.com/blog/chatgpt`

[2] `https://shorturl.at/eEHIP`

relação aos seus equivalentes de código fechado. Eles proporcionam maior flexibilidade, pois permitem que os pesquisadores modifiquem e adaptem o código de acordo com suas necessidades específicas. Eles oferecem mais controle sobre o processo de treinamento e uso do modelo, o que pode levar a melhores resultados.

Os modelos de código aberto tratam questões importantes de privacidade e segurança, pois permitem que os usuários vejam exatamente como o modelo está processando e usando seus dados. Um aspecto crucial que nos levou a escolher modelos de código aberto é a possibilidade de treiná-los com dados abertos. Isso é particularmente útil no contexto de experimentos científicos, pois permite que os pesquisadores direcionem o treinamento do modelo para tarefas e contextos específicos, tal qual esse experimento de modelos direcionados a tarefa de perguntas e respostas.

Outro aspecto fundamental do nosso presente estudo é a natureza multilíngue dos modelos de linguagem que estamos utilizando. Modelos multilíngues são capazes de entender e gerar texto em várias línguas, tornando-os ferramentas valiosas para uma ampla gama de aplicações, desde a tradução automática até a análise de sentimentos em diferentes idiomas, superando a barreira idiomática, que anteriormente poderia ser um empecilho em sistemas que lidam com mais de um idioma.

Este relatório está organizado da seguinte forma: A Seção 2 apresenta trabalhos relacionados a este estudo. A Seção 3 descreve a metodologia do trabalho. Apresentamos os experimentos organizados e as estratégias da avaliação experimental. A Seção 4.2 relata os resultados obtidos; A Seção 5 desenvolve uma discussão sobre os resultados. Por fim, apresentamos uma conclusão na Seção 6.

2 Síntese da Literatura

2.1 Sistemas de Perguntas e Respostas

Nos últimos anos, inúmeros experimentos foram realizados envolvendo modelos de linguagem e a tarefa de *question-answering* [2]. Esta tarefa esta relacionada a construção de sistemas capazes de responder a perguntas formuladas em linguagem natural. Esses sistemas utilizam técnicas de Processamento de Linguagem Natural (NLP) e compreensão de texto para identificar a resposta correta dentro de um conjunto de dados ou corpus. Um exemplo de tarefa de Question Answering (QA) pode ser o seguinte: Suponha que temos um corpus de texto que inclui o seguinte paragrafo: "**A Torre Eiffel, localizada em Paris, França, é uma das estruturas mais reconhecíveis do mundo. Foi nomeada em homenagem a Gustave Eiffel, cuja empresa projetou e construiu a torre.**". A pergunta feita ao sistema poderia ser: "**Quem projetou a Torre Eiffel?**". O sistema de QA, então, analisaria o corpus de texto e retornaria a resposta: "**A torre Eiffel foi projetada por Gustave Eiffel**".

2.2 SQuAD

Um dos estudos mais significativos relacionados a essa tarefa de P&R é o SQuAD [4] (*Stanford Question Answering Dataset*). Este projeto, desenvolvido na Universidade de Stanford, revolucionou a maneira como os modelos de linguagem são treinados para a tarefa de *question-answering*. O SQuAD é um conjunto de dados composto por perguntas formuladas por humanos e respostas extraídas de artigos da Wikipedia[3]. Foi projetado para treinar modelos de linguagem para responder perguntas de maneira precisa e contextualizada. O sucesso do SQuAD foi tão grande que se tornou o padrão de referência para a avaliação de modelos de *question-answering*, especificamente na tarefa de *Extractive QA*. Apesar da qualidade desse conjunto de dados, na época de seu lançamento, os modelos de linguagem utilizavam a mesma tecnologia atual, *Transformer*, mas possuíam dezenas ou até centenas de vezes menos parâmetros. Por exemplo, os modelos BERT [5], que possuem em média centenas de milhões de parâmetros, são pequenos em comparação com os modelos GPT, que possuem em média centenas de bilhões de parâmetros. Esta diferença é fundamental para entender a qualidade e o nível de humanização das respostas geradas por esses novos modelos, que foram desenvolvidos anos após o advento desse *dataset*.

2.3 FAQuAD

Um estudo que se destacou no treinamento de modelos de linguagem em línguas alternativas foi o FAQuAD [6]. Este projeto criou um conjunto de dados no mesmo formato do SQuAD, mas com um objetivo específico. Os autores coletaram dados de documentos oficiais da Universidade Federal do Mato Grosso do Sul, com o intuito de treinar modelos de linguagem capazes de responder perguntas feitas por novos alunos da universidade.

O aspecto mais relevante deste estudo é o seu pioneirismo no treinamento de modelos de linguagem em línguas com recursos limitados. Este foi um dos principais focos do experimento, pois é essencial que o modelo seja capaz de responder perguntas nas duas línguas presentes no conjunto de dados. O trabalho do FAQuAD demonstrou a importância e a viabilidade de desenvolver tecnologias de linguagem para línguas além do inglês, abrindo caminho para futuras pesquisas e aplicações nesta área.

2.4 Sabiá

O Sábia [7], um modelo generativo pré-treinado em português, é de grande relevância para o nosso estudo atual. Este modelo compartilha uma arquitetura semelhante à do modelo escolhido para o nosso estudo, particularmente no que diz respeito ao seu

[3] `https://www.wikipedia.org/`

componente de decodificação. O decodificador é a parte do modelo que recebe a representação codificada de um texto de entrada e gera a próxima palavra na sequência, levando em consideração todas as palavras anteriores. A arquitetura do Sábia é baseada no modelo Llama [8], que, assim como o GPT, também utiliza uma rede *Transformer*. Treinado especificamente para o português, o Sábia alcançou resultados notáveis na tarefa de extração de respostas. Ele está disponível em duas versões, que diferem na quantidade de parâmetros: uma com 7 bilhões e outra com 65 bilhões. Esses números destacam a capacidade do *Sábia* de lidar com tarefas complexas de processamento de linguagem natural, demonstrando o potencial dos modelos de linguagem treinados em línguas específicas. Um dos principais pontos de conexão deste trabalho é a capacidade demonstrada pelo Sábia na tarefa de *Extractive Question-Answering* utilizando o FAQuAD. Esta capacidade é particularmente relevante para o nosso estudo, pois ilustra a eficácia dum modelo, arquiteturalmente semelhante ao escolhido por nós, na tarefa e língua especificada.

2.5 Fundamentação Tecnológica

Para realizar nosso presente estudo, adotamos a biblioteca *Transformer*[4] como componente fundamental. Essa ferramenta destaca-se por fornecer um *framework* especializado no treinamento de "*Large language models*"(LLMs)[5]. De maneira mais específica, simplifica consideravelmente o processo, abstraindo as complexidades das bibliotecas de treinamento de modelos e disponibilizando implementações prontas para uso imediato.

