CENTRO UNIVERSITÁRIO FEI

LUCAS MATEUS DE MORAES

APLICAÇÃO DE TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA AVALIAÇÃO AUTOMÁTICA DE QUESTÕES DISSERTATIVAS

São Bernardo do Campo

2024

LUCAS MATEUS DE MORAES

APLICAÇÃO DE TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA AVALIAÇÃO AUTOMÁTICA DE QUESTÕES DISSERTATIVAS

Trabalho de conclusão de curso, apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Ciência da Computação. Orientado pelo Professor Dr. Charles Henrique Porto Ferreira.

São Bernardo do Campo

2024

Dedico este trabalho especialmente aos Professores Dr. Guilherme Wachs e Dr. Charles Ferreira. Ao primeiro, pela inspiração que me despertou um interesse genuíno na ciência da computação, e ao segundo, pela orientação para realizar este trabalho. Aprendi com ambos como problemas complexos podem ser resolvidos de forma elucidativa. Por isso, nutro profunda gratidão e respeito por eles e pelos demais docentes da FEI.

AGRADECIMENTOS

Neste momento de conclusão da graduação, agradeço a Deus pela oportunidade de ter estudado em uma instituição de ensino de grande excelência como a FEI. Dentro dessa instituição, gostaria de agradecer com profunda gratidão os professores da FEI, especialmente aos professores do Departamento de Ciência da Computação, por todo o ensino durante essa formação acadêmica. Por último, mas não menos importante, gostaria de agradecer especialmente ao professor Dr. Charles Ferreira pela paciência, auxílio e disponibilidade que ofereceu ao orientar este trabalho. Sem o seu suporte, não seria possível que este projeto fosse concluído com os resultados finais que obtivemos. Seu exemplo como docente e acadêmico com certeza será um modelo que terei em mente em minha jornada daqui para frente. Guardo com zelo todo o aprendizado dos últimos anos e realmente vejo o valor enriquecedor para a compreensão de questões técnicas e científicas. Além desse aprendizado, os exemplos dos professores são uma inspiração edificante como guias para a jornada de um aluno, que também guardarei com afinco.

"As raízes do estudo são amargas, mas seus frutos são doces."

Aristóteles

"Ciência da Computação está tão relacionada aos computadores quanto a Astronomia aos telescópios, Biologia aos microscópios, ou Química aos tubos de ensaio. A Ciência não estuda ferramentas. Ela estuda como nós as utilizamos, e o que descobrimos com elas."

Edsger Dijkstra

RESUMO

No contexto educacional, a correção de avaliações, principalmente as dissertativas, demanda um tempo considerável dos docentes. Este trabalho propõe um algoritmo para gerar avaliações automáticas de respostas a questões dissertativas, utilizando uma média ponderada de diferentes fatores extraídos do texto através de técnicas de Processamento de Linguagem Natural (PLN). Cada fator recebe um peso específico, determinado por regressão linear treinada com bases em diferentes idiomas, resultando em pesos distintos para cada versão do algoritmo. Os resultados demonstraram acurácia de 63,43% para a versão em português (*dataset* de questões de Biologia), 83,58% para a versão em espanhol (*dataset* de questões de Literatura) e 81,63% para a versão em inglês (*dataset* de questões de Ciência da Computação). De forma geral, o algoritmo apresentou desempenho regular, gerando avaliações próximas às realizadas por docentes na maior parte dos casos. Os resultados indicam que, com aprofundamento, essa abordagem pode ser promissora para soluções relacionadas à problemática abordada neste trabalho.

Palavras-chave: Similaridade Semântica; Distância de Levenshtein; Frequência de Palavras; Processamento de Linguagem Natural; Avaliação de Respostas Dissertativas; Regressão Linear

ABSTRACT

In the educational context, grading assessments, especially essay questions, is a time-consuming task for teachers. This work proposes an algorithm to generate automatic evaluations of essay question answers using a weighted average that evaluates different factors extracted from the text using Natural Language Processing (NLP) techniques. Each factor receives a specific weight, determined by linear regression trained with bases in different languages, resulting in distinct weights for each version of the algorithm. The results showed an accuracy of 63.43% for the version trained with a Portuguese dataset, whose question theme was Biology; 83.58% for the version trained with Spanish data, with questions about Literature; and 81.63% for the version trained with English data, with questions about Computer Science. Thus, in general, the algorithm demonstrated the ability to generate evaluations with values close to those made by teachers, showing regular performance in most cases. These results indicate that, with further deepening, this approach can be a promising direction for solutions involving the problem addressed in this work.

Keywords: Semantic Similarity; Levenshtein Distance; Word Frequency; Natural Language Processing; Evaluation of Descriptive Answers; Linear Regression

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Na atualidade das instituições de ensino, uma variedade de métodos de avaliação são empregados para mensurar o aprendizado dos alunos, destacando-se entre eles as questões dissertativas, nas quais os alunos devem fornecer suas respostas de maneira textual. Esse método proporciona ao professor uma compreensão mais aprofundada da linha de raciocínio do aluno durante a correção, possibilitando assim, uma avaliação mais precisa do nível de aprendizado alcançado e um melhor acompanhamento da evolução do aluno ao longo do tempo (OLIVEIRA; SANTOS, 2005).

Embora a precisão da avaliação seja uma vantagem desse método, o mesmo exige maior tempo de leitura, análise e compreensão de cada questão por parte do professor, uma vez que ele precisará analisar integralmente o conteúdo dos textos fornecidos como resposta pelos alunos.

Tal dificuldade pode levar o docente a preferir questões objetivas, que, por possuírem um gabarito, demandam um tempo menor de correção. Entretanto, é importante ressaltar que as questões objetivas não atingem o mesmo nível de precisão na avaliação em comparação com as questões dissertativas, uma vez que possibilitam que o aluno escolha uma alternativa de maneira aleatória e possa obter a resposta correta mesmo sem ter nenhum conhecimento da mesma, ao contrário das questões dissertativas, em que essa possibilidade não existe.

A utilização de avaliações dissertativas é utilizado em apenas 30% das formas de avaliação aplicadas aos alunos, conforme demonstrado em um estudo realizado por Oliveira e Santos (2005). Este estudo aborda as vantagens das avaliações dissertativas e sua maior adequação para a avaliação do desempenho e acompanhamento do progresso dos alunos no processo de aprendizado. Considerando esse contexto, seria benéfico para as instituições de ensino adotar mais frequentemente avaliações dissertativas (OLIVEIRA; SANTOS, 2005). No entanto, como mencionado anteriormente, um dos principais impeditivos da adoção desse método seria o aumento na carga de trabalho dos professores responsáveis pelas correções.

Por conta disso, o presente trabalho propõe uma abordagem para realizar avaliação automática das respostas dissertativas dos alunos, comparando-as com uma resposta padrão fornecida por um especialista da área como modelo do conhecimento esperado para aquela questão.

De forma geral, o algoritmo desenvolvido neste trabalho avalia o "grau de similaridade"entre as respostas das questões levando em consideração propriedades da parte léxica, da parte de sintaxe e da parte semântica dos textos. No fim, o algoritmo utiliza uma métrica de

média ponderada para mensurar a similaridade entre as respostas e as referências de professores ou especialistas e gerar uma avaliação automática das respostas fornecidas para determinadas questões. Para que essa avaliação fosse efetivamente realizada, alguns fatores eram extraídos do texto, os quais podemos considerar como elementos relevantes na composição do texto de uma resposta. Esses fatores foram:

a) Similaridade semântica
b) Frequência de termos
c) Distância de Levenshtein

1.1 OBJETIVO

O objetivo final do trabalho foi desenvolver e implementar o algoritmo para uma abordagem automatizada de avaliação para respostas dissertativas. A proposta teve como alvo simplificar o processo de correção manual dessas respostas, promovendo uma análise dos fatores extraídos do texto e sua relação com as avaliações geradas. As metas planejadas no trabalho foram especificadas nos seguintes tópicos:

a) Desenvolver um algoritmo para mensurar a similaridade semântica entre respostas dissertativas e uma resposta padrão de um professor como referência.
b) Considerar os fatores de similaridade semântica, frequência de termos e distância de Levenshtein.
c) Validar a eficácia do algoritmo comparando-o com dados de avaliações já corrigidas.
d) Como última meta, planejada para o caso das anteriores serem alcançadas, foi feito um protótipo para testes práticos do algoritmo visando elucidar os conceitos abordados no trabalho.

1.2 QUESTÕES DE PESQUISA

O tema abordado levanta importantes questões de pesquisa, nas quais o presente trabalho buscou responder questões tais como:

a) Quais parâmetros podemos extrair como fatores do texto que devem ser considerados como componentes relevantes para pontuar a similaridade semântica entre as respostas e as referências?
b) Esses parâmetros podem definir uma pontuação que funcione de maneira geral quando aplicada a casos práticos de correções dissertativas?

c) Como os fatores devem ser ponderados dentro de uma métrica para as avaliações?

d) Como a acurácia da métrica para as avaliações varia conforme os eventuais pesos, dados para treino e dados para teste dos fatores também variam?

2 TRABALHOS RELACIONADOS

Para busca de trabalhos relacionados foram utilizados as ferramentas de pesquisas para artigos científicos do *Google Scholar* (https://scholar.google.com/), *IEEE Xplore* (https://ieeexplore.ieee.org/Xplore/home.jsp), *ScienceDirect* (https://www.sciencedirect.com/) e *Semantic Scholar* (https://www.semanticscholar.org/).

Como palavras-chaves na busca foram utilizados termos em inglês, sendo eles, *"semantic similarity between texts"*, *"measure degree of paraphrase"*, *"paraphrase detection"*, *"natural language processing"*, *"measure semantic similarity between answers"* e *"evaluation of descriptve answers"*. Os termos que trouxeram os melhores resultados e estavam presentes nos melhores artigos selecionados foram *"semantic similarity"*, *"natural language processing"* e *"evaluation of descriptve answers"*.

Inicialmente foram selecionados 26 artigos que poderiam ser relevantes para o presente trabalho com base nos temas, a sumarização dos artigos pode ser vista na Tabela 1 em que os artigos selecionados no final estão destacados na cor cinza.

Título	Fonte	Referência
Determining Degree of Relevance of Reviews Using a Graph-Based Text Representation	IEEE	(RAMACHANDRAN; GEHRINGER, 2011)
A Chinese text paraphrase detection method based on dependency tree	IEEE	(JIANG; HAO; ZHU, 2016)
Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity	IEEE	(GUSTAFSON; PERA; NG, 2008)
Enhanced Text Matching Based on Semantic Transformation	IEEE	(ZHANG et al., 2020)
Semantic similarity based assessment of descriptive type answers	IEEE	(MEENA; LAWRANCE, 2016)
A comparative analysis of various approaches for automated assessment of descriptive answers	IEEE	(KAUR; SASIKUMAR, 2017)
A reliable approach to automatic assessment of short answer free responses	SSL	(BACHMAN et al., 2002)
A Descriptive Answer Evaluation System Using Cosine Similarity Technique	IEEE	(THALOR, 2021)
An Intelligent System for Evaluation of Descriptive Answers	IEEE	(BAGARIA et al., 2020)
Application Research of Similarity Algorithm in the Design of English Intelligent Question Answering System	IEEE	(ZHANG, 2022)
Near duplicate text detection using graph depiction	IEEE	(POULOS, 2016)
Recognition of Parallelism Sentence Based on Recurrent Neural Network	IEEE	(DAI et al., 2018)
LSGC: An Interactive Text Matching Model Combined with Enhanced Encoding	IEEE	(WANG et al., 2022)
A software system for determining the semantic similarity of short texts in Serbian	IEEE	(BATANOVIĆ; FURLAN; NIKOLIĆ, 2011)
Arabic Semantic Textual Similarity Identification based on Convolutional Gated Recurrent Units	IEEE	(MAHMOUD; ZRIGUI, 2021)
A Chinese text paraphrase detection method based on dependency tree	IEEE	(JIANG; HAO; ZHU, 2016)
Using paraphrases to improve tweet classification: Comparing WordNet and word embedding approaches	IEEE	(LI et al., 2016)
SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection	SSL	(SCHLECHTWEG et al., 2020)
SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation	SSL	(CER et al., 2017)
Use of Syntactic Similarity Based Similarity Matrix for Evaluating Descriptive Answer	IEEE	(PAUL; PAWAR, 2014)
Chapter 16 - Semantic similarity–based descriptive answer evaluation	SCD	(SHAUKAT et al., 2021)
A Study of Automated Evaluation of Student's Examination Paper using Machine Learning Techniques	IEEE	(SANUVALA; FATIMA, 2021)
Online Examination with short text matching	IEEE	(KUDI et al., 2014)
Towards Automated Evaluation of Handwritten Assessments	IEEE	(ROWTULA; OOTA; C.V, 2019)
Automatic Short Answer Grading (ASAG) using Attention-Based Deep Learning MODEL	IEEE	(AMUR; HOOI; SOOMRO, 2022)
Semantic similarity–based descriptive answer evaluation	SSL	(SHAUKAT et al., 2021)

Tabela 1 – Tabela de revisão bibliográfica sumarizada.

Após uma filtragem com base nos resumos e palavras-chaves, tendo como critério, a semelhança dos termos e a similaridade de outros artigos em comparação com a proposta do presente trabalho, foram selecionados quatro artigos como base para uma visão geral do estado da arte, seu direcionamento para o tema específico tratato no presente trabalho e uma revisão bibliográfica. Os artigos foram então destacados, como consta na Tabela 2, para leitura integral de seu conteúdo.

Base	Total encontrados	Após remoção dos Duplicados	Após análise do resumo
IEEE	21	21	2
Science Direct	1	1	1
Semantic Scholar	4	3	1

Tabela 2 – Tabela do funil de leitura.

O artigo proposto por Schlechtweg et al. (2020) faz o uso de *embeddings* de tipo (*type embeddings*) e *embeddings* contextualizados (*token embeddings*) para representar as palavras. Primeiro, é introduzida a distribuição de frequência de termos e uma detecção de mudança binária é definida em termos de limiares de frequência. Após isso, a distância de Jensen-Shannon entre as distribuições normalizadas de frequência é utilizada para medir a mudança efetuada.

No artigo proposto por Paul e Pawar (2014), podemos destacar que o trabalho utiliza a técnica de Análise Semântica Latente (LSA), que é comumente usada para determinar a similaridade de documentos, mas ressalta suas limitações em documentos curtos. O artigo destaca a ausência de abordagens anteriores que se concentrem na avaliação automática de respostas descritivas usando vetores de ordem de palavras. O método proposto utiliza uma matriz de similaridade entre vetores de ordem de palavras para avaliar respostas descritivas. A similaridade entre os vetores é calculada por meio de uma métrica de similaridade sintática, ou seja, baseada na ordem das palavras. Os resultados indicam que abordagens baseadas em ordem de palavras e frequência de termos são promissoras para a avaliação automática de respostas descritivas. A matriz de similaridade é apresentada como uma ferramenta eficaz para computar as notas de cada pergunta.

Na proposta do artigo escrito pelos autores Shaukat et al. (2021), pode-se destacar que a pesquisa faz uso de Processamento de Linguagem Natural (NLP) para automatizar o processo de avaliação, especialmente a similaridade de cosseno e índices de similaridade, são empregados para atribuir notas às respostas.

Os autores Sanuvala e Fatima (2021) incorporam uma abordagem que emprega ferramentas de Reconhecimento Óptico de Caracteres (OCR) para extrair texto de respostas manuscritas digitalizadas. A ênfase principal, no entanto, recai sobre o emprego de técnicas avançadas de processamento de linguagem natural para aprimorar a avaliação. O estudo destaca a importância de etapas como a tokenização, remoção de stop words e verificação de sinônimos e antônimos no pré-processamento das respostas. Além disso, aborda a criação de modelos semânticos e o cálculo de similaridade sem mencionar explicitamente as métricas de Machine Learning utilizadas.

