Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Avaliação do Desempenho de Modelos LLaMA e Gemini na Correção de Redações do ENEM

Daniel Silva Lopes da Costa

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Prof. Dr. Denis Deratani Mauá
Cossupervisor: Prof. Msc. Igor Cataneo Silveira

São Paulo
2024

A todos que não ignoram o cego
mascando chiclete no bonde.

Agradecimentos

O correr da vida embrulha tudo,
a vida é assim: esquenta e esfria,
aperta e daí afrouxa,
sossega e depois desinquieta.
O que ela quer da gente é coragem
— Guimarães Rosa

A conclusão deste trabalho seria impossível sem o apoio e a dedicação de várias pessoas e instituições que estiveram comigo ao longo dessa jornada.

Aos meus pais e ao meu irmão, meu mais profundo agradecimento. Sua presença constante, paciência e incentivo foram fundamentais em cada etapa desta caminhada. A confiança e o amor que recebi de vocês me deram forças para enfrentar desafios e buscar sempre o melhor de mim.

Aos colegas de curso, agradeço pela parceria e pelas trocas enriquecedoras que tornaram esta experiência mais leve e produtiva. Aos professores e mentores, minha gratidão pela orientação, paciência e generosidade em compartilhar conhecimentos. Suas lições vão além das páginas dos livros e continuarão a me inspirar.

Por fim, gostaria de expressar meu reconhecimento à Universidade de São Paulo, uma instituição de excelência e gratuita, que possibilita a realização de sonhos como o meu. A USP é um exemplo de como a educação pública pode transformar vidas e contribuir para um futuro mais justo e promissor.

A todos, meu mais sincero obrigado. Este trabalho é também um reflexo do apoio e da confiança de cada um de vocês.

Resumo

Daniel Silva Lopes da Costa. **Avaliação do Desempenho de Modelos LLaMA e Gemini na Correção de Redações do ENEM**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

Este trabalho avalia o desempenho dos modelos de linguagem LLaMA e Gemini na tarefa de correção automática de redações do ENEM, focando na atribuição de notas para as cinco competências exigidas. A análise foi estruturada em duas fases principais: a exploração de diferentes padrões de prompts e a utilização dos modelos como insumos para algoritmos supervisionados de aprendizado de máquina.

Os resultados indicaram que ajustes no formato dos prompts, especialmente os padrões contextualizados e em cadeias de pensamento, podem melhorar significativamente a precisão e a consistência das respostas geradas pelos modelos. Além disso, as saídas desses modelos, combinadas com métricas derivadas por ferramentas como o NILC-Metrix, mostraram-se relevantes para a construção de modelos supervisionados mais robustos.

O trabalho também introduz um dataset estendido, com a inclusão de redações nota mil, para mitigar desequilíbrios nas distribuições de notas e enriquecer a base de dados disponível para a pesquisa. Apesar dos avanços, desafios como a escassez de redações com notas muito baixas e a necessidade de maior transparência nos modelos utilizados permanecem.

Contribuindo para o desenvolvimento de ferramentas mais eficientes e acessíveis, este estudo reforça o potencial dos grandes modelos de linguagem na transformação da avaliação educacional, apontando caminhos para pesquisas futuras em técnicas de aprendizado por supervisionado e engenharia de prompts.

Palavras-chave: Correção automática. ENEM. LLaMA. Gemini. Engenharia de Prompts. Aprendizado de Máquina.

Abstract

Daniel Silva Lopes da Costa. **Performance Evaluation of LLaMA and Gemini Models in Automated Grading of ENEM Essays**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

This study evaluates the performance of the LLaMA and Gemini language models in the automated grading of ENEM essays, focusing on scoring the five required competencies. The analysis was structured into two main phases: exploring different prompt patterns and leveraging the models as inputs for supervised machine learning algorithms.

The results indicated that adjustments in prompt formatting, particularly contextualized and chain-of-thought patterns, can significantly improve the precision and consistency of the models' outputs. Furthermore, the outputs of these models, combined with derived metrics from tools like NILC-Metrix, proved valuable for building more robust supervised models.

The study also introduced an extended dataset, including perfect-score essays, to mitigate imbalances in score distributions and enrich the available research dataset. Despite the advancements, challenges such as the scarcity of low-score essays and the need for greater transparency in the utilized models remain.

Contributing to the development of more efficient and accessible tools, this research reinforces the potential of large language models to transform educational assessment, paving the way for future studies in reinforcement learning techniques and prompt engineering.

Keywords: Automated Essay Scoring. ENEM. LLaMA. Gemini. Prompt Engineering. Machine Learning.

Lista de Abreviaturas

IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo
QWK	Quadratic Weighted Kappa
RMSE	Root Mean Square Error (Erro Médio Quadrático da Raiz)
PIA	Precisão do Intervalo Absoluto
AES	Avaliação Automática de Redações (*Automated Essay Scoring*)
LLM	Modelo de Linguagem de Grande Escala (*Large Language Model*)
ENEM	Exame Nacional do Ensino Médio
MEC	Ministério da Educação
NILC	Núcleo Interinstitucional de Linguística Computacional
GXBoost	Extreme Gradient Boosting
BrWaC	Corpus Brasileiro na Web (*Brazilian Web Corpus*)
IA	Inteligência Artificial (*Artificial Intelligence*)

Lista de Figuras

Lista de Tabelas

Lista de Programas

Sumário

Capítulo 1

Introdução

O Exame Nacional do Ensino Médio (ENEM) representa o maior teste realizado no âmbito nacional no Brasil, contando com milhões de participantes anualmente. Uma das etapas centrais dessa avaliação consiste na produção de uma redação argumentativa-dissertativa, cujo propósito é avaliar o domínio da língua, o senso crítico e o repertório dos estudantes. Nessa etapa, são disponibilizados textos e dados de apoio relacionados a um tema revelado durante o exame.

Diante dos avanços nos modelos de Inteligência artificial generativa, é válido perguntar: Qual seria a performace desses modelos na correção automática dessas redações? Seria esses modelos capazes extrair de forma quantitativa as determinadas competências exigidas aos estudantes? Esse trabalho pretende principalmente endereçar essas perguntas.

Com datasets de redações reais, será avaliada a capacidade dos Modelos Gemini do Google e Llama da Meta, de atribuir notas para cada competência. Serão implementadas duas fases de análise. Em primeiro lugar, diversos experimentos com diferentes prompts serão avaliados. Na sequência, o efeito das estimativas desses modelos, em outro modelo de aprendizado de máquina como uma feature adicional, também será avaliado.

1.1 Contextualização

No primeiro dias da prova, com 90 questões da área de humanidade e linguagem, o aluno precisa escrever um texto dissertativo argumentativo. Esse texto avaliado, sobre a luz de cinco competência básicas, que são elas:

- **Competência 1 - Domínio da norma culta da língua escrita:** O candidato deve demonstrar o domínio da língua portuguesa em sua modalidade padrão, respeitando as normas gramaticais e as convenções da escrita. São avaliados aspectos como ortografia, concordância verbal e nominal, regência, pontuação e adequação lexical.
- **Competência 2 - Compreensão da proposta de redação:** Aqui é avaliada a capacidade do candidato de interpretar corretamente o tema proposto, apresentando um texto que atenda à estrutura dissertativo-argumentativa. Além disso, é analisada a coerência temática e a relevância das ideias desenvolvidas.

- **Competência 3 - Seleção e organização de argumentos:** Nesta competência, é avaliada a habilidade do candidato de construir uma argumentação consistente, organizada e bem fundamentada. O texto deve apresentar informações, fatos, dados ou opiniões que sustentem os argumentos de maneira lógica e coesa.
- **Competência 4 - Demonstração de conhecimento dos mecanismos linguísticos:** O foco aqui está na coesão textual. O candidato deve demonstrar o uso correto de conectivos e articuladores que garantam a progressão e a continuidade das ideias no texto, contribuindo para a clareza e fluidez da redação.
- **Competência 5 - Elaboração de uma proposta de intervenção:** Essa competência avalia a capacidade do candidato de propor uma solução viável, detalhada e respeitosa para o problema discutido no tema da redação. A proposta deve considerar os agentes envolvidos e as possíveis consequências das ações sugeridas.

Essas competências são avaliadas em uma escala de 0 a 200 pontos cada, totalizando 1.000 pontos. O objetivo é verificar não apenas a habilidade de escrita, mas também o pensamento crítico, a capacidade de argumentação e a criatividade do candidato ao propor soluções para problemas sociais. As redações são corrigidas por dois avaliadores, e caso haja uma discrepância significativa entre as notas atribuídas, um terceiro avaliador entra em cena para revisão. É relevante ressaltar que esse processo é dispendioso e moroso, e por vezes injusto, uma vez que os professores podem cometer erros. Dessa forma, fica clara a importância do desenvolvimento de técnicas que permitam a automatização dessa tarefa, visando alcançar resultados ainda mais imparciais e eficientes.

1.2 Justificativa

Esse trabalho é justificável pelo seu impacto em três frentes principais: o avanço no entendimento de Grandes Modelos de Linguagem Conversacional (LLMs), a introdução de novas técnicas na correção automática de texto e, por fim, o impacto social e educacional para a democratização do aprendizado.

Grandes Modelos de Linguagem Conversacional: Os modelos baseados na arquitetura Transformer, especialmente os do tipo Encoder e Decoder, consolidaram-se como ferramentas poderosas no processamento de linguagem natural (PLN), particularmente após o lançamento do GPT-3 em 2020. Esse avanço representou um marco no campo da inteligência artificial, mas, apesar disso, ainda há muito a ser explorado em termos de otimização e compreensão desses modelos. Este trabalho contribui para esse campo emergente ao avaliar sistematicamente a eficácia de dois grandes modelos (LLaMA e Gemini) em tarefas complexas de correção de redações. A análise considera diferentes padrões de prompt, uma área de pesquisa que vem ganhando destaque por sua capacidade de influenciar significativamente o desempenho dos modelos. Estudos como este permitem não apenas medir a eficiência de cada modelo em tarefas específicas, mas também identificar os fatores que tornam os prompts mais eficientes e generalizáveis.

Correção automática de texto: A correção automática é uma área consolidada em idiomas como inglês, com ferramentas amplamente difundidas, mas ainda é insipiente no contexto da língua portuguesa. Um dos principais desafios está relacionado à escassez de

corpora robustos e bem anotados para treinamento e avaliação de modelos. Além disso, há uma limitada diversidade de técnicas exploradas para lidar com as particularidades do português brasileiro. Este trabalho aborda essas lacunas ao introduzir um novo dataset estendido, incluindo redações nota mil, e ao aplicar metodologias inovadoras, como engenharia de prompts e análise baseada em métricas específicas para avaliação de textos. Assim, ele avança tanto na criação de recursos quanto no desenvolvimento de estratégias que podem ser replicadas e ampliadas em estudos futuros.

Educação e impacto social: Segundo o Ministério da Educação (MEC), em 2024 mais de dois milhões de estudantes participaram do Exame Nacional do Ensino Médio (ENEM), sendo 1,6 milhões provenientes de escolas públicas. Para muitos desses alunos, a redação representa um dos maiores desafios, pois nem sempre têm acesso a suporte adequado para praticar e receber feedback estruturado. Ferramentas de correção automática, baseadas em modelos de linguagem, têm o potencial de democratizar o acesso à educação de qualidade, permitindo que alunos de diferentes contextos socioeconômicos possam revisar e aprimorar suas habilidades de escrita de forma independente. Além disso, os métodos aqui explorados podem ser utilizados não apenas por estudantes, mas também por instituições de ensino, professores e plataformas educacionais, promovendo uma abordagem mais inclusiva e escalável. Este trabalho reforça a importância da tecnologia como um catalisador para reduzir desigualdades e melhorar o aprendizado no Brasil.

1.3 Objetivo

O **objetivo** desta monografia é explorar a eficiência de **Grandes Modelos de Linguagem Conversacional** na correção automática de textos. Para tanto, serão utilizados os modelos **LLaMA** e **Gemini** na tarefa de revisão de redações do ENEM, fornecendo notas para cada uma das cinco competências.

Essa análise exploratória será dividida em duas partes:

Fase 1: Serão avaliados diferentes padrões de *prompt*, os quais serão comparados entre si, a fim de investigar o efeito de seis diferentes entradas sobre a eficiência dos modelos. Estudos têm mostrado que diferenças no formato da entrada podem ter um impacto significativo na capacidade do modelo de realizar a tarefa consistentemente. Grande parte dos padrões implementados é baseada no trabalho White *et al.*, 2023.

Fase 2: Na sequência, os modelos serão comparados com uma abordagem mais tradicional, que utiliza características do texto como entrada em uma rede neural. Para tanto, serão utilizados modelos consolidados para extrair informações do texto. O objetivo é **comparar** a eficiência dos modelos de linguagem com técnicas mais tradicionais usando redes neurais e também usar as notas dos modelos como parâmetros para analisar o impacto no algoritmo. Dessa forma, **pretende-se** explorar a capacidade dos modelos em extrair *features* mais complexas do texto. Essa estratégia é uma replicação estendida do trabalho Mizumoto e Eguchi, 2023.

Com isso, espera-se contribuir para o campo de correção automática de redações, bem como para a área de **engenharia de *prompts***, por meio da avaliação de diferentes *prompts* na tarefa de correção de texto.

Um objetivo adicional é a introdução de um novo *dataset* contendo todas as redações nota mil disponíveis publicamente. Atualmente, as fontes de dados de redações possuem poucos textos com nota máxima; a maioria dos resultados tende a valores médios de nota. Com a introdução desta nova base de dados, espera-se rebalancear o *dataset* original, contribuindo para análises mais próximas da realidade e com cobertura de uma maior diversidade de cenários.

Capítulo 2

Revisão literária

2.1 Processamento de Linguagem Natural (PLN)

2.2 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é uma subárea da inteligência artificial que estuda como as máquinas podem entender, interpretar e gerar a linguagem humana. O PLN envolve uma ampla gama de tarefas, desde a análise sintática até a compreensão semântica e a geração de texto. Alguns dos avanços recentes na área foram possibilitados pelo uso de modelos de aprendizado profundo, como redes neurais recorrentes (RNNs), redes de memória de longo prazo (LSTMs) e, mais recentemente, Transformers Vaswani, 2017.

A evolução do PLN pode ser dividida em três fases principais:

1. **Métodos baseados em regras**: Durante as décadas de 1950 e 1960, os sistemas de PLN eram construídos manualmente, utilizando gramáticas formais e dicionários. Esses métodos eram limitados devido à dificuldade de capturar a complexidade e a ambiguidade da linguagem natural Chomsky, 1957.

2. **Métodos estatísticos**: Nos anos 1990, com o aumento da disponibilidade de dados digitais e o avanço do poder computacional, surgiram métodos baseados em modelos probabilísticos. Algoritmos como os *Hidden Markov Models* (HMM) e *Naive Bayes* foram amplamente utilizados em tarefas como análise de sentimentos e tradução automática Jurafsky, 2000.

3. **Aprendizado profundo**: Nos últimos anos, o PLN foi revolucionado pelo uso de redes neurais profundas, que permitem o aprendizado de representações distribuídas das palavras (*word embeddings*) e da estrutura da linguagem. Modelos como *Word2Vec* Mikolov, 2013 e *GloVe* Pennington *et al.*, 2014 marcaram o início dessa fase, seguidos pelo advento dos *Transformers* e suas variantes, que atualmente dominam a pesquisa em PLN.

Com a introdução de arquiteturas como *BERT* Devlin, 2018 e *GPT* Radford, 2018, a

capacidade dos modelos de PLN de capturar o contexto e as dependências de longo alcance em textos melhorou significativamente. Esses avanços têm sido aplicados em diversas áreas, como geração de texto, tradução automática e sistemas de diálogo.

2.3 Tranformers

O modelo **Transformer** é uma arquitetura de *deep learning* introduzida por Vaswani, 2017 e amplamente utilizada em tarefas de Processamento de Linguagem Natural (PLN), visão computacional e outras áreas que envolvem dados sequenciais ou com dependências complexas. Sua principal característica é o uso de uma técnica de atenção chamada **self-attention**, que permite ao modelo aprender relações entre elementos de uma sequência independentemente de sua posição, superando as limitações de modelos como redes neurais recorrentes (RNNs) e LSTMs, que processam dados sequencialmente. Essa capacidade de atenção torna os Transformers extremamente eficientes para processar longas sequências de dados, pois permite o processamento paralelo das informações.

A arquitetura do Transformer é composta por duas partes principais: **encoder** e **decoder**. Ambas as partes utilizam múltiplas camadas de atenção e camadas *feed-forward*. No entanto, *encoders* e *decoders* desempenham papéis distintos dentro do modelo:

1. **Encoder**: A função do *encoder* é processar a entrada e gerar uma representação contextualizada da sequência, capturando as relações entre os elementos. O *encoder* é composto por várias camadas que aplicam *self-attention* e redes neurais *feed-forward* em paralelo. Durante esse processo, cada *token* da entrada "atenta" para todos os outros *tokens*, resultando em uma matriz de atenção que destaca as conexões entre diferentes partes da sequência. Em tarefas de tradução de idiomas, por exemplo, o *encoder* é responsável por codificar a sentença no idioma de origem em uma representação intermediária.

