UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GUILHERME SALIM MONTEIRO DE CASTRO PAES
Orientador: Pedro Henrique Lopes Silva

DETECÇÃO DE TEXTOS GERADOS POR LLMS EM PORTUGUÊS

Ouro Preto, MG
2024

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GUILHERME SALIM MONTEIRO DE CASTRO PAES

DETECÇÃO DE TEXTOS GERADOS POR LLMS EM PORTUGUÊS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Pedro Henrique Lopes Silva

Ouro Preto, MG
2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

P126d Paes, Guilherme Salim Monteiro de Castro.
Detecção de textos gerados por LLMs em português. [manuscrito] / Guilherme Salim Monteiro de Castro Paes. - 2025.
49 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Pedro Henrique Lopes Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Inteligência artificial. 2. Aprendizado do computador. 3. Modelos de Linguagem de Grande Escala. I. Silva, Pedro Henrique Lopes. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.8

Bibliotecário(a) Responsável: Soraya Fernanda Ferreira e Souza - SIAPE: 1.763.787

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Guilherme Salim Monteiro de Castro Paes

Detecção de textos gerados por LLMs em português

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 31 de Março de 2025.

Membros da banca

Pedro Henrique Lopes Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Amanda da Silva Oliveira (Examinadora) - Mestre - Blip
Augusto Ferreira Guilarducci (Examinador) - Bacharel - PPGCC - UFOP

Pedro Henrique Lopes Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 31/03/2025.

Documento assinado eletronicamente por **Pedro Henrique Lopes Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 31/03/2025, às 15:36, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0885587** e o código CRC **F2D19DBF**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Resumo

Com o aumento gradativo da disponibilização e uso de modelos de Inteligência Artificial (IA) generativa, levanta-se a preocupação com os perigos do seu mau uso. Apesar de terem sido desenvolvidos para atuar como ferramentas que facilitam o cotidiano popular, problemas como plágio e desinformação acabam escalando em razão do uso indevido ou mal-intencionado desses modelos. Por serem recentes e extremamente poderosos, ainda há certa dificuldade em identificar textos gerados pelos chamados Large Language Models (LLMs). Com isso em mente, propõe-se um *dataset* com exemplos de textos humanos, artificiais e textos originalmente humanos porém reescritos por IA. Além disso, foram propostos um conjunto de cinco algoritmos classificadores baseados nos LLMs da família Llama e BERT e uma *Recurrent Neural Network*, baseada em camadas LSTM bi-direcionais. Os classificadores atingiram resultados positivos, alcançando acurácias de até 98,18% e 97,7%, nas classificações de duas (escrito e não escrito por uma LLM) e três classes (escrito, não escrito e reescrito por uma LLM), respectivamente, no conjunto de teste proposto.

Palavras-chave: *Large Language Models*. Aprendizado de Máquina. Classificação.

Abstract

With the gradual increase in the availability and use of generative Artificial Intelligence (AI) models, concerns about the dangers of their misuse are being raised. Although they were developed as tools to facilitate everyday life, issues such as plagiarism and misinformation have escalated due to the improper or malicious use of these models. Because they are recent and extremely powerful, there is still some difficulty in identifying texts generated by so-called Large Language Models (LLMs). With this in mind, we propose a dataset containing examples of human-written texts, AI-generated texts, and originally human-written texts that have been rewritten by AI. Additionally, a set of five classification algorithms based on Llama and BERT family LLMs, as well as a Recurrent Neural Network based on bi-directional LSTM layers, was proposed. The classifiers achieved positive results, reaching accuracies of up to 98.18% and 97.7% in binary classification (written or not written by an LLM) and three-class classification (written, not written, and rewritten by an LLM), respectively, on the proposed test set.

Keywords: *Large Language Models. Machine Learning. Classification.*

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

API *Application Programming Interface*. iv, 12, 20–22, 28

BERT *Bidirectional Encoder Representations from Transformers*. 2, 16, 22, 23, 28, 33

GPT Generative Pre-trained Transformer. 1, 13, 23

IA Inteligência Artificial. 1–3, 8, 9, 13, 15, 17–19, 23, 29–31, 34

LLM *Large Language Model*. iv, vii, 1–3, 5, 8, 10–20, 22, 23, 28, 33–35

LoRA *Low-Rank Adaptation*. iv, vii, 13–15, 23, 24

LSTM *Long Short-Term Memory*. 24

NLP *Natural Language Processing*. 5

PLN Processamento de Linguagem Natural. iv, vii, 1–3, 5–8, 10, 11, 13, 15, 16, 23