No contexto da revisão bibliográfica feita com os trabalhos relacionados, alguns conceitos importantes foram retirados dos artigos selecionados, para contribuir com o presente trabalho. Dentre esses, destacam-se a análise da similaridade semântica com distância de cossenos, a consideração dos vetores de frequência de termos e a consideração da importância de fatores da sintaxe do texto, como a ordem das palavras.

A análise de similaridade semântica com distância de cossenos é uma técnica importante, conforme evidenciado nos artigos revisados (SHAUKAT et al., 2021). Essa abordagem é frequentemente empregada para medir a proximidade semântica entre textos.

O uso da frequência de termos em um dos artigos indica a importância específica da distribuição da frequência das palavras no texto. Esse conceito pode contribuir para o trabalho, sendo relevante para fatores de aspectos léxicos e fatores de aspecto da sintaxe do texto.

A utilização de vetores de ordem de palavras, como abordado em um dos artigos, resalta a importância da ordem das palavras na avaliação de respostas descritivas. Essa técnica pode superar limitações associadas à Análise Semântica Latente (LSA) em documentos curtos, oferecendo uma abordagem promissora para a avaliação automática.

Em síntese, a revisão bibliográfica gerou a necessidade de explicitar conceitos fundamentais à análise de similaridade semântica com distância de cossenos, a distribuição de frequência de palavras e da exploração de vetores de ordem de palavras ou um fator que avalia a parte léxica do texto como pontos relevantes nos estudos revisados.

3 CONCEITOS

Nesta seção serão apresentados alguns conceitos fundamentais para o entendimento da porposta desse projeto. Serão abordados conceitos de processamento de lingugagem natural, formas de representação de texto em formato númerico e metricas para comparação de textos e avalialção de desempenho.

3.1 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural (PLN) refere-se à aplicação de técnicas computacionais para a interpretação e manipulação de linguagem humana. Envolve o desenvolvimento de algoritmos e modelos que capacitam computadores a compreender, analisar e gerar texto de maneira semelhante ao entendimento humano.

3.2 REPRESENTAÇÃO DE TEXTOS

A Representação de Textos é crucial para permitir que algoritmos compreendam palavras e documentos. Duas técnicas comuns são TFIDF (Term Frequency-Inverse Document Frequency) e Word Embedding. O TFIDF avalia a importância de uma palavra em um documento, enquanto o Word Embedding mapeia palavras em vetores contínuos, capturando relações semânticas.

3.2.1 Word embedding

O Word Embedding é uma técnica que mapeia palavras em vetores de números reais, capturando relações semânticas e contextuais. Essa representação densa permite que algoritmos de processamento de linguagem natural compreendam a similaridade e a semântica entre palavras (EISENSTEIN, 2019). Considere as palavras "rei"e "rainha."Se estiverem bem representadas por embeddings, a subtração dos vetores "rei"e "homem"deve ser aproximadamente igual à subtração dos vetores "rainha"e "mulher,"refletindo a relação semântica de gênero.

3.3 SIMILARIDADE DE TEXTOS

A Similaridade de Textos é fundamental para comparar documentos ou palavras. Diversas métricas são empregadas para avaliar essa similaridade, como Distância de Jaccard, Distância de Cosseno, Distância Euclidiana e Modelos de Linguagem.

3.3.1 Frequência de Termos e Count Vectorization

A frequência de termos (TF) e a *Count Vectorization* são técnicas fundamentais em processamento de linguagem natural para representar e analisar textos. A TF mede a frequência com que cada termo (palavra) aparece em um documento, enquanto a count vectorization transforma um texto em um vetor numérico que representa a frequência de cada termo.

3.3.2 Distância de Cosseno

A Distância de Cosseno mede o ângulo entre dois vetores de palavras, representando a similaridade direcional entre eles. Quanto menor o ângulo, maior a similaridade. Considere dois vetores de palavras representando documentos. Se esses vetores apontarem na mesma direção, a distância de cosseno indicará alta similaridade. Se apontarem em direções opostas, a distância indicará baixa similaridade.

3.3.3 Distância de Levenshtein

A Distância de Levenshtein, também conhecida como Edição de Texto ou Distância de Edição, é uma métrica fundamental em processamento de linguagem natural para medir a similaridade entre sequências de caracteres, como palavras ou frases. Ela quantifica a diferença mínima entre duas sequências, considerando o número de operações de edição (inserção, deleção ou substituição) necessárias para transformá-las uma na outra. Além de medir um valor de similaridade através da edição de texto, seu valor também é relevanta para analisar mudanças necessárias na sintaxe e na parte léxica de um texto em relação a outro.

3.3.4 Language Models

Os Modelos de Linguagem, como os de Parafraseamento, buscam entender a similaridade semântica entre frases ou documentos, indo além da análise baseada em palavras. Supondo que um modelo de linguagem deve prever palavras em frases. Na frase "O gato está na", o modelo de linguagem pode prever as palavras "casa", "árvore"e rua por exemplo. Com base em um conjunto de dados de treinamento a probabilidade da palavra "casa"pode ser maior. Eles também vão além da análise baseada em palavras, buscando capturar as nuances da linguagem e a similaridade semântica entre frases ou documentos. Podemos destacar para o presente trabalho os seguintes modelos:

a) **BERT**: O BERT (*Bidirectional Encoder Representations from Transformers*) é um dos modelos de linguagem muito usado em processamento de linguagem natural. Ele é treinado em um enorme conjunto de dados de texto e código, permitindo que ele aprenda a representar a linguagem de forma contextual e bidirecional (DEVLIN et al., 2018).

b) **BERTimbau**: O BERTimbau é a versão brasileira do BERT, treinada em um conjunto de dados de texto em português brasileiro. Ele foi aprimorado por pesquisadores brasileiros e oferece melhor desempenho em tarefas específicas para o português brasileiro, como tradução automática, resposta a perguntas, similaridade de senteças textuais e sumarização de texto (SOUZA; NOGUEIRA; LOTUFO, 2020).

c) **BETO**: O BETO é versão espanhola do BERT, treinada com um conjunto de dados de texto em espanhol. Foi desenvolvido por um grupo de pesquisadores da Universidade do Chile. Ele oferece desempenho aprimorado nas mesmas tarefas que os outros modelos, mas voltado especificamente para o idioma espanhol (CAÑETE et al., 2020).

Todos os três modelos possuem uma versão *Base* (12 camadas e 110 milhões de parâmetros) e uma versão Large (24 camadas e 335 milhões de parâmetros). Esses modelos geram *embeddings* das frases, que são representações vetoriais que capturam o significado semântico das frases, permitindo a comparação dos valores para verificar a similaridade semântica entre elas.

3.4 GERAÇÃO DE VALORES DE PESO

É preciso determinar para as avaliações geradas pelo algoritmo os valores dos pesos mais adequados em representar a influência de cada fator dentro de uma métrica. Os pesos podem ser inicializados com valores aleatórios dentro de um intervalo predefinido. Essa é uma técnica simples e comumente utilizada como ponto de partida para outras técnicas. Para melhorar a acurácia da métrica com pesos determinados de forma mais precisa, técnicas dentro do conjunto de Inteligência Artificial e análise de dados podem ser usadas, como por exemplo a regressão linear.

3.4.1 Regressão Linear

Na regressão linear, os pesos representam a importância relativa de cada variável independente na previsão da variável dependente. Em geral, a técnica tem como objetivo tratar de um valor que não se consegue estimar inicialmente. Para isso ela prevê o valor de dados desconhecidos usando valores de dados relacionados e conhecidos, assim, modelando matematicamente a variável desconhecida ou dependente e a variável conhecida ou independente como uma equação linear (MARSLAND, 2014). Por isso a escolha dos valores de peso tem um impacto determinante no desempenho do algoritmo nas avaliações feitas ao final.

3.5 MÉTRICAS DE AVALIAÇÃO

As Métricas de Avaliação quantificam o desempenho de modelos de processamento de linguagem natural. A acurácia é uma medida fundamental, representando a proporção de predições corretas em relação ao total. Outras métricas, como erro médio, erro quadrático médio (EQM) e erro médio absoluto (EMA) são essenciais.

3.5.1 Acurácia

A acurácia é uma métrica fundamental de avaliação, comumente usada para medir o desempenho geral de modelos de processamento de linguagem natural. Uma alta acurácia indica bom desempenho geral, mas pode mascarar problemas em classes minoritárias. A medida representa a proporção de predições corretas em relação ao total de predições. Embora seja uma medida direta, a Acurácia pode ser limitada em cenários desbalanceados, sendo complementada por métricas adicionais, como precisão, revocação e F1-Score, para avaliação mais abrangente do desempenho do modelo. A Acurácia geral também pode ser expressa em forma de uma porcentagem

3.5.2 Erro Médio

A soma dos erros absolutos de todas as predições, dividida pelo número total de predições. Mede a magnitude média do erro.

3.5.3 Erro Quadrático Médio (EQM)

A média dos erros quadrados de todas as predições. Mede a magnitude média do erro ao quadrado, penalizando erros maiores mais fortemente.

3.5.4 Erro Médio Absoluto (EMA)

A média dos erros absolutos de todas as predições. Mede a magnitude média do erro sem considerar o sinal, útil para avaliar a distância entre valores preditivos e reais.

4 METODOLOGIA

Nesta seção, descrevemos a metodologia adotada para desenvolver e testar o algoritmo de avaliação automática de respostas dissertativas. O processo abrange desde a formatação dos dados até a comparação dos resultados gerados pelo algoritmo com as avaliações realizadas pelos docentes. De forma visual o processo geral pode ser visto na Figura 1

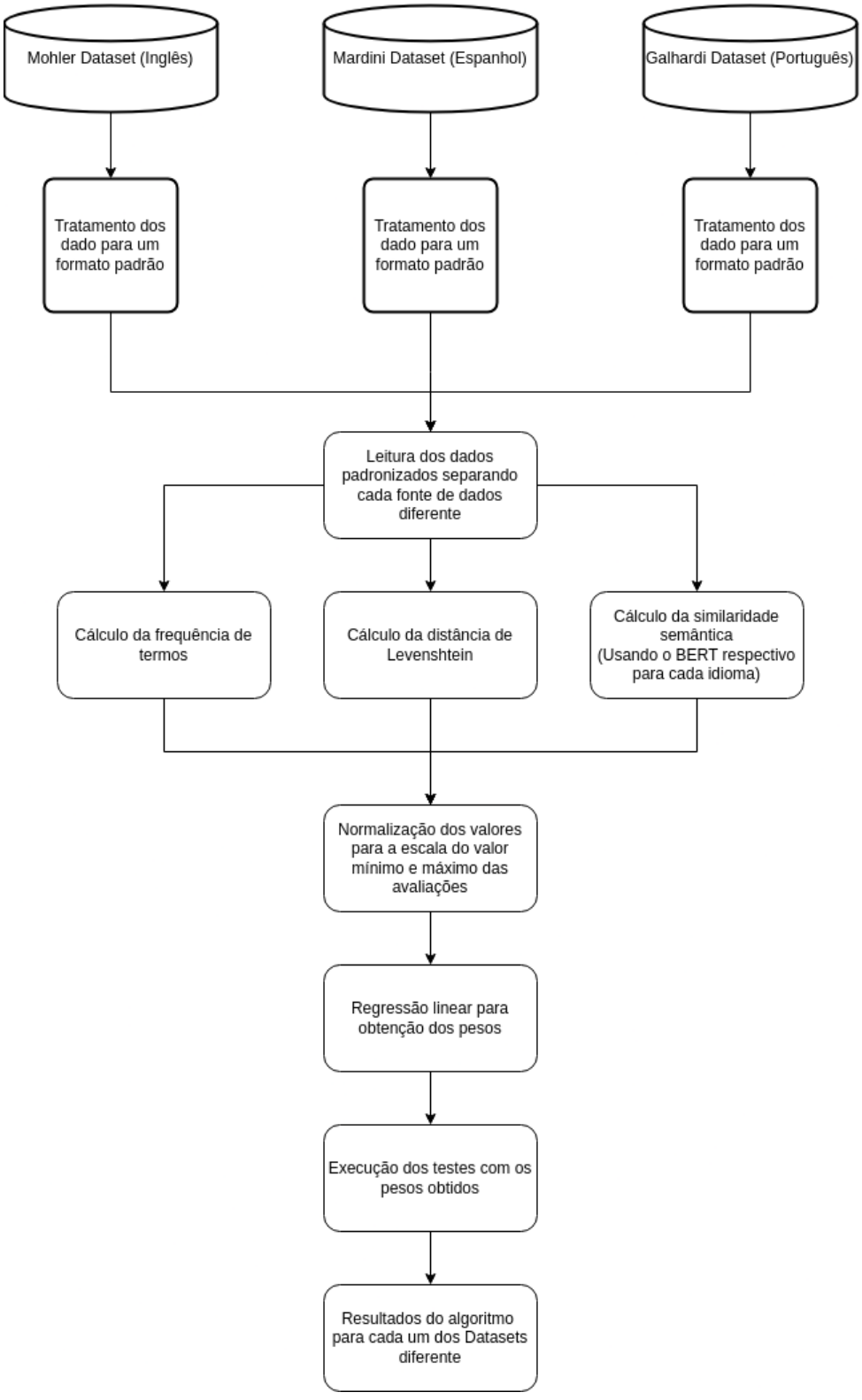

Figura 1 – Diagrama do fluxo completo do algoritmo

Na metodologia, de forma geral, foram retirados dos textos disponíveis nas bases de dados os três fatores que são usados regressão linear e posteriormente nos testes de geração de avaliações. Esses fatores são:

a) **As propriedades léxicas do texto:** O primeiro fator será levado em consideração fazendo uso das frequências de termos de cada texto.

b) **As propriedades da sintaxe do texto:** O segundo fator será levado em consideração fazendo uso da distância de Levenshtein.

c) **As propriedades da semântica do texto:** O terceiro fator será levado em consideração fazendo uso dos valores de similaridade semântica fornecidos pelos embeddings gerados pelo modelo BERT.

A métrica utilizada na metodologia pode ser matemáticamente descrita como o exemplo da Equação 1, que contém a equação de uma média ponderada.

$$Métrica = \frac{fator1 \times peso_1 + fator2 \times peso_2 + fator3 \times peso_3}{\sum_{i=1}^{3} peso_i} \quad (1)$$

4.1 PIPELINE DE EXECUÇÃO DO ALGORITMO

O pipeline de execução do algoritmo compreende várias etapas, desde a preparação dos dados até a análise dos resultados. As etapas podem ser detalhadas da seguinte forma:

4.1.1 Formatação dos Dados

A primeira etapa envolve a formatação dos dados oriundos de diferentes bases, convertendo-os em um modelo comum.

Uma das principais dificuldades encontradas inicialmente no presente trabalho foi o tratamento de dados provenientes de múltiplas bases em diferentes idiomas, cada uma com formatos e estruturas distintas.

Isso exigiu a implementação de algoritmos de pré-processamento e normalização de texto para garantir a consistência dos dados antes da análise. Isso foi fundamental para desenvolver abordagens eficazes de tratamento de dados em inglês, espanhol e português.

Isso garante a padronização dos dados, facilitando o processamento subsequente. A estrutura de dados é definida como demonstrado no diagrama da Figura 2, onde cada entrada é tipada e preparada para a análise, no fim essas informações são salvas em arquivo no formato *.json* para facilitar a leitura com auxílio de bibliotecas prontas na linguagem de programação *Python*.