A escolha da *Transformer* viabilizou uma integração eficiente com a biblioteca *PyTorch*[6], selecionada para a construção do modelo. Essa decisão foi motivada pela capacidade única do *PyTorch* em oferecer diferenciabilidade e facilitar cálculos vetoriais e matriciais, características essenciais para o desenvolvimento do modelo em questão. Além de simplificar a construção do modelo, essa biblioteca desempenhou um papel crucial na otimização e no gerenciamento de recursos durante o treinamento. Através de suas funcionalidades integradas, conseguimos otimizar o desempenho do modelo, ajustar hiper-parâmetros de maneira eficaz e monitorar o progresso do treinamento em tempo real.

O ecossistema da biblioteca *Transformer* ainda oferece facilitadores adicionais para o processo experimental, como a biblioteca *Datasets*[7]. Essa biblioteca simplifica o acesso e compartilhamento de conjuntos de dados de processamento de linguagem natural (NLP), especificamente os conjuntos de dados escolhidos. Esses dados estão

[4] `https://huggingface.co/docs/transformers/index`
[5] `https://openai.com/research/better-language-models`
[6] `https://pytorch.org/`
[7] `https://huggingface.co/docs/hub/datasets-usage`

armazenados no *Hugging Face Hub*[8], e com o auxílio dessa biblioteca, os processos de geração de divisões para treinamento, avaliação e teste são facilitados.

3 Treinando Modelos para Realizar a Tarefa de Perguntas e Respostas

A Figura **??** apresenta o processo metodológico de nosso experimento. Desde a parte inicial da captura dos dados, a estruturação deles, a combinação deles com o modelo base no processo de ajuste fino e o resultado final dos modelos já treinados.

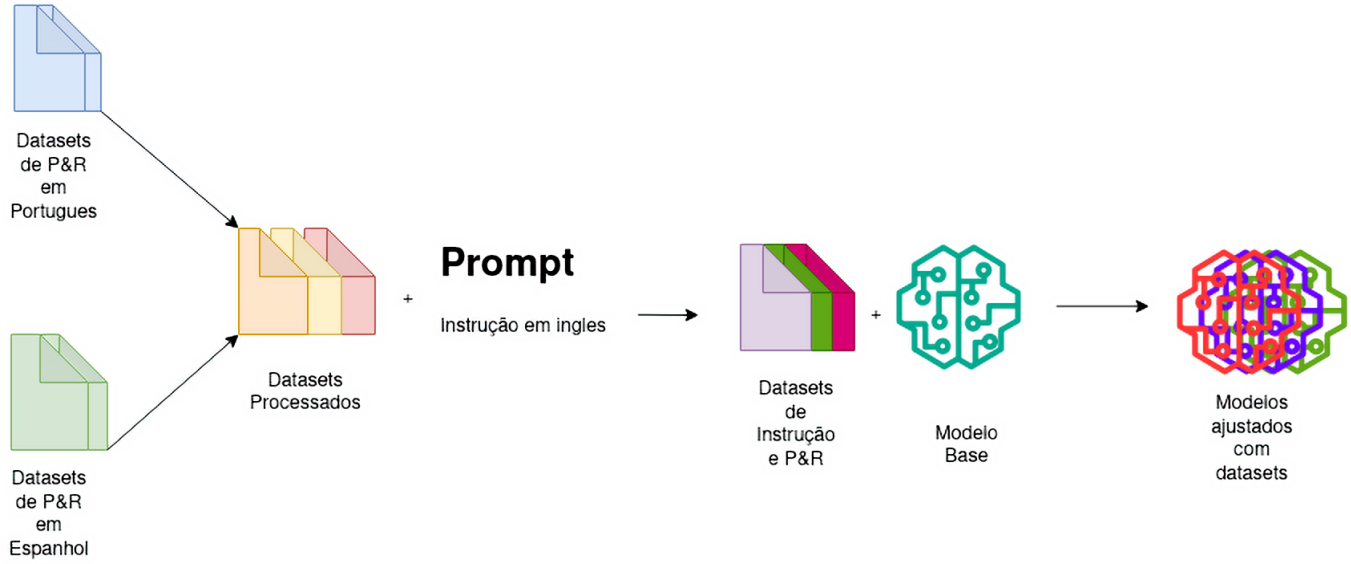

Figura 1: Metodologia do experimento proposto

3.1 Dados

Para o processo de ajuste fino do modelo nas duas línguas alvo - português e espanhol - selecionamos vários conjuntos de dados de perguntas e respostas. Para o português, utilizamos os conjuntos de dados ECQuAD[9], SQuAD pt-br[10], FAQuAD[6] e Dolly 15k[11]. Para o espanhol, recorremos aos conjuntos de dados SQAC[12] e MLQA[13]. A tabela 1 traz um resumo da quantidade de exemplos em cada conjunto de dados e o idioma principal de cada um.

[8] `https://huggingface.co/datasets`

Tabela 1: Conjuntos de dados que foram utilizados nesse estudo: nome, quantidade de exemplos e Idioma

Dataset	**Quantidade de Exemplos**	**Idioma**
SQuAD v1 pt-br	87.500	pt-BR
ECQuAD	8317	pt-BR
FAQuAD	900	pt-BR
Dolly 15k	3546	pt-BR
SQAC	18.810	es-ES
MLQA	5753	es-ES

3.1.1 SQuAD pt-br

Recentemente, o grupo *Deep Learning* Brasil [9] traduziu o conjunto de dados SQuAD v.1.1 para o português usando o *Google Translate*. Como esse mecanismo de tradução automática inevitavelmente propaga seus erros de decodificação para o mecanismo de perguntas e respostas, o grupo também passou cerca de dois meses fazendo correções no conjunto de dados traduzido antes de lançar o conjunto de dados SQuAD-pt-BR. Um exemplo de como os dados desse conjunto são estruturados, pode ser visto na Figura 2, que contêm um contexto, uma pergunta e uma resposta, contendo o texto e o índice de inicio da resposta dentro do contexto.

[9] `http://www.deeplearningbrasil.com.br/`

Figura 2: Exemplo de dado de conjunto extrativo

```
{
    title: 'University of Notre Dame',
    context: Arquitetonicamente, a escola tem um carater catolico. No
       topo da cupula de ouro do edificio principal esta uma estatua
       de ouro da Virgem Maria. Imediatamente em frente ao edificio
       principal e de frente para ele, esta uma estatua de Cristo de
       cobre com os bracos erguidos com a lenda "Venite Ad Me Omnes".
       Ao lado do edificio principal e a Basilica. Imediatamente atras
        da basilica fica a Gruta e uma replica da gruta de Lourdes, na
        Franca, onde a Virgem Maria apareceu para Santa Bernadette
       Soubirous em 1858.
    question: A quem a Virgem Maria supostamente apareceu em 1858 em
       Lourdes, Franca?,
    answer: {
        text: Santa Bernadette Soubirous,
        answer_start: 533
    }
}
```

3.1.2 ECQuAD

O conjunto de dados ECQuAD é um é um conjunto de dados de compreensão de leitura para responder perguntas em plataformas de *e-commerce* brasileiras. Consiste em perguntas anotadas por trabalhadores de *crowdsourcing* em um conjunto de descrições de produtos. Ele segue o formato SQuAD, explicitado na Figura 2.