2. **Decoder**: A função do *decoder* é gerar a saída (por exemplo, uma sequência traduzida) a partir da representação fornecida pelo *encoder*. O *decoder* possui uma estrutura semelhante à do *encoder*, com camadas de *self-attention*, mas também inclui uma etapa de **cross-attention**, que permite ao *decoder* focar na representação intermediária gerada pelo *encoder*. Assim, o *decoder* utiliza a representação da entrada enquanto processa a saída já gerada, *token* por *token*, ao longo das camadas. Em tarefas de tradução, o *decoder* gera a sentença no idioma de destino com base na representação codificada e nos *tokens* já gerados até aquele momento.

A diferença essencial entre o *encoder* e o *decoder* é que o *encoder* foca apenas na entrada, enquanto o *decoder* processa tanto a entrada quanto a saída gerada, permitindo que ele seja utilizado de maneira *autoregressiva*. A arquitetura Transformer tem sido a base para muitos modelos de sucesso, como o *BERT* Devlin, 2018 (que usa apenas o *encoder*) e o *GPT* Radford, 2018 (que usa apenas o *decoder*), entre outros.

2.4 Correção automática de redações

Automated Essay Scoring (AES) é uma tecnologia que utiliza sistemas computacionais para avaliar e pontuar redações escritas com base em critérios predefinidos, como correção linguística, riqueza lexical, coerência, sintaxe e relevância semântica. Desde seu início há mais de 50 anos, o AES tem atraído a atenção de pesquisadores e profissionais, principalmente por sua capacidade de minimizar riscos associados à avaliação humana, como a subjetividade e a demora no processo de correção (Attali, 2013; Dikli, 2006).

Os primeiros sistemas de AES surgiram na década de 1960, com destaque para o Project Essay Grade (PEG) desenvolvido por Page, 1967. O PEG utilizava análise de regressão múltipla para prever pontuações com base em características mensuráveis do texto, como o comprimento médio das sentenças, número de preposições e de vírgulas. Apesar da inovação, esses sistemas foram criticados por focarem excessivamente em aspectos superficiais da linguagem, tornando-os suscetíveis a estratégias de manipulação, como o aumento arbitrário do número de palavras e pontuações (Attali, 2013; Dikli, 2006).

Nos anos 1990, avanços na área de Processamento de Linguagem Natural (NLP), permitiram o desenvolvimento de sistemas mais robustos. Um exemplo notável é o e-rater, criado pelo Educational Testing Service (ETS) em 1998. Este sistema combina métodos estatísticos e baseados em regras para analisar aspectos mais profundos do texto, incluindo sintaxe, morfologia e semântica. Estudos indicam que o e-rater apresenta alta confiabilidade e validade, sendo utilizado em exames de grande impacto, como o GRE e o TOEFL (Burstein *et al.*, 2013; Attali e Burstein, 2004).

Paralelamente, a funcionalidade de feedback automático presente em sistemas de AES impulsionou pesquisas em Avaliação Escrita Automatizada (AWE). Ferramentas como Criterion (baseada no e-rater) e MY Access! (do IntelliMetric) têm sido avaliadas em contextos educacionais, com foco na eficácia do feedback automático no desenvolvimento da escrita dos alunos (Cotos, 2014; Koltovskaia, 2020). Embora, alguns pesquisadores questionem a adequação dessas ferramentas às práticas pedagógicas (Condon, 2013), outros argumentam que elas auxiliam os aprendizes a revisar e melhorar seus textos (C.-M. Chen e Cheng, 2008; Dikli e Bleyle, 2014).

2.4.1 AES em Português

A avaliação automática de textos (AES) em Português, especialmente no contexto do ENEM, apresenta uma área em crescimento, mas ainda com lacunas significativas em termos de corpus e técnicas empregadas. De acordo com estudos recentes, como Amorim e Veloso, 2017 e Marinho *et al.*, 2022, o número limitado de corpora anotados e publicamente disponíveis para o português brasileiro é uma das principais barreiras para o avanço da área.

Os trabalhos mais proeminentes nessa área variam amplamente em suas abordagens. Por exemplo, Amorim e Veloso, 2017 utilizaram métodos de regressão linear com um conjunto de características específicas para prever as notas atribuídas a redações do ENEM, enquanto Marinho *et al.*, 2022 propuseram um corpus ampliado, o "Essay-BR", com mais de 2.000 redações anotadas, sendo o maior corpus público do Brasil para essa tarefa. Essas

iniciativas representam passos importantes para consolidar a pesquisa no Brasil.

Os estudos atuais mostram que ainda existe um espaço considerável para a exploração de técnicas mais avançadas, como o uso de Modelos de Linguagem de Grande Escala (LLMs). Apesar de esses modelos não terem sido amplamente empregados nessa tarefa ainda, eles oferecem oportunidades promissoras para superar limitações atuais, como a necessidade de grandes volumes de dados anotados. Por exemplo, os LLMs podem ser ajustados para contextos de baixa disponibilidade de dados, como demonstra o uso de técnicas de zero-shot ou few-shot learning.

Além disso, desafios específicos do ENEM, como a avaliação em múltiplas competências e a necessidade de transparência nos critérios de correção, continuam a oferecer questões abertas para a pesquisa. Estudos como o de Amorim et al. Amorim, Veloso e Araújo, 2018 destacam a importância de lidar com o viés humano nas avaliações e de incorporar métodos que possam analisar e mitigar esses efeitos.

Embora os sistemas de correção automática não substituam completamente os avaliadores humanos, sua integração ao longo das últimas décadas mostra grande potencial para complementar processos educacionais e avaliações de larga escala. Estudos continuam a explorar como equilibrar a objetividade dos algoritmos com as nuances qualitativas inerentes à escrita humana.

Capítulo 3

Ferramentas

3.1 Modelos

3.1.1 Gemini 1.5-Flash

O Gemini 1.5-Flash é um modelo de linguagem multimodal desenvolvido pela Google, projetado para oferecer alto desempenho com eficiência computacional em tarefas que envolvem texto, áudio, vídeo e imagens. Parte da família Gemini 1.5, o Flash é uma variante mais leve e otimizada para utilização eficiente de recursos, sem sacrificar a qualidade nas principais tarefas.

De acordo com Team, Georgiev *et al.*, 2024, o Gemini 1.5-Flash apresenta as seguintes características:

- **Janela de Contexto:** Suporte para até 2 milhões de tokens, permitindo o processamento de documentos extensos, análises multimodais longas e consultas complexas.
- **Desempenho Multimodal:** Capacidade de integrar texto, imagens, áudio e vídeo em um único fluxo de entrada, processando dados mistos de forma nativa.
- **Eficiência Computacional:** Projetado para execução em TPUs com baixa latência, utilizando técnicas de distilação online e cálculo paralelo para atenção e camadas feedforward.
- **Dados de Treinamento:** Treinado em uma base de dados multimodal abrangente, incluindo documentos da web, código, imagens, áudio e vídeo, com aplicação de filtros avançados de qualidade e segurança. De acordo com Team, Anil *et al.*, 2023

O Gemini 1.5-Flash mantém um desempenho competitivo em benchmarks comparado a modelos maiores, como o Gemini 1.5 Pro, sendo uma opção eficiente e ágil para aplicações que demandam processamento extensivo de dados multimodais e longos.

3.1.2 LLaMA-3.1 70B

O LLaMA-3.1 70B é um modelo de linguagem de última geração, parte da família de modelos desenvolvidos pela Meta AI. Este modelo possui 70 bilhões de parâmetros,

utilizando uma arquitetura Transformer densa com dimensões cuidadosamente ajustadas para balancear desempenho e eficiência computacional.

O modelo foi acessado por meio da API gratuita da Groq, uma empresa especializada em aceleração de inferência de cargas de trabalho de Inteligência Artificial (IA). A Groq desenvolve hardware específico para IA, como a *Language Processing Unit* (LPU), um circuito integrado projetado para otimizar o desempenho em tarefas que exigem processamento intensivo, como execução de grandes modelos de linguagem (LLMs). A plataforma GroqCloud possibilita que desenvolvedores testem modelos como o LLaMA com baixíssima latência e alta taxa de geração de tokens, graças à arquitetura determinística das LPUs.

O LLaMA-3.1 foi projetado para suportar uma ampla variedade de tarefas em Processamento de Linguagem Natural (PLN), como tradução de idiomas, geração de código e resolução de problemas complexos. De acordo com Dubey *et al.*, 2024 temos as seguintes especificações Técnicas do modelo:

- **Parâmetros:** 70 bilhões, organizados em 80 camadas com 64 cabeças de atenção.
- **Dimensão do Modelo:** 8.192, com dimensão da Rede Neural Feed-Forward de 28.672.
- **Janela de Contexto:** 128.000 tokens, incluindo suporte aprimorado para idiomas não-ingleses. Com isso, modelo pode lidar com documentos extensos e análises de prompts complexos. Essa característica é particularmente útil em tarefas que requerem contextualização de longo alcance, como resumo de documentos e respostas a perguntas técnicas detalhadas.
- **Dados de Treinamento:** Base multilinguística composta por 15T de tokens, incluindo 50% de conhecimento geral, 25% de dados matemáticos e de raciocínio, 17% de código e 8% de tokens multilinguísticos. Essa abordagem amplia sua capacidade de compreensão em domínios técnicos e linguísticos diversos.

A arquitetura e os dados de treinamento foram otimizados para maximizar a eficiência e a escalabilidade do modelo. O uso de filtros de alta qualidade no pré-processamento dos dados assegurou um conjunto de treinamento limpo e diversificado, enquanto a utilização de técnicas como rejeição amostral no ajuste fino aprimorou a capacidade de alinhamento do modelo com instruções humanas.

3.1.3 NILC Metrix

NILC-Metrix é uma ferramenta desenvolvida pelo Núcleo Interinstitucional de Linguística Computacional (NILC), com o objetivo de analisar aspectos de coesão, coerência e complexidade textual em português brasileiro. A ferramenta incorpora mais de 200 métricas, agrupadas em 14 categorias, que avaliam desde características básicas, como contagem de palavras e sentenças, até métricas mais complexas baseadas em análises sintáticas e semânticas. Ela evoluiu a partir do projeto Coh-Metrix-Port Scarton e Aluisio, 2010 e de outras iniciativas, como Adapt2Kids Hartmann e Aluísio, 2020 e PorSimples Candido Jr *et al.*, 2009, que contribuíram para a construção de modelos de predição de complexidade textual (S. E. Leal *et al.*, 2023).

Entre suas aplicações, destacam-se a análise de textos voltados para crianças, a predição de complexidade textual em narrativas e a adaptação automática de textos. Além disso, a ferramenta emprega recursos como o parser Palavras e o modelo LX-parser para análise linguística, utilizando grandes corpora, como BrWaC e Corpus Brasileiro, para treinar modelos de frequência lexical e análise semântica. Uma das inovações recentes é o uso de Análise Semântica Latente (LSA) para calcular coesão semântica, com base em corpora contendo centenas de milhões de palavras (AL., 2023, S. LEAL E TEAM, 2021).

NILC-Metrix possui código-fonte aberto, licenciado sob AGPLv3, e está disponível tanto para uso em interface web quanto para integração via API. Suas métricas têm sido amplamente utilizadas em estudos de processamento de linguagem natural, educação e ciências cognitivas, consolidando-se como uma ferramenta versátil para pesquisadores e desenvolvedores (S. LEAL E TEAM, 2021).

3.1.4 XGBoost

O XGBoost (eXtreme Gradient Boosting) é um algoritmo de aprendizado de máquina baseado em árvores de decisão e amplamente utilizado para resolver problemas de classificação e regressão devido à sua eficiência, flexibilidade e desempenho superior em competições de aprendizado de máquina. Ele é uma implementação otimizada do algoritmo de boosting baseada em gradientes, que utiliza técnicas como a paralelização e a poda para melhorar a performance e evitar overfitting T. CHEN E GUESTRIN, 2016.

Quando aplicado a tarefas de multiclassificação, o XGBoost utiliza estratégias como o *one-vs-all* ou *softmax objective*, dependendo da configuração. A estratégia *softmax*, em particular, é amplamente utilizada para lidar com múltiplas classes simultaneamente, ajustando o modelo para minimizar uma função de perda logarítmica generalizada. Isso permite ao XGBoost capturar eficientemente relações complexas entre as variáveis preditoras e as classes de destino FRIEDMAN, 2001. Além disso, o modelo oferece suporte para regularização (L1 e L2), o que ajuda a reduzir a complexidade do modelo em tarefas com muitas classes.

Outra vantagem do XGBoost em tarefas de multiclassificação é sua capacidade de lidar com conjuntos de dados desequilibrados, aplicando pesos diferentes às classes na função de perda. Esse recurso, aliado à capacidade de ajustar hiperparâmetros como a profundidade máxima das árvores (*max_depth*) e a taxa de aprendizado (*learning_rate*), permite ao modelo atingir alta acurácia mesmo em cenários desafiadores. Sua aplicação em problemas de multiclassificação inclui diagnósticos médicos, sistemas de recomendação e reconhecimento de padrões em imagens T. CHEN E HE, 2015.

3.2 Datasets

3.2.1 AES ENEM Dataset - Fonte A

AES ENEM Dataset I. C. SILVEIRA *et al.*, 2024 usado neste trabalho é um novo benchmark para avaliação automática de redações em português brasileiro. Ele foi criado para mitigar problemas como inconsistências de anotação, viés de dados e falta de documentação em

corpora anteriores. A seguir, apresenta-se uma descrição detalhada da Fonte A, que compõe o núcleo deste conjunto de dados:

Fonte A

A Fonte A consiste em 1629 redações coletadas entre agosto de 2015 e março de 2020. Estas redações foram extraídas de um website que simulava a avaliação do Exame Nacional do Ensino Médio (ENEM). Para cada mês no período mencionado, um novo tema de redação, acompanhado de textos de apoio, era disponibilizado, e os textos submetidos no mês anterior eram anotados. No entanto, devido a problemas como escalas de avaliação inconsistentes e formatos diferentes de proposta textual (por exemplo, textos no formato de carta), 474 redações foram descartadas durante o processo de limpeza, resultando no conjunto final.

Cada redação na Fonte A é anotada com:

- **Pontuação por Competência**: Avaliações detalhadas em cinco competências, conforme o modelo do ENEM (como fluência linguística, argumentação e uso adequado de conectores textuais), com valores possíveis de {0, 40, 80, 120, 160, 200}.
- **Comentários por Competência**: Explicações qualitativas fornecidas pelos anotadores para justificar a pontuação atribuída.
- **Propostas de Redação e Textos de Apoio**: As propostas de redação e os textos de apoio correspondentes, apresentados separadamente.

Além disso, dois especialistas independentes reavaliaram um subconjunto de 200 redações desta fonte. Essa reavaliação possibilitou medir a concordância entre os anotadores originais e os especialistas, destacando a qualidade das anotações. Foi observado que as redações da Fonte A apresentavam uma distribuição de notas mais próxima do padrão oficial do ENEM, com menor viés em relação a outras fontes.

Na Tabela 3.1, temos a divisão do dataset com os conjuntos de treino, teste e validação.

Divisão dataset	**Número de redações**
Treino	744
Teste	216
Validação	195
Total	**1155**

Tabela 3.1: *Divisão Dataset: Corpus Essay-BR - Fonte A*

3.2.2 Redações Nota Mil

Foram coletadas todas as redações nota mil disponíveis em fontes públicas. Parte dos textos foi obtida manualmente de reportagens dos veículos de comunicação O Globo Globo, 2019 e G1 g1, 2024. Além disso, outras redações foram extraídas de cartilhas para estudantes disponibilizadas pelo engenheiro de software Lucas Felpi Felpi, 2019.

A extração dos textos das notícias foi realizada manualmente. Para os arquivos PDF fornecidos por Felpi, 2019, utilizou-se o modelo Gemini para processar e identificar os textos, convertendo-os para o formato de texto. Todo o processo foi validado por um avaliador humano para garantir a precisão.

Com os textos coletados, utilizou-se a métrica de Levenshtein para remover textos duplicados. Como as redações estavam disponíveis em diferentes formatos, poderiam existir variações causadas por espaçamentos distintos ou símbolos adicionais. A métrica de Levenshtein foi empregada para identificar e eliminar essas duplicidades.

A métrica de Levenshtein, também conhecida como distância de edição, mede a similaridade entre dois textos ao calcular o número mínimo de operações necessárias para transformar uma string em outra. Essas operações incluem inserções, deleções e substituições de caracteres Levenshtein, 1966. Uma variação amplamente utilizada é a razão de Levenshtein (*Levenshtein Ratio*), que normaliza o valor da distância, representando-o como uma proporção entre 0 e 1:

$$\text{Levenshtein Ratio} = \frac{\text{Comprimento da String} - \text{Distância de Levenshtein}}{\text{Comprimento da String}} \tag{3.1}$$

Essa métrica é amplamente usada em áreas como correção ortográfica, sistemas de recomendação e detecção de plágio Navarro, 2001. Para este trabalho, utilizou-se a biblioteca `python-Levenshtein` Zverovich, 2023 para calcular a distância entre os textos, removendo todos aqueles cuja razão de Levenshtein fosse maior que 0.2.