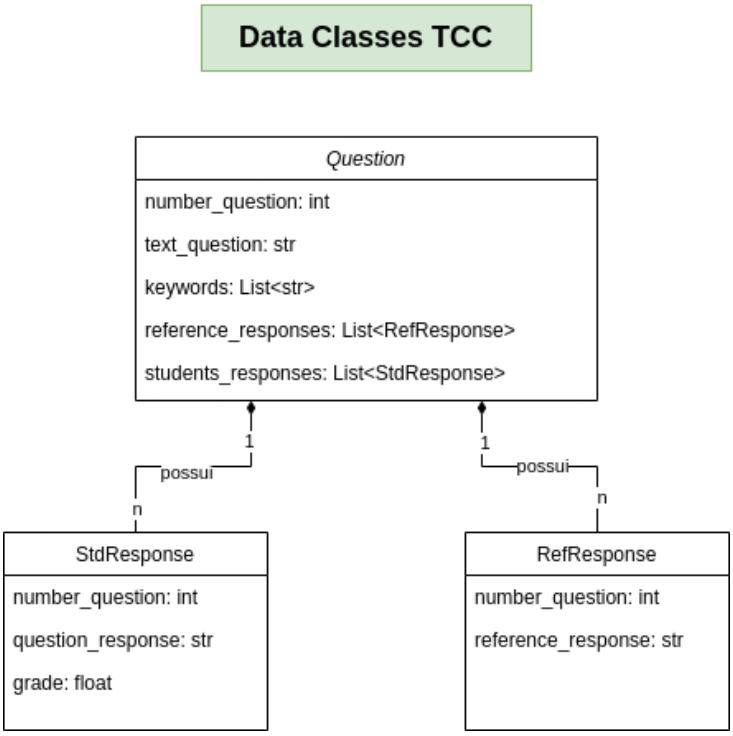

Figura 2 – Modelo de padronização dos dados representados como classes

4.1.2 Extração e Normalização dos Fatores

Após a formatação, os dados são processados pelo algoritmo para a extração de três fatores principais que influenciam a avaliação. Primeiramente, é feito o cálculo da distância de Levenshtein entre o texto da resposta de referência do professor e a resposta de um aluno disponível na base de dados, retornando um valor para avaliar a proximidade da sintaxe dos dois textos. Em seguida, o cálculo da frequência de termos usando *Count Vectorization* é feito para ambos os textos, e a comparação entre os vetores gerados para eles é feita usando a distância de cosseno, gerando um valor para avaliar a semelhança léxica dos textos.

Depois disso, utilizando o modelo BERT, os embeddings dos textos são calculados. O BERT utiliza uma técnica chamada "WordPiece tokenization"para dividir o texto em pedaços menores, chamados de "tokens". Cada token é então convertido em um vetor de números reais através de uma camada de embeddings. Esses embeddings capturam o contexto e o significado das palavras no texto. Posteriormente, os embeddings de cada texto são organizados em matrizes e manipulados para calcular a similaridade entre eles utilizando a distância de cosseno, o que resulta em um valor que representa a similaridade semântica entre os dois textos.

Esse processo é repetido algumas vezes apenas para o fator semântico, pois existem modelos diferentes do BERT em cada idioma e versões diferentes (*Base* e *Large*). É necessário

utilizar os dados gerados por cada um dos modelos para avaliar o desempenho do algoritmo com cada um deles.

No fim, todos os fatores são normalizados com base nos valores mínimos e máximos das notas fornecidas pelos professores. A normalização é crucial para garantir que os fatores se situem dentro da mesma escala na qual as notas devem ser avaliadas.

4.1.3 Regressão Linear

Com os fatores normalizados, a próxima etapa é a aplicação de uma regressão linear. A regressão é treinada comparando os fatores extraídos com as notas das avaliações dos professores. Isso permite determinar os pesos específicos para cada fator. Os pesos obtidos são essenciais para a fase subsequente, onde serão usados para gerar novas avaliações. Além dos pesos, com a regressão também é possível obter os valores do EQM (*Mean Squared Error* ou MSE) e do EMA (*Mean Absolute Error* ou MAE).

4.1.4 Geração de Avaliações

Utilizando os pesos determinados pela regressão linear, o algoritmo gera avaliações automáticas para um conjunto de dados separados. Esses dados são as respotas dos alunos em comparadas com as referências dos professsores, porém nesse momento os três fatores são retirados dos textos e logo em seguida uma avaliação de nota é gerando fazendo o cálculo da média ponderada, para isso são usados, em conjunto com os valores de cada fator, os valores de cada peso que foram determinados anteriormente na regressão linear. Após isso, ainda nesta etapa, os valores das avaliações geradas pelo algoritmo são guardados para serem comparados com as avaliações feitas pelos próprios professores para as mesmas respotas de alunos disponíveis nas bases de dados, assim podermos constatar se há semelhança entre os valores de ambas as avaliações, do algoritmo e dos professores.

4.1.5 Comparação e Cálculo de Acurácia

Finalmente, as avaliações geradas pelo algoritmo são comparadas com as notas originais fornecidas pelos professores na base de dados. Essas comparações são feitas dividindo o conjunto de dados em diferentes quantias de valores. Primeiramente, 60% dos dados são usados para o treino da regressão na etapa anterior e 40% dos dados são usados para testes na etapa atual. Em seguida, os valores de treino e testes são ajustados para 70% e 30%, respectivamente. Essa

alteração é feita sucessivamente para os valores de 80% e 20% até os valores finais de 90% e 10%. Esse ciclo é repetido para todas as bases de dados e todas as versões dos modelos *BERT*. A precisão dessas avaliações é verificada em termos de acurácia percentual. A acurácia média é então calculada para todas as avaliações geradas nos testes, permitindo avaliar o desempenho do algoritmo.

4.1.6 Execução do algoritmo em diferentes bases e modelos

Todo o ciclo de metodologia é executado múltiplas vezes para cada uma das bases de dados e das diferentes versões dos modelos, afim de obter a maior quantidade de dados possíveis para avaliação da acurácia do algoritmo no final em vários casos e observar em quais casos os resultados obtidos foram melhores.

4.1.7 Otimização do algoritmo

No decorres do desenvolvimento do presente trabalho, uma dificuldade enfrentada foi otimizar a execução do algoritmo para lidar com grandes volumes de dados de forma eficiente. Isso exigiu a implementação de estratégias avançadas de otimização, como a paralelização com *multi-threading*, a programação dinâmica, uso de protocolos como *Open MPI* e a execução dos modelos na *GPU* com *CUDA*.

A paralelização com *multi-threading* permitiu que o algoritmo executasse múltiplas tarefas em paralelo, acelerando o processamento de grandes conjuntos de dados. A programação dinâmica otimizou o uso de recursos computacionais e reduziu a complexidade algorítmica, garantindo uma execução mais eficiente do algoritmo e menos repetições desnecessárias de trechos de código. A divisão de processos com protocolo *MPI* permitiu que diferentes partes do algoritmo trocassem dados de forma assíncrona e coordenada, dividindo a carga de trabalho. Já a execução dos modelos na *GPU* com *CUDA* acelerou o processamento de dados, aproveitando o poder de processamento das *GPU's* para lidar com grandes volumes de dados de forma eficiente.

Essas técnicas de otimização garantiram uma execução eficiente do algoritmo, permitindo lidar com grandes volumes de dados e garantir tempos de execução adequados. Antes desse processo de otimização, o tempo estimado de execução do algoritmo estava em cerca de 10 dias cada vez que era rodado. Após as otimizações do código, o algoritmo pode ser executado completamente em cerca de 40 minutos. A experiência técnica e o conhecimento especializado

obtidos ao longo da graduação foram fundamentais para superar os desafios enfrentados durante o desenvolvimento do algoritmo e garantir sua eficácia e otimização nos tempos de execução.

4.2 *DATASETS UTILIZADOS*

Para este projeto de avaliação automática de respostas curtas, foram utilizadas três bases de dados distintas, cada uma em um idioma específico. A primeira base de dados consiste em um conjunto de questões de Biologia em português, a segunda base de dados é composta por questões de Literatura em espanhol e a terceira base de dados contém questões de Ciência da Computação em inglês.

A base de dados em português foi apresentada por Galhardi, Souza e Brancher (2020) criada em 2020, contendo cerca de 15 questões e 23350 respostas de alunos.

Além disso, o conjunto de dados em espanhol apresentado por Mardini G. et al. (2023) em 2023, contem cerca de 20 questões e 3770 respostas de alunos.

Por fim, a base de dados em inglês empregada foi apresentada por Mohler e Mihalcea (2009) em 2009, contando com 85 questões e 3645 respostas de alunos.

5 RESULTADOS OBTIDOS

5.1 DESEMPENHO DO ALGORITMO

Os resultados obtidos demonstram o desempenho do algoritmo em diferentes configurações de treinamento e teste. Abaixo estão os resultados para cada conjunto de dados de treinamento e teste:

5.1.1 Resultados do treinamento com o *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*

Os resultados do treinamento com o *dataset Galhardi* em português utilizando o modelo *BERTimbau Base* mostram uma acurácia média variando de 59.35% a 63.43%. A acurácia mais alta foi alcançada com 80% dos dados de treinamento. Apesar de a acurácia média ser menor em comparação com os modelos de inglês e espanhol que serão exibidos em após os resultados em portugês.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia média**
60%	14026	9352	[0.6540, 0.2649, 0.2679]	0.5211	0.5284	59.35%
70%	16364	7014	[0.6064, 0.2940, 0.2973]	0.5165	0.5262	61.46%
80%	18702	4676	[0.5772, 0.2574, 0.3362]	0.5385	0.5375	63.43%
90%	21040	2338	[0.4971, 0.2798, 0.4261]	0.5950	0.5686	62.26%

Tabela 3 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*

Nos gráfico das Figuras 3, 4, 5 e 6 é possível ver uma concetração dos dados na faixa de 40% até 60%, com uma queda dessa faixa mais notável na 6.

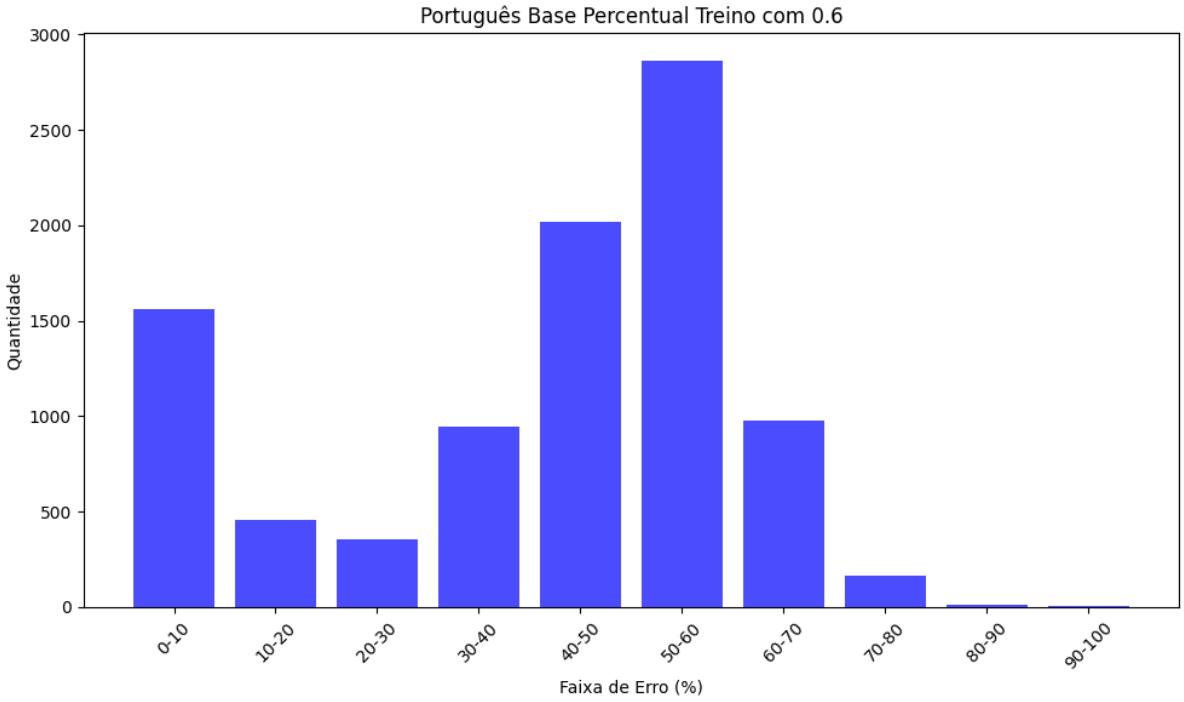

Figura 3 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*)

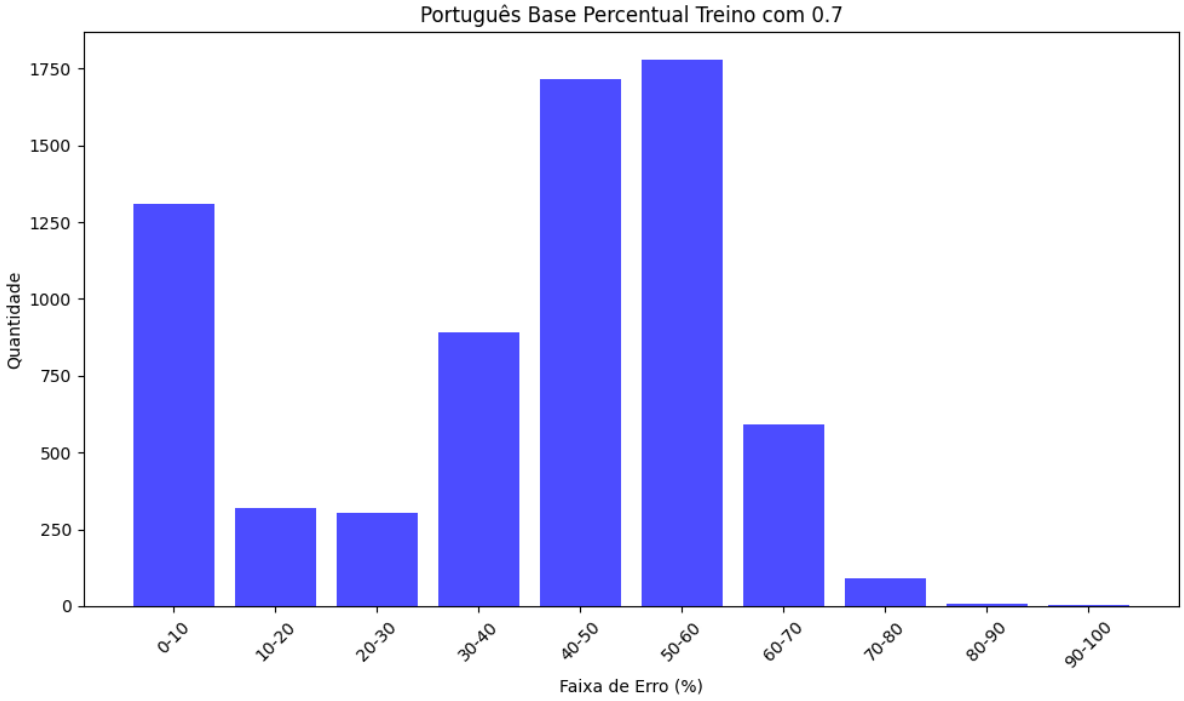

Figura 4 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*)

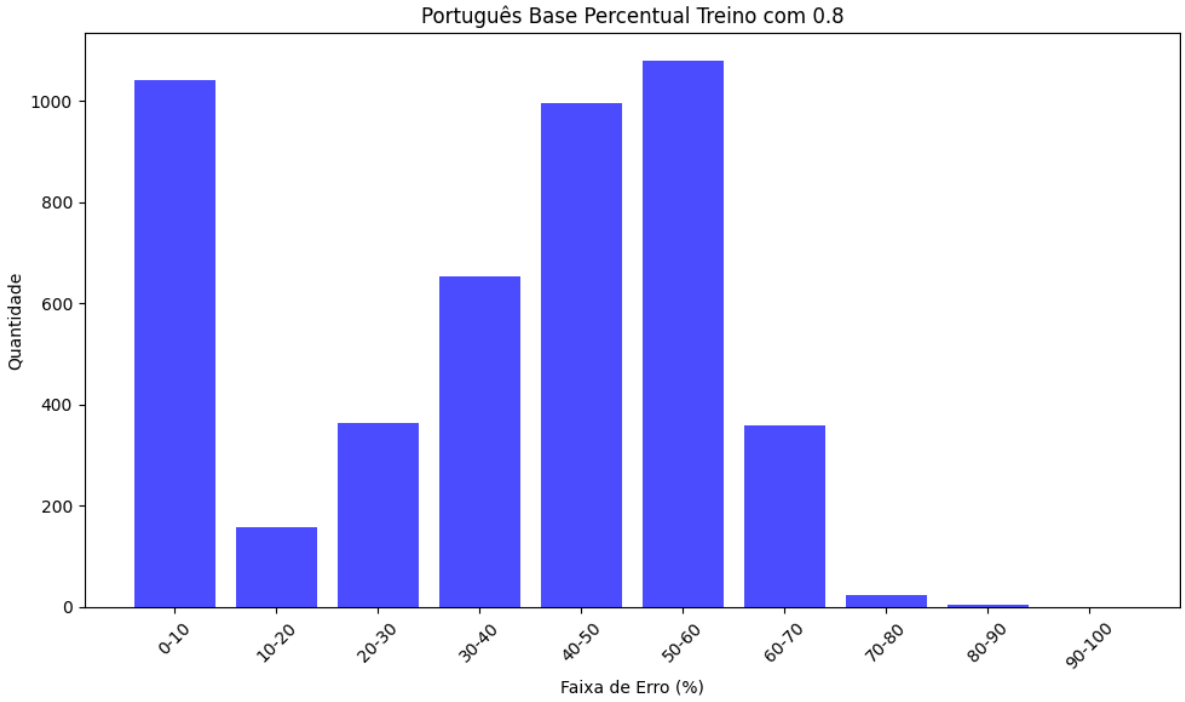

Figura 5 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*)