3.1.3 Dolly 15k

Os dados do Dolly, como exemplificado pela Figura 3, têm uma fonte única e diversificada, proveniente de instruções geradas por milhares de funcionários do *Databricks*. A versão do *dataset* Dolly utlizada nesse experimento foi a traduzida para português[10].

Para a tradução foi utilizado LibreTranslate[11], que é uma ferramenta de tradução automática de código aberto. Ela permite a tradução de textos entre vários idiomas, sem a necessidade de compartilhar dados com serviços de terceiros. A ferramenta pode ser usada online ou instalada localmente, oferecendo privacidade e controle sobre os dados de tradução. Além disso, *LibreTranslate* suporta uma API, permitindo que desenvolvedores integrem a funcionalidade de tradução em suas próprias aplicações ou

[10] `https://huggingface.co/datasets/Gustrd/dolly-15k-libretranslate-pt`
[11] `https://libretranslate.com/`

serviços. Um exemplo do conjunto pode ser visto na Figura 3, que tal qual a Figura 2, contêm os mesmos elementos principais: contexto, pergunta e resposta, porém sem o indicador do inicio da resposta, demonstrando a diferença entre as sub-tarefas dentro do domínio de *question-answering*.

Figura 3: Exemplo de dado do conjunto Dolly 15k

```
{
    context: Virgin Australia, o nome comercial da Virgin Australia
       Airlines Pty Ltd, e uma companhia aerea australiana. E a maior
       companhia aerea por tamanho da frota para usar a marca Virgin.
       Iniciou em 31 de agosto de 2000 como Virgin Blue, com duas
       aeronaves em uma unica rota. De repente encontrou-se como uma
       grande companhia aerea no mercado interno da Australia apos o
       colapso da Ansett Australia em setembro de 2001. A companhia
       aerea cresceu para servir diretamente 32 cidades na Australia,
       a partir de hubs em Brisbane, Melbourne e Sydney.,
    question:  Quando a Virgin Australia iniciou ?
    answer: A Virgin Australia iniciou em 31 de agosto de 2000 como
       Virgin Blue, com duas aeronaves em uma unica rota.
}
```

3.1.4 SQAC

O conjunto de dados SQAC (*Spanish Question-Answering Corpus*) é um dataset de perguntas e respostas extraídas voltado para o idioma espanhol. Faz parte do projeto MarIA [14], que se concentra no desenvolvimento de modelos de linguagem em espanhol. As fontes dos contextos incluem artigos enciclopédicos da Wikipedia em espanhol, artigos de notícias do Wikinews e textos de agências de notícias e literatura do corpus AnCora. O objetivo principal do SQAC é fornecer um recurso para avaliar e treinar modelos de processamento de linguagem natural em espanhol, especialmente no contexto de perguntas e respostas extraídas. Ele segue a mesma estrutura da Figura 2.

3.1.5 MLQA

O MLQA (*MultiLingual Question Answering*) [13] é um conjunto de dados de referência criado para avaliar o desempenho de respostas a perguntas em várias línguas. Composto por mais de 5 mil instâncias de perguntas e respostas em sete idiomas, incluindo inglês, árabe, alemão, espanhol, hindi, vietnamita e chinês simplificado, o MLQA foi construído usando uma estratégia de alinhamento de contexto em artigos

da Wikipedia. O conjunto de dados, que segue o formato SQuAD, é dividido em conjuntos de desenvolvimento e teste, sendo destinado principalmente como um corpus de avaliação. Tal qual o outro conjunto em espanhol extrativo, ele segue a mesma estrutura da Figura 2.

3.1.6 Estrutura dos dados

Todos esses conjuntos de dados, com exceção do Dolly 15k, seguem a estrutura do SQuAD [4]. Esses dados são estruturados em trios: contexto, pergunta e resposta. Esta estrutura, que pode ser observada na Figura 2 permite que os modelos dominem a tarefa fundamental que é a compreensão de leitura de máquina, que é a capacidade do modelo de entender o texto e identificar informações relevantes.

Embora o conjunto de dados Dolly siga a mesma estrutura básica de contexto, pergunta e resposta, ele se distingue de outros conjuntos de dados em um aspecto crucial: a natureza das respostas. Em muitos conjuntos de dados, a resposta é tipicamente um trecho extraído diretamente do texto. No entanto, no Dolly, as respostas às perguntas apresentam uma maior contextualização. Isso se deve ao fato de que o Dolly foi originalmente desenvolvido para treinar modelos generativos, que são notórios por produzir respostas mais detalhadas e humanizadas.

Os outros conjuntos de dados são construídos a partir da extração de trechos de diferentes contextos, seguida pela formulação de perguntas com base nesses trechos e pela identificação de respostas dentro do texto. Esse processo resulta em conjuntos de dados ricos em informações factuais que abrangem uma ampla variedade de tópicos.

Apesar da diversidade de tópicos, as respostas são extraídas diretamente do texto podem não refletir a mesma profundidade de compreensão e contextualização presente nos conjuntos de dados criados pela empresa responsável pela construção desse conjunto de dados.

Nesses últimos conjuntos, há mais do que um simples trecho do texto; há criatividade e interpretação humana na elaboração da resposta, contribuindo para a qualidade das respostas de um modelo generativo. Portanto, a inclusão do conjunto de dados Dolly em nosso experimento não apenas enriquece a capacidade do nosso modelo de gerar respostas mais naturais e contextualmente relevantes, mas também aprimora a robustez e a generalização do nosso modelo. Isso permite que ele lide com uma variedade mais ampla de textos e contextos, proporcionando benefícios significativos para a qualidade e diversidade das respostas geradas.

3.2 Pré-processamento dos dados

A Figura 4 apresenta a divisão e junção dos conjuntos de dados escolhidos para o estudo, de forma que a as setas representam as cisões e junções dos *datasets* de treinamento e teste.