Após a remoção de duplicatas, foi gerada uma versão consolidada do dataset. Este dataset poderá ser usado futuramente em tarefas de classificação binária, para determinar se uma redação obteve nota máxima ou não. No contexto deste trabalho, as redações nota mil foram incorporadas ao dataset AES ENEM Dataset - Fonte A, com o objetivo de balancear as classes e aumentar o número de redações com nota 200 em cada competência.

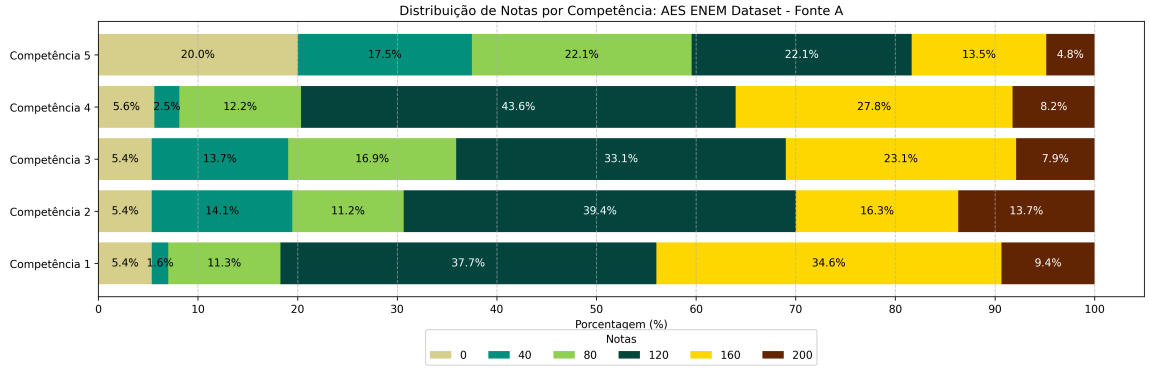
As redações nota mil coletadas e utilizadas neste trabalho estão disponíveis em um repositório público no GitHub, podendo ser acessadas por meio do link https://github.com/dslcosta1/EssaysENEM1000GradeDataset. O repositório inclui todos os textos utilizados, devidamente processados e organizados, permitindo sua reutilização em pesquisas futuras e validação dos resultados apresentados neste trabalho.

A Tabela 3.2 apresenta a divisão do dataset em conjuntos de treino, teste e validação, seguindo as mesmas proporções do dataset original.

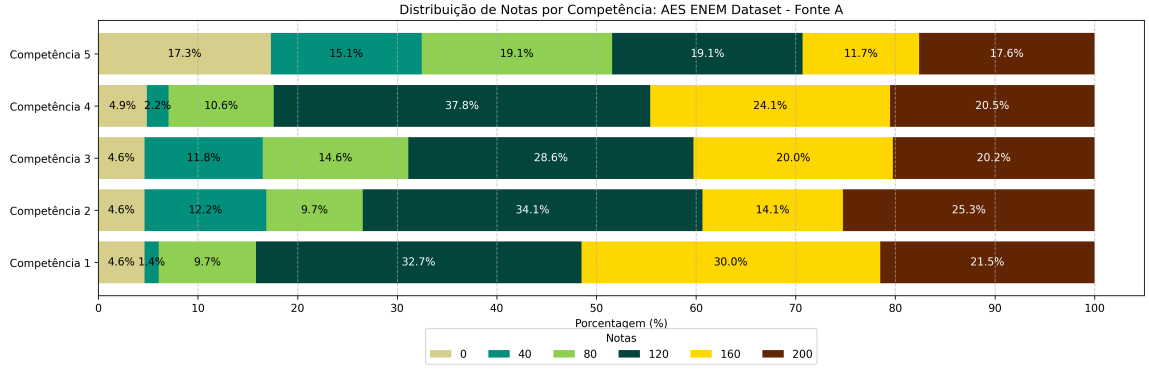
Divisão do Dataset	**# Redações Nota Mil**	**# Redações Nota Mil + Essay-BR**
Treino	129	873
Teste	27	243
Validação	23	219
Total	**179**	**1334**

Tabela 3.2: *Divisão do Dataset: Redações Nota Mil de 2010 a 2023*

Já na Figura 3.1, são apresentados os gráficos com as distribuições de notas por competência no dataset original e no dataset estendido, que inclui as redações nota mil. Observa-se que a inclusão dessas redações resultou em uma distribuição mais balanceada para notas altas, corrigindo parcialmente um dos problemas do dataset original: a escassez de redações com nota 200. Contudo, o problema da ausência de textos com notas 0 e 40 persiste, podendo ser abordado em trabalhos futuros.



(a) *Distribuição de Nota por Competência: Dataset AES ENEM Dataset - Fonte A*



(b) *Distribuição de Nota por Competência: Dataset AES ENEM Dataset - Fonte A + Redações Nota Mil*

Figura 3.1: *Comparação das Distribuições de Notas por Competência*

Por simplicidade, quando nos referirmos ao Dataset AES ENEM Dataset - Fonte A + Redações Nota Mil, utilizaremos o termo "Dataset Estendido". Todos os experimentos apresentados nesse trabalho usaram esse Dataset Estendido.

3.3 Métricas

3.3.1 Quadratic Weighted Kappa

A métrica **Quadratic Weighted Kappa (QWK)** é uma medida de concordância que quantifica o nível de similaridade entre duas classificações categóricas ordinais. É frequentemente utilizada para avaliar a correspondência entre as previsões de um modelo e os rótulos de referência (como classificações fornecidas por especialistas). Ao contrário de outras métricas de concordância, como a Kappa simples, a QWK atribui pesos quadráticos às discordâncias, penalizando mais fortemente as discrepâncias maiores entre as categorias.

Essa característica torna a QWK particularmente útil em tarefas de classificação ordinal, onde a ordem das categorias é significativa.

Formalmente, o cálculo da **QWK** é dado pela seguinte fórmula:

$$\kappa = 1 - \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}} \tag{3.2}$$

onde:

- $O_{i,j}$ representa a matriz de confusão observada normalizada entre as categorias i e j;
- $E_{i,j}$ representa a matriz de confusão esperada sob a hipótese de independência entre as classificações;
- $W_{i,j}$ é uma matriz de pesos quadráticos definida por:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{3.3}$$

Nesse contexto, N é o número total de categorias possíveis. A QWK varia entre -1 e 1, onde:

- $\kappa = 1$: indica concordância perfeita;
- $\kappa = 0$: indica concordância esperada pelo acaso;
- $\kappa < 0$: indica concordância pior do que o acaso.

A Tabela 3.3 apresenta os intervalos de valores da QWK e seus significados correspondentes.

Intervalo de QWK	**Significado**
$0{,}81 - 1{,}00$	Concordância quase perfeita
$0{,}61 - 0{,}80$	Concordância substancial
$0{,}41 - 0{,}60$	Concordância moderada
$0{,}21 - 0{,}40$	Concordância razoável
$0{,}00 - 0{,}20$	Concordância leve
$< 0{,}00$	Discordância (pior que o acaso)

Tabela 3.3: *Intervalos de QWK e seus significados*

Essa métrica foi originalmente proposta por Cohen Cohen, 1968.

3.3.2 Média do Error Quadrático

A métrica **Root Mean Square Error (RMSE),** ou Erro Quadrático Médio da Raiz, é amplamente utilizada para avaliar a precisão de modelos de previsão, especialmente em regressão. Ela mede a média das diferenças quadráticas entre os valores previstos e os valores reais, oferecendo uma indicação de quão distante, em média, estão as previsões em

relação aos valores observados. Uma característica importante da RMSE é que, ao elevar as diferenças ao quadrado, penaliza-se mais fortemente grandes erros, o que a torna sensível a discrepâncias maiores entre previsões e observações.

A RMSE é definida pela fórmula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.4}$$

onde:

- n representa o número total de observações,
- y_i é o valor observado para a i-ésima instância,
- $\hat{y}_i$ é o valor previsto para a i-ésima instância.

Valores de RMSE mais baixos indicam previsões mais próximas dos valores observados, enquanto valores mais altos indicam maiores discrepâncias.

3.3.3 Acurácia

A métrica **Acurácia** é uma medida amplamente utilizada para avaliar o desempenho de modelos em tarefas de classificação. Ela é definida como a proporção de predições corretas em relação ao total de predições realizadas. No contexto deste trabalho, a acurácia foi calculada considerando a correspondência exata entre as notas previstas pelo modelo e as notas reais de referência.

Formalmente, a acurácia é definida pela fórmula:

$$\text{Acurácia} = \frac{\text{Número de predições corretas}}{\text{Número total de predições}} \tag{3.5}$$

No código implementado, a acurácia foi calculada para cada conjunto de notas, utilizando a função `accuracy_score` da biblioteca `scikit-learn`.

Resultados mais próximos de 1 indicam que o modelo possui maior precisão na previsão exata das notas.

3.3.4 Precisão do Intervalo Absoluto (PIA)

A métrica **Precisão do Intervalo Absoluto (PIA)** é uma medida desenvolvida para avaliar a proximidade entre as notas previstas pelo modelo e as notas reais em um intervalo de tolerância predefinido. Essa métrica considera uma predição como precisa se a diferença absoluta entre a nota prevista ($\hat{y}$) e a nota real (y) for menor ou igual à tolerância especificada.

O cálculo da PIA segue os seguintes passos:

1. Calcular a diferença absoluta entre as notas reais e as notas previstas:

$$\text{Diferença Absoluta} = |y - \hat{y}| \tag{3.6}$$

2. Contar o número de predições em que a diferença absoluta é menor ou igual à tolerância definida (τ):

$$\text{PIA} = \frac{\text{Número de predições precisas}}{\text{Número total de predições}} \times 100 \tag{3.7}$$

No código implementado, a PIA foi calculada para cada conjunto de notas, utilizando uma tolerância padrão de 80. A métrica é especialmente útil para cenários em que pequenas discrepâncias entre as notas reais e previstas são aceitáveis, proporcionando uma avaliação mais flexível do desempenho do modelo.

O valor 80 foi utilizado como tolerância padrão, pois é o valor adotado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) para detectar divergências entre dois corretores reais. Se a diferença na nota de uma dada competência for maior que 80, um terceiro avaliador revisa o texto. Por conta disso, utilizou-se o mesmo valor para mensurar em quantos casos ocorre essa divergência significativa entre as notas do corretor real e a nota predita pelo modelo.

3.3.5 Teste de Kruskal-Wallis

O teste de **Kruskal-Wallis** é um teste estatístico não paramétrico utilizado para determinar se existem diferenças significativas entre as medianas de dois ou mais grupos independentes. Ele é particularmente útil quando os dados não atendem aos pressupostos de normalidade ou homogeneidade de variâncias, requeridos por testes paramétricos como a ANOVA.

Formalmente, o teste de Kruskal-Wallis calcula a estatística de teste H com base nos postos dos dados, em vez dos valores brutos, e segue a seguinte hipótese:

- H_0: As medianas dos grupos são iguais.
- H_1: Pelo menos uma mediana é diferente.

A estatística do teste H é definida como:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1) \tag{3.8}$$

onde:

- N: número total de observações;
- k: número de grupos;
- n_i: número de observações no grupo i;

- R_i: soma dos postos do grupo i.

O resultado do teste é acompanhado por um valor-p (p) que indica a significância estatística do resultado. No código implementado, o teste de Kruskal-Wallis foi aplicado a cada competência avaliada, gerando uma estatística H e o correspondente valor-p.

—

Valor-p e seu significado

O **valor-p** (*p-value*) é uma medida estatística que indica a probabilidade de obter resultados tão extremos quanto os observados, assumindo que a hipótese nula (H_0) é verdadeira. Em termos práticos, ele é usado para decidir se rejeitamos ou não a hipótese nula em testes de significância.

Os critérios gerais para interpretar o valor-p são: - Se $p < 0.05$: Há evidências suficientes para rejeitar a hipótese nula, indicando que há uma diferença estatisticamente significativa entre os grupos analisados. - Se $p \geq 0.05$: Não há evidências suficientes para rejeitar a hipótese nula, indicando que não foi encontrada diferença estatisticamente significativa.

No contexto deste trabalho, um valor-p menor que 0.05 sugere que há diferenças significativas entre os grupos comparados (por exemplo, as competências avaliadas).

Capítulo 4

Metodologia

4.1 Fase 1 - Análise e Engenharia de Prompts

Os padrões de prompt, conforme descrito por White *et al.*, 2023, são soluções reutilizáveis para problemas recorrentes na interação com modelos de linguagem. Cada padrão estabelece diretrizes específicas para a formatação das interações e dos outputs, promovendo maior eficiência e personalização. Inspirados em padrões de software, esses prompts ajudam a documentar e transferir conhecimento em diversos contextos, permitindo um uso mais estratégico e adaptado dos modelos de linguagem. Nesta seção, implementamos e avaliamos diferentes padrões de prompt para correção de redações do ENEM, explorando sua eficácia em múltiplas configurações.

É importante destacar que todos os modelos foram utilizados no modo *zero-shot*, ou seja, os modelos realizaram as tarefas solicitadas sem nenhum treinamento ou ajuste prévio especificamente para o contexto do ENEM. Esse tipo de abordagem avalia diretamente a capacidade dos modelos em lidar com instruções explícitas baseadas em sua base de treinamento geral. Além disso, todas as interações com os modelos foram realizadas através de chamadas às APIs oficiais disponibilizadas pelas empresas detentoras dos modelos LLaMA e Gemini.

4.1.1 Experimento 1 - Prompt Base

Este experimento utilizou o padrão "Template", que define uma estrutura fixa e direta para o output. A simplicidade do prompt base visa avaliar como o modelo performa sem ajustes detalhados ou adição de informações contextuais, confiando exclusivamente em sua base de treinamento prévia.

Experimento 1 - Prompt A

Corrija a seguinte redação, seguindo o método de avaliação do ENEM. Entregue apenas a nota quantitativa entre 0 e 200 para cada competência e a nota final no seguinte formato: [x, y, z, w, t, x+y+z+w+t]. Não quero texto na resposta, apenas as notas numéricas para cada competência e o total.

Tema:<TEMA>
Textos de suporte: <SUPORTE>
Redação: <REDAÇÃO>

4.1.2 Experimento 2 - Prompt Base em Inglês

Este experimento seguiu o mesmo padrão de "Template"do Experimento 1, porém com o prompt traduzido para o inglês. O objetivo foi investigar o impacto do idioma das instruções no desempenho dos modelos multilíngues, considerando que muitos deles possuem maior treinamento em dados de língua inglesa. Esta abordagem avalia se o uso de prompts em inglês pode melhorar a precisão e consistência das respostas geradas.

Experimento 2 - Prompt A

Correct the following writing, following the ENEM assessment method. Only deliver the quantitative grade between 0 and 200 for each competency and the final grade in the following format: [x, y, z, w, t, x+y+z+w+t]. I don't want text in the answer, just the numerical grades for each competency and the total.

Theme: <TEMA>
Supporting texts: <SUPORTE>
Writing: <REDAÇÃO>

4.1.3 Experimento 3 - Prompt Base Melhorado

Este experimento introduziu aprimoramentos estruturais ao prompt base do Experimento 1, utilizando a ferramenta **PromptPerfect**. O **PromptPerfect** é uma plataforma projetada para otimizar prompts utilizados em interações com modelos de linguagem natural, como GPTs e outros grandes modelos de linguagem. Ele oferece sugestões para aumentar a clareza, reduzir ambiguidades e estruturar os prompts de maneira eficiente, com base em boas práticas observadas no uso prático de modelos generativos. Mais informações sobre a ferramenta estão disponíveis no site oficial: https://promptperfect.jina.ai/ (acessado em 30 de novembro de 2024).

No contexto deste trabalho, o **PromptPerfect** foi utilizado para sugerir melhorias específicas ao prompt base. As mudanças implementadas incluem uma separação clara entre os textos motivadores, o tema e a redação, além de um detalhamento mais preciso do formato esperado para o output. O objetivo foi avaliar se essas modificações poderiam melhorar a consistência e a precisão das respostas geradas pelos modelos, mantendo-se fiel ao padrão base.

Experimento 3 - Prompt A

Corrija a redação abaixo seguindo o método de avaliação do ENEM. Forneça apenas as notas quantitativas entre 0 e 200 para cada competência, e a nota final, no seguinte formato:[C1,C2,C3,C4,C5,Total], onde "Total"é a soma das notas das cinco competências.

Tema:
“““
<TEMA>
”””

Textos de suporte:
“““
<SUPORTE>
”””

Redação:
“““
<REDAÇÃO>
”””

Não forneça texto na resposta, apenas as notas numéricas para competência e o total.

4.1.4 Experimento 4 - Prompt Persona

O padrão "Persona"consiste em configurar o modelo para adotar um papel específico, como o de uma profissão ou personalidade, com o objetivo de influenciar a forma e o tom da resposta gerada. Este experimento explorou a capacidade do modelo de incorporar a perspectiva de um professor de ensino médio especializado na correção de redações do ENEM. A hipótese era de que o uso de uma persona alinhada ao contexto da tarefa poderia melhorar a adequação e a precisão das respostas, especialmente ao integrar terminologias e práticas específicas do domínio educacional.

Por meio dessa técnica, buscou-se não apenas moldar o tom das respostas, mas também simular o comportamento de um avaliador humano experiente, capaz de interpretar os critérios de avaliação com maior fidelidade.