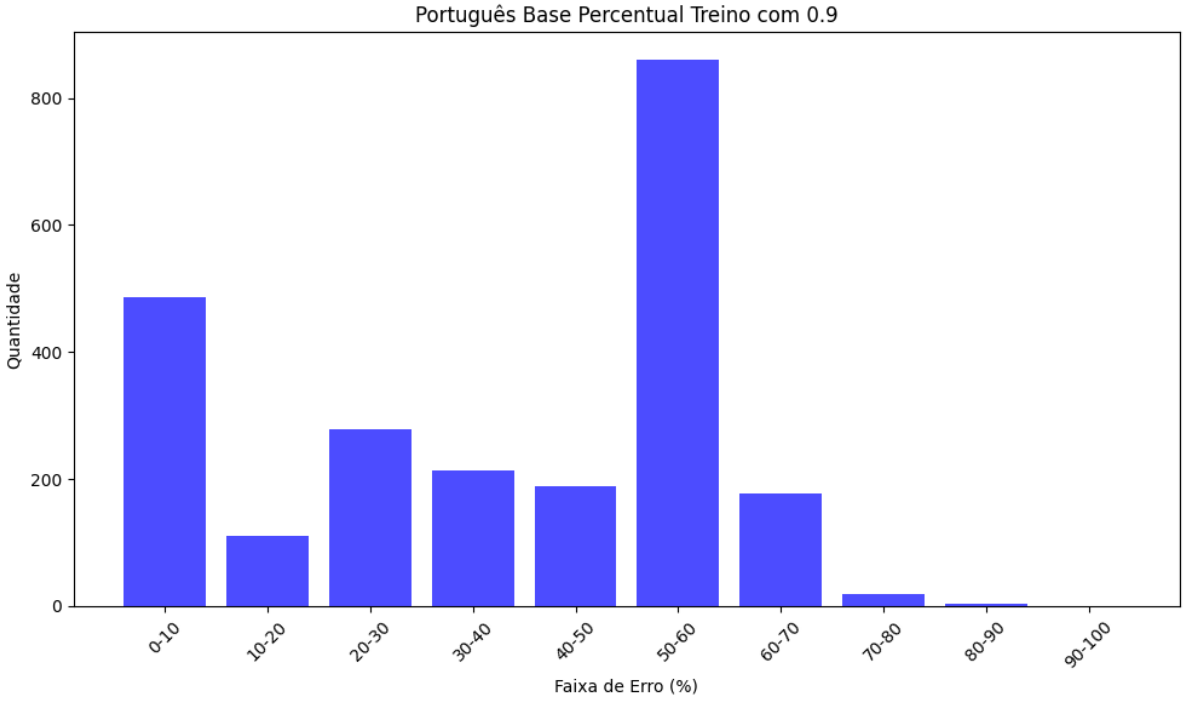

Figura 6 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Base*)

5.1.2 Resultados do treinamento com o *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*

Os resultados para o *dataset Galhardi* em português utilizando o modelo *BERTimbau Large* indicam uma acurácia média variando entre 55.4% e 59.83%. A acurácia mais alta foi obtida com 80% dos dados de treinamento. Embora a acurácia média seja menor do que a observada com o modelo *BERTimbau Base*, os valores de EQM e EMA sugerem que o modelo *BERTimbau Large* pode precisar de mais ajustes para alcançar um desempenho superior em tarefas de linguagem natural em português, pelo menos quando usado na tarefa de avaliar similaridade semântica.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia média**
60%	14026	9352	[0.7045, 0.2960, 0.5349]	0.5230	0.5318	55.4%
70%	16364	7014	[0.6538, 0.3434, 0.5882]	0.5201	0.5317	58.14%
80%	18702	4676	[0.6335, 0.3061, 0.6643]	0.5438	0.5443	59.83%
90%	21040	2338	[0.5673, 0.3484, 0.8416]	0.6037	0.5765	57.91%

Tabela 4 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*

Nos gráfico das Figura 7 é possível ver uma concetração dos dados na faixa de 50% até 70%, embora na Figura 10 o erro pareça menor, ainda é possível notar que esse faixa possui mais concentração que as outras.

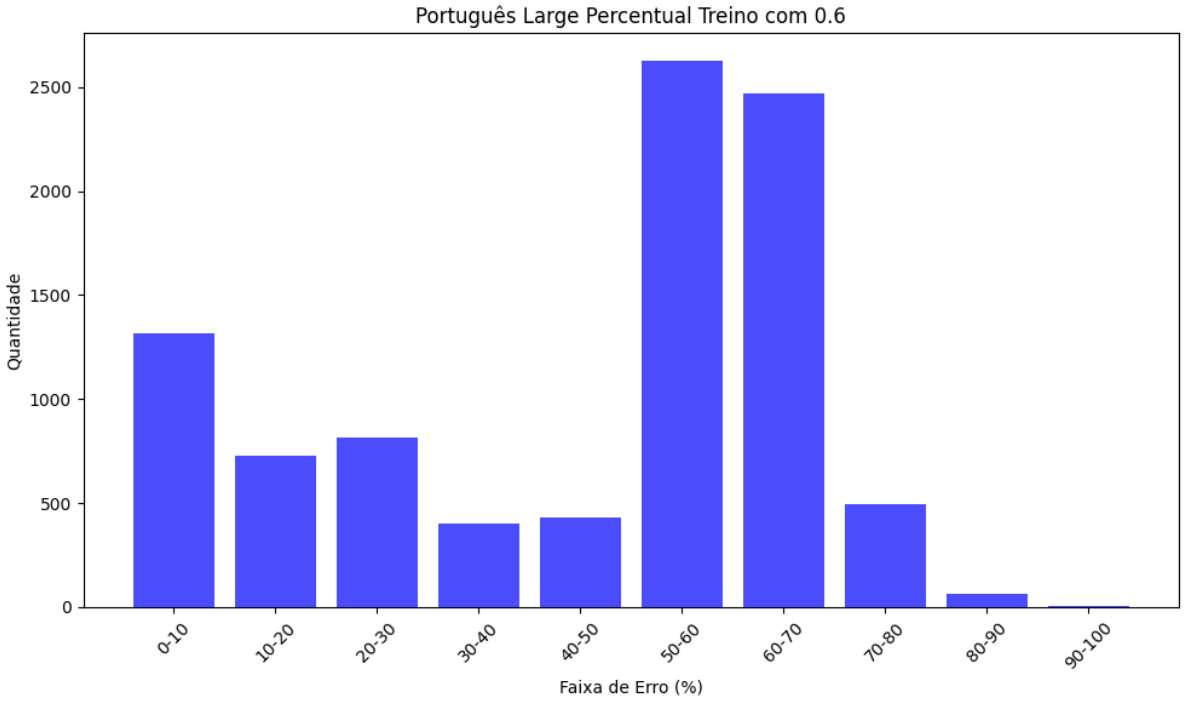

Figura 7 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*)

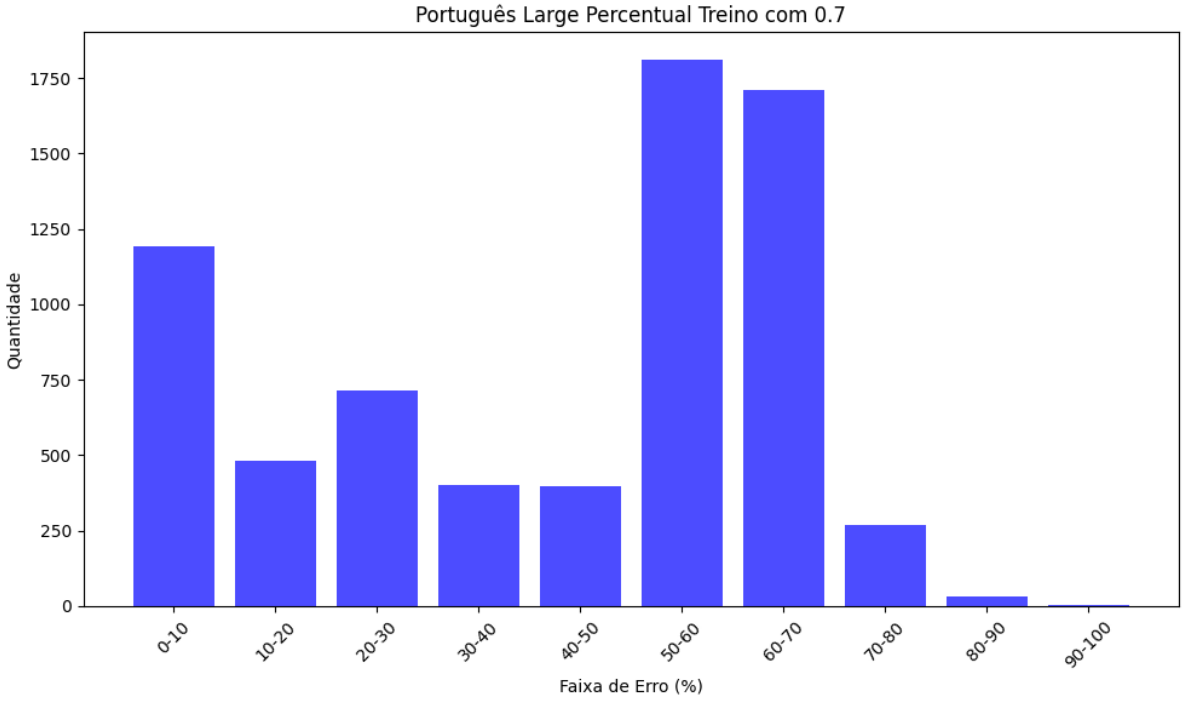

Figura 8 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*)

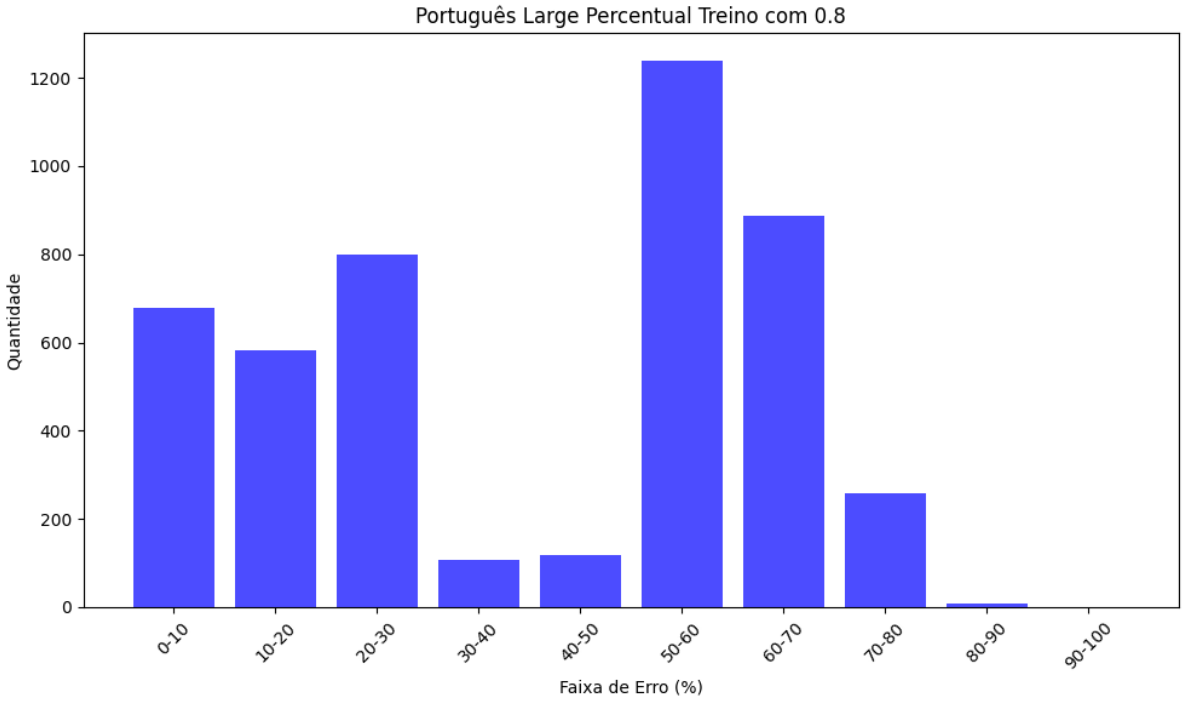

Figura 9 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*)

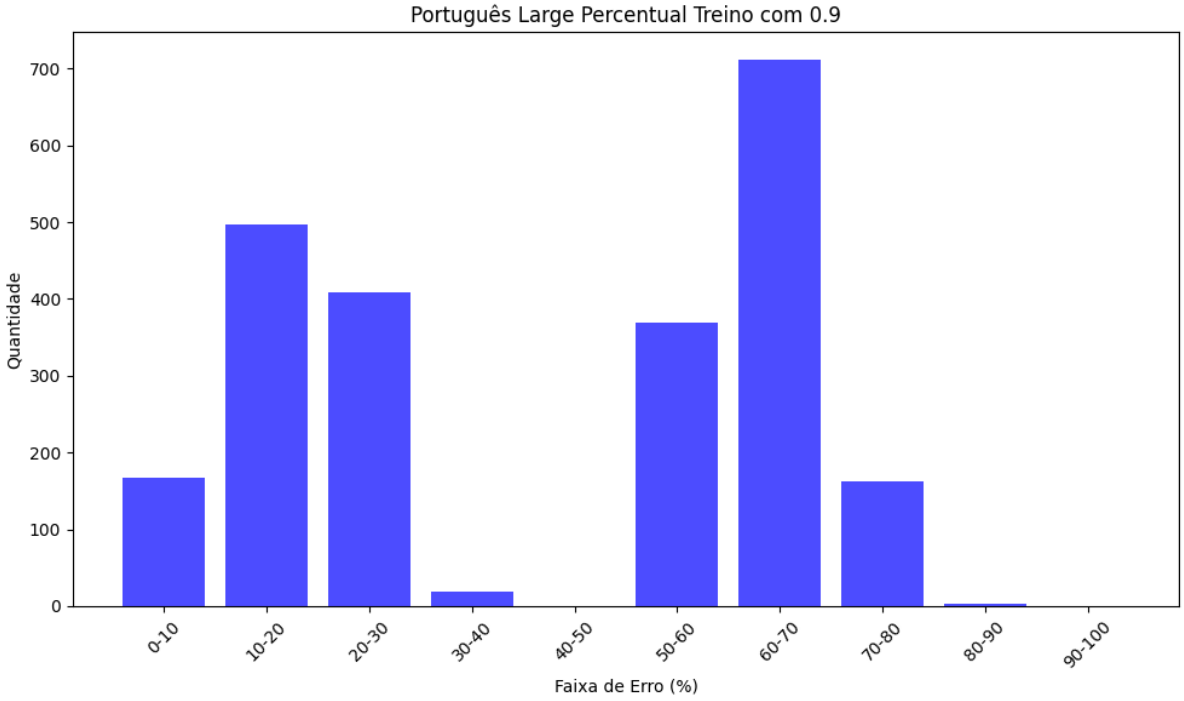

Figura 10 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Galhardi* (Português) usando o Modelo *BERTimbau Large*)

5.1.3 Resultados do treinamento com o *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

Os resultados para o *dataset Mardini* em espanhol utilizando o modelo *BETO Base* demonstram uma acurácia média variando entre 65.12% e 71.7%. A acurácia mais alta foi alcançada com 90% dos dados de treinamento. Os valores de EQM e EMA são relativamente estáveis, com uma leve melhoria à medida que a quantidade de dados de treinamento aumenta. Estes resultados indicam que o modelo *BETO Base* é eficaz para tarefas em espanhol, embora haja espaço para melhorias no uso de outra versão do modelo, como veremos na comparação a versão *Large*.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia média**
60%	2263	1509	[0.0838, 0.0349, 0.4526]	0.8955	0.7457	65.12%
70%	2640	1132	[0.1119, 0.0387, 0.4619]	0.9010	0.7536	66.58%
80%	3017	755	[0.1077, 0.0453, 0.4709]	0.8702	0.7390	66.28%
90%	3394	378	[0.1034, 0.0486, 0.4504]	0.8822	0.7449	71.7%

Tabela 5 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

Nos gráfico das Figuras 11 é possível ver uma quantidade grande de erros distribuídos em varias faixas, ao passo que na Figura 14 esses erros diminuíram, e a faixa mais concetrada é a de entre 20% e 30% de divergência nas avaliações do algoritmo comparadas com as avaliações dos professores.