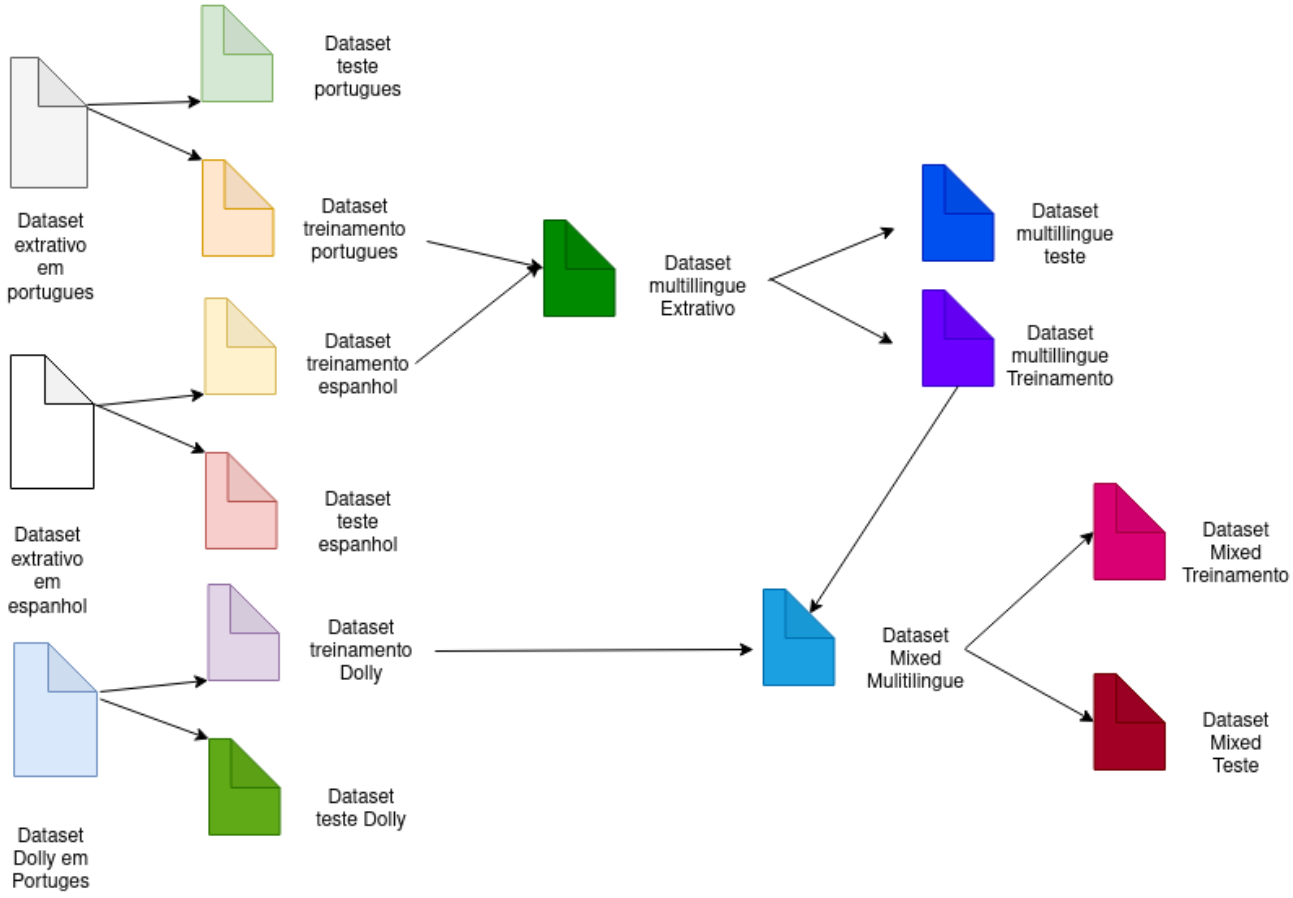

Figura 4: Divisão e junção dos conjuntos de dados

Dado a seleção de dados, é possível explicar melhor o processo de organização deles dentro do contexto de treinamento multi-lingue. No estudo em questão, analisamos a distribuição de dados na Tabela 1. Notamos um desequilíbrio entre a quantidade de dados em português e espanhol. Para lidar com essa discrepância, adotamos uma estratégia meticulosa na divisão dos dados, com foco particular nos conjuntos de treino, teste e validação.

Inicialmente, formamos um conjunto de dados em espanhol, que foi dividido em duas partes. A primeira, destinada ao treinamento, continha aproximadamente 80% dos dados, ou seja, 22106 exemplos. A segunda parte, composta pelos 20% restantes (2457 exemplos), foi reservada para testar a capacidade do modelo em espanhol. Com base no número de exemplos no conjunto de testes em espanhol, selecionamos uma quantidade igual de exemplos do conjunto de dados de teste em português (2457

exemplos) para manter a consistência. Para o conjunto de treinamento em português, combinamos os datasets ECQuAD e FAQuAD, resultando em 9127 exemplos. Os exemplos restantes foram selecionados aleatoriamente dos dados do SQuAD pt-br, gerando um conjunto com 22106 exemplos.

Os conjuntos de treinamento em espanhol e português foram criados com o objetivo de serem combinados em um grande conjunto de contextos, perguntas e respostas. Após a concatenação, o dataset de treinamento totalizou 44212 exemplos. Esses foram divididos em três partes: treino, teste e validação, seguindo a proporção 80/10/10, com 35369, 4421 e 4422 exemplos, respectivamente.

Para o nosso estudo, inicialmente separamos 10% do conjunto de dados Dolly (cerca de 355 exemplos) como um conjunto de testes independente para avaliações futuras do modelo. Os 90% restantes foram divididos em duas partes: 80% para treinamento e o restante para validação. Essa divisão foi feita com o objetivo de avaliar o desempenho do modelo durante o treinamento com este conjunto de dados.

Esses 90% de treinamento e validação foram adicionados ao *dataset* multilíngue de treinamento, o qual anteriormente havia apenas exemplos de extração de resposta. Este conjunto foi embaralhado aleatoriamente e dividido em três partes, com base na classe do exemplo, sendo elas: extrativa e dolly. Assim, esses três conjuntos mantêm a mesma distribuição de classes e contêm, respectivamente, 30848 exemplos para treino, 3856 para teste e 3856 para validação.

Tabela 2: Quantidade de exemplos em cada *dataset* gerado pós-processamento

Dataset	**Treino**	**Teste**	**Validação**
Espanhol	22106	2457	-
Português	22106	2457	-
Dataset extrativo multilíngue	35369	4421	4422
Dolly	2836	355	355
Dataset multilíngue + Dolly	30848	3856	3856

3.3 Modelo base

Dada a necessidade de utilizar um modelo de código aberto que permitisse a repetição de múltiplos experimentos, optamos pelo *Bloomz* [15]. Esse é um modelo de linguagem de grande porte baseado em *Transformer*, desenvolvido por mais de 1.000 pesquisadores de IA como parte da iniciativa *BigScience*[12]. O treinamento desse modelo de base para nosso estudo ocorreu de março a julho de 2022, envolvendo aproximadamente 366 bilhões de tokens. O *Bloomz* foi treinado utilizando dados de 46 línguas naturais e 13 linguagens de programação, totalizando 1,6 terabytes de texto pré-processado convertido em 350 bilhões de tokens únicos. Ele é capaz de

[12] `https://bigscience.huggingface.co/`

compreender instruções humanas em dezenas de idiomas, incluindo o Português e o Espanhol, que são relevantes para este experimento.

O modelo escolhido possui 1.7 bilhões de parâmetros, o que o torna significativamente menor em comparação com outros modelos citados neste trabalho. Essa característica o torna ideal para ambientes de treinamento com capacidade computacional limitada, como o utilizado neste experimento. No caso, empregamos uma única GPU, especificamente a NVIDIA A100[13]. Embora seja uma GPU excelente, é importante destacar seu alto custo de uso, o que impõe limitações na capacidade de explorar diversas combinações de modelos e variações de hiper-parâmetros.