RNN *Recurrent Neural Network*. iv, 1, 2, 24–26, 28, 29, 33, 35

Sumário

1 Introdução

Em anos recentes, tornou-se notável o grande avanço no desenvolvimento de modelos de Inteligência Artificial (IA) generativa como o ChatGPT[1] e Gemini[2]. Os modelos atuais, chamados de *Large Language Models* (LLMs), devido à grande quantidade de parâmetros que possuem, destoam de métodos tradicionais como *Recurrent Neural Networks* (RNNs) e também em virtude do grande aumento da capacidade de compreensão e geração dos mais diversos tipos de tarefas, como responder perguntas, gerar textos, reconhecer e produzir diferentes tipos de mídia, como imagens, vídeos e áudios, além de gerar códigos em múltiplas linguagens (Gemini Team Group, 2024).

Somada à melhora da performance de tais modelos, outro ponto de mudança em tempos recentes foi a criação e disponibilização pública de ferramentas que fazem uso dos *Large Language Models* (LLMs). Tais ferramentas, devido às suas capacidades de identificar, criar e rever conteúdos como textos e imagens, fornecem à população uma alternativa de fácil acesso e uso para automatizar e melhorar tarefas cotidianas como escrever grandes textos (ROSSONI; CHAT, 2022) e até auxiliar no ensino (SANT et al., 2023). Comumente, os LLMs são utilizados no desenvolvimento de *chatbots*, ferramentas que mimetizam interações entre humanos e assistentes, respondendo perguntas e simulando conversas em linguagem natural. Um exemplo da alta popularização destes produtos é o ChatGPT, *chatbot* da OpenAI, que utiliza a família de modelos Generative Pre-trained Transformer (GPT) e registrou um milhão de usuários apenas uma semana após seu lançamento (VALLANCE, 2022).

Além das melhorias para a produtividade de tarefas do cotidiano das pessoas, a disponibilização pública de modelos de linguagem pré-treinados também é responsável por facilitar e produzir grandes avanços na área de Processamento de Linguagem Natural (PLN). Devido à grande quantidade de informações com as quais são treinados, estes modelos possuem grande capacidade de generalização de contexto, se mostrando uma alternativa mais eficiente do que treinar pequenos modelos de contexto específico em

[1] OpenAI, *ChatGPT*, <https://openai.com/chatgpt>. Acesso em: 03/04/2025

[2] Google DeepMind, *Gemini AI*, <https://deepmind.google/technologies/gemini/>. Acesso em: 03/04/2025

casos com poucos dados de treino disponíveis (YUAN et al., 2023).

Apesar desta mudança de paradigma se mostrar promissora e inovadora, o mau uso destas ferramentas pode gerar problemas como plágio não identificado e, mais preocupantemente, disseminação de informação científica falsa a partir de artigos gerados em grande parte (ou em sua completude) por IA. O real perigo destes artigos não autorais é a possibilidade de haver artigos e pesquisas baseados neles, que consequentemente conteriam também informação falsa (ELSE, 2023) e se mostram prejudiciais para a sociedade, especialmente quando extrapolados para áreas de pesquisa sensíveis como a Medicina (DONATO; ESCADA; VILLANUEVA, 2023) que lidam com a saúde e segurança das pessoas. Desta forma, fica evidente a necessidade de avanços científicos na área de detecção do uso indevido e mal intencionado destas ferramentas.

Outro fator agravante da falta de métodos confiáveis de detectar textos artificiais é a falta de estudos da área relativos à língua portuguesa. Fatores socioeconômicos, somados à dificuldade intrínseca da análise gramatical do português, são os principais contribuintes para a ausência de estudos neste escopo (CASELI; NUNES, 2023). Uma vez que a língua é a quarta mais falada no mundo (CAMOES, 2022), mostram-se necessários os avanços na criação de conjuntos de dados e classificadores para esta tarefa, sob o escopo do idioma.