Experimento 4 - Prompt A

Como um professor de ensino médio especializado na correção de redações do ENEM, corrija a redação abaixo seguindo o método de avaliação do ENEM. Forneça apenas as notas quantitativas entre 0 e 200 para cada competência, e a nota final, no seguinte formato: [C1,C2,C3,C4,C5,Total], onde "Total"é a soma das notas das cinco competências.

Tema:
“““
<TEMA>
”””

Textos de suporte:
“““
<SUPORTE>
”””

Redação:
“““
<REDAÇÃO>
”””

Não forneça texto na resposta, apenas as notas numéricas para competência e o total.

4.1.5 Experimento 5 - Prompt Cadeia de Pensamento

O padrão "Cadeia de Pensamento"(*Chain-of-Thought*) explora a capacidade dos modelos de linguagem de dividir problemas complexos em etapas menores e sequenciais, facilitando a geração de respostas mais precisas e coerentes. Estudos recentes, como o de Wei *et al.*, 2022, demonstram que essa abordagem melhora significativamente o desempenho em tarefas que requerem raciocínio lógico ou análises detalhadas. Neste experimento, a correção de redações do ENEM foi estruturada em três etapas principais: compreensão do método de avaliação, correção detalhada com justificativas e, por fim, a geração do output final em formato padronizado.

A abordagem em etapas foi adotada para promover maior transparência no raciocínio do modelo e para avaliar se a divisão do processo de avaliação poderia resultar em uma melhoria na qualidade das notas geradas. O experimento foi implementado utilizando três prompts distintos, cada um com um objetivo específico.

Prompt A: Compreensão do método de avaliação

A primeira etapa visa assegurar que o modelo compreenda os critérios oficiais utilizados para a correção das redações do ENEM. Ao solicitar que o modelo descreva o método de avaliação, esta fase busca alinhar a interpretação do modelo às diretrizes oficiais, reduzindo possíveis ambiguidades e garantindo consistência nas etapas subsequentes.

Experimento 5 - Prompt A

Qual o método de avaliação utilizado na correção de redações do ENEM?

Prompt B: Correção detalhada com justificativas

Na segunda etapa, o modelo é solicitado a realizar a correção da redação, atribuindo notas para cada uma das cinco competências do ENEM, acompanhadas de justificativas detalhadas para cada nota. Este passo permite avaliar a capacidade do modelo de interpretar os critérios de avaliação e de produzir explicações consistentes para as notas atribuídas. Além disso, a explicitação das justificativas pode ser útil tanto para validar o processo quanto para identificar possíveis falhas ou inconsistências no raciocínio do modelo.

Experimento 5 - Prompt B

Corrija a redação abaixo seguindo o método de avaliação do ENEM. Atribua notas quantitativas entre 0 e 200 para cada uma das cinco competências e forneça uma explicação detalhada que justifique cada nota.

Tema:
“““
<TEMA>
”””

Textos de suporte:
“““
<SUPORTE>
”””

Redação:
“““
<REDAÇÃO>
”””

Prompt C: Output final no formato específico

A etapa final consolida os resultados das fases anteriores, fornecendo apenas o output final no formato padronizado [C1, C2, C3, C4, C5, Total], onde "Total"representa a soma das notas das cinco competências. Este prompt avalia a capacidade do modelo de seguir instruções precisas e de produzir uma saída objetiva e formatada de acordo com o esperado.

Experimento 5 - Prompt C

Com base nas respostas anteriores, forneça apenas as notas quantitativas entre 0 e 200 para cada competência, e a nota final, no seguinte formato: [C1,C2,C3,C4,C5,Total], onde "Total"é a soma das notas das cinco competências. Não forneça texto na resposta, apenas as notas numéricas.

4.1.6 Experimento 6 - Prompt Contextualizado

Neste experimento, investigou-se o impacto de enriquecer o prompt com informações detalhadas sobre a tarefa, prática conhecida como "Contextual Prompting". Segundo Zhou *et al.*, 2023, o "Contextual Prompting"refere-se ao fornecimento de informações adicionais relevantes e específicas ao modelo para melhorar sua capacidade de compreender e executar a tarefa. Essa técnica é particularmente eficaz em contextos que demandam precisão e fidelidade, como tarefas de avaliação ou diagnóstico, pois ajuda o modelo a alinhar suas respostas às expectativas do usuário.

No contexto deste experimento, foram incluídas descrições detalhadas das cinco competências avaliadas na redação do ENEM, bem como os critérios para atribuição de notas em cada uma delas. A hipótese era de que, ao receber informações explícitas, o modelo seria capaz de produzir respostas mais fundamentadas e consistentes.

Experimento 6 - Prompt A Parcial

Conheça as cinco competências cobradas pelo Inep na redação do ENEM:

1. Domínio da escrita formal da língua portuguesa
... [conteúdo resumido das competências e critérios para pontuação] ...

Corrija a redação abaixo seguindo o método de avaliação do ENEM. Forneça apenas as notas quantitativas entre 0 e 200 para cada competência, e a nota final, no seguinte formato: [C1,C2,C3,C4,C5,Total], onde "Total"é a soma das notas das cinco competências.

Tema:
“““
<TEMA>
”””

Textos de suporte:
“““
<SUPORTE>
”””

Redação:
“““

<REDAÇÃO>
"""

Não forneça texto na resposta, apenas as notas numéricas para competência e o total.

As descrições detalhadas fornecidas no prompt tiveram como objetivo aumentar a clareza e a especificidade das respostas do modelo, alinhando-se aos princípios descritos em Zhou *et al.*, 2023, que destacam a importância de enriquecer prompts com contexto explícito para melhorar a qualidade das interações com modelos de linguagem.

Além dos padrões específicos descritos acima, um padrão geral de formatação foi aplicado a todos os prompts. Isso inclui a padronização do formato de output, com foco em respostas estruturadas e específicas, como listas de notas quantitativas no formato esperado. Este cuidado busca garantir consistência e clareza nos resultados gerados pelos modelos.

O prompt completa do experimento 6, com todo o contexto passado para o modelo, pode ser encontrado no Apêndice A. As informações sobre cada competência foram retirados do guia disponibilizado pelo governo para os estudantes, disponíveis no site oficial: http://portal.mec.gov.br/ultimas-noticias/418-enem-946573306/81381-conheca-as-cinco-competencias-cobradas-na-redacao-do-enem (acessado em 20 de outubro de 2024).

4.2 Fase 2 - Análise e Engenharia de Características

A segunda fase do trabalho focou na extração de características para serem utilizadas como insumos em modelos de aprendizado de máquina. O objetivo principal dessa etapa foi avaliar como as diferentes métricas extraídas dos textos podem contribuir para a tarefa de correção automática de redações, utilizando abordagens supervisionadas.

4.2.1 Passo 1 - NILC-Metrix: Extração de Características Base

Inicialmente, utilizamos o NILC-Metrix para extrair 72 métricas textuais de cada redação. Estas métricas incluem estatísticas linguísticas, lexicais e de estrutura textual que são amplamente utilizadas para tarefas de avaliação automática de textos. Na Tabela 4.1, apresentamos um exemplo das métricas extraídas ao avaliar a frase: "Liberdade é muito pouco, o que eu quero ainda não tem nome.".

No Anexo A Tabela A.1, temos a descrição de cada uma das métricas utilizadas.

Essas métricas foram posteriormente normalizadas e organizadas em um conjunto de dados que serviu como entrada para o modelo de aprendizado de máquina.

Métrica	Valor
adjective_ratio	0.0
adverbs	0.33333
content_words	0.75
flesch	88.605
function_words	0.25
sentences_per_paragraph	1.0
syllables_per_content_word	2.0
words_per_sentence	12.0
noun_ratio	0.16667
paragraphs	1
sentences	1
words	12
pronoun_ratio	0.25
verbs	0.25
logic_operators	0.08333
and_ratio	0.0
if_ratio	0.0
or_ratio	0.0
negation_ratio	0.08333
cw_freq	1341415.33333
cw_freq_brwac	6.01556
cw_freq_bra	5.83889
min_cw_freq	65261.0
min_cw_freq_brwac	5.135

(a) *Primeira metade das métricas.*

Métrica	Valor
min_freq_brwac	5.135
min_cw_freq_bra	4.963
min_freq_bra	4.963
freq_brwac	6.14508
freq_bra	6.08708
hypernyms_verbs	0.0
brunet	5.20226
honore	1079.18125
personal_pronouns	0.08333
ttr	1.0
conn_ratio	0.08333
add_neg_conn_ratio	0.08333
add_pos_conn_ratio	0.0
cau_neg_conn_ratio	0.0
cau_pos_conn_ratio	0.0
log_neg_conn_ratio	0.08333
log_pos_conn_ratio	0.0
tmp_neg_conn_ratio	0.0
tmp_pos_conn_ratio	0.0
adjectives_ambiguity	0
adverbs_ambiguity	4.75
nouns_ambiguity	4.0
verbs_ambiguity	16.0
yngve	2.14286

(b) *Segunda metade das métricas.*

Tabela 4.1: *Métricas extraídas de amostra, usando NILC-Metrix programa.*

4.2.2 Passo 2 - Extração com Modelos LLaMA e Gemini

Além das métricas do NILC-Metrix, utilizamos as saídas dos modelos LLaMA e Gemini para extrair características adicionais das redações. Essas saídas foram obtidas utilizando os melhores prompts da Fase 1, especificamente os Prompts Cadeia de Pensamento e Contextual. Cada modelo avaliou o conjunto de textos uma única vez, devido a restrições de custo das operações.

O objetivo dessa etapa foi verificar se as informações geradas pelos modelos LLaMA e Gemini podem servir como características relevantes para treinar modelos supervisionados. As saídas consistiram em escores por competência, detalhamentos intermediários e a nota final atribuída pelos modelos.

4.2.3 Passo 3 - Grid Search para Otimização de Hiperparâmetros

Com o conjunto de dados consolidado, utilizamos o algoritmo XGBoost para treinar e testar modelos supervisionados. Antes disso, foi realizada uma busca sistemática pelos melhores hiperparâmetros utilizando a técnica de *Grid Search*, implementada pela biblioteca Scikit-learn. O objetivo era maximizar a métrica de *Quadratic Weighted Kappa* (QWK), utilizada para avaliar a consistência das notas preditas em relação às notas verdadeiras.

O Programa 4.1 foi usado para realizar essa busca pelos hiperparâmetros. É valido destacar que o QWK foi utilizado como métrica de otimização. No código também é possível notar todos os hiperparâmetros que foram testados, o modelo foi executado 19683 com cada combinação de hiper-parâmetros a fim de chegar na combinação ideal.

A Tabela 4.2 apresenta os parâmetros otimizados para o XGBoost, enquanto o Código 4.1 ilustra o processo de busca.

Parâmetro	Valor	Descrição
`colsample_bytree`	1.0	Fração de colunas a serem amostradas aleatoriamente para cada árvore.
`learning_rate`	0.01	Taxa de aprendizado para o treinamento do modelo; valores menores tornam o treinamento mais lento, mas podem melhorar o desempenho.
`max_depth`	11	Profundidade máxima de cada árvore, controla a complexidade do modelo e pode evitar overfitting.
`n_estimators`	100	Número de árvores (ou estimadores) no modelo.
`subsample`	0.4	Fração de amostras a serem utilizadas para o treinamento de cada árvore.
`gamma`	0	Valor mínimo de redução de perda necessário para uma divisão adicional; ajuda no controle da complexidade.
`max_delta_step`	0	Incremento máximo permitido no peso da árvore em cada iteração.
`scale_pos_weight`	1	Peso relativo das classes positivas em relação às negativas, usado em datasets desbalanceados.
`min_child_weight`	1	Peso mínimo somado das instâncias em um nó para que ele seja dividido.

Tabela 4.2: *Hiperparâmetros Otimizados para o XGBoost*

4.2.4 Passo 4 - Execução dos Experimentos

Realizamos sete experimentos, variando os conjuntos de dados de entrada. Os detalhes estão na Tabela 4.3. Cada experimento utilizou diferentes combinações de características baseadas nas métricas do NILC-Metrix e nas saídas dos modelos LLaMA e Gemini.

Os resultados mostraram que a combinação de características do NILC-Metrix com as saídas do LLaMA e Gemini frequentemente superou os modelos base, evidenciando o benefício da integração de múltiplas fontes de dados.

Programa 4.1 Grid Search do GXBoost com QWK como métrica de otimização.

```
# Define a custom scorer for Cohen's Kappa
def cohen_kappa_scorer(y_true, y_pred):
    return cohen_kappa_score(y_true, y_pred, weights="quadratic")

kappa_scorer = make_scorer(cohen_kappa_scorer, greater_is_better=True)

# Define parameter grid
param_grid = {
    "learning_rate": [0.01, 0.05, 0.1], # Finer tuning of learning rate
    "max_depth": [7, 9, 11], # Explore deeper trees
    "n_estimators": [50, 100, 200], # Increase the number of trees
    "subsample": [0.4, 0.6, 0.8], # Explore different subsets of data
    "colsample_bytree": [0.6, 0.8, 1.0], # Column sampling
    "min_child_weight": [1, 3, 5], # Regularization for splits
    "gamma": [0, 1, 5], # Minimum loss reduction
    "scale_pos_weight": [1, 2, 3], # Class balancing
    "max_delta_step": [0, 1, 5] # Improves convergence for imbalanced data
}

xgb_model = xgb.XGBClassifier(objective="multi:softmax", num_class=6,
    eval_metric="mlogloss")

# Set up GridSearchCV
grid_search = GridSearchCV(
    estimator=xgb_model,
    param_grid=param_grid,
    scoring=kappa_scorer,
    cv=3, # 3-fold cross-validation
    verbose=2,
    n_jobs=-1
)

# Fit GridSearchCV
grid_search.fit(X_train_c1, y_train_c1)

# Get the best parameters and best score
print("Best Parameters:", grid_search.best_params_)
print("Best QWK Score:", grid_search.best_score_)
```

Experimento	Dados de Treinamento
1	Apenas métricas do NILC-Metrix
2	Métricas do NILC-Metrix + Resultados Gemini (Prompts Cadeia de Pensamento e Contextual)
3	Métricas do NILC-Metrix + Resultados LLaMA (Prompts Cadeia de Pensamento e Contextual)
4	Métricas do NILC-Metrix + Resultados Gemini e LLaMA (Prompts Cadeia de Pensamento e Contextual)
5	Resultados Gemini (Prompts Cadeia de Pensamento e Contextual)
6	Resultados LLaMA (Prompts Cadeia de Pensamento e Contextual)
7	Resultados Gemini e LLaMA (Prompts Cadeia de Pensamento e Contextual)

Tabela 4.3: *Descrição dos dados de treinamento por experimento*

4.2.5 Repositório Público

O código e os dados utilizados neste trabalho estão disponíveis no repositório https://github.com/dslcosta1/AutomatedENEMEssaysCorrectionUsingChatBots Costa, 2024.

Capítulo 5

Resultados

5.1 Fase 1

5.1.1 Gemini 1.5-Flash

A Figura 5.1 apresenta os resultados obtidos para o modelo **Gemini 1.5-Flash**, com as métricas de desempenho avaliadas para os diferentes padrões de prompt testados.

De forma geral, o experimento **Cadeia de Pensamento** apresentou o melhor desempenho em termos de concordância, precisão e robustez das predições em várias competências. Este padrão foi particularmente eficaz nas competências 2, 3, 4 e 5, alcançando os maiores valores de *QWK* e baixos valores de *RMSE*. Além disso, os resultados obtidos por este experimento indicaram que as predições frequentemente estavam dentro do intervalo aceitável de 80 pontos definido pelo *PIA*, demonstrando alinhamento consistente com as notas de referência.

Para a Competência 1, que avalia o domínio da norma culta, o melhor desempenho foi alcançado pelo experimento **Base Persona**. Foi solicitado que o modelo agisse como um professor de lingua portuguesa, o que provavelmente levou o modelo a ter mais atenção a gramática e uso da língua que são avaliados nessa competência. Já na Competência 2, relacionada à compreensão da proposta, o experimento **Cadeia de Pensamento** destacou-se devido à sua capacidade de dividir a tarefa em etapas lógicas, permitindo que o modelo interpretasse com mais precisão a relação entre o tema e o desenvolvimento do texto.

Na Competência 3, que avalia a seleção e organização de argumentos, o experimento **Cadeia de Pensamento** também obteve os melhores resultados, sugerindo que a estruturação em subtarefas beneficiou o modelo ao lidar com aspectos mais complexos da argumentação. Para a Competência 4, que trata da coesão e progressão textual, o padrão **Contextual** foi o mais eficaz quando avaliando o PIA, mas o Cadeia de Pensamento continuou se destacando no QWK, Acurácia e RMSE.

Por fim, na Competência 5, que exige a elaboração de uma proposta de intervenção, o padrão **Cadeia de Pensamento** mais uma vez obteve o melhor desempenho, evidenciando que a divisão da tarefa em etapas auxilia o modelo na avaliação da viabilidade e

detalhamento das soluções propostas pelo candidato.