3.4 Estruturação e Preparação dos Dados para Treinamento do Modelo

Para a preparação dos dados visando o treinamento do modelo, adotamos uma estruturação cuidadosa para garantir a adequação ao processo e otimizar os resultados com os recursos disponíveis. Decidimos utilizar uma janela de contexto de treinamento de 512 *tokens*, que são as menores unidades de texto que o modelo pode compreender e processar. A escolha desse tamanho de janela foi influenciada pela capacidade de memória RAM da GPU, que é de 80GB. Com uma janela de contexto de 512 *tokens*, o treinamento consumia cerca de 78GB, permitindo assim a maximização do tamanho do contexto sem exceder os recursos disponíveis.

Em processamento de linguagem natural (NLP), tokenizar é o processo de dividir o texto em palavras, frases, símbolos ou outros elementos significativos, chamados *tokens*[14]. Os tokens podem ser palavras, partes de palavras ou até mesmo caracteres, dependendo de como o tokenizador foi construído. Uma decisão significativa em nosso estudo foi influenciada por uma limitação do modelo selecionado. Apesar de sua capacidade multilíngue, o modelo apresenta um desempenho superior ao processar instruções em inglês, conforme evidenciado pelas referências no *HuggingFace hub*[15].

Inspirados pelo modelo Dolly [11], optamos por estruturar um *prompt* de instrução em inglês. O *prompt* é uma instrução clara e concisa que orienta o modelo de linguagem a gerar uma resposta adequada com base nos dados de entrada fornecidos. Esta abordagem permite que o modelo compreenda melhor a tarefa e produza resultados mais precisos e relevantes. Ao utilizar instruções na língua de melhor desempenho, o modelo pode aproveitar ao máximo seu conhecimento pré-treinado e fornecer respostas de alta qualidade, mesmo em contextos multilíngues. A Figura 5 apresenta nossa solução de combinar indicações em inglês das estruturas básicas com a tarefa de *question-answering* (Contexto, pergunta e resposta). A estrutura básica é deri-

[13] `https://www.nvidia.com/pt-br/data-center/a100/`
[14] `https://brasileiraspln.com/livro-pln/1a-edicao/parte3/cap4/cap4.html`
[15] `https://huggingface.co/bigscience/bloomz-1b7`

vada do conjunto de dados Alpaca [16], um *dataset* de instruções para modelos de linguagem.

Figura 5: *Exemplo de prompt* para treinamento do modelo com a saída

```
Below is an instruction that describes a task. Write a response that
   appropriately completes the request.

### Instruction:

## Context
Lixa Para Madeira E Parede / A257 / G80 / Norton / 50 Pecas
CARACTERISTICAS:
Marca: Norton;
Seco / umido / Ambos: Seco;
Abrasivo: Metal;
Linha: A257;
Grano: G80;
Costado: Papel;
Costado leve;
Excelente acabamento final na pintura;

## Question
Boa tarde, qual a marca do produto?

### Response:
A marca do produto e Norton.
```

3.5 Métricas

Na análise dos resultados dos treinamentos, adotamos duas métricas fundamentais: ROUGE[18] e BLEU [17].

A ROUGE destaca-se como uma métrica crucial na avaliação de sumarizações e traduções, comparando o texto gerado automaticamente com um conjunto de textos de referência criados por humanos.

Por sua vez, a BLEU é uma métrica amplamente empregada na avaliação de traduções. Seus escores são inicialmente calculados para segmentos específicos do texto, comparando-os com os textos de referência. Posteriormente, uma média é obtida sobre o corpo completo do texto, resultando em um escore final que representa a qualidade global da tradução.

Ambas as métricas variam entre 0 e 1, e neste estudo, optamos por apresentar

esses valores em termos percentuais para facilitar a interpretação dos resultados.

4 Avaliação Experimental

A Subseção 4.1 apresenta o procedimento experimental e a Subseção 4.2 apresenta os resultados dos experimentos.

4.1 Procedimento experimental

Com a definição das tecnologias e os conjunto de dados, iniciamos o processo de experimentação, diferenciando os *datasets* para cada treinamento. Empregamos a técnica de *EarlyStopping*, monitorando a função de perda durante o treinamento do modelo usando o conjunto de dados de validação. Esta técnica interrompe o treinamento se a função de perda parar de melhorar após um número determinado de passos, neste caso, 100 passos. Isso é útil para evitar o *overfitting*, que ocorre quando o modelo se ajusta demais aos dados de treinamento e perde a capacidade de generalizar para novos dados.

Um fator importante para o treinamento foi a utilização da técnica de treinamento com precisão mista, mais especificamente usando bf16[16], que é um tipo de precisão que permite que o modelo seja treinado mais rapidamente e com menos uso de memória utilizando a GPU escolhida. A precisão mista combina o uso de tipos de dados de 32 bits e 16 bits para reduzir a demanda de memória durante o treinamento, permitindo que o modelo seja treinado com um tamanho de lote maior e, portanto, convergir mais rapidamente.

A seguir explicamos as configurações dos experimentos que conduzimos. Em cada um deles partimos do mesmo modelo base, *Bloomz 1.7B*, sem qualquer tipo de ajuste fino anterior.

- **Experimento #1**

  - Treinamento do modelo com *dataset* de extração de respostas;

  No primeiro experimento, partimos do modelo base e ajustamos as configurações para o tamanho do contexto e a estruturação dos dados. O objetivo foi treinar o modelo no maior tamanho de lote (*batch size*) possível, dadas as limitações da máquina virtual no estudo. Assim, escolhemos um tamanho de lote de 16, com uma taxa de aprendizado (*learning rate*) de 1e-5, e treinamos o modelo por duas épocas. A decisão sobre a taxa de aprendizado e o número de épocas foi baseada em experimentos anteriores e na literatura, especificamente a do modelo BERT[5], que sugerem que taxas de aprendizado muito altas podem

[16] `https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus`

levar a uma rápida convergência para soluções subótimas, enquanto taxas de aprendizado muito baixas podem fazer com que o treinamento demore muito para convergir ou fique preso. Portanto, optamos por uma taxa de aprendizado de 1e-5, que é considerada pequena, mas ainda capaz de permitir a convergência do modelo. Treinamos o modelo por duas épocas, um número relativamente baixo, mas suficiente para observar o comportamento do modelo e ajustar os hiper-parâmetros conforme necessário. Durante o treinamento, observamos que o modelo tende a *overfitting* após um certo número de épocas, indicando que a técnica de *EarlyStopping* contribui em prevenir o problema de memorização dos dados.

- **Experimento #2**

  - Treinamento do modelo com *dataset Dolly*-15k de perguntas e respostas

  No segundo experimento, o qual só contêm o *dataset* da *Databricks*, optamos por utilizar os mesmos hiper-parâmetros do Experimento #1, pois nele houve uma convergência rápida do modelo e uma estabilidade do treinamento, de tal maneira que a função de perda em combinação com o conjunto de validação permitiu a compreensão do fenômeno de *overfitting*. As mesmas técnicas empregadas no experimento anterior foram usadas nesse também.