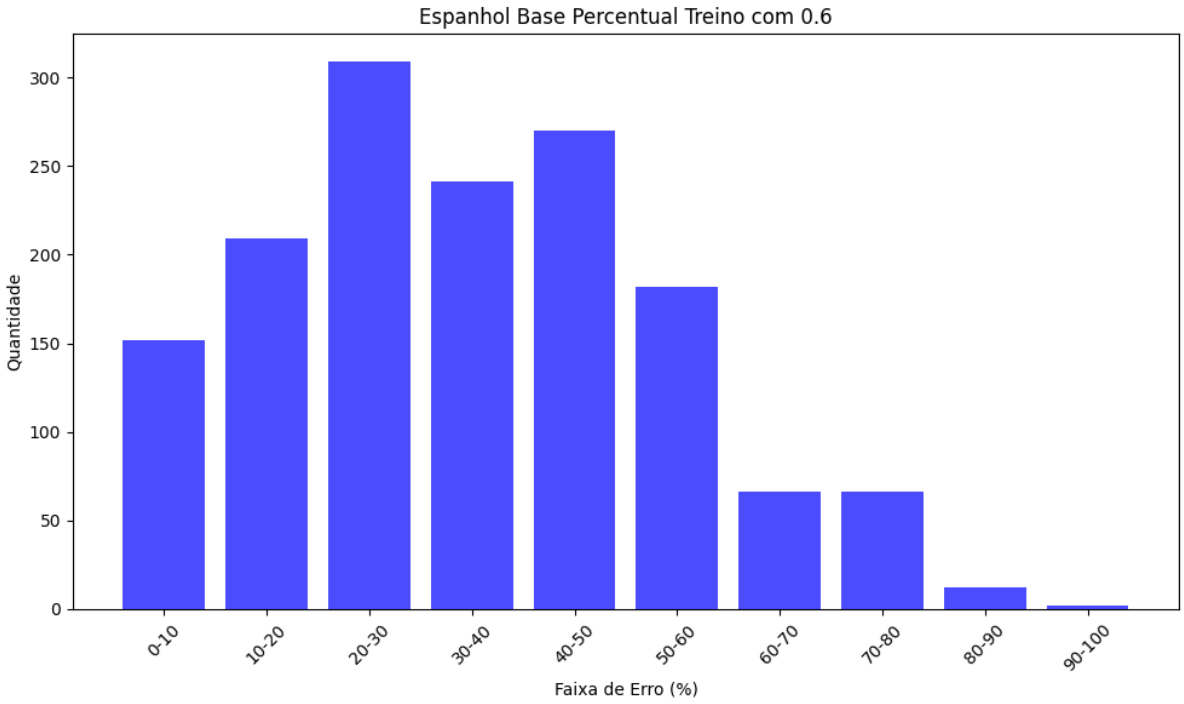

Figura 11 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

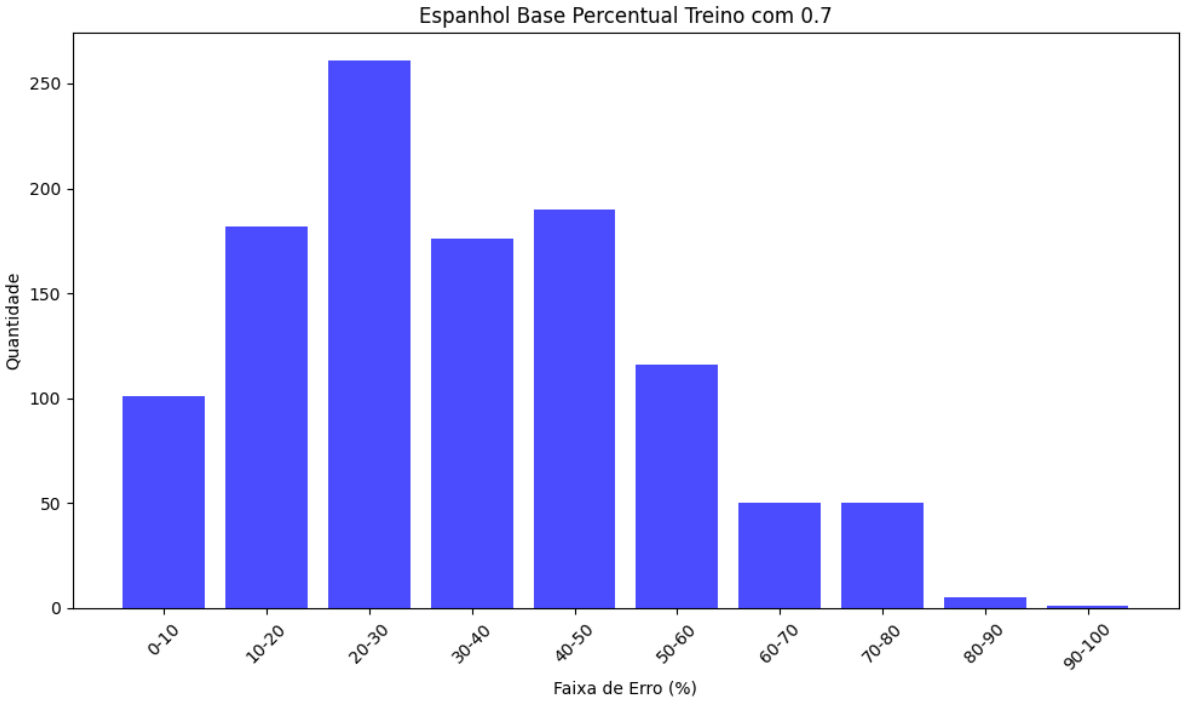

Figura 12 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

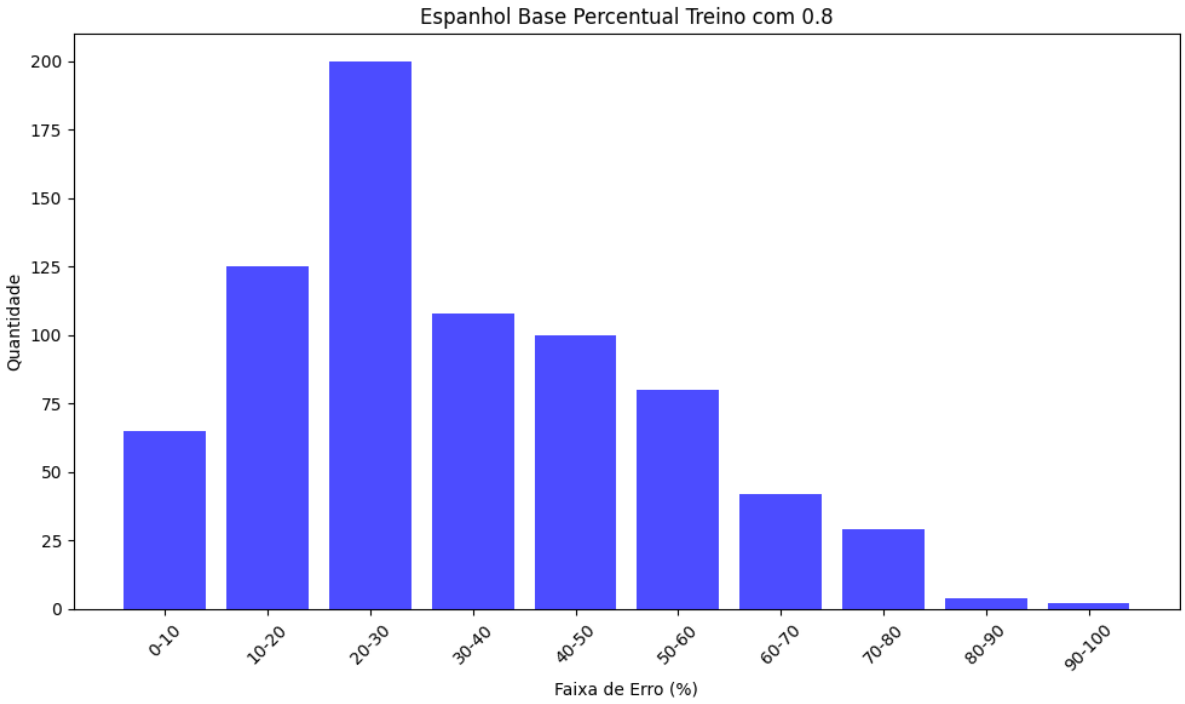

Figura 13 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

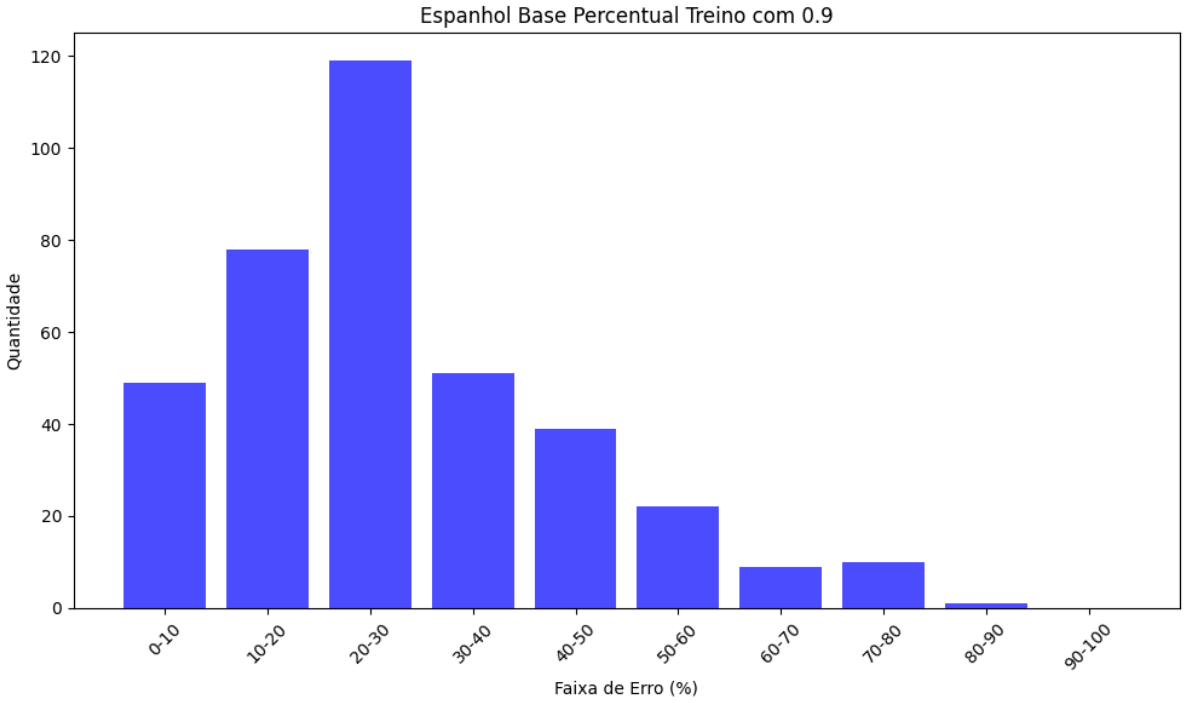

Figura 14 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Base*

5.1.4 Resultados do treinamento com o *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

Para o *dataset Mardini* em espanhol utilizando o modelo *BETO Large*, os resultados mostram uma acurácia média superior, variando de 78.14% a 83.58%. A acurácia mais alta foi obtida com 90% dos dados de treinamento. Os valores de EQM e EMA são ligeiramente melhores comparados ao modelo *BETO Base*, confirmando que o *BETO Large* oferece um desempenho aprimorado em tarefas de processamento de linguagem natural em espanhol.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia média**
60%	2263	1509	[0.1412, 0.0081, 0.1289]	0.9622	0.7773	78.14%
70%	2640	1132	[0.1634, 0.0049, 0.1366]	0.9703	0.7835	78.74%
80%	3017	755	[0.1638, 0.0010, 0.1300]	0.9450	0.7754	79.18%
90%	3394	378	[0.1509, 0.0085, 0.1439]	0.9441	0.7783	83.58%

Tabela 6 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

Nos gráfico das Figuras 15, 16, 17 e 18 é possível ver que a maiora das avaliações se mantém na faixa de acurácia acima de 70%, com uma mudança na distribuiçao comparando com o modelo *Base*.