Por outro lado, o experimento **Base** apresentou o pior desempenho em quase todas as métricas e competências avaliadas. Os valores de *PIA* foram consistentemente baixos, enquanto o *QWK* e Acurácia apresentaram alta dispersão, indicando falta de concordância com as notas reais.

Conclui-se, portanto, que padrões mais estruturados, como **Cadeia de Pensamento** e **Contextual**, são mais eficazes na tarefa de correção de redações do ENEM, especialmente para competências que demandam maior nível de interpretação e análise, como organização de argumentos e coesão textual. Já padrões simples ou sem alinhamento com o contexto cultural, como **Base e o Base Inglês**, são menos adequados, apresentando desempenho inferior em praticamente todas as métricas avaliadas.

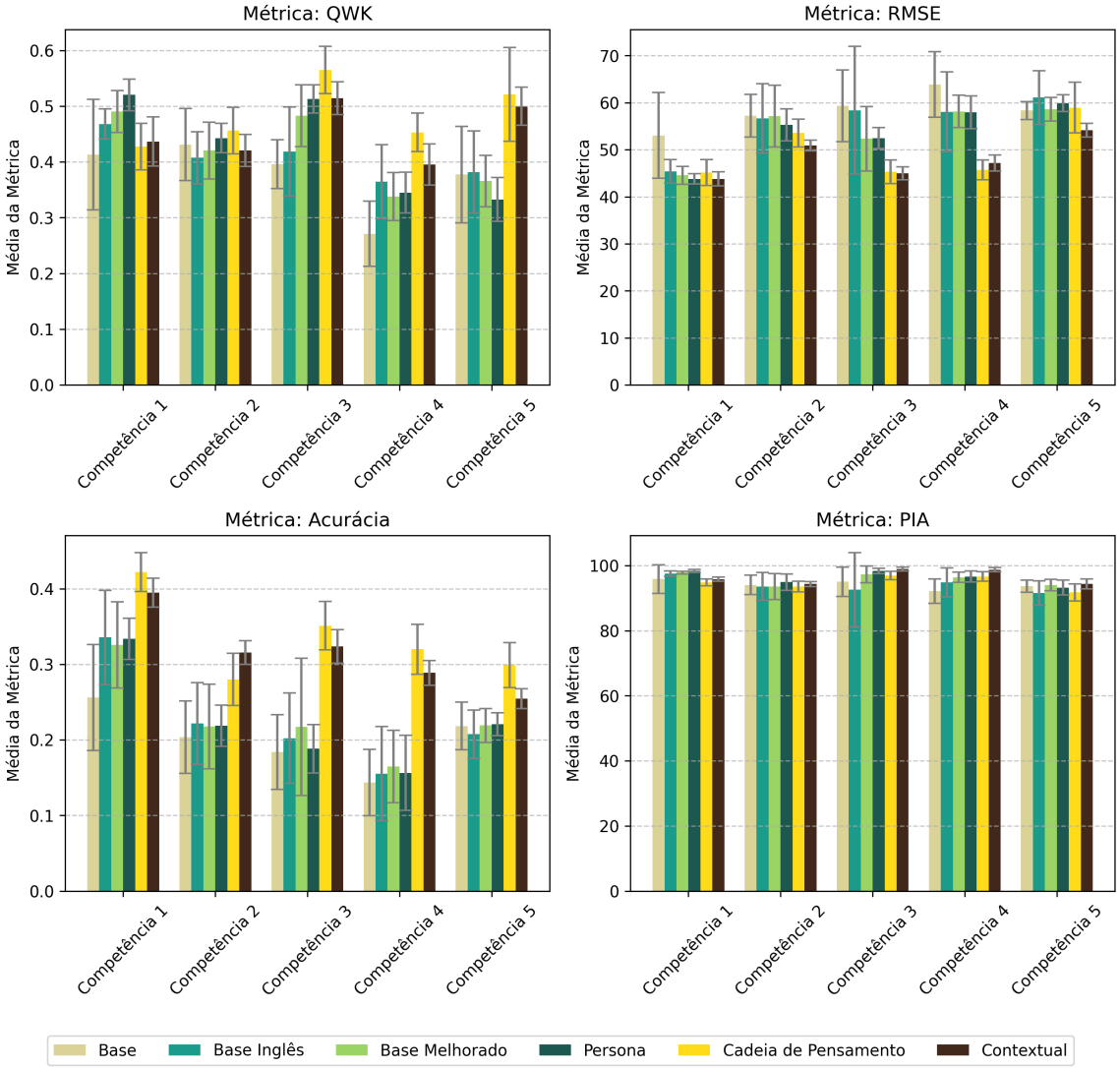


Figura 5.1: *Gemini: Média e Desvio Padrão para QWK, RMSE, Acurácia e PIA dos Resultados*

5.1.2 LLaMA

A Figura 5.2 apresenta os resultados obtidos para o modelo **LLaMA**, com as métricas de desempenho avaliadas para os diferentes padrões de prompt. Assim como no modelo Gemini, foram analisadas as métricas *QWK* (Quadratic Weighted Kappa), *RMSE* (Root Mean Square Error), Acurácia e *PIA* (Precisão do Intervalo Absoluto) para as cinco competências do ENEM.

De forma geral, o experimento **Contextual** apresentou os melhores resultados para o modelo LLaMA, sendo superior em diversas competências e métricas. Esse experimento destacou-se particularmente nas competências 3, 4 e 5, onde alcançou os maiores valores de *QWK* e *PIA*, além de um *RMSE* mais baixo em comparação com outros padrões. O enriquecimento dos prompts com descrições detalhadas sobre as competências avaliadas no ENEM demonstrou ser uma estratégia eficaz para alinhar as respostas do modelo aos critérios de avaliação humana.

Na Competência 1, que avalia o domínio da norma culta, o experimento **Persona** também apresentou resultados sólidos, com desempenho comparável ao experimento **Contextual** nas métricas de *QWK* e *RMSE*. É válido destacar um comportamento observado na Competência 1, mas que se repetiu para as demais: a clara melhoria nos resultados com os prompts **Base Melhorado** e **Persona**, o que evidencia, também, a importância da estrutura e do formato do pedido para a eficácia do modelo.

Nas Competências 2, 3, 4 e 5, que envolvem a seleção e organização de argumentos, o padrão **Contextual** foi claramente o mais robusto, destacando-se em todas as métricas.

Por outro lado, os experimentos **Base** e **Base Inglês** novamente apresentaram o pior desempenho em quase todas as métricas e competências. Os resultados de *QWK* foram baixos e apresentaram maior dispersão, enquanto o *RMSE* foi consistentemente elevado para esses casos. Esse comportamento reforça a hipótese de que o uso de prompts em inglês é menos eficaz para modelos avaliando tarefas cultural e linguisticamente específicas, como as redações do ENEM. Além disso, a simplicidade do prompt pode ter contribuído para a incapacidade do modelo de compreender plenamente os critérios de avaliação ou identificar claramente o texto que deveria ser avaliado.

O prompt **Cadeia de Pensamento** também apresentou um dos piores desempenhos. Uma possível explicação para isso pode estar relacionada à arquitetura e ao treinamento do modelo. Embora a abordagem de dividir a tarefa em etapas seja amplamente reconhecida por melhorar o desempenho de LLMs em outros contextos, o LLaMA pode não ter sido suficientemente ajustado para lidar com prompts que exigem raciocínio em múltiplas etapas de maneira eficiente. Além disso, o tamanho da janela de contexto do LLaMA, embora grande, pode não ter sido otimizado para processar as informações adicionais geradas por prompts mais longos e detalhados. Outro fator relevante pode ser a falta de alinhamento entre o treinamento do LLaMA e tarefas específicas em português, o que pode ter prejudicado a compreensão e o raciocínio necessários para interpretar e aplicar os critérios do ENEM apresentados em etapas sucessivas. Por fim, a complexidade adicional do formato **Cadeia de Pensamento** pode ter introduzido ambiguidades que confundiram o modelo, especialmente na integração das respostas intermediárias com o

output final, resultando em desempenho inferior em comparação a padrões mais diretos, como o **Contextual**.

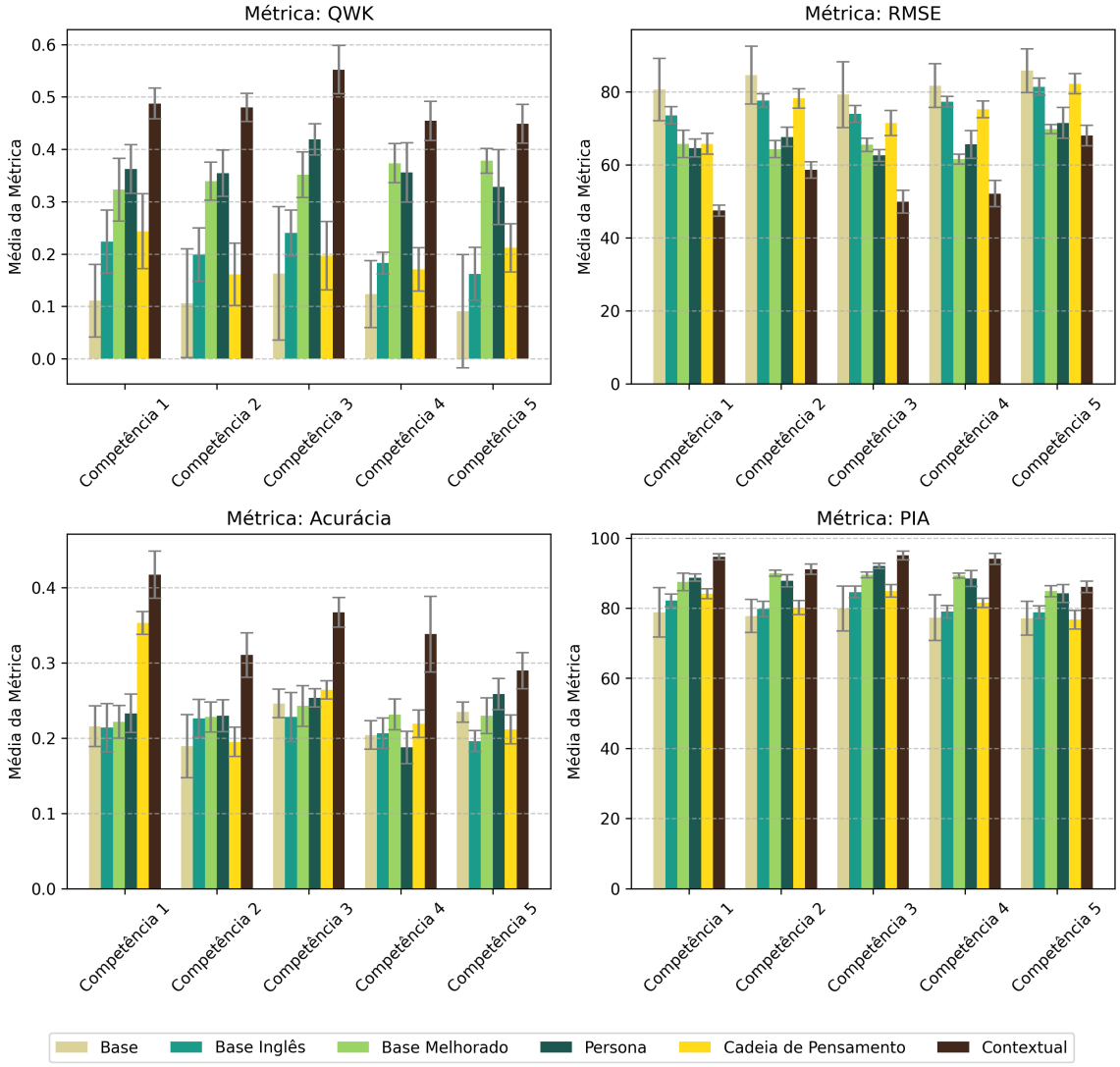


Figura 5.2: *LLaMA: Média e Desvio Padrão para QWK, RMSE, Acurácia e PIA dos Resultados*

5.1.3 Comparação: Gemini vs LLaMA

A Tabela 5.1.3 apresenta uma comparação detalhada entre os modelos **Gemini** e **LLaMA** para cada uma das cinco competências do ENEM, avaliadas com base nas métricas *QWK*, *RMSE*, acurácia e *PIA* em diferentes experimentos. A seguir, discutimos os principais achados.

De forma geral, o modelo **Gemini** apresentou desempenho superior em todas as competências para as métricas *QWK* e *PIA*, enquanto o **LLaMA** demonstrou resultados promissores em experimentos específicos, particularmente com o padrão **Contextual** (Experimento 6). O Gemini foi mais consistente, com valores mais altos de *QWK* e menores *RMSE* em grande parte das competências, refletindo maior precisão e alinhamento com as notas reais.

O **LLaMA**, por outro lado, apresentou maior variabilidade de resultados, sugerindo que sua eficácia depende significativamente do tipo de prompt utilizado. Os melhores resultados para o LLaMA foram obtidos com o Experimento 6 (Contextual), onde o modelo mostrou bom desempenho, especialmente nas métricas de *RMSE* e *PIA*, reduzindo a diferença em relação ao Gemini.

Na **Competência 1** (Domínio da norma culta), o Gemini superou o LLaMA em todas as métricas. O padrão **Persona** (Experimento 4) foi o melhor para o Gemini, com um *QWK* de 0.520 e *RMSE* de 43.801. Para o LLaMA, o melhor desempenho foi no Experimento 6, com um *QWK* de 0.487 e *RMSE* de 47.474, mostrando que prompts detalhados melhoram a avaliação.

Na **Competência 2** (Compreensão da proposta), o Gemini manteve desempenho superior em *QWK* (0.456 no Experimento Cadeia de Pensamento) e *RMSE* (50.914 no Experimento 6), enquanto o LLaMA apresentou seu melhor resultado no Experimento Contextual, com *QWK* de 0.480 e *RMSE* de 58.640.

Na **Competência 3** - Seleção e organização de argumentos - o padrão **Cadeia de pensamento** foi o melhor. O Gemini alcançou *QWK* de 0.565 e *PIA* de 98.930 com esse prompt, enquanto o LLaMA atingiu *QWK* de 0.552 e *PIA* de 95.062 com o prompt contextual. A superioridade do Gemini nesta competência reflete sua robustez em lidar com tarefas de organização argumentativa.

Na **Competência 4** (Coesão textual), os resultados foram semelhantes. O Gemini obteve *QWK* de 0.453 e *RMSE* de 45.725 no Experimento 5, enquanto o LLaMA alcançou *QWK* de 0.454 e *RMSE* de 52.124 no Experimento 6. Apesar da diferença em *RMSE*, o *QWK* próximo sugere que ambos os modelos conseguem capturar bem a coesão textual com prompts detalhados. Esse foi a úncia competência onde o LLaMA consegui ganhar no QWK.

Na **Competência 5** (Proposta de intervenção), o Gemini mais uma vez liderou, com *QWK* de 0.521 no Experimento 5 e *RMSE* de 54.134 no Experimento 6. O LLaMA apresentou seu melhor desempenho no Experimento 6, com *QWK* de 0.449 e *PIA* de 86.091. Esses resultados reforçam a capacidade do Gemini de lidar com competências complexas, enquanto o LLaMA se beneficia significativamente de prompts enriquecidos.

Os resultados mostram que o modelo **Gemini-1.5 Flash** é mais consistente, com desempenho superior em quase todas as métricas e competências. Sua capacidade de lidar bem com prompts variados sugere um alinhamento mais robusto com os critérios de avaliação do ENEM. Por outro lado, o **LLaMA 3.2 70B** demonstrou que é capaz de alcançar resultados competitivos quando combinado com padrões de prompt detalhados, como o **Contextual**.

O Gemini, parece estar mais bem ajustado para tarefas como a avaliação de redações, que exigem interpretação ampla e flexível. O LLaMA, apesar de seu tamanho e complexidade, pode não ter sido tão amplamente treinado em português ou em tarefas relacionadas ao ENEM, o que pode explicar sua maior dependência de prompts contextualiados.

A análise comparativa reforça a eficácia do **Gemini** em tarefas de avaliação de redações, destacando sua consistência e robustez em diferentes configurações de prompt. O

LLaMA, embora menos consistente, mostrou-se promissor, especialmente quando utilizado com padrões detalhados como o **Contextual**, indicando que sua performance pode ser significativamente melhorada com ajustes e otimizações específicas para o contexto brasileiro.