Visto que o avanço no desenvolvimento dos LLMs é recente e de larga escala, a classificação de textos gerados por estes modelos possui dificuldades como superar sua alta performance em tarefas textuais, além da falta de dados que permitam capturar as características de tais textos. Porém, uma vez que os *Large Language Models* possuem alta capacidade em diversas tarefas de PLN além da geração de textos, levanta-se a hipótese da eficácia de seu uso como classificador de textos gerados por outros LLMs. Desta forma, visando abordar o problema do uso não identificado de modelos de texto generativos, em especial em português, propõe-se neste trabalho a criação de um conjunto de dados composto por exemplos de texto humano e texto artificial, para ser usado em métodos de detecção de textos gerados por LLM. Além disso, foram propostos modelos classificadores, baseados em LLMs das famílias Llama e *Bidirectional Encoder Representations from Transformers* (BERT), além de uma RNN, aplicados ao conjunto de dados definido, que obtiveram acurácias de 98,18% e 97,7%, nas classificações de duas (escrito e não escrito por LLM) e três classes (escrito, não escrito e reescrito por LLM), respectivamente, do

conjunto de teste proposto.

1.1 Justificativa

O desenvolvimento de métodos de detecção de textos gerados por LLMs se mostra uma tarefa necessária no âmbito científico e acadêmico, uma vez que permite o controle da disseminação de informação incorreta em artigos falsos gerados por IA e de plágio em trabalhos científicos e escolares (IBRAHIM et al., 2023).

Além disso, um dos desafios para a realização desta tarefa é a dificuldade de aplicação de métodos multilinguísticos, ou seja, métodos que possuam efetividade comprovada em múltiplas línguas, sendo comumente desenvolvidos e testados apenas para o inglês (WU et al., 2023), demonstrando assim, ser uma tarefa em aberto no campo do PLN em português.

1.2 Objetivos

O principal objetivo deste trabalho é propor a criação de um conjunto de dados e uma abordagem aplicáveis para detecção de textos gerados por LLM na língua portuguesa. Para tal, são propostos os seguintes objetivos específicos:

1. Fazer uma análise da bibliografia para definir as principais soluções atuais empregadas na literatura e relacioná-las com o português;
2. Definição e aplicação da metodologia para a produção do *dataset* visando que ele seja eficiente e bem balanceado;
3. Definição e implementação de algoritmos de classificação baseados em LLMs
4. Realização de testes dos classificadores;
5. Análise quantitativa e qualitativa dos resultados e definição de sua relevância;

1.3 Organização do Trabalho

O presente trabalho é organizado de acordo com a seguinte estrutura: Capítulo 1 apresenta a introdução do problema, justificativa e objetivos. Capítulo 2 aborda uma fundamentação teórica dos conceitos considerados mais relevantes para a compreensão do trabalho, além de uma revisão da literatura de acordo com trabalhos relacionados ao problema abordado. Já o Capítulo 3 define a metodologia que será aplicada para o desenvolvimento do trabalho. Enquanto que o Capítulo 4 traz os resultados obtidos após o desenvolvimento do trabalho e uma discussão sobre sua relevância. Por fim, o Capítulo 5 conclui o trabalho apresentado.

2 Revisão Bibliográfica

Este capítulo apresenta uma contextualização de conceitos importantes para a compreensão do texto na Seção 2.1, além de uma revisão de trabalhos previamente publicados cujos temas possuem relação com o objetivo deste trabalho, presente na Seção 2.2.

2.1 Fundamentação Teórica

Esta seção faz breves introduções e explicações aos diversos conceitos considerados relevantes que são abordados durante o desenvolvimento do trabalho. A Seção 2.1.1 aborda conceitos gerais sobre a área de Processamento de Linguagem Natural e suas dificuldades, enquanto que as Seção 2.1.2 e Seção 2.1.3 apresentam os conceitos técnicos que envolvem o problema de detecção de textos gerados por *Large Language Model*s (em tradução livre, modelos de linguagem de larga escala). Além disso, são definidas nas Seção 2.1.4 e Seção 2.1.5 conceitos específicos que serão aplicados durante o desenvolvimento do trabalho.

2.1.1 Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN) (ou *Natural Language Processing* (NLP), em inglês) é o nome dado para a área da computação que visa estudar as formas de tratar computacionalmente problemas relacionados às diferentes formas de entender, aprender e produzir conteúdo em linguagem humana (ou linguagem natural) seja ela por sons, palavras, textos, etc (HIRSCHBERG; MANNING, 2015). A Figura 2.1 demonstra os diferentes ramos os quais esta área pode se dividir devido à sua extensa gama de possibilidades, como recuperação de informação, tradução e interpretação de textos, análise de sentimentos, etc (VIEIRA; LOPES, 2010).