- **Experimento #3**

  - Treinamento do modelo com a combinação dos *datasets* dos experimentos #1 e #2

  Neste experimento, optamos por explorar uma ampla gama de dados, abrangendo diversos idiomas e variados tipos de tarefas, com o objetivo de realizar um extenso treinamento por meio da variação de hiper-parâmetros. Realizamos quatro treinamentos neste conjunto de dados, todos apresentando a mesma situação de interrupção devido à degradação de desempenho, similar ao que ocorreu nos experimentos com conjuntos de dados não misturados. Na primeira fase do experimento, treinamos dois modelos com o mesmo número de épocas, mas variando a taxa de aprendizado em uma ordem de magnitude. Na segunda fase, aumentamos o número de épocas para três, e em um dos modelos, alteramos a taxa de aprendizado, aumentando-a cinco vezes.

Tabela 3 apresenta os detalhes dos experimentos conduzidos.

Tabela 3: Detalhes sobres os experimentos realizados

Num of Epochs	Batch Size	Learning Rate	Dataset
2	16	1e-5	QA extrativo
2	16	1e-5	Dolly 15k
2	16	1e-6	Mix
2	16	1e-5	Mix
3	16	1e-5	Mix
3	16	5e-5	Mix

4.2 Resultados

4.2.1 Experimento #1: Treinamento com conjunto de dados de QA Extrativo

A Tabela 4 e a Tabela 5 apresentam os resultados do treinamento com um conjunto de dados de QA Extrativo. Cada linha da tabela representa um conjunto de dados diferente usado para a etapa de teste. O conjunto de dados "Dolly"apresentou os menores valores em todas as métricas. Esses valores estão significativamente abaixo da média aritmética das métricas de avaliação. Isso indica um desempenho inferior do conjunto de dados "Dolly"em comparação com os outros conjuntos de dados. Em contraste, o conjunto de dados "Multilingue + Dolly"apresentou os maiores valores para as métricas escolhidas. Esses valores estão acima da média aritmética das métricas de avaliação, sugerindo um desempenho superior nessas áreas. No entanto, o valor de BLEU para "Multilingue + Dolly"é 0.2328, que é significativamente menor do que a média de BLEU (0.3486), e menor do que os outros conjuntos de dados, exceto "Dolly".

Tabela 4: Desempenho do modelo pós-treinamento com esse conjunto de dados extrativos

Dataset	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	BLEU
Dolly	0.1882	0.0882	0.1840	0.1845	0.2134
Português	0.5657	0.3439	0.5656	0.5645	0.4252
Espanhol	0.5787	0.3919	0.5769	0.5773	0.4633
Extrativo Multilíngue	0.5457	0.3657	0.5429	0.5417	0.4087
Multilingue + Dolly	0.5879	0.4215	0.5874	0.5862	0.2328

Os conjuntos de dados "Português", "Espanhol"e "Extrativo Multilíngue"apresentaram desempenhos semelhantes, com "Espanhol"tendo ligeiramente melhores resultados nas métricas ROUGE e BLEU. Em geral, esses resultados sugerem que o modelo testado com o conjunto de dados "Multilingue + Dolly"teve o melhor desempenho em termos de métricas ROUGE, mas não em termos de métrica BLEU. Isso pode indicar que o modelo é bom em recuperar informações relevantes (como indicado pe-

las métricas ROUGE), mas pode não ser tão bom em produzir respostas fluentes e coerentes (como indicado pela métrica BLEU).

Tabela 5: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 4

Métrica	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.4932	0.3222	0.4913	0.4908	0.3486

4.2.2 Experimento #2: Treinamento com *dataset Dolly* 15k

A Tabela 6 exibe os resultados de teste com o modelo treinado exclusivamente com o *dataset Dolly 15k*. O conjunto de dados "Dolly"obteve os menores valores para todas as métricas, tal qual aconteceu com o Experimento #1, demonstrando que apesar da maior semelhança dos dados de treinamento a parte de teste, ainda sim eles não foram suficientes para atingir resultados expressivos. Estes resultados são substancialmente inferiores à média aritmética das métricas de avaliação conforme apresentado na tabela 7. Os conjuntos de dados "Português"e "Espanhol"mostraram desempenhos comparáveis e robustos, com valores de ROUGE-1, ROUGE-2, ROUGE-L e ROUGE-Lsum todos acima de 0.5 e valores de BLEU acima de 0.3, indicando que o modelo treinado com esses conjuntos de dados foi capaz de gerar respostas que estão mais alinhadas com as referências humanas. O conjunto de dados "Extrativo Multilingue"apresentou valores ligeiramente menores que "Português"e "Espanhol", mas ainda assim superiores ao "Dolly".

O conjunto de dados "Multilingue + Dolly"destacou-se com o maior valor para a métrica ROUGE-2, com 0.4014, sugerindo que o modelo foi particularmente eficiente em capturar bigramas neste conjunto de dados. No entanto, o valor de BLEU para "Multilingue + Dolly"foi de apenas 0.2271, significativamente abaixo da média, o que pode indicar que, embora o modelo seja capaz de identificar palavras e frases relevantes, ele pode não estar gerando respostas corretas quanto os outros conjuntos de dados. Em resumo, o teste com os conjuntos de dados "Português", "Espanhol"e "Extrativo Multilingue"resultou em boas métricas finais, com o "Multilingue + Dolly"mostrando força particular na captura de bigramas, mas com limitações na produção de respostas com boa qualidade, conforme evidenciado pelos valores de BLEU.

Tabela 6: Desempenho do modelo pós-treinamento com esse conjunto de dados

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Dolly	0.1719	0.0883	0.1660	0.1664	0.1142
Português	0.5651	0.3480	0.5646	0.5646	0.4562
Espanhol	0.5580	0.3570	0.5554	0.5562	0.3893
Extrativo Multilingue	0.5330	0.3463	0.5301	0.5315	0.3994
Multilingue + Dolly	0.5756	0.4014	0.5725	0.5710	0.2271

Tabela 7: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 6

Métrica	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.4807	0.3082	0.4777	0.4779	0.3172

4.2.3 Experimento #3: Treinamento com junção dos *datasets*

O Experimento #3 analisou a efetividade de modelos treinados com a combinação dos *datasets* extrativo e *Dolly*, que foram previamente testados de forma independente nos Experimentos #1 e #2, os resultados podem ser vistos nas tabelas de . A análise dos resultados revela padrões distintos de desempenho em diferentes configurações de treinamento. Na configuração inicial com duas épocas, *batch size* de 16 e *learning rate* de 1e-05, o *dataset* Espanhol se sobressaiu, alcançando os maiores valores em todas as métricas, sugerindo um desempenho superior.