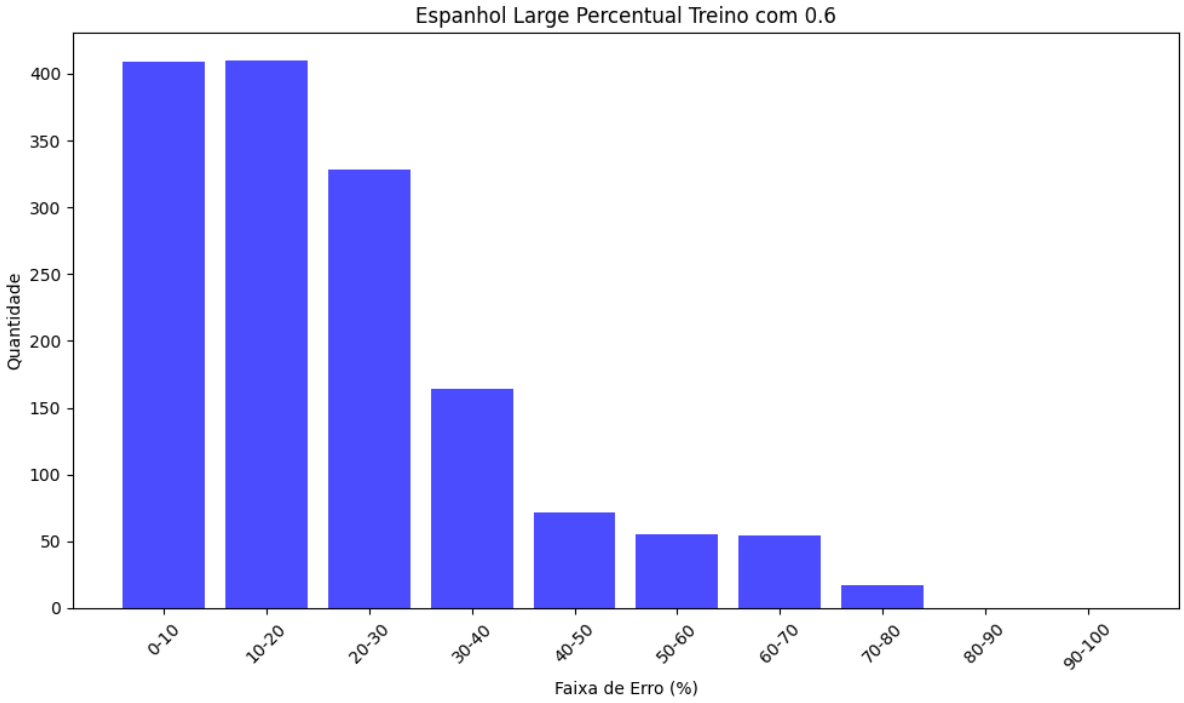

Figura 15 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

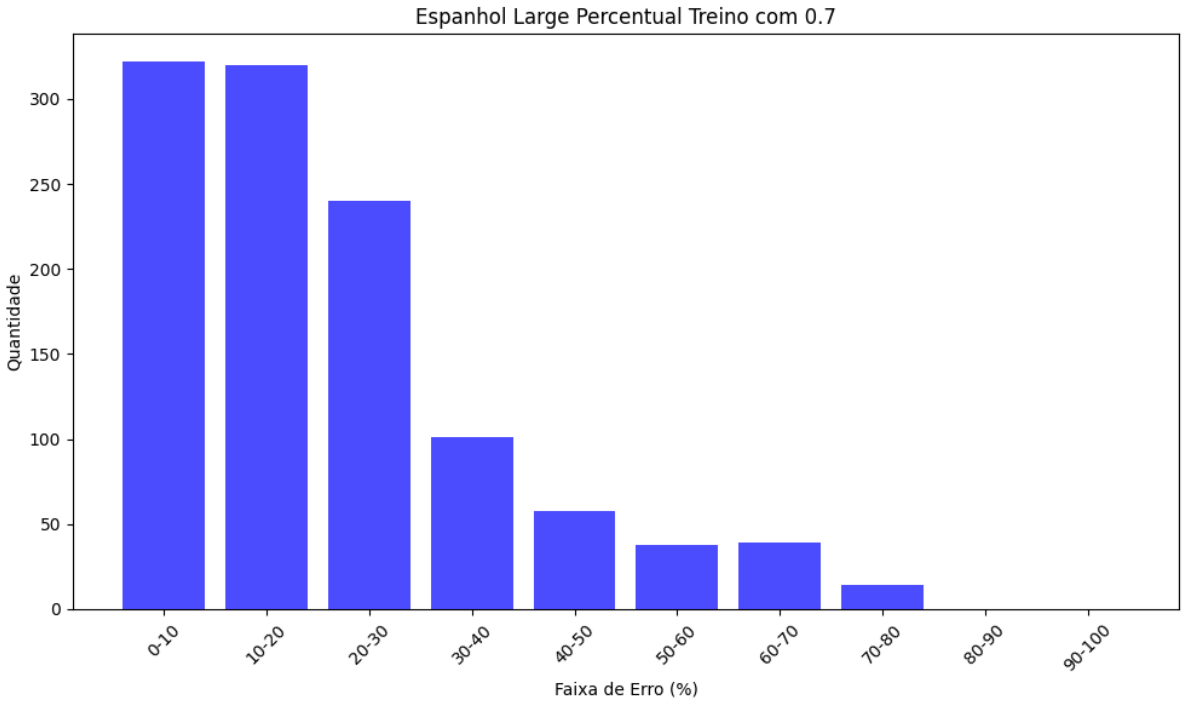

Figura 16 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

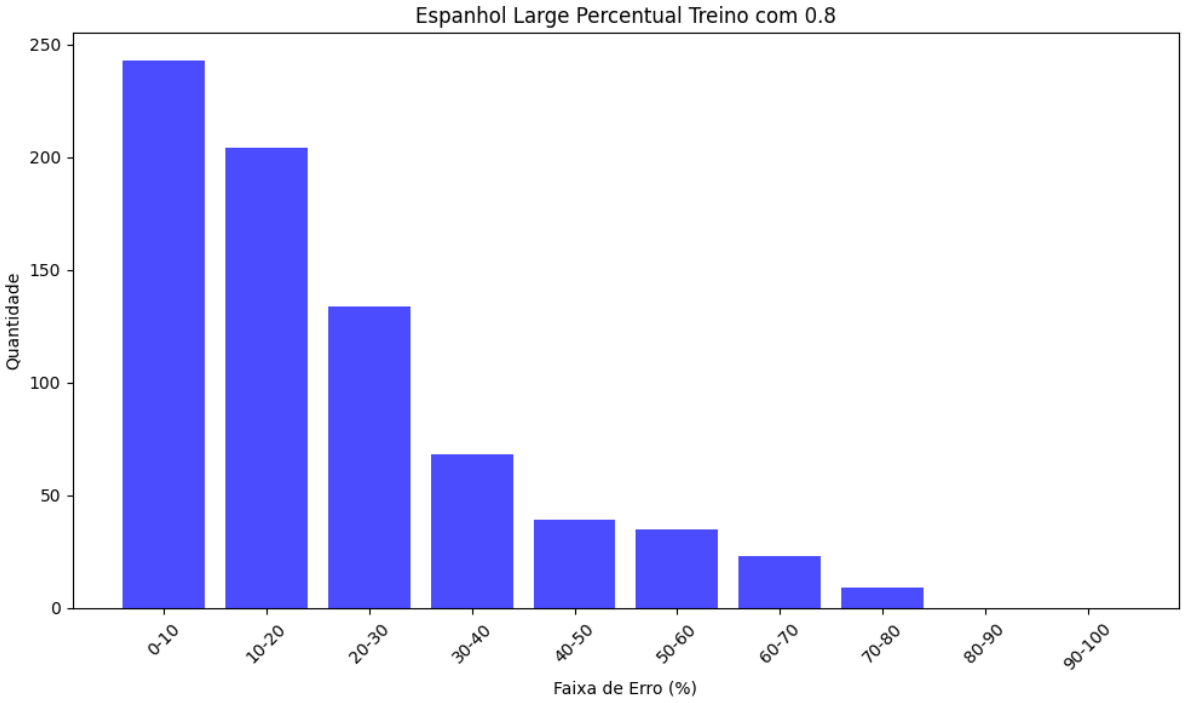

Figura 17 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

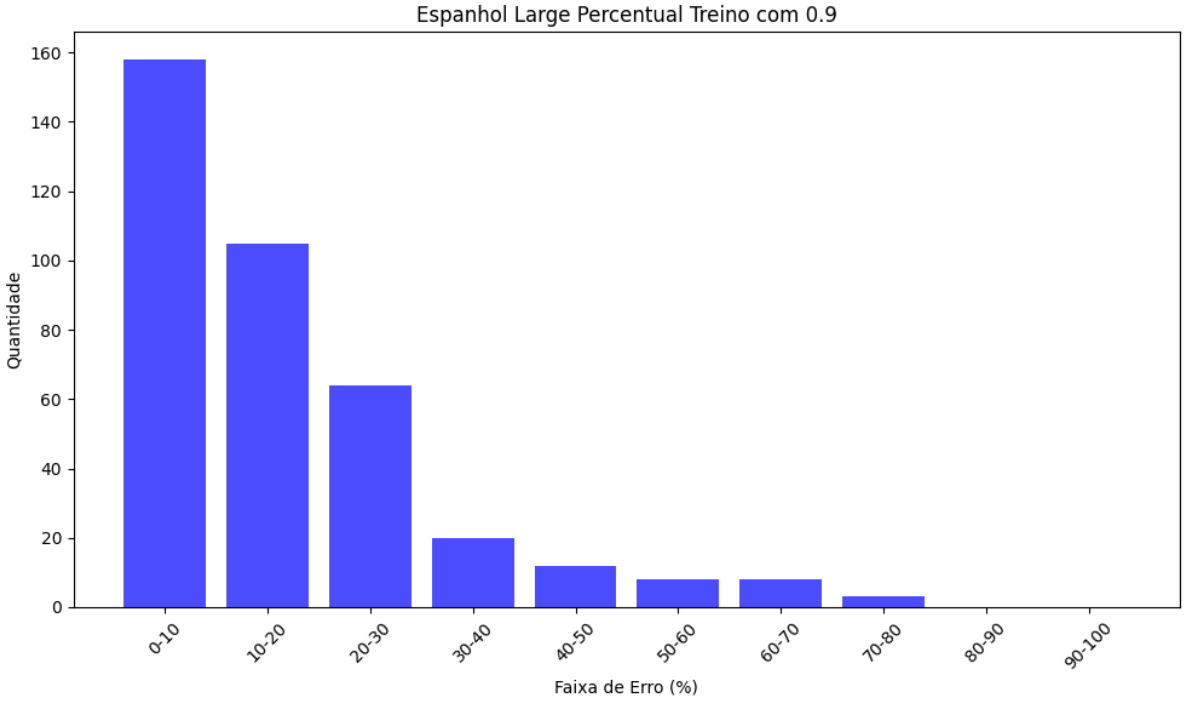

Figura 18 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Mardini* (Espanhol) usando o Modelo *BETO Large*

5.1.5 Resultados do treinamento com o *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

Os resultados do treinamento com o *dataset Mohler* em inglês utilizando o modelo *BERT Base* mostram que a acurácia média varia entre 78.15% e 80.13% conforme aumenta a quantidade de dados de treinamento. Observa-se que o EQM e o EMA diminuem ligeiramente à medida que mais dados são usados para treinamento, indicando uma melhoria na precisão do modelo. A melhor acurácia média foi alcançada com 70% dos dados de treinamento.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia média**
60%	2187	1459	[0.4459, 0.1509, 0.1015]	1.1724	0.8516	79.81%
70%	2552	1094	[0.4643, 0.1622, 0.1140]	1.1672	0.8456	80.13%
80%	2916	730	[0.4469, 0.1705, 0.1459]	1.1086	0.8242	78.79%
90%	3281	365	[0.4312, 0.1681, 0.1418]	1.1034	0.8236	78.15%

Tabela 7 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

Nos gráficos das Figuras 19, 20, 21 e 22 é possível ver que a maiora das avaliações estão em uma faixa de acurácia acima de 80%.