Comp	Métrica	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
C1	QWK Gemini	0.413	0.468	0.490	**0.520**	0.427	0.437
	QWK Llama	0.111	0.224	0.323	0.362	0.243	**0.487**
	RMSE Gemini	53.026	45.400	44.553	**43.801**	45.179	43.808
	RMSE Llama	80.659	73.615	65.763	64.657	65.797	**47.474**
	Acurácia Gemini	0.256	0.336	0.326	0.334	**0.422**	0.395
	Acurácia Llama	0.216	0.214	0.221	0.233	0.353	**0.417**
	PIA Gemini	95.863	97.568	97.822	**98.313**	94.832	95.802
	PIA Llama	78.848	82.222	87.490	88.724	84.115	**94.650**
C2	QWK Gemini	0.431	0.407	0.420	0.443	**0.456**	0.420
	QWK Llama	0.106	0.199	0.339	0.354	0.161	**0.480**
	RMSE Gemini	57.201	56.681	57.133	55.288	53.547	**50.914**
	RMSE Llama	84.622	77.658	64.334	67.662	78.214	**58.640**
	Acurácia Gemini	0.204	0.222	0.218	0.219	0.280	**0.315**
	Acurácia Llama	0.189	0.226	0.228	0.230	0.195	**0.310**
	PIA Gemini	94.019	93.583	93.571	**94.897**	93.504	94.300
	PIA Llama	77.778	79.835	89.959	87.819	80.165	**91.111**
C3	QWK Gemini	0.396	0.419	0.483	0.513	**0.565**	0.515
	QWK Llama	0.163	0.240	0.351	0.419	0.197	**0.552**
	RMSE Gemini	59.318	58.387	52.328	52.477	45.312	**45.002**
	RMSE Llama	79.276	73.970	65.496	62.612	71.508	**49.895**
	Acurácia Gemini	0.184	0.202	0.217	0.188	**0.351**	0.324
	Acurácia Llama	0.246	0.228	0.243	0.253	0.264	**0.367**
	PIA Gemini	95.014	92.545	97.304	98.354	96.929	**98.930**
	PIA Llama	79.918	84.609	89.547	92.016	84.938	**95.062**
C4	QWK Gemini	0.271	0.365	0.338	0.345	**0.453**	0.395
	QWK Llama	0.124	0.183	0.373	0.356	0.171	**0.454**
	RMSE Gemini	63.879	58.075	58.138	57.929	**45.725**	47.175
	RMSE Llama	81.738	77.290	61.580	65.623	75.243	**52.124**
	Acurácia Gemini	0.144	0.155	0.165	0.156	**0.320**	0.289
	Acurácia Llama	0.204	0.207	0.231	0.188	0.219	**0.338**
	PIA Gemini	92.127	94.825	96.379	96.626	96.617	**98.704**
	PIA Llama	77.284	79.012	89.300	88.477	81.564	**94.074**

Continua na próxima página

Continuação da Tabela 5.1.3

Comp	Métrica	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
C5	QWK Gemini	0.377	0.382	0.366	0.333	**0.521**	0.500
	QWK Llama	0.091	0.162	0.378	0.328	0.212	**0.449**
	RMSE Gemini	58.323	61.092	58.607	59.900	58.949	**54.134**
	RMSE Llama	85.821	81.426	69.810	71.532	82.258	**68.046**
	Acurácia Gemini	0.218	0.207	0.219	0.221	**0.299**	0.255
	Acurácia Llama	0.235	0.196	0.230	0.258	0.212	**0.290**
	PIA Gemini	93.635	91.575	93.988	93.210	91.728	**94.403**
	PIA Llama	77.119	78.848	84.856	84.198	76.708	**86.091**

Tabela 5.1: *Comparação Métricas para as Competências por experimento, Gemini e Llama.*

5.1.4 Teste estatístico

Como os modelos utilizados não são determinísticos, cada execução de avaliação para o mesmo texto e com o mesmo prompt pode gerar notas diferentes. Isso levanta a questão sobre a confiabilidade dos resultados apresentados nesta seção, especialmente considerando que, devido a limitações de recursos e restrições das APIs, foi possível realizar apenas 20 execuções para cada prompt no Gemini e 5 no LLaMA para o mesmo texto. O objetivo inicial era atingir pelo menos 100 execuções para cada modelo, avaliando a mesma redação.

Para assegurar que os resultados obtidos são representativos e não apenas valores atípicos (*outliers*) de execuções isoladas, foi realizado o teste estatístico de Kruskal-Wallis. Esse teste avalia a consistência dos modelos, verificando se a mediana das diferenças absolutas entre as notas reais e preditas permanece próxima nas execuções realizadas. Isso permite inferir que, mesmo com um número limitado de execuções, há alta probabilidade de que os resultados observados se mantenham consistentes em um número maior de iterações, como 100 ou 1000 execuções.

Os resultados do teste de Kruskal-Wallis estão apresentados na Tabela 5.2. As colunas indicam os valores do teste estatístico e o valor-p (*p-value*) para cada modelo, experimento e competência. Valores-p menores que 0.05 foram destacados, indicando que a hipótese nula — de que os resultados são consistentes entre as execuções — é rejeitada. Isso significa que, nesses casos específicos, a consistência do modelo não pode ser garantida.

Para o modelo **Gemini**, os resultados foram amplamente positivos, com a maioria das execuções passando no teste de Kruskal-Wallis em quase todas as competências e experimentos. Esse desempenho reforça a robustez e a confiabilidade do Gemini, mesmo com o número limitado de repetições realizadas. Apenas no experimento **Base** com a **Competência 4** foram observados resultados que podem não se repetir em futuras execuções.

Já para o modelo **LLaMA**, os resultados foram mais variados, com várias execuções

falhando no teste. Um destaque negativo é a **Competência 3** (Seleção e organização de argumentos), onde todos os experimentos falharam no teste de Kruskal-Wallis. Isso indica alta variabilidade nos resultados do modelo para essa competência, sugerindo que ele pode não estar processando essa dimensão de forma consistente. Casos como este são preocupantes, pois a falta de consistência compromete parte da validade dos resultados apresentados para essa competência. Embora os resultados ainda sejam relevantes, por refletirem o comportamento observado em experimentos reais, não há garantia estatística robusta de que eles se repetirão em futuras execuções.

No **Experimento 2**, a **Competência 5** também apresentou inconsistências significativas para o LLaMA, com um valor-p de 0.003, sugerindo que os resultados podem variar consideravelmente com mais execuções. Esse comportamento é menos evidente no modelo Gemini, que manteve resultados consistentes em praticamente todos os cenários avaliados.

Vale ressaltar que o teste de Kruskal-Wallis foi escolhido em detrimento da ANOVA devido ao tamanho desbalanceado dos grupos e à ausência de normalidade na distribuição dos dados. Apesar disso, a ANOVA foi realizada como verificação adicional, e os resultados mostraram-se semelhantes aos obtidos com o Kruskal-Wallis. Detalhes sobre os resultados da ANOVA podem ser encontrados no apêndice.

Tabela 5.2: *Teste de Kruskal-Wallis para Gemini(Ge) e LLaMA(LL) por Experimentos e Competências*

Experimentos	Comp.	Ge. Kruskal-W	Ge. Valor-P	LL. Kruskal-W	LL. Valor-P
Experimento 1	C1	7.200	0.126	9.686	**0.046**
	C2	1.908	0.753	4.382	0.357
	C3	6.885	0.142	9.777	**0.044**
	C4	11.712	0.020	2.554	0.635
	C5	6.447	0.168	5.600	0.231
Experimento 2	C1	0.081	0.999	3.983	0.408
	C2	0.022	1.000	14.971	**0.005**
	C3	0.842	0.933	12.555	**0.014**
	C4	0.343	0.987	3.001	0.558
	C5	0.000	1.000	15.905	**0.003**
Experimento 3	C1	0.000	1.000	7.121	0.130
	C2	1.473	0.831	9.565	**0.048**
	C3	0.000	1.000	11.314	**0.023**
	C4	4.622	0.328	0.811	0.937
	C5	0.000	1.000	4.099	0.393
Experimento 4	C1	3.200	0.525	10.651	**0.031**
	C2	0.842	0.933	0.667	0.955
	C3	4.800	0.308	17.172	**0.002**
	C4	6.153	0.188	2.579	0.631
	C5	0.842	0.933	5.684	0.224

Continua na próxima página

Continuação da Tabela 5.2

Experimentos	Comp.	Ge. Kruskal-W	Ge. Valor-P	LL. Kruskal-W	LL. Valor-P
Experimento 5	C1	1.525	0.822	7.314	0.120
	C2	1.050	0.902	0.000	1.000
	C3	3.684	0.450	11.584	**0.021**
	C4	2.078	0.721	0.162	0.997
	C5	1.600	0.809	1.117	0.892
Experimento 6	C1	0.521	0.971	0.688	0.953
	C2	0.000	1.000	0.548	0.969
	C3	0.533	0.970	15.990	**0.030**
	C4	0.257	0.992	0.122	0.998
	C5	8.348	0.080	7.320	0.120

5.2 Fase 2

Nesta seção, analisamos os resultados obtidos a partir dos experimentos realizados na Fase 2, cujos dados estão sintetizados na Figura 5.3 e na Tabela 5.3. O objetivo principal foi avaliar como as diferentes combinações de dados de treinamento - incluindo as métricas do NILC-Metrix e as saídas dos modelos LLaMA e Gemini - impactaram o desempenho do modelo XGBoost nas tarefas de predição das notas das cinco competências avaliadas no ENEM.

5.2.1 Impacto dos Dados do LLaMA e Gemini

Os resultados indicam que a adição das características extraídas pelos modelos LLaMA e Gemini resultou em melhorias significativas nas métricas de desempenho, especialmente quando combinadas com as métricas base do NILC-Metrix. No Experimento 4 (NILC + Gemini + LLaMA), por exemplo, observou-se o maior desempenho geral em termos de *QWK* e *RMSE*, destacando-se particularmente nas Competências 2 e 3, com valores de *QWK* de 0.601 e 0.696, respectivamente. Isso demonstra o potencial complementar das características extraídas pelos dois LLMs, que capturam aspectos textuais não contemplados pelas métricas base.

Adicionalmente, os experimentos com dados exclusivamente derivados do LLaMA e Gemini (Experimentos 5, 6 e 7) mostraram que, mesmo sem as métricas do NILC-Metrix, esses modelos são capazes de capturar informações suficientes para treinar um modelo de aprendizado de máquina. O Experimento 7 (Gemini + LLaMA) alcançou *QWK* de 0.619 na Competência 5, superando os experimentos que incluíam as métricas do NILC-Metrix. Isso sugere que as características geradas pelos LLMs são ricas e robustas o suficiente para prever com alta precisão a competência de Proposta de Intervenção (C5), onde aspectos contextuais e criativos têm mais peso.

Embora a adição de dados tenha, em geral, resultado em melhorias, alguns experimentos revelaram quedas de desempenho em métricas específicas. Um exemplo claro ocorreu na competência C4, onde o *PIA* caiu de 93.827 (Experimento 1) para 91.770 (Experimento

6), indicando que a inclusão de características derivadas apenas do LLaMA pode ter introduzido ruído no treinamento. Esse comportamento pode ser explicado pelo fato de que as saídas dos LLMs, embora ricas em informações, também podem conter inconsistências ou variações não alinhadas com o objetivo da tarefa, impactando negativamente o modelo de aprendizado.

Além disso, o Experimento 3 (NILC + LLaMA) apresentou *QWK* inferior ao Experimento 4 (NILC + Gemini + LLaMA) em todas as competências, confirmando que a introdução de características adicionais precisa ser cuidadosamente balanceada para evitar sobrecarga de dados irrelevantes. A competência 1, por exemplo, teve o desempenho de *QWK* reduzido de 0.566 (Experimento 4) para 0.493 (Experimento 6), demonstrando que o uso isolado das características do LLaMA não foi suficiente para superar as métricas base e as combinações mais robustas.

As competências C2 (Compreensão da Proposta), C3 (Seleção e Organização de Argumentos) e C5 (Proposta de Intervenção) apresentaram os melhores resultados gerais em termos de *QWK*, especialmente nos experimentos que utilizaram ambas as fontes de dados (Gemini e LLaMA). Esses resultados podem ser atribuídos à capacidade dos LLMs de capturar aspectos semânticos e estruturais cruciais para essas competências.

Os resultados da Fase 2 destacam o potencial das características extraídas pelos LLMs na melhoria do desempenho de modelos tradicionais de aprendizado de máquina, como o XGBoost. A combinação das métricas do NILC-Metrix com as saídas dos modelos LLaMA e Gemini demonstrou ser particularmente eficaz, capturando nuances textuais que antes passavam despercebidas.

No entanto, a introdução de características adicionais exige cuidado, uma vez que a inclusão de dados irrelevantes ou inconsistentes pode introduzir ruído e, consequentemente, reduzir o desempenho em métricas específicas. Esses achados reforçam a importância de realizar análises exploratórias cuidadosas e testes sistemáticos para identificar as melhores combinações de dados.

Por fim, os experimentos com dados exclusivamente derivados dos LLMs mostram que esses modelos têm o potencial de substituir parcialmente as métricas base em alguns contextos, especialmente em tarefas mais subjetivas e criativas. Isso abre espaço para futuras investigações sobre o papel dos LLMs em sistemas de correção automática e outras aplicações de processamento de linguagem natural.

5.3 Comparação entre os Resultados da Fase 1 e Fase 2

Ao comparar os resultados da Fase 1, onde apenas os modelos LLaMA e Gemini foram utilizados em cenários zero-shot, com os da Fase 2, que envolveu o uso de aprendizado de máquina (XGBoost) com diferentes combinações de dados, observa-se que o XGBoost superou ambos os modelona maioriate das métricas e competências. Essa diferença é especialmente notável nas métricas *QWK*, *RMSE*, e *Acurácia*, o que demonstra que o uso de algoritmos de aprendizado supervisionado pode aproveitar melhor as características extraídas pelos modelos de linguagem para predizer as notas das competências. Vale

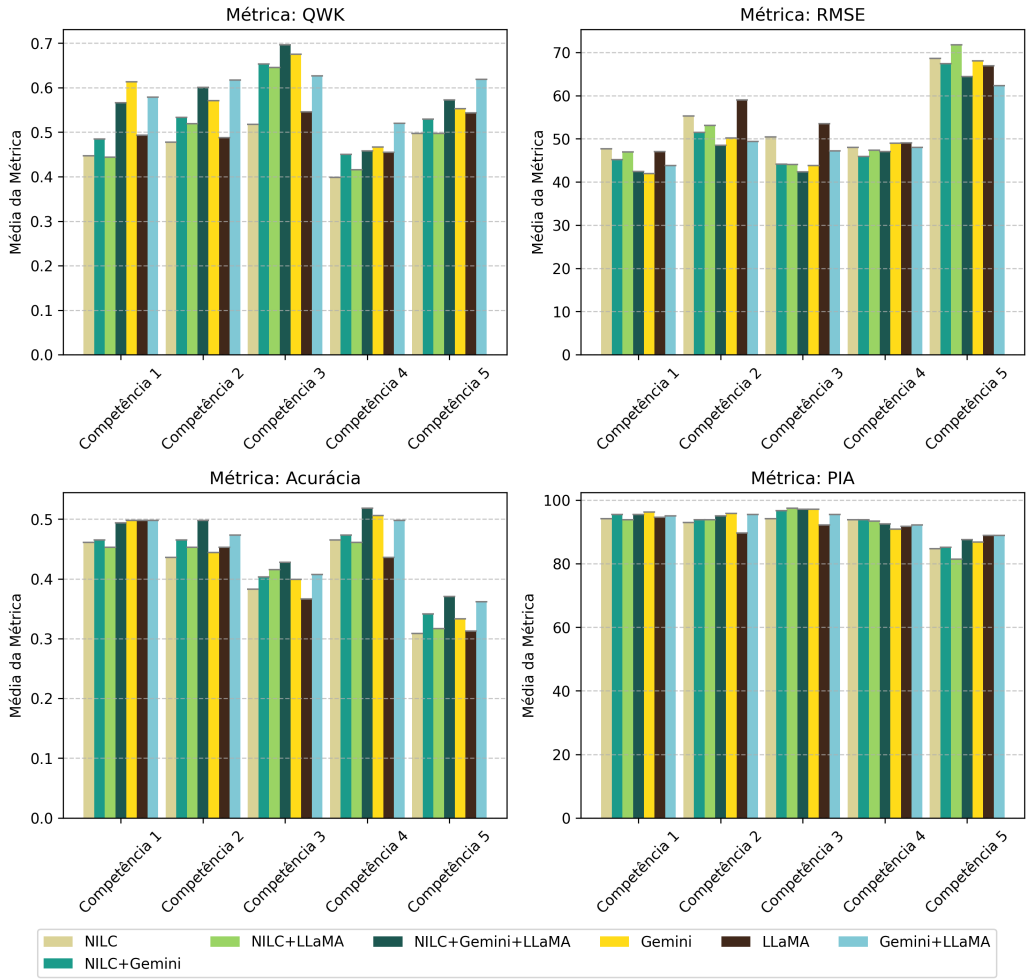


Figura 5.3: *GXBoost: QWK, RSME, Acurácia e PIA dos Resultados*

ressaltar que o desempenho competitivo dos LLMs em um cenário zero-shot sugere que, em situações com poucos dados disponíveis, esses modelos podem ser uma boa alternativa, fornecendo resultados próximos aos alcançados por abordagens com treinamento supervisionado.

Competência 1 (C1): Na métrica *QWK*, o Experimento 5 da Fase 2 (XGBoost com Gemini) obteve 0.613, superando os melhores resultados da Fase 1 para Gemini (0.520) e LLaMA (0.487). Usando apenas as métricas tradicionais do NILC-Metrix (Experimento 1), o XGBoost alcançou 0.447, que é infeior ao desempenho do LLaMA e do Gemini. Em termos de *RMSE*, o XGBoost no Experimento 5 foi o melhor (41.929), enquanto o uso exclusivo do NILC-Metrix teve 47.730, superando o LLaMA, mas não o Gemini. A *Acurácia* do XGBoost com todas as combinações de dados atingiu 0.498, o que é superior aos dois modelos da Fase 1, enquanto o uso exclusivo do NILC-Metrix alcançou 0.461, também superando o LLaMA e o Gemini sozinhos.

Competência 2 (C2): O melhor *QWK* na Fase 2 foi 0.617 (Experimento 7), superando significativamente os 0.456 de Gemini e 0.480 de LLaMA na Fase 1. O uso exclusivo do NILC-Metrix atingiu 0.478. O *RMSE* foi menor na Fase 2 (48.483 no Experimento 4), enquanto os melhores valores na Fase 1 foram 50.914 (Gemini) e 58.640 (LLaMA). Mesmo com apenas

Comp	Métrica	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7
C1	QWK	0.447	0.485	0.444	0.566	**0.613**	0.493	0.579
	RMSE	47.730	45.179	46.966	**42.475**	41.929	47.036	43.848
	Acurácia	0.461	0.465	0.453	0.494	**0.498**	**0.498**	**0.498**
	PIA	94.239	95.473	93.827	95.473	**96.296**	94.650	95.062
C2	QWK	0.478	0.534	0.519	0.601	0.571	0.488	**0.617**
	RMSE	55.273	51.512	53.086	**48.483**	50.218	59.018	49.358
	Acurácia	0.436	0.465	0.453	0.498	0.444	0.453	**0.473**
	PIA	93.004	93.827	93.827	95.062	95.885	89.712	**95.473**
C3	QWK	0.517	0.653	0.646	**0.696**	0.675	0.546	0.626
	RMSE	50.479	44.147	44.073	**42.397**	43.848	53.518	47.175
	Acurácia	0.383	0.403	0.416	**0.428**	0.399	0.366	0.407
	PIA	94.239	96.708	97.531	**97.119**	**97.119**	92.181	95.473
C4	QWK	0.399	0.450	0.416	0.458	0.466	0.455	**0.520**
	RMSE	48.005	**45.902**	47.384	47.036	49.023	49.090	48.005
	Acurácia	0.465	0.473	0.461	**0.519**	0.506	0.436	0.498
	PIA	**93.827**	**93.827**	93.416	92.593	90.947	91.770	92.181
C5	QWK	0.497	0.530	0.497	0.572	0.553	0.543	**0.619**
	RMSE	68.614	67.403	71.756	64.457	68.035	66.913	**62.328**
	Acurácia	0.309	0.342	0.317	**0.370**	0.333	0.313	0.362
	PIA	84.774	85.185	81.481	87.654	86.831	**88.889**	**88.889**

Tabela 5.3: *Métricas para as Competências por experimento, usando aprendizado de máquina.*

as métricas NILC-Metrix, o *RMSE* foi competitivo (55.273). A *Acurácia* do XGBoost com dados adicionais alcançou 0.473 (Experimento 7), superando os modelos puros, mas o NILC-Metrix isolado ficou em 0.436.

Competência 3 (C3): Na métrica *QWK*, o XGBoost obteve 0.675 (NILC com Gemini), próximo ao maior valor alcançado por Gemini na Fase 1 (0.696). Usando apenas o NILC-Metrix, o *QWK* foi 0.517, não superando os valores do LLaMA ou do Gemini. No entanto, em *RMSE*, o XGBoost foi superior com 42.397 (Experimento 4), enquanto os melhores valores na Fase 1 foram 45.002 (Gemini) e 49.895 (LLaMA). Isso indica que o aprendizado de máquina conseguiu resultados mais consistentes, mesmo quando o *QWK* não foi o maior. A *Acurácia* com todas as combinações de dados foi 0.428, maior que os valores da Fase 1, enquanto o NILC-Metrix isolado teve 0.383, superior aos dois LLMs em cenário de zero-shot.

Competência 4 (C4): Para *QWK*, o XGBoost no Experimento 7 (NILC-Metrix, Gemini e LLaMA) atingiu 0.520, superando os 0.453 e 0.454 dos melhores resultados de Gemini e

LLaMA, respectivamente. O uso exclusivo do NILC-Metrix, no entanto, alcançou 0.399, ficando abaixo de ambos os modelos da Fase 1. Em *RMSE*, o Gemini obteve o melhor resultado (45.002 no Experimento 2), enquanto o NILC-Metrix isolado teve 48.005, sendo levemente inferior ao Gemini.

Competência 5 (C5): O Experimento 7 da Fase 2 obteve o melhor resultado geral em *QWK* (0.619) e *RMSE* (62.328), superando ambos os modelos da Fase 1. O uso exclusivo do NILC-Metrix obteve *QWK* de 0.497, superior ao LLaMA, mas inferior ao Gemini.

Os resultados confirmam que o XGBoost, quando combinado com as métricas extraídas por Gemini e LLaMA, supera os modelos de linguagem pura na maioria das competências. Entretanto, os modelos de linguagem LLaMA e Gemini foram testados em zero-shot, sem nenhum dado de treinamento e foram competitivos em vários casos, indicando sua aplicabilidade em cenários com poucos dados disponíveis. Essa estratégia é uma forma de otimizar os resultados do LLM, sem a necessidade de fazer o fine tunning do modelo original, o que exige um bastante recurso de hardware para treinar o modelo localmente.

Além disso, o uso exclusivo das métricas tradicionais do NILC-Metrix superou os resultados do LLaMA em algumas competências competências, mas raramente superou o Gemini, o que destaca a importância dos grandes modelos generativos conversacionais em capturar características mais relevantes para a tarefa. Esses achados reforçam que a combinação de métricas tradicionais e saídas de LLMs pode oferecer um desempenho superior e mais robusto e também evidencia o poder o LLMs em cenários com poucos dados, onde não é possível fazer o pre-processamento.

Os resultados de todos os experimentos podem ser acessado em Costa, 2024.

5.4 Comparação com resultados de estudos anteriores.

As técnicas exploradas nesse artigo também se mostram relevantes quando comparadas como estudo anteriores realizados nessa tarefa de correção automática de redações do ENEM.

Os resultados alcançados em meu trabalho superam os de outras abordagens, como os apresentados por Amorim e Veloso, 2017, que utilizaram técnicas tradicionais de regressão linear para prever as notas em múltiplos aspectos da redação do ENEM. Por exemplo, no estudo deles, os melhores valores de *QWK* para competências individuais ficaram em 0.424 para linguagem formal (Competência 1) e 0.335 para compreensão da tarefa (Competência 2), mesmo após o oversampling. Em contraste, meu modelo baseado em XGBoost combinado com saídas dos modelos LLaMA e Gemini alcançou valores significativamente superiores, como 0.613 para Competência 1 e 0.617 para Competência 2. Apesar de utilizarmos datasets diferentes, o desempenho aprimorado no meu trabalho reflete a eficácia de integrar modelos generativos de linguagem com algoritmos supervisionados avançados, evidenciando um avanço significativo no estado da arte da correção automática de redações em português brasileiro.

Ao comparar os resultados obtidos com Mayer, 2023, que utilizou o modelo BERTimbau

para avaliação automática de redações no formato do ENEM, destacam-se melhorias significativas em várias competências. Por exemplo, no estudo de Mayer, os valores de *QWK* atingiram 0.595 para Competência 1, 0.562 para Competência 2 e 0.621 para Competência 4. Em contraste, meus resultados com o modelo XGBoost combinado com as saídas dos modelos LLaMA e Gemini superaram esses números em diversas competências.

Ao comparar os resultados alcançados em meu trabalho com os reportados por I. Silveira *et al.*, 2024, que investigaram modelos baseados em Transformers para correção automática de redações (incluindo Gemini e Phi-3), observamos diferenças importantes no desempenho em termos de *QWK*. No estudo deles, os valores máximos de *QWK* para as competências foram de 0.46 para a Competência 1, 0.52 para a Competência 3 e 0.61 para a Competência 5 utilizando o modelo Phi-3, enquanto o Gemini mostrou desempenho mais estável, mas geralmente inferior, com resultados entre 0.35 e 0.41 para todas as competências. Em contraste, meu trabalho alcançou valores superiores em várias competências, como 0.613 para Competência 1, 0.617 para Competência 2 e 0.675 para Competência 3, ao combinar o modelo XGBoost com as saídas dos modelos LLaMA e Gemini. Embora os datasets utilizados sejam diferentes, minha abordagem demonstrou maior eficácia na extração e uso das características textuais, destacando a vantagem de integrar modelos de aprendizado supervisionado com LLMs para tarefas de avaliação de textos. Essa comparação reforça o potencial de avanços na área quando modelos gerativos e algoritmos supervisionados são utilizados em conjunto.

Na Tabela 5.4 é possível ver a comparação dos resultados encontrados com os outros trabalhos na área. É válido destacar que o XGBoost usando dados do NILC-Metrix, Gemini e LLaMA foi superior a todos os trabalhos anteriores em 4 das 5 competências.

Modelo	C1	C2	C3	C4	C5
XGBoost(NILC-Metrix, Gemini e LLaMA)	**0.613**	**0.617**	**0.696**	0.520	**0.619**
Gemini	0.520	0.456	0.565	0.453	0.521
LLaMA	0.487	0.480	0.552	0.454	0.449
Phi-3 I. Silveira *et al.*, 2024	0.463	0.351	0.522	0.292	0.610
BERTimbau Mayer, 2023	0.595	0.562	0.539	**0.621**	0.548
Técnicas Clássicas Amorim e Veloso, 2017	0.424	0.335	0.182	0.273	0.154

Tabela 5.4: *Comparação dos melhores valores de QWK por competência entre diferentes trabalhos.*

Capítulo 6

Conclusão

Este trabalho apresentou uma avaliação abrangente do desempenho dos modelos LLaMA e Gemini na correção automática de redações do ENEM, com foco na capacidade desses modelos em atribuir notas para cada uma das cinco competências exigidas pela avaliação oficial. A análise foi dividida em duas fases principais: exploração de diferentes padrões de prompts e utilização dos modelos como insumos para técnicas de aprendizado de máquina.

Os resultados obtidos destacaram a relevância de ajustes no formato e conteúdo dos prompts para aumentar a eficiência e precisão das respostas geradas pelos modelos. Entre os padrões explorados, a utilização de prompts contextualizados e em cadeias de pensamento se mostrou especialmente eficaz, indicando que informações mais detalhadas e estruturadas podem orientar os modelos a produzir saídas mais consistentes.

Na segunda fase, a integração das saídas dos modelos LLaMA e Gemini em um modelo supervisionado demonstrou que as informações geradas por esses grandes modelos de linguagem podem contribuir significativamente para a tarefa de correção automática. A introdução de métricas derivadas, aliadas às características extraídas por ferramentas como NILC-Metrix, ampliou a compreensão das redações avaliadas e forneceu uma base sólida para melhorias futuras na área.

Adicionalmente, a construção de um dataset estendido, com a inclusão de redações nota mil, contribui para o avanço da pesquisa em correção automática de textos em português. Este recurso não apenas enriquece a base de dados disponível, mas também corrige desequilíbrios nas distribuições de notas observadas nos conjuntos anteriores, favorecendo análises mais robustas e abrangentes.

Apesar dos avanços, ainda há desafios a serem superados, como a escassez de redações com notas muito baixas e a necessidade de modelos mais transparentes e interpretáveis. O estudo aponta caminhos promissores para futuras investigações, como a execução dos experimentos mais vezes para garantir maior robustez estatística, a exploração de outras técnicas de engenharia de prompt e a expansão do uso de dados multimodais.

Em suma, este trabalho reafirma o potencial dos grandes modelos de linguagem para transformar o campo da avaliação de textos, oferecendo ferramentas mais eficientes e aces-

síveis para apoiar estudantes e educadores em larga escala. Acredita-se que os resultados apresentados aqui não apenas contribuem para o desenvolvimento da área de correção automática, mas também fortalecem a base para a construção de sistemas mais justos, precisos e inclusivos no futuro.

Apêndice A

Experimento 6 - Prompt A Completo

Experimento 3 - Prompt A

Conheça as cinco competências cobradas pelo Inep na redação do ENEM:

1. Domínio da escrita formal da língua portuguesa

É avaliado se a redação do participante está adequada às regras de ortografia, como acentuação, ortografia, uso de hífen, emprego de letras maiúsculas e minúsculas e separação silábica. Ainda são analisadas a regência verbal e nominal, concordância verbal e nominal, pontuação, paralelismo, emprego de pronomes e crase.

São seis níveis de desempenho:
200 pontos - Demonstra excelente domínio da modalidade escrita formal da língua portuguesa e de escolha de registro. Desvios gramaticais ou de convenções da escrita serão aceitos somente como excepcionalidade e quando não caracterizarem reincidência.
160 pontos - Demonstra bom domínio da modalidade escrita formal da língua portuguesa e de escolha de registro, com poucos desvios gramaticais e de convenções de escrita.
120 pontos - Demonstra domínio mediano da modalidade escrita formal da língua portuguesa e de escolha de registro, com alguns desvios gramaticais e de convenções da escrita.
80 pontos - Demonstra domínio insuficiente da modalidade escrita formal da língua portuguesa, com muitos desvios gramaticais, de escolha de registro e de convenções da escrita.
40 pontos - Demonstra domínio precário da modalidade escrita formal da língua portuguesa, de forma sistemática, com diversificados e frequentes desvios gramaticais, de escolha de registro e de convenções da escrita.
0 ponto - Demonstra desconhecimento da modalidade escrita formal da língua

portuguesa.

—-

2. Compreender o tema e não fugir do que é proposto

Avalia as habilidades integradas de leitura e de escrita do candidato. O tema constitui o núcleo das ideias sobre as quais a redação deve ser organizada e é caracterizado por ser uma delimitação de um assunto mais abrangente.

Eis os seis níveis de desempenho:
200 pontos - Desenvolve o tema por meio de argumentação consistente, a partir de um repertório sociocultural produtivo e apresenta excelente domínio do texto dissertativo-argumentativo.
160 pontos - Desenvolve o tema por meio de argumentação consistente e apresenta bom domínio do texto argumentativo-dissertativo, com proposição, argumentação e conclusão
120 pontos - Desenvolve o tema por meio de argumentação previsível e apresenta domínio mediano do texto dissertativo-argumentativo, com proposição, argumentação e conclusão
80 pontos - Desenvolve o tema recorrendo à cópia de trechos de textos motivadores ou apresenta domínio insuficiente do texto dissertativo-argumentativo, não atendendo à estrutura com proposição, argumentação e conclusão.
40 pontos - Apresenta o assunto, tangenciando o tema, ou demonstra domínio precário do texto dissertativo-argumentativo, com traços constantes de outros tipos textuais. 0 ponto - Fuga ao tema/não atendimento à estrutura dissertativo-argumentativa. Nestes casos a redação recebe nota zero e é anulada.

—-

3. Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista

O candidato precisa elaborar um texto que apresente, claramente, uma ideia a ser defendida e os argumentos que justifiquem a posição assumida em relação à temática da proposta da redação. Trata da coerência e da plausibilidade entre as ideias apresentadas no texto, o que é garantido pelo planejamento prévio à escrita, ou seja, pela elaboração de um projeto de texto.

Eis os seis níveis de desempenho:

200 pontos - Apresenta informações, fatos e opiniões relacionados ao tema proposto, de forma consistente e organizada, configurando autoria, em defesa de um ponto de vista.
160 pontos - Apresenta informações, fatos e opiniões relacionados ao tema, limitados aos argumentos dos textos motivadores e pouco organizados, em defesa de um ponto de vista.

120 pontos - Apresenta informações, fatos e opiniões relacionados ao tema, limitados aos argumentos dos textos motivadores e pouco organizados, em defesa de um ponto de vista.
80 pontos - Apresenta informações, fatos e opiniões relacionados ao tema, mas desorganizados ou contraditórios e limitados aos argumentos dos textos motivadores, em defesa de um ponto de vista.
40 pontos - Apresenta informações, fatos e opiniões pouco relacionados ao tema ou incoerentes e sem defesa de um ponto de vista.
0 ponto - Apresenta informações, fatos e opiniões não relacionados ao tema e sem defesa de um ponto de vista.

—- 4. Conhecimento dos mecanismos linguísticos necessários para a construção da argumentação

São avaliados itens relacionados à estruturação lógica e formal entre as partes da redação. A organização textual exige que as frases e os parágrafos estabeleçam entre si uma relação que garanta uma sequência coerente do texto e a interdependência entre as ideias.

Preposições, conjunções, advérbios e locuções adverbiais são responsáveis pela coesão do texto porque estabelecem uma inter-relação entre orações, frases e parágrafos. Cada parágrafo será composto por um ou mais períodos também articulados. Cada ideia nova precisa estabelecer relação com as anteriores.

Abaixo, seguem os seis níveis de desempenho:
200 pontos - Articula bem as partes do texto e apresenta repertório diversificado de recursos coesivos.
160 pontos - Articula as partes do texto, com poucas inadequações, e apresenta repertório diversificado de recursos coesivos.
120 pontos - Articula as partes do texto, de forma mediana, com inadequações, e apresenta repertório pouco diversificado de recursos coesivos.
80 pontos - Articula as partes do texto, de forma insuficiente, com muitas inadequações e apresenta repertório limitado de recursos coesivos.
40 pontos - Articula as partes do texto de forma precária.
0 ponto - Não articula informações.

—-
5 – Proposta de Intervenção
Na última competência, o candidato deve escrever uma proposta de intervenção para a temática, ou seja, apontar uma iniciativa para enfrentar o problema proposto. Para a proposta de intervenção ser considerada completa, é preciso ter cinco elementos, sendo eles: agente, ação, meio, finalidade e detalhamento, além de estar conforme os direitos humanos.

Os seis níveis de desempenho da Competência 5 são os seguintes:

200 pontos - Elabora muito bem proposta de intervenção, detalhada, relacionada ao tema e articulada à discussão desenvolvida no texto.
160 pontos - Elabora bem proposta de intervenção relacionada ao tema e articulada à discussão desenvolvida no texto.
120 pontos - Elabora, de forma mediana, proposta de intervenção relacionada ao tema e articulada com a discussão desenvolvida no texto.
80 pontos - Elabora, de forma insuficiente, proposta de intervenção relacionada ao tema, ou não articulada com a discussão desenvolvida no texto.
40 pontos - Apresente proposta de intervenção vaga, precária ou relacionada apenas ao assunto.
0 ponto - Não apresenta proposta de intervenção ou apresenta proposta não relacionada ao tema ou ao assunto.

—- Motivos que podem zerar a redação do Enem em todas as competências:

Fuga total do tema;
Desrespeito ao formato dissertativo-argumentativo;
Parte da redação desconectada deliberadamente do tema;
Apresentação de impropérios, desenhos e outras formas propositais de anulação, como números ou sinais fora do texto;
Texto redigido predominante ou integralmente em língua estrangeira;

Corrija a redação abaixo seguindo o método de avaliação do ENEM. Forneça apenas as notas quantitativas entre 0 e 200 para cada competência, e a nota final, no seguinte formato:[C1,C2,C3,C4,C5,Total], onde "Total"é a soma das notas das cinco competências.

Tema:
“““
<TEMA>
”””

Textos de suporte:
“““
<SUPORTE>
”””

Redação:
“““
<REDAÇÃO>
”””

Não forneça texto na resposta, apenas as notas numéricas para competência e o

total.

Anexo A

Descrição Nilc-Metrix

Métrica	Descrição
adjective_ratio	Proporção de adjetivos em relação às palavras do texto.
adverbs	Proporção de advérbios em relação às palavras do texto.
content_words	Proporção de palavras de conteúdo (substantivos, verbos, adjetivos e advérbios).
flesch	Índice de legibilidade Flesch, indicando facilidade de leitura.
function_words	Proporção de palavras funcionais (artigos, preposições, pronomes, etc.).
sentences_per_paragraph	Média de sentenças por parágrafo.
syllables_per_content_word	Média de sílabas por palavra de conteúdo.
words_per_sentence	Média de palavras por sentença.
noun_ratio	Proporção de substantivos em relação às palavras do texto.
paragraphs	Número de parágrafos no texto.
sentences	Número total de sentenças no texto.
words	Número total de palavras no texto.
pronoun_ratio	Proporção de pronomes em relação às palavras do texto.
verbs	Proporção de verbos em relação às palavras do texto.
logic_operators	Proporção de operadores lógicos (e.g., "e", "ou", "se").
and_ratio	Proporção de ocorrências da conjunção "e".
if_ratio	Proporção de ocorrências da conjunção "se".
or_ratio	Proporção de ocorrências da conjunção "ou".
negation_ratio	Proporção de palavras de negação (e.g., "não").

cw_freq	Frequência média das palavras de conteúdo no texto.
cw_freq_brwac	Frequência média das palavras de conteúdo no corpus BrWaC.
cw_freq_bra	Frequência média das palavras de conteúdo no corpus Brasileiro.
min_cw_freq	Frequência mínima de palavras de conteúdo no texto.
min_cw_freq_brwac	Frequência mínima de palavras de conteúdo no BrWaC.
min_cw_freq_bra	Frequência mínima de palavras de conteúdo no corpus Brasileiro.
freq_brwac	Frequência média geral de palavras no BrWaC.
freq_bra	Frequência média geral de palavras no corpus Brasileiro.
hypernyms_verbs	Média de hiperônimos por verbo.
brunet	Índice de Brunet, usado para medir legibilidade.
honore	Estatística de Honoré para medir a diversidade lexical.
personal_pronouns	Proporção de pronomes pessoais em relação ao total de palavras.
ttr	Type-Token Ratio, medindo diversidade lexical.
conn_ratio	Proporção de conectivos no texto.
add_neg_conn_ratio	Proporção de conectivos aditivos negativos.
add_pos_conn_ratio	Proporção de conectivos aditivos positivos.
adjectives_ambiguity	Média de ambiguidade dos adjetivos.
adverbs_ambiguity	Média de ambiguidade dos advérbios.
nouns_ambiguity	Média de ambiguidade dos substantivos.
verbs_ambiguity	Média de ambiguidade dos verbos.
yngve	Índice de Yngve, que mede complexidade sintática.
frazier	Índice de Frazier, relacionado ao processamento sintático.
dep_distance	Distância média de dependências sintáticas.
cross_entropy	Entropia cruzada de sentenças no texto.
content_density	Densidade de palavras de conteúdo no texto.
lsa_adj_mean	Média de similaridade semântica entre sentenças adjacentes.
lsa_adj_std	Desvio padrão da similaridade semântica entre sentenças adjacentes.
lsa_all_mean	Média de similaridade semântica entre todas as sentenças.
lsa_all_std	Desvio padrão da similaridade semântica entre todas as sentenças.

lsa_givenness_mean	Similaridade semântica de informações prévias em sentenças.
lsa_span_mean	Média de similaridade semântica no intervalo de sentenças.
lsa_span_std	Desvio padrão da similaridade semântica no intervalo de sentenças.
negative_words	Proporção de palavras com polaridade negativa.
positive_words	Proporção de palavras com polaridade positiva.
ratio_function_to_content_words	Proporção de palavras funcionais em relação às palavras de conteúdo.

Tabela A.1: *Descrição das métricas de NILC-Metrix S. E. Leal et al., 2023.*

Referências

[AL. 2023] Sidney Leal et AL. *NILC-Metrix: assessing the complexity of written and spoken language in Brazilian Portuguese*. 2023. URL: https://arxiv.org/pdf/2201.03445.pdf (citado na pg. 11).

[AMORIM e VELOSO 2017] Everton AMORIM e Adriano VELOSO. "Automatic essay scoring in brazilian portuguese". Em: *Proceedings of the Brazilian Symposium on Artificial Intelligence*. 2017, pgs. 231–245 (citado nas pgs. 7, 43, 44).

[AMORIM, VELOSO e ARAÚJO 2018] Everton AMORIM, Adriano VELOSO e Jéssica ARAÚJO. "Addressing bias in automatic essay scoring systems for the enem". Em: *International Journal of Educational Technology* 8.2 (2018), pgs. 89–102 (citado na pg. 8).

[ATTALI 2013] Yigal ATTALI. "What can automated essay scoring do?" Em: *Educational Measurement: Issues and Practice* 32.4 (2013), pgs. 38–50 (citado na pg. 7).

[ATTALI e BURSTEIN 2004] Yigal ATTALI e Jill BURSTEIN. "Automated essay scoring with e-rater v. 2". Em: *Journal of Technology, Learning, and Assessment* 4.3 (2004), pgs. 3–30 (citado na pg. 7).

[BURSTEIN *et al.* 2013] Jill BURSTEIN, Martin CHODOROW e Claudia LEACOCK. *Automated essay scoring: A cross-disciplinary perspective*. Routledge, 2013 (citado na pg. 7).

[CANDIDO JR *et al.* 2009] Arnaldo CANDIDO JR *et al.* "Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese". Em: *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*. 2009, pgs. 34–42 (citado na pg. 10).

[C.-M. CHEN e CHENG 2008] Chun-Mei CHEN e Wei-Yuan CHENG. "Exploring the use of automated writing evaluation in english as a foreign language". Em: *Computers and Education* 51.2 (2008), pgs. 1174–1184 (citado na pg. 7).

[T. CHEN e GUESTRIN 2016] Tianqi CHEN e Carlos GUESTRIN. "Xgboost: a scalable tree boosting system". Em: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pgs. 785–794 (citado na pg. 11).

[T. Chen e He 2015] Tianqi Chen e Tong He. "Xgboost: reliable large-scale machine learning". Em: *The 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015 (citado na pg. 11).

[Chomsky 1957] Noam Chomsky. "Logical structure in language". Em: *Journal of the American Society for Information Science* 8.4 (1957), pg. 284 (citado na pg. 5).

[Cohen 1968] Jacob Cohen. "Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit". Em: *Psychological Bulletin* 70.4 (1968), pgs. 213–220 (citado na pg. 15).

[Condon 2013] William Condon. "Large-scale assessment, locally-developed measures, and automated scoring: challenges and opportunities". Em: *Assessing Writing* 18.1 (2013), pgs. 100–121 (citado na pg. 7).

[Costa 2024] Daniel Costa. *Automated ENEM Essays Correction Using ChatBots*. Accessed: 30 November 2024. 2024. URL: https://github.com/dslcosta1/AutomatedENEMEssaysCorrectionUsingChatBots (citado nas pgs. 29, 43).

[Cotos 2014] Elena Cotos. "Awe tools in writing instruction: developing an evaluation approach". Em: *Journal of Writing Research* 6.2 (2014), pgs. 141–170 (citado na pg. 7).

[Devlin 2018] Jacob Devlin. "Bert: pre-training of deep bidirectional transformers for language understanding". Em: *arXiv preprint arXiv:1810.04805* (2018) (citado nas pgs. 5, 6).

[Dikli 2006] Semire Dikli. "An overview of automated scoring of essays". Em: *The Journal of Technology, Learning and Assessment* 5.1 (2006) (citado na pg. 7).

[Dikli e Bleyle 2014] Semire Dikli e Susan Bleyle. "Automated essay scoring feedback: how do students engage?" Em: *Assessing Writing* 22 (2014), pgs. 1–17 (citado na pg. 7).

[Dubey *et al.* 2024] Abhimanyu Dubey *et al.* "The llama 3 herd of models". Em: *arXiv preprint arXiv:2407.21783* (2024) (citado na pg. 10).

[Felpi 2019] Lucas Felpi. *Cartilha Redação a Mil*. 2019. URL: https://www.lucasfelpi.com.br/redamil (citado nas pgs. 12, 13).

[Friedman 2001] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". Em: *Annals of statistics* 29.5 (2001), pgs. 1189–1232 (citado na pg. 11).

[g1 2024] g1. *Redação do Enem: leia 100 textos que tiraram nota mil*. 2024. URL: https://g1.globo.com/educacao/enem/2024/noticia/2024/10/26/redacao-do-enem-leia-100-textos-que-tiraram-nota-mil.ghtml (citado na pg. 12).

[Globo 2019] O Globo. *Enem: leia 44 redações que tiraram nota 1000 nos últimos anos*. 2019. URL: https://oglobo.globo.com/brasil/educacao/enem-e-vestibular/enem-leia-44-redacoes-que-tiraram-nota-1000-nos-ultimos-anos-23993654 (citado na pg. 12).

[Hartmann e Aluísio 2020] Nathan Siegle Hartmann e Sandra Maria Aluísio. "Adaptação lexical automática em textos informativos do português brasileiro para o ensino fundamental". Em: *Linguamática* 12.2 (2020), pgs. 3–27 (citado na pg. 10).

[Jurafsky 2000] Daniel Jurafsky. *Speech and language processing*. 2000 (citado na pg. 5).

[Koltovskaia 2020] Svetlana Koltovskaia. "Scaffolding learner autonomy with automated writing evaluation: integrating criterion in english writing courses". Em: *Computers and Composition* 55 (2020), pg. 102544 (citado na pg. 7).

[S. Leal e team 2021] Sidney Leal e team. *NILC-Metrix*. 2021. URL: https://github.com/nilc-nlp/nilcmetrix (citado na pg. 11).

[S. E. Leal *et al.* 2023] Sidney Evaldo Leal, Magali Sanchez Duran, Carolina Evaristo Scarton, Nathan Siegle Hartmann e Sandra Maria Aluísio. "Nilc-metrix: assessing the complexity of written and spoken language in brazilian portuguese". Em: *Lang Resources & Evaluation* (2023). URL: https://doi.org/10.1007/s10579-023-09693-w (citado nas pgs. 10, 55).

[Levenshtein 1966] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". Em: *Soviet physics doklady* 10.8 (1966), pgs. 707–710 (citado na pg. 13).

[Marinho *et al.* 2022] Mateus Marinho, José Lucas Silva e Daniel Moreira. "Essay-br: a large-scale corpus for essay scoring in brazilian portuguese". Em: *Brazilian Journal of Artificial Intelligence* 18.3 (2022), pgs. 123–137 (citado na pg. 7).

[Mayer 2023] José Lucas Silva Mayer. "Avaliação Automática de Redações no Modelo do ENEM por meio do fine-tuning do BERTimbau". Trabalho de Formatura Supervisionado, Supervisor: Prof. Dr. Denis Deratani Mauá, Coorientador: Igor Cataneo Silveira. Monografia de Bacharelado. São Paulo, Brasil: Universidade de São Paulo, Instituto de Matemática e Estatística, 2023. URL: https://linux.ime.usp.br/~josemayer/mac0499/ (citado nas pgs. 43, 44).

[Mikolov 2013] Tomas Mikolov. "Efficient estimation of word representations in vector space". Em: *arXiv preprint arXiv:1301.3781* 3781 (2013) (citado na pg. 5).

[Mizumoto e Eguchi 2023] Atsushi Mizumoto e Masaki Eguchi. "Exploring the potential of using an ai language model for automated essay scoring". Em: *Research Methods in Applied Linguistics* 2.2 (2023), pg. 100050 (citado na pg. 3).

[Navarro 2001] Gonzalo Navarro. "A guided tour to approximate string matching". Em: *ACM computing surveys (CSUR)* 33.1 (2001), pgs. 31–88 (citado na pg. 13).

[Page 1967] Ellis Batten Page. "The application of the electronic computer to the scoring of essays". Em: *ETS Research Bulletin Series* 1967.2 (1967), pgs. i–47 (citado na pg. 7).

[Pennington *et al.* 2014] Jeffrey Pennington, Richard Socher e Christopher D Manning. "Glove: global vectors for word representation". Em: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pgs. 1532–1543 (citado na pg. 5).

[Radford 2018] Alec Radford. "Improving language understanding by generative pre-training". Em: (2018) (citado nas pgs. 5, 6).

[Scarton e Aluisio 2010] Carolina Scarton e Sandra Maria Aluisio. "Coh-metrix-port: a readability assessment tool for texts in brazilian portuguese". Em: *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language, Extended Activities Proceedings, PROPOR*. Vol. 10. 1. sn. 2010 (citado na pg. 10).

[I. Silveira *et al.* 2024] Igor Silveira, André Barbosa, Daniel Costa e Denis Mauá. "Investigating universal adversarial attacks against transformers-based automatic essay scoring systems". Em: *Proceedings of the 24th Brazilian Conference on Intelligent Systems (BRACIS)*. São Paulo, Brazil: Springer, 2024. doi: 10.1007/XXXXXXXXX (citado na pg. 44).

[I. C. Silveira *et al.* 2024] Igor Cataneo Silveira, André Barbosa e Denis Deratani Mauá. "A new benchmark for automatic essay scoring in Portuguese". Em: *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*. Ed. por Pablo Gamallo *et al.* Santiago de Compostela, Galicia/Spain: Association for Computational Lingustics, mar. de 2024, pgs. 228–237. url: https://aclanthology.org/2024.propor-1.23 (citado na pg. 11).

[Team, Anil *et al.* 2023] Gemini Team, Rohan Anil *et al.* "Gemini: a family of highly capable multimodal models". Em: *arXiv preprint arXiv:2312.11805* (2023) (citado na pg. 9).

[Team, Georgiev *et al.* 2024] Gemini Team, Petko Georgiev *et al.* "Gemini 1.5: unlocking multimodal understanding across millions of tokens of context". Em: *arXiv preprint arXiv:2403.05530* (2024) (citado na pg. 9).

[Vaswani 2017] A Vaswani. "Attention is all you need". Em: *Advances in Neural Information Processing Systems* (2017) (citado nas pgs. 5, 6).

[Wei *et al.* 2022] Jason Wei *et al.* "Chain-of-thought prompting elicits reasoning in large language models". Em: *arXiv preprint arXiv:2201.11903* (2022). url: https://arxiv.org/abs/2201.11903 (citado na pg. 22).

[White *et al.* 2023] Jules White *et al.* "A prompt pattern catalog to enhance prompt engineering with chatgpt". Em: *arXiv preprint arXiv:2302.11382* (2023) (citado nas pgs. 3, 19).

[Zhou *et al.* 2023] Wei Zhou, Sheng Zhang, Hoifung Poon e Mingwei Chen. "Context-faithful prompting for large language models". Em: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pgs. 14544–14556. URL: https://aclanthology.org/2023.findings-emnlp.968/ (citado nas pgs. 24, 25).

[Zverovich 2023] Artem Zverovich. *Python-Levenshtein: The Levenshtein Python C extension module*. https://github.com/ztane/python-Levenshtein. Acessado em 15 de outubro de 2023. 2023 (citado na pg. 13).

Índice Remissivo