Para ser capaz de reconhecer e extrair informações da linguagem natural, é feita uma representação lógica da sentença. Para tal, primeiro é feito o processo de tokenização, que consiste em separar palavras e frases em subunidades denominadas *tokens* que podem

Figura 2.1 – Processamento de Linguagem Natural e suas diversas áreas.

Fonte: Retirada de <https://www.ontotext.com/blog/top-5-semantic-technology-trends-2017/> (adaptada). Acessado em: 23/04/2024.

ser lidas, reconhecidas e interpretadas por um computador. Após a tokenização, é feita a análise da linguagem em diversos níveis de conhecimento (definidos por Gonzalez e Lima (2003) de acordo com os conteúdos da Tabela 2.1), desde sua estrutura sintática até a extração de sua representação semântica, que pode ser interpretada como uma forma de representar o texto, definindo seu significado para um falante da língua. (ABEND; RAPPOPORT, 2017).

Tabela 2.1 – Níveis de conhecimento linguístico.

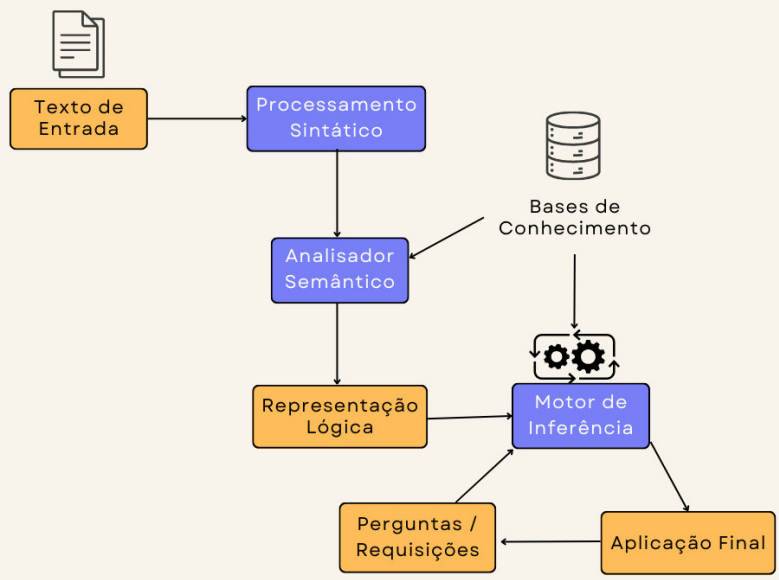
Nível de Conhecimento	**Definição**
Fonético e fonológico	Do relacionamento das palavras com os sons que produzem.
Morfológico	Da construção das palavras a partir de unidades de significado primitivas e de como classificá-las em categorias morfológicas.
Sintático	Do relacionamento das palavras entre si, cada uma assumindo seu papel estrutural nas frases, e de como as frases podem ser partes de outras, constituindo sentenças.
Semântico	Do relacionamento das palavras com seus significados e de como eles são combinados para formar os significados das sentenças.
Pragmático	Do uso de frases e sentenças em diferentes contextos, afetando o significado.

Fonte: (GONZALEZ; LIMA, 2003).

Além do processo de análise estrutural e semântica de uma frase, ainda pode ser necessário fazer diversos processamentos linguísticos como reconhecimento de entida-

des nomeadas, por exemplo, que consiste em identificar e classificar automaticamente entidades como nomes de pessoas, organizações, locais, datas e outros em um texto. Para realizar estes processamentos, se mostra necessária a utilização de uma chamada base de conhecimento. Tal base de conhecimento pode ser definida como uma base de dados que contém diferentes tipos de informações, regras, etc., que representam alguma forma de conhecimento (geral ou de domínio específico) que são considerados relevantes para a tarefa proposta (CASELI; NUNES, 2023). A Figura 2.2 apresenta um exemplo de um sistema de PLN que segue esta arquitetura.

Figura 2.2 – Exemplo de estrutura de um sistema de entendimento de Linguagem Natural.



Fonte: (CASELI; NUNES, 2023).