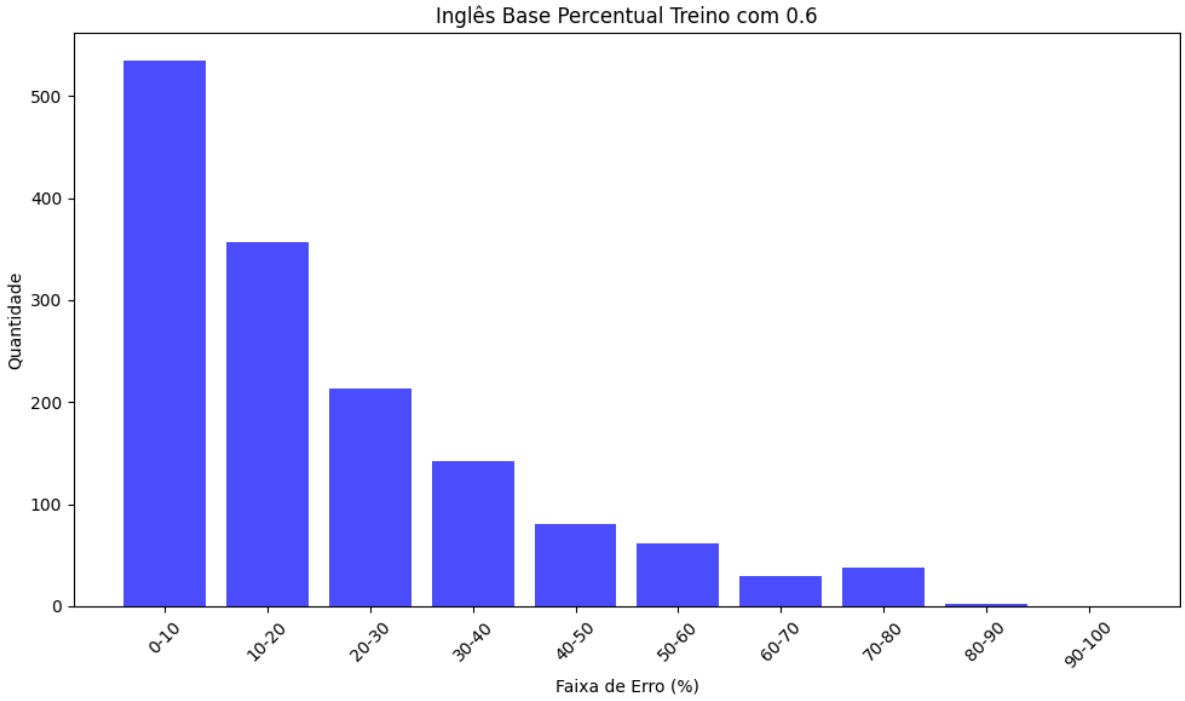

Figura 19 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

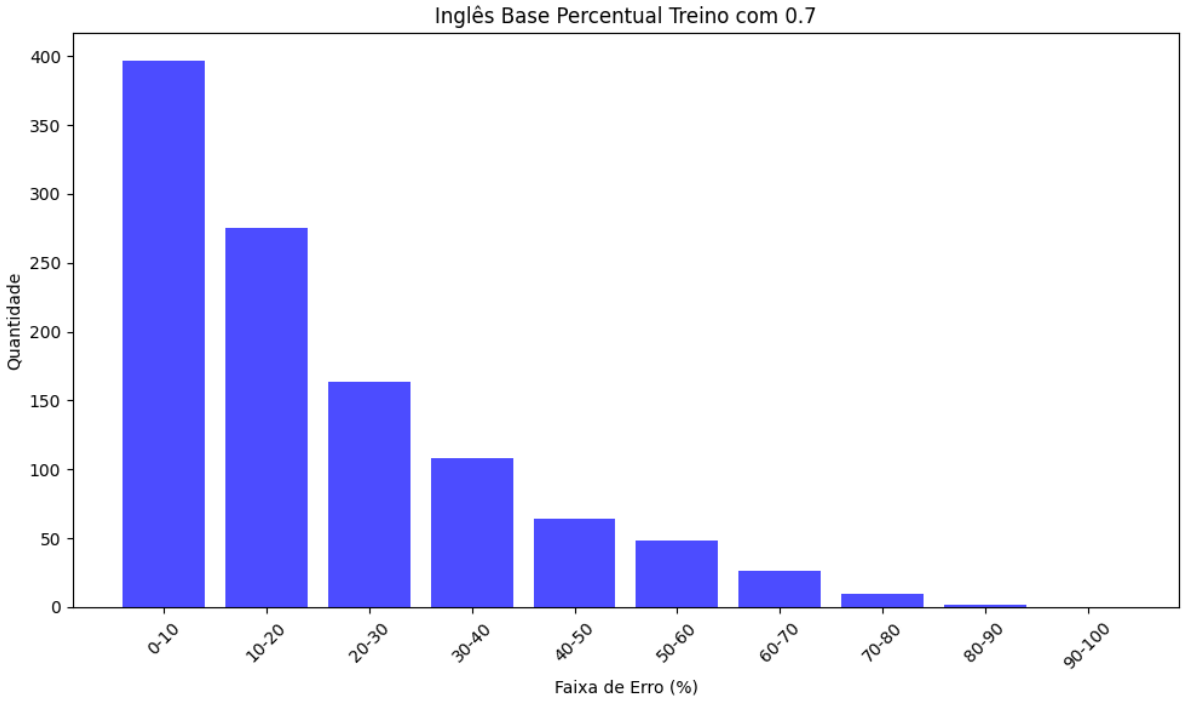

Figura 20 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

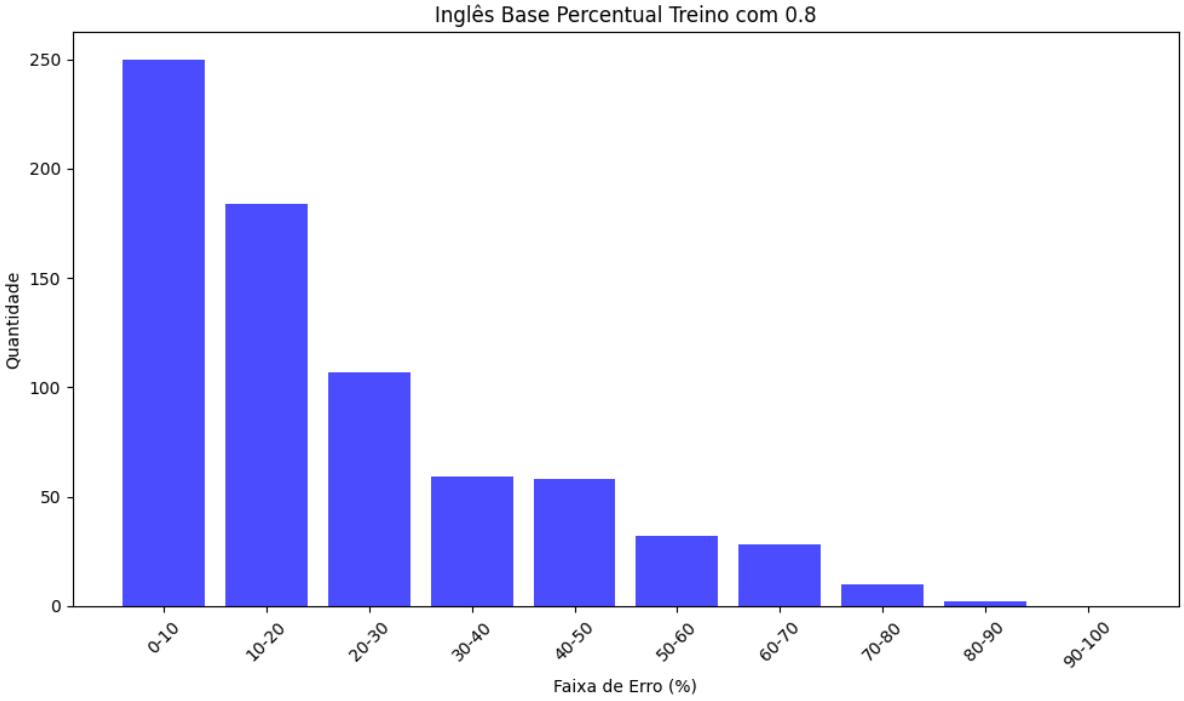

Figura 21 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

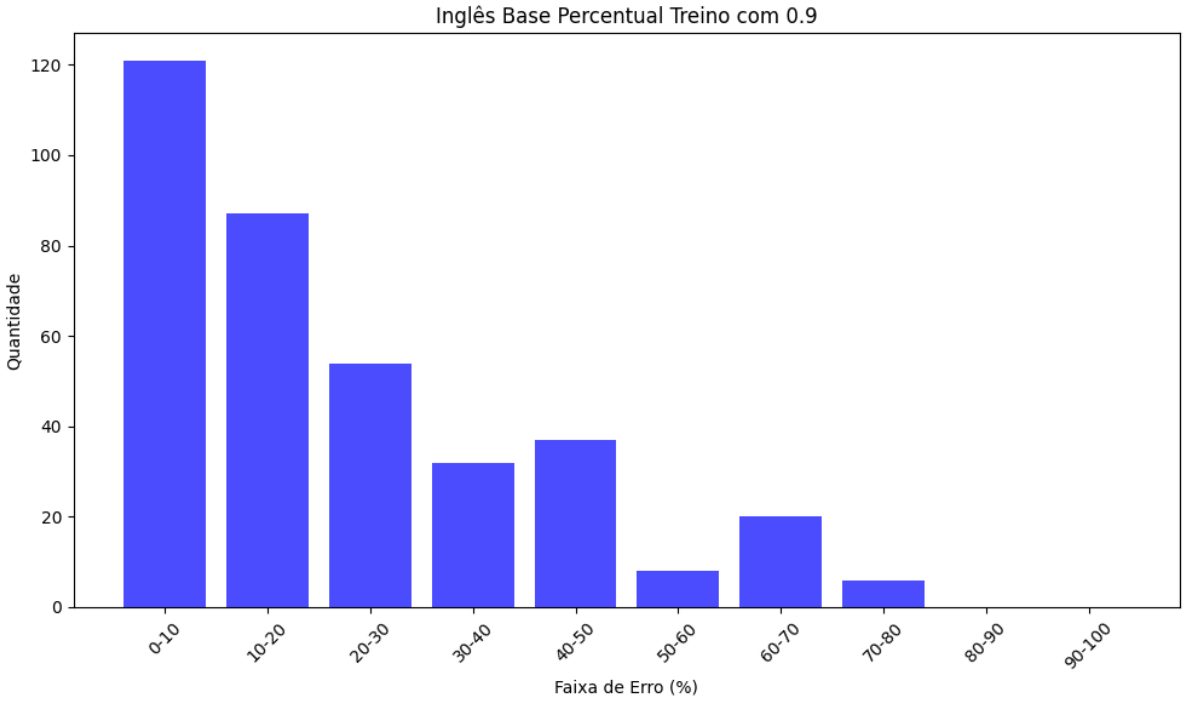

Figura 22 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Mohler* (Inglês) usando o Modelo *BERT Base*

5.1.6 Resultados do treinamento com o *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

Os resultados do treinamento com o *dataset Mohler* em inglês utilizando o modelo *BERT Large* indicam uma melhoria geral em comparação ao modelo *BERT Base*. A acurácia média varia de 79.72% a 81.63%, mostrando uma maior estabilidade e melhor desempenho com diferentes percentuais de dados de treinamento. Os valores de EQM e EMA também são consistentemente melhores, reforçando a eficácia do modelo *BERT Large*.

Percentual de dados para o treinamento	**Qtd. Treino**	**Qtd. Teste**	**Pesos [Fator 1, Fator 2, Fator 3]**	**EQM**	**EMA**	**Acurácia**
60%	2187	1459	[0.4473, 0.1388, 0.1930]	1.1667	0.8519	81.19%
70%	2552	1094	[0.4698, 0.1487, 0.1814]	1.1642	0.8451	81.63%
80%	2916	730	[0.4530, 0.1494, 0.1987]	1.1072	0.8240	80.68%
90%	3281	365	[0.4388, 0.1487, 0.1727]	1.1034	0.8237	79.72%

Tabela 8 – Resultados de Regressão para Diferentes Percentuais de Treino com o *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

Nos gfaficos das Figuras 23, 24, 25 e 26 é possível ver que a maiora das avaliações se mantém na faixa de acurácia acima de 80%, assim como no modelo *Base*.

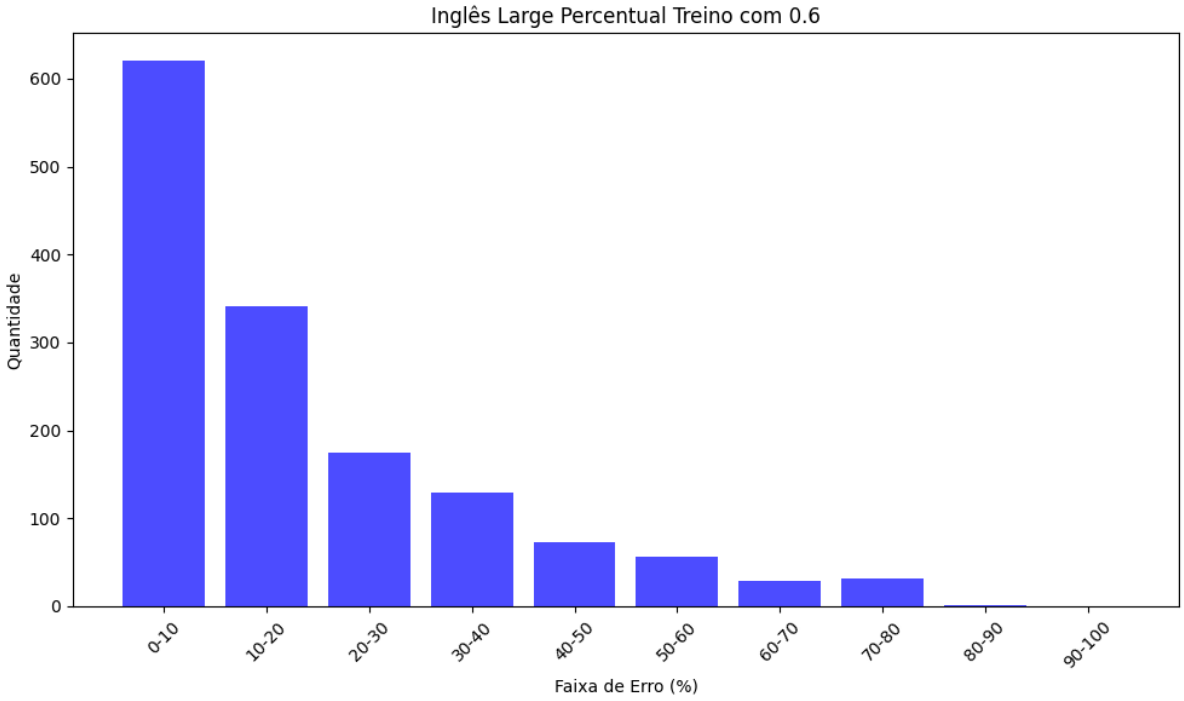

Figura 23 – Quantidade de respostas por faixas de erro percentual dos testes com 40% do *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

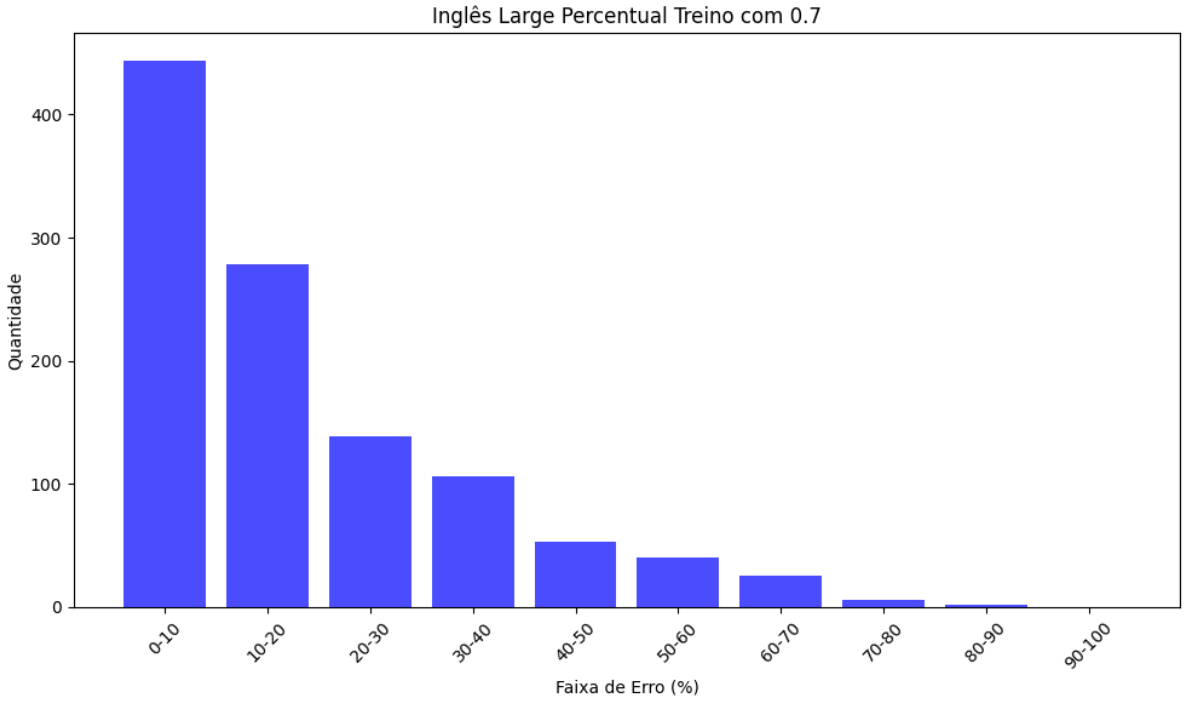

Figura 24 – Quantidade de respostas por faixas de erro percentual dos testes com 30% do *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

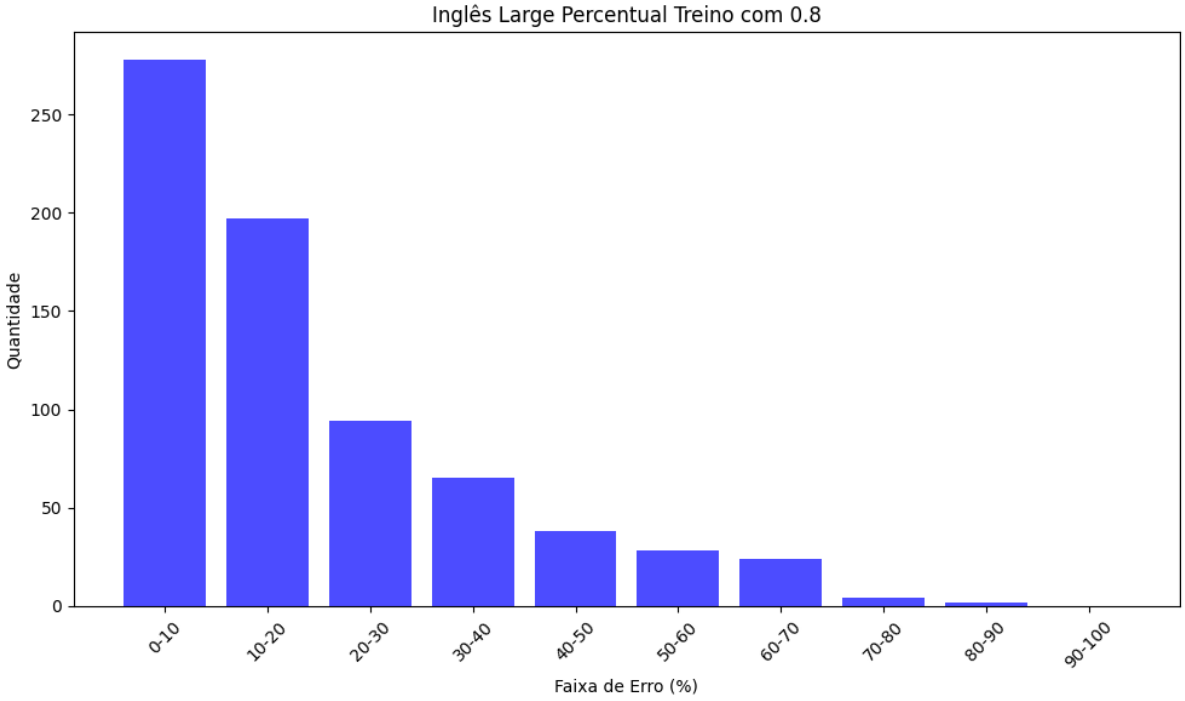

Figura 25 – Quantidade de respostas por faixas de erro percentual dos testes com 20% do *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

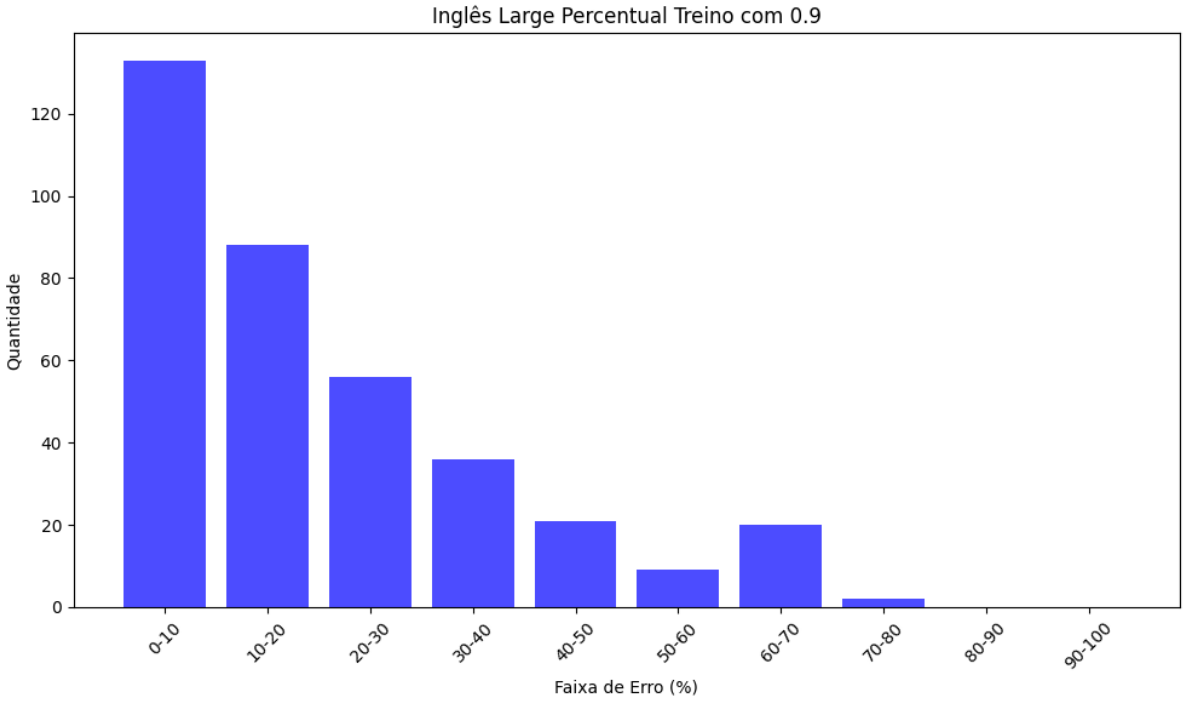

Figura 26 – Quantidade de respostas por faixas de erro percentual dos testes com 10% do *dataset Mohler* (Inglês) usando o Modelo *BERT Large*

5.1.7 Análise dos Resultados

Os resultados obtidos indicam uma melhoria no desempenho do algoritmo à medida que mais dados de treinamento são utilizados. No entanto, mesmo com uma quantidade substancial de dados de treinamento, ainda há espaço para melhorias.