O *dataset* Extrativo Multilíngue mostrou resultados sólidos, enquanto o Dolly apresentou o desempenho mais fraco, indicando possíveis desafios na generalização a partir desse *dataset* específico. Quando o *learning rate* foi reduzido para 1e-6, houve uma queda no desempenho geral, o que pode sugerir que essa taxa de aprendizado mais baixa não foi suficiente para otimizar o modelo de forma eficaz. Por outro lado, ao aumentar o número de épocas para três com um *learning rate* de 1e-05, observou-se uma melhoria nos resultados, especialmente para os *datasets* Espanhol e Extrativo Multilíngue, indicando que um maior número de épocas pode ser benéfico para o aprendizado. No entanto, um aumento no *learning rate* para 5e-5 resultou em uma diminuição significativa do desempenho, o que pode apontar para uma otimização excessiva ou um ajuste inadequado do modelo. Isso sugere que um *learning rate* intermediário de 1e-05 e um número de épocas entre dois e três podem oferecer um equilíbrio mais eficaz para o treinamento.

Em uma comparação de alto nível, o *dataset* Espanhol consistentemente apresentou os melhores resultados, como pode ser visto nas tabelas 8, 10, 12 e 14, o que pode refletir a eficácia do modelo para esse idioma ou a qualidade dos dados. Em contraste, o *dataset Dolly* consistentemente mostrou o menor desempenho nessas tabelas citadas, levantando questões sobre a adequação desse conjunto para o treinamento ou sobre a complexidade das tarefas que ele contém. Os outros *datasets*, relacionados à extração e à combinação, tendem a apresentar resultados próximos em todas as métricas, o que é coerente com a natureza compartilhada desses dados, e o modelo multilíngue parece ter aprendido de forma semelhante, independentemente do idioma predominante no *dataset*. Confirmando essa ideia, as melhores métricas médias para *Rouge-1*, *Rouge-L* e *Rouge-L sum* foram observadas nas tabelas 9. A melhor métrica para *Rouge-2* foi encontrada na tabela 11. Além disso, um valor de BLEU próximo ao melhor resultado foi observado no experimento da tabela 9, o qual pode ser comparado ao resultado visto na tabela 5. Esses achados demonstram que a combinação

de conjuntos de dados possui um grande potencial para melhorar a capacidade desses modelos decodificadores.

Essas observações destacam a importância de um ajuste cuidadoso dos hiperparâmetros durante o treinamento de modelos de aprendizado de máquina, considerando as características específicas de cada *dataset* e as métricas de avaliação relevantes.

Tabela 8: Performance do modelo com 2 épocas, *batch size* 16 e *learning rate* 1e-05

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Dolly	0.4471	0.3198	0.4166	0.4169	0.1407
Espanhol	0.6620	0.4657	0.6604	0.6612	0.4885
Português	0.5869	0.3703	0.5875	0.5857	0.2734
Extrativo Multilingue	0.6212	0.4249	0.6207	0.6203	0.4294
Multilingue + Dolly	0.5841	0.4153	0.5794	0.5783	0.3206

Tabela 9: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 8

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.58026	0.3992	0.57292	0.57248	0.33052

Tabela 10: Desempenho do modelo treinado por 2 épocas, com *batch size* 16 e *learning rate* 1e-6

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Dolly	0.2571	0.1358	0.2231	0.2243	0.0862
Espanhol	0.4295	0.2930	0.4266	0.4265	0.0678
Português	0.4026	0.2489	0.3983	0.3995	0.0371
Extrativo Multilingue	0.4549	0.3126	0.4520	0.4522	0.0763
Multilingue + Dolly	0.4360	0.3013	0.4288	0.4292	0.0944

Tabela 11: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 10

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.3960	0.2581	0.3862	0.3867	0.0720

Tabela 12: Desempenho do modelo treinado por 3 épocas com *batch size* 16 e *learning rate* 1e-5

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Dolly	0.4444	0.3209	0.4131	0.4137	0.1730
Espanhol	0.6427	0.4415	0.6395	0.6405	0.4088
Português	0.5540	0.3447	0.5520	0.5519	0.1487
Extrativo Multilingue	0.6125	0.4224	0.6118	0.6107	0.3729
Multilingue + Dolly	0.5647	0.3945	0.5593	0.5587	0.2586

Tabela 13: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 12

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.5643	0.3858	0.5613	0.5611	0.2722

Tabela 14: Performance do modelo treinado por 3 épocas com *batch size* 16 e *learning rate* 5e-5

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Dolly	0.3269	0.2209	0.2999	0.3009	0.1181
Espanhol	0.2369	0.1376	0.2360	0.2365	0.1261
Português	0.1865	0.1128	0.1858	0.1865	0.041
Extrativo Multilingue	0.2294	0.1396	0.2264	0.2266	0.1203
Multilingue + Dolly	0.2267	0.1385	0.2225	0.2235	0.1154

Tabela 15: Média aritmética das métricas de avaliação utilizadas no Experimento com modelo da tabela 14

Dataset	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**	**BLEU**
Média	0.2411	0.1497	0.2341	0.2346	0.1042

5 Discussão

Em geral, os resultados obtidos corroboram a hipótese de que a fusão de diferentes conjuntos de dados pode otimizar a efetividade do modelo. Esta conclusão foi respaldada pelos experimentos realizados, que, em média, apresentaram melhores resultados ao utilizar a combinação de conjuntos de dados. Isso sugere que a integração de exemplos de diversas sub-tarefas pode ser mais eficaz do que treiná-los individualmente. A variação dos hiper-parâmetros se mostrou crucial no processo de treinamento, apontando para possíveis melhorias em futuros experimentos.

As métricas Rouge e BLEU, que avaliam a qualidade das respostas geradas pelo modelo, indicaram que ainda há margem para aprimoramento. A métrica Rouge, que avalia a sobreposição de *n-gramas* entre a saída do sistema e a referência, revelou que mais da metade das respostas do modelo estão alinhadas com as respostas de referência. No entanto, a métrica BLEU, que também avalia a sobreposição de *n-gramas*, mas prioriza a precisão, sugere que o modelo pode estar enfrentando dificuldades em gerar respostas precisas e gramaticalmente corretas.

A combinação de conjuntos de dados é essencial para entender a variação dessas métricas, pois permite uma análise mais detalhada da performance do modelo em diferentes contextos. A diversidade dos conjuntos de dados utilizados proporciona uma visão mais completa da capacidade do modelo de lidar com variadas tarefas e tipos de linguagem. A análise das métricas revelou que a escolha do conjunto de dados tem um impacto direto na eficácia do modelo. Conjuntos de dados com maior diversidade de tarefas tendem a resultar em modelos com melhor desempenho em termos de generalização, pois são expostos a uma gama mais ampla de exemplos durante o treinamento. Isso sugere que a seleção cuidadosa dos conjuntos de dados para treinamento pode ser uma estratégia eficaz para melhorar a performance do modelo.

Apesar dos avanços significativos alcançados com a combinação de conjuntos de dados, ainda existem desafios a serem superados. Um desses desafios é a necessidade de melhorar a precisão das respostas geradas pelo modelo. Embora o modelo esteja gerando respostas relevantes, a precisão dessas respostas ainda precisa ser aprimorada.