Aliada à dificuldade de representação lógica de uma sentença, outro grande desafio do PLN é lidar com a ambiguidade, ou seja, a falta da possibilidade de definir um significado único para uma determinada frase ou expressão (PINTO, 2015). Um dos maiores complicadores da ambiguidade é sua capacidade de se manifestar em qualquer nível de conhecimento durante a análise da sentença (JACKSON; MOULINIER, 2007), como exemplificado na Tabela 2.2. Normalmente, em um exemplo de comunicação utilizando linguagem natural, a maior parte dos casos de ambiguidade é facilmente reconhecida por um falante, seja por contexto, entonação ou até mesmo uso do senso comum; porém, representar tais valores computacionalmente se mostra um desafio.

Tabela 2.2 – Exemplos de frases com ambiguidades textuais

Frase	**Tipo de ambiguidade**	**Diferentes significados**
Eu li a notícia sobre a greve na faculdade.	Sintática	Pode significar que li a notícia enquanto estava na faculdade ou que li a notícia sobre a greve da faculdade.
Na minha casa, tem uma mangueira no jardim.	Léxica	A palavra mangueira pode representar tanto a árvore quanto o tubo utilizado para regagem.
Maria bateu em José com seu próprio caderno	Semântica	A palavra 'seu' pode significar que o caderno é de Maria ou de José.

Fonte: Elaborado pelo autor

O problema do desenvolvimento de técnicas de detecção de textos gerados por LLMs se encaixa no contexto de PLN uma vez que os textos que os modelos de IA generativa produzem são em linguagem natural e altamente similares a textos produzidos por humanos, desta forma, identificar os padrões destas escritas artificiais se torna um problema de análise das estruturas de conteúdo em linguagem natural.

2.1.2 Aprendizado de Máquina e Redes Neurais

Aprendizado de Máquina (*Machine Learning*, em inglês) é uma sub-área de estudo de IA definida por Alpaydin (2020) como uma área que estuda o desenvolvimento e aplicação de algoritmos para treinar um modelo computacional capaz de aprender padrões baseados em exemplos de dados de treinamento e experiências passadas. Segundo os autores, tais modelos podem ser preditivos, ou seja, fazem previsão de situações futuras baseadas nos padrões de exemplos passados, descritivos, que dizem respeito a modelos que ganham conhecimento a partir de dados, ou ambos.

Existem diferentes técnicas de aprendizado de máquina que podem ser aplicadas para diferentes objetivos e cada uma segue diferentes etapas (exemplificadas na Figura 2.3). As principais classes são:

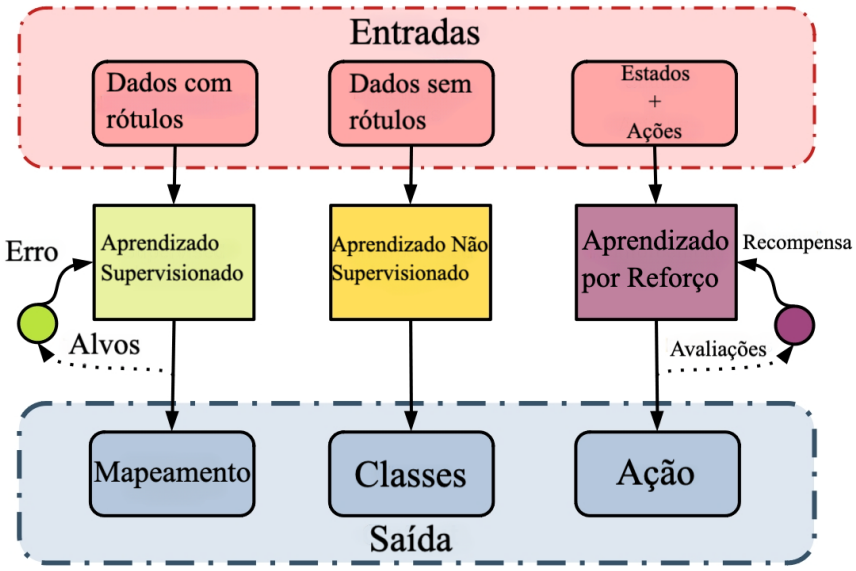
Aprendizado Supervisionado: Consiste em treinar um modelo com pares de valores de entrada e rótulo associado para que o modelo aprenda a mapear o valor de uma saída para sua entrada correta. Utilizada em problemas de classificação como detecção de imagens ou regressão, como previsão de ações do mercado;

Aprendizado Não-Supervisionado: Consiste em prover o modelo com dados de en-

trada sem fornecer rótulos e tendo como saída esperada do modelo dados com padrões similares entre seus exemplos. Utilizado em aplicações como sistemas de recomendação e segmentação de *marketing*;

Aprendizado por reforço: Consiste em posicionar o modelo (agente) em um ambiente e maximizar uma recompensa acumulada dada de acordo com o *feedback* de determinada ação no ambiente dado. Utilizado comumente em ferramentas de *chatbot* e aplicações de Inteligência Artificial em jogos.

Figura 2.3 – Exemplos das principais técnicas de *Machine Learning* com suas entradas e saídas.

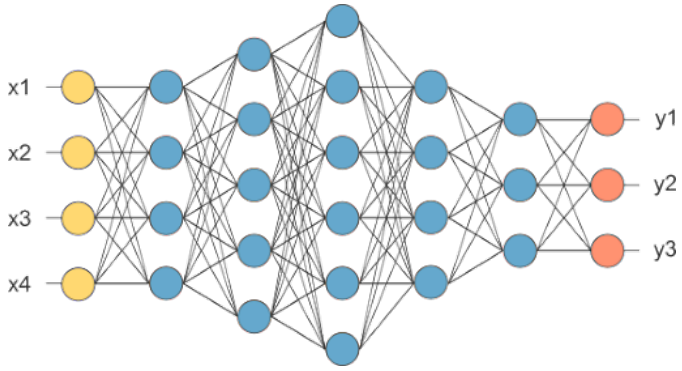


Fonte: Retirado de <https://starship-knowledge.com/supervised-vs-unsupervised-vs-reinforcement#What_is_reinforcement_learning> (adaptado). Acessado em 24/05/2024

No campo de aprendizado de máquina estão presentes métodos dos chamados aprendizados de representação, que consistem em métodos que permitem o modelo descobrir e aprender as representações úteis para extrair os padrões entre os exemplos de entrada (BENGIO; COURVILLE; VINCENT, 2013). Um método de aprendizado de representação é o chamado aprendizado em profundidade, definido como um método com múltiplos níveis de representação que se dão a partir da composição de módulos não

lineares (chamados neurônios) capazes de transformar a representação em um nível mais abstrato (LECUN; BENGIO; HINTON, 2015). A junção de vários neurônios interconectados em múltiplas camadas define uma Rede Neural Artificial (chamada desta forma pois simula o comportamento dos neurônios humanos) que são modelos de aprendizado de máquina capazes de aprender representações de alta especificidade em um dado de entrada. A Figura 2.4 detalha um exemplo da estrutura em camadas de um modelo de Rede Neural Artificial.

Figura 2.4 – Exemplo de arquitetura básica de uma Rede Neural Artificial.



Fonte: Retirado de <https://www2.decom.ufop.br/imobilis/fundamentos-de-redes-neurais/>. Acessado em 24/05/2024

A alta eficácia destas redes para diversos tipos de tarefas (LECUN; BENGIO; HINTON, 2015), fez com que fossem eventualmente usadas em tarefas de PLN e, mais recentemente, no desenvolvimento de modelos de geração de textos como os LLMs. Desta forma, no contexto de detecção de textos gerados por LLMs o aprendizado de máquina e as redes neurais constituem um fator importante para a compreensão não apenas dos métodos de detecção, mas também dos modelos que geram tais textos.

2.1.3 *Large Language Models* e geração de textos

O grande aumento recente na produção e consumo de ferramentas de *chatbots* passa pelos avanços no desenvolvimentos dos modelos que são o pilar deste tipo de sistema: *Large Language Model* (LLM). Os LLMs consistem em modelos matemáticos de geração de textos pela predição estatística feita a partir da distribuição de acordo com o escopo de dados que foram treinados (SHANAHAN, 2024). Em outras palavras, são modelos generativos uma vez que geram informação textual de acordo com as palavras

mais prováveis de ocorrerem dado o contexto do dado de entrada e dos *tokens* já gerados. Apesar de não ser o único método de geração de textos e de PLN, o desenvolvimento de LLMs é o atual estado da arte devido à sua alta capacidade de generalizar aspectos linguísticos complexos (BLANK, 2023).

A capacidade de tais modelos se dá devido à utilização da arquitetura de *Transformers* (VASWANI et al., 2017). Esta arquitetura é responsável pela transferência de conhecimento facilitada e eficiente entre unidades neurais (ZHOU et al., 2023). A eficiência de tal arquitetura ocorre pela utilização dos chamados mecanismos de atenção (VASWANI et al., 2017), que consistem em mecanismos que atribuem um peso a cada parte do dado de entrada, desta forma, o modelo de Redes Neurais pode se concentrar em aprender as partes consideradas mais importantes da entrada (ou com maior peso). Existem diversos tipos de mecanismo de atenção para diferentes problemas, porém o utilizado em LLMs é o chamado *self-attention* (ou auto-atenção, em tradução livre). Este mecanismo é responsável por conectar diferentes partes da sentença de entrada para criar uma representação da sentença livre dos limites posicionais das palavras no texto, permitindo que o modelo seja capaz de aprender palavras e características de uma sentença independentemente de sua posição relativa à outras partes da sentença (GALASSI; LIPPI; TORRONI, 2021).

A performance dos LLMs em tarefas distintas demonstra uma alta capacidade de adaptabilidade de tais modelos. Tal adaptabilidade permite que os modelos não precisem ter todos os pesos recalculados para serem aplicados. Métodos de treinamento e ajuste alternativos como Ajuste Fino, *Few-shot*, etc, definidos por Brown et al. (2020) de acordo com os conteúdos da Tabela 2.3, se mostram mais eficientes que um treinamento completo de um modelo mais simples (YUAN et al., 2023). A aplicação de um algoritmo de adaptação dos modelos, como exemplificado pela Figura 2.5, demonstra a capacidade de realizar diversas tarefas complexas como responder perguntas, completar textos, etc (CHANG et al., 2023).

2.1.4 Métodos de detecção de caixa preta e caixa branca

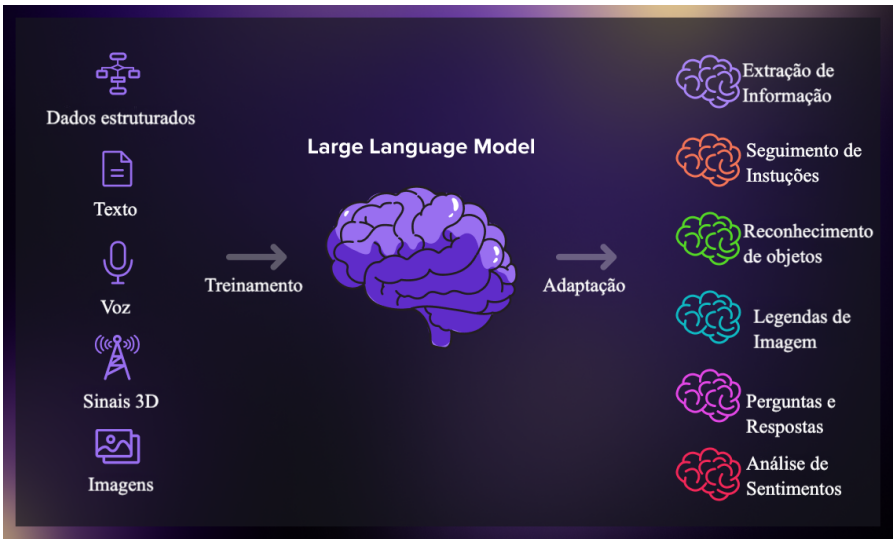
Quando se trata do estudo sobre o uso dos LLMs, existem diferentes formas de detectar um texto gerado por tal modelo baseadas em como se dá a interação entre um usuário final e o modelo. Tang, Chuang e Hu (2024) define as duas possíveis formas

Tabela 2.3 – Métodos de aprendizado em contexto.

Método de aprendizado em contexto	**Definição**
Ajuste Fino	Atualiza os pesos de um modelo pré-treinado a partir do treinamento do modelo em exemplos supervisionados do contexto específico.
Few-Shot	São passados ao modelo K exemplos da tarefa desejada durante a inferência, porém sem atualizar os pesos do modelo.
One-Shot	Similar ao *Few-Shot* porém com apenas 1 exemplo passado ao modelo, ou seja, K = 1.
Zero-Shot	Similar ao *Few-shot* porém invés de exemplos, é passada uma descrição em linguagem natural da tarefa ao modelo durante a inferência

Fonte: (BROWN et al., 2020).

Figura 2.5 – Exemplo de extrutura simplificada de treinamento e possibilidades de adaptação de LLMs.



Fonte: Retirado de