5.1.7.1 Comparação entre os Conjuntos de Dados

Ao comparar os resultados entre os conjuntos de dados em português, inglês e espanhol, observa-se que:

a) Para o conjunto de dados em português, o melhor resultado obtido foi de 63,43%, enquanto para o conjunto de dados em espanhol, foi de 83.58%. Já para o conjunto de dados em inglês, a taxa de erro geral foi de 81.63%.

b) As diferenças nos pesos otimizados entre os três conjuntos de dados podem indicar variações nas características mais importantes para cada idioma e também na quantidade de dados disponíveis no *dataset* de cada um deles, o que pode ter afetado o resultado da regressão linear.

c) Outras tendências ou discrepâncias grandes em valores de avaliações, podem ser exploradas para entender melhor como o algoritmo entendeu o fator para aquele texto e buscar compreender o sentido por trás das notas geradas nesses caso. Em tipos específicos de respostas, um fator pode estar pesando mais do que deveria.

5.2 REPRODUTIBILIDADE DA EXPERIMENTAÇÃO

Para que o experimento realizado no presente trabalho tenha transparência e integridade na sua condução, todos os códigos-fonte desenvolvidos ao longo do processo, juntamento com os dados de treinamento, testes e resultados obtidos estão todos disponibilizados públicamente no Github[1]. Assim, qualque um pode executar os códigos para validação dos resultados apresentados neste trabalho ou, eventualmente, para comparação com outras possíveis abordagens propostas futuramente relacionadas a mesma temática deste trabalho.

[1] O repositório do Github pode ser acessado através do link: https://github.com/luca-moraes/relatorioParcialTCC

6 CONCLUSÃO

Levando em consideração tudo o que foi exposto ao longo de todos os capítulos, pode-se concluir, deste trabalho, que propôs uma abordagem para automatizar o processo de avaliação de respostas dissertativas, que os resultados obtidos apontam para uma direção interessante e promissora. Foi possível gerar uma grande quantidade de avaliações com resultados próximos aos de avaliações de professores. A métrica proposta, que levou em consideração a frequência de termos, distância de Levenshtein e similaridade semântica obtidas através de modelos BERT, visando simplificar o processo de correção manual para possibilitar a análise do desempenho dos alunos mais rapidamente, demonstrou seu funcionamento com resultados razoáveis e compreensíveis dentro dos limites do que este trabalho teve como proposta. Ao utilizar os *DataSets* de diferentes temáticas e idiomas, foi possível estimar pesos para os fatores que compõem a métrica de avaliação. Com eles, gerar avaliações automáticas e coletar os dados de acurácia da métrica, que apresentaram um resultado que pode ser considerado satisfatório para os fatores presentes e a quantidade de dados disponibilizados, além do tempo disponível para conclusão deste trabalho.

Algumas limitações foram observadas nos resultados, em que no caso de alguns *datasets*, foi possível ver uma quantidade grande de erros em faixas percentuais muito altas. Avaliações com uma diferença de valor alta quando comparadas com as avaliações dos professores. Uma explicação para muitos desses casos é a falta de compreensão de mais fatores relativos ao texto que o algoritmo ainda não possui a capacidade de avaliar.

Tendo isso em vista, como possibilidades de trabalhos futuros para melhoria dos resultados, algumas abordagens podem ser consideradas, como por exemplo:

a) **Inclusão de mais fatores no algoritmo**: Avaliar a inclusão de mais fatores ou características no algoritmo pode ajudar a capturar melhor a complexidade das respostas dissertativas em diferentes idiomas. Fatores como identificação de paráfrases e mais características da semântica do texto podem ajudar a alcançar uma métrica mais acertiva.

b) **Uso de outros modelos ou redes neurais**: Comparar o desempenho do algoritmo de regressão linear com outros modelos ou redes neurais pode fornecer boas considerações sobre qual abordagem é mais eficaz para determinação dos pesos.

Essas abordagens podem ajudar a aprimorar a precisão e a generalização do algoritmo para qualquer temática, tornando-o mais robusto e eficaz na avaliação de respostas dissertativas em português, inglês, espanhol ou até outros idiomas, eventualmente.

Em relação às questões de pesquisa levantadas para este trabalho, podemos considerar que:

a) Os parâmetros extraídos foram capazes de avaliar uma grande quantidade de respostas, embora com a complementação de mais fatores e mais dados para treinamento com outros modelos, os resultados podem ser melhorados.
b) Esses parâmetros foram capazes de definir uma pontuação que funcionou de maneira geral para avaliação de respostas dissertativas, embora houvessem limitações na acurácia da métrica.
c) Os fatores devem ser ponderados com pesos, que podem ser otimizados com um volume maior de dados ou outros algoritmos de inteligência artificial, como redes neurais, para determiná-los.
d) Podemos constatar que a acurácia da métrica variou consideravelmente comparando os idiomas e também o volume de dados para treinamento e teste. Isso fez com que os valores de acurácia do algoritmo fossem mais próximos quando treinados com os *datasets* em inglês e espanhol.

Em última análise, este trabalho almejou não apenas uma solução para as limitações atuais na avaliação de respostas dissertativas, mas também contribuir para a determinação de quais parâmetros são relevantes em uma métrica para esse problema de pesquisa. Com a determinação mais detalhada e profunda desses parâmetros em trabalhos futuros, abre-se a possibilidade do surgimento de uma contribuição significativa para os processos educacionais, proporcionando aos professores uma ferramenta valiosa para avaliação e acompanhamento do progresso dos alunos.

REFERÊNCIAS

AMUR, Zaira Hassan; HOOI, Yew Kwang; SOOMRO, Gul Muhammad. Automatic Short Answer Grading (ASAG) using Attention-Based Deep Learning MODEL, p. 1–7, 2022. DOI: 10.1109/ICDI57181.2022.10007187. Disponível em: https://ieeexplore.ieee.org/document/10007187.

BACHMAN, Lyle F. et al. A Reliable Approach to Automatic Assessment of Short Answer Free Responses. In: INTERNATIONAL Conference on Computational Linguistics. [S.l.: s.n.], 2002. Disponível em: https://api.semanticscholar.org/CorpusID:27889503.

BAGARIA, Vinal et al. An Intelligent System for Evaluation of Descriptive Answers, p. 19–24, 2020. DOI: 10.1109/ICISS49785.2020.9316110. Disponível em: https://ieeexplore.ieee.org/document/9316110.

BATANOVIĆ, Vuk; FURLAN, Bojan; NIKOLIĆ, Boško. A software system for determining the semantic similarity of short texts in Serbian, p. 1249–1252, 2011. DOI: 10.1109/TELFOR.2011.6143778. Disponível em: https://ieeexplore.ieee.org/document/6143778.

CAÑETE, José et al. Spanish Pre-Trained BERT Model and Evaluation Data. In: PML4DC at ICLR 2020. [S.l.: s.n.], 2020.

CER, Daniel Matthew et al. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation, 2017. Disponível em: https://api.semanticscholar.org/CorpusID:4421747.

DAI, Yange et al. Recognition of Parallelism Sentence Based on Recurrent Neural Network, p. 148–151, 2018. DOI: 10.1109/ICSESS.2018.8663734. Disponível em: https://ieeexplore.ieee.org/document/8663734.

DEVLIN, Jacob et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. **CoRR**, abs/1810.04805, 2018. arXiv: 1810.04805. Disponível em: http://arxiv.org/abs/1810.04805.

EISENSTEIN, Jacob. **Introduction to natural language processing**. [S.l.]: MIT press, 2019.

GALHARDI, Lucas; SOUZA, Rodrigo de; BRANCHER, Jacques. Automatic Grading of Portuguese Short Answers Using a Machine Learning Approach. In: ANAIS Estendidos do XVI Simpósio Brasileiro de Sistemas de Informação. Evento Online: SBC, 2020. P. 109–124. DOI: 10.5753/sbsi.2020.13133. Disponível em: https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/13133.

GUSTAFSON, Nathaniel; PERA, Maria Soledad; NG, Yiu-Kai. Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity. v. 1, p. 690–696, 2008. DOI: 10.1109/WIIAT.2008.16. Disponível em: https://ieeexplore.ieee.org/document/4740531.

JIANG, Yipeng; HAO, Yu; ZHU, Xiaoyan. A Chinese text paraphrase detection method based on dependency tree, p. 1–5, 2016. DOI: 10.1109/ICNSC.2016.7479003. Disponível em: https://ieeexplore.ieee.org/document/7479003.

KAUR, Amarjeet; SASIKUMAR, M. A comparative analysis of various approaches for automated assessment of descriptive answers. In: 2017 International Conference on Computational Intelligence in Data Science(ICCIDS). [S.l.: s.n.], 2017. P. 1–7. DOI: 10.1109/ICCIDS.2017.8272650. Disponível em: https://ieeexplore.ieee.org/document/8272650.

KUDI, Pooja et al. Online Examination with short text matching, p. 56–60, 2014. DOI: 10.1109/GCWCN.2014.7030847. Disponível em: https://ieeexplore.ieee.org/document/7030847.

LI, Quanzhi et al. Using paraphrases to improve tweet classification: Comparing WordNet and word embedding approaches, p. 4014–4016, 2016. DOI: 10.1109/BigData.2016.7841094. Disponível em: https://ieeexplore.ieee.org/document/7841094.

MAHMOUD, Adnen; ZRIGUI, Mounir. Arabic Semantic Textual Similarity Identification based on Convolutional Gated Recurrent Units, p. 1–7, 2021. DOI: 10.1109/INISTA52262.2021.9548576. Disponível em: https://ieeexplore.ieee.org/document/9548576.

MARDINI G., Ivan D. et al. A deep-learning-based grading system (ASAG) for reading comprehension assessment by using aphorisms as open-answer-questions. **Education and Information Technologies**, Kluwer Academic Publishers, USA, v. 29, n. 4, p. 4565–4590, jul. 2023. ISSN 1360-2357. DOI: 10.1007/s10639-023-11890-7. Disponível em: https://doi.org/10.1007/s10639-023-11890-7.

MARSLAND, Stephen. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2nd. [S.l.]: Chapman & Hall/CRC, 2014. ISBN 1466583282.

MEENA, K; LAWRANCE, R. Semantic similarity based assessment of descriptive type answers. In: 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16). [S.l.: s.n.], 2016. P. 1–7. DOI: 10.1109/ICCTIDE.2016.7725366. Disponível em: https://ieeexplore.ieee.org/document/7725366.

MOHLER, Michael; MIHALCEA, Rada. Text-to-text semantic similarity for automatic short answer grading. In: PROCEEDINGS of the 12th Conference of the European Chapter of the ACL (EACL 2009). [S.l.: s.n.], 2009. P. 567–575. Disponível em: https://www.kaggle.com/code/abdokamr/question-answering-with-t5.

OLIVEIRA, Katya Luciane de; SANTOS, Acácia Aparecida Angeli dos Santos. Avaliação da aprendizagem na universidade. pt. **Psicologia Escolar e Educacional**, scielopepsic, v. 9, p. 37–46, jun. 2005. ISSN 1413-8557. Disponível em: http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1413-85572005000100004&nrm=iso.

PAUL, Dimple V.; PAWAR, Jyoti D. Use of Syntactic Similarity Based Similarity Matrix for Evaluating Descriptive Answer, p. 253–256, 2014. DOI: 10.1109/T4E.2014.60. Disponível em: https://ieeexplore.ieee.org/document/7009583.

POULOS, Marios. Near duplicate text detection using graph depiction, p. 1–6, 2016. DOI: 10.1109/IISA.2016.7785368. Disponível em: https://ieeexplore.ieee.org/document/7785368.

RAMACHANDRAN, Lakshmi; GEHRINGER, Edward F. Determining Degree of Relevance of Reviews Using a Graph-Based Text Representation, p. 442–445, 2011. DOI: 10.1109/ICTAI.2011.72. Disponível em: https://ieeexplore.ieee.org/document/6103362.

ROWTULA, Vijay; OOTA, Subba Reddy; C.V, Jawahar. Towards Automated Evaluation of Handwritten Assessments, p. 426–433, 2019. DOI: 10.1109/ICDAR.2019.00075. Disponível em: https://ieeexplore.ieee.org/document/8977982.

SANUVALA, Ganga; FATIMA, Syeda Sameen. A Study of Automated Evaluation of Student's Examination Paper using Machine Learning Techniques, p. 1049–1054, 2021. DOI: 10.1109/ICCCIS51004.2021.9397227.

SCHLECHTWEG, Dominik et al. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. **ArXiv**, abs/2007.11464, 2020. Disponível em: https://api.semanticscholar.org/CorpusID:220686630.

SHAUKAT, Mohammad Shaharyar et al. Chapter 16 - Semantic similarity–based descriptive answer evaluation. Edição: Sarika Jain, Vishal Jain e Valentina Emilia Balas. Academic Press, p. 221–231, 2021. DOI: https://doi.org/10.1016/B978-0-12-822468-7.00014-6. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780128224687000146.

SOUZA, Fábio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: 9TH Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear). [S.l.: s.n.], 2020.

THALOR, Meenakshi A. A Descriptive Answer Evaluation System Using Cosine Similarity Technique. In: 2021 International Conference on Communication information and Computing Technology (ICCICT). [S.l.: s.n.], 2021. P. 1–4. DOI: 10.1109/ICCICT50803.2021.9510170. Disponível em: https://ieeexplore.ieee.org/document/9510170.

WANG, Chenglin et al. LSGC: An Interactive Text Matching Model Combined with Enhanced Encoding, p. 1–7, 2022. DOI: 10.1109/IJCNN55064.2022.9889792. Disponível em: https://ieeexplore.ieee.org/document/9889792.

ZHANG, Jing. Application Research of Similarity Algorithm in the Design of English Intelligent Question Answering System. In: 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC). [S.l.: s.n.], 2022. P. 1–4. DOI: 10.1109/ICMNWC56175.2022.10031708.

ZHANG, Shutao et al. Enhanced Text Matching Based on Semantic Transformation. **IEEE Access**, v. 8, p. 30897–30904, 2020. DOI: 10.1109/ACCESS.2020.2973206. Disponível em: https://ieeexplore.ieee.org/document/8993831.