Nossa análise sobre as métricas também apontou para a necessidade de desenvolver métodos mais robustos para avaliar a qualidade das respostas geradas pelo modelo. As métricas atuais, como Rouge e BLEU, embora úteis, têm suas limitações e podem não capturar completamente a qualidade das respostas do modelo. Portanto, a proposta e o desenvolvimento de novas métricas ou a combinação de múltiplas métricas pode

ser uma área promissora para futuras pesquisas. Consideramos a necessidade de desenvolver métodos mais robustos para avaliar a qualidade das respostas geradas pelo modelo. As métricas atuais, como Rouge e BLEU, embora úteis, têm suas limitações e podem não capturar completamente a qualidade das respostas do modelo. Portanto, o desenvolvimento de novas métricas ou a combinação de múltiplas métricas pode ser uma área promissora para futuras pesquisas.

Em futuros trabalhos relacionados ao treinamento de modelos, nosso objetivo é realizar uma maior quantidade de experimentos, explorando a variação de modelos e de hiper-parâmetros. Conseguimos coletar um volume significativo de exemplos para treinamento e validação, no entanto, a limitação na variação desses elementos restringiu a abrangência do nosso experimento. A principal barreira encontrada foi o custo de treinamento, conforme mencionado na subseção sobre tecnologia. Embora tivéssemos acesso a uma máquina de alta capacidade, o tempo disponível para utilizá-la foi limitado. Com o aprendizado adquirido sobre a eficácia da combinação de conjuntos de dados de diferentes tipos, pretendemos iniciar os próximos experimentos já com essa abordagem, permitindo um foco mais direcionado. Isso pode potencializar os resultados, uma vez que diferentes tipos de dados podem complementar-se mutuamente, proporcionando uma visão mais completa e, consequentemente, melhorando a performance do modelo.

Em suma, a hipótese geral de que o treinamento de modelos com a combinação de dados de diferentes sub-tarefas dentro de *question-answering* indicou que há possíveis ganhos nesse caminho, mas são necessários mais experimentos e técnicas melhores de entendimento de quão boas são as respostas geradas às perguntas. Uma possibilidade seria a implementação desses modelos em um ambiente de produção, onde os usuários poderiam fazer perguntas e receber respostas, fornecendo assim um *feedback* valioso para a validação geral do sistema. Outro ponto relevante é que, apesar de o modelo ser multilíngue, seus resultados não foram tão satisfatórios quanto esperávamos inicialmente, especialmente em relação à validação. Isso destaca a importância de entender as limitações do experimento proposto e de continuar aprimorando o modelo e as técnicas de treinamento.

Em conclusão, a combinação de diferentes conjuntos de dados para treinamento de modelos de *question-answering* parece ser um caminho promissor para melhorar a performance do modelo. No entanto, são necessários mais experimentos e uma melhor compreensão das respostas geradas para validar completamente essa abordagem. Além disso, apesar de o modelo ser multilíngue, ainda há espaço para melhorias.

6 Conclusão

O refinamento de modelos para a tarefa de questão e resposta é um desafio de pesquisa em aberto. Este trabalho objetivou estudar a combinação de diferentes conjunto de

dados para o treinamento de modelos de *question-answering*. Encontramos que a combinação dos conjunto de dados é um caminho promissor para melhorar a efetividade de resposta dos modelos. Este estudo alcançou os objetivos propostos em termos de treinamento, levando em consideração a organização dos conjuntos de dados e a quantidade de exemplos utilizados. Para a próxima etapa do estudo, planejamos dedicar mais tempo à elaboração e realização de testes, o que poderia contribuir para a obtenção de resultados mais satisfatórios. Planejamos ainda aprimorar a seleção de hiper-parâmetros e explorar diferentes arquiteturas de modelos também pode ser útil para melhorar a performance. Por fim, a análise de possíveis problemas nos dados, que na sua maioria são extrativos podem ter influenciado na performance final desses modelos. O desempenho final do modelo, mesmo no experimento que apresentou as melhores métricas, não foi excepcional.

Agradecimentos

Este trabalho foi conduzido em colaboração com o projeto "Construção e uso de bases de conhecimento em sistemas automatizados de questão e respostas" – convênio 93454 – entre Unicamp e a GoBots Soluções Inteligentes LTDA. Agradecemos toda a colaboração dos envolvidos na empresa no contexto desse projeto e fornecimento dos créditos para o treinamento dos modelos utilizados. Este trabalho teve igualmente apoio do aluno de Mestrado do IC/UNICAMP, Rodrigo Oliveira Caus.

Referências

[1] Ecoffet, A., & Others. (2023). GPT-4 Technical Report. arXiv:2303.08774

[2] Philipp Cimiano; Christina Unger; John McCrae (1 March 2014). Ontology-Based Interpretation of Natural Language. Morgan & Claypool Publishers. ISBN 978-1-60845-990-2.

[3] 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

[4] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:1606.05250

[5] Devlin, J., Chang, M-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805

[6] Sayama, Helio & Araujo, Anderson & Fernandes, Eraldo. (2019). FaQuAD: Reading Comprehension Dataset in the Domain of Brazilian Higher Education. 443-448. 10.1109/BRACIS.2019.00084.

[7] Pires, R., Abonizio, H., Rogério, T., & Nogueira, R. (2023). Sabiá: Portuguese Large Language Models. arXiv preprint arXiv:2304.07880.

[8] Izacard, G., Le, Q. V., Raffel, C., Shazeer, N., Vaswani, A., Parmar, N., Uszkoreit, J., Kaiser, L., Noam, S., Shlens, J., Jones, L., Gomez, A. N., & Kaiser, L. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971

[9] Rodrigo Caus, Arthur Baia, Victor Ávila, & Victor Hochgreb. (2022). E-Commerce Question Answering Dataset (2.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7314395

[10] DeepLearningBrasil.Squad v1.1 automatically translated to portuguese and reviewed, http://tinyurl.com/2rfdwtsx2021. Last accessed:2023-07-14

[11] Databricks. (2023). Databricks Dolly 15K Dataset. Retrieved from https://www.kaggle.com/datasets/snehilsanyal/databricks-dolly-15k-dataset

[12] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Armentano-Oller, C., Rodriguez-Penagos, C., Gonzalez-Agirre, A., & Villegas, M. (2022). MarIA: Spanish Language Models. Procesamiento del Lenguaje Natural, 68(0), 39-60.

[13] Lewis, P., Ouguz, B., Rinott, R., Riedel, S., & Schwenk, H. (2019). MLQA: Evaluating Cross-lingual Extractive Question Answering. arXiv preprint arXiv:1910.07475

[14] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Maria: Spanish language models. arXiv preprint arXiv:2107.07253.

[15] Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. L., Bari, M. S., Shen, S., Yong, Z., Schoelkopf, H., ... & Raffel, C. (2023). Crosslingual Generalization through Multitask Finetuning. arXiv preprint arXiv:2211.01786

[16] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023). Stanford Alpaca: An Instruction-following LLaMA model. GitHub repository. Retrieved from https://github.com/tatsu-lab/stanford_alpaca

[17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

[18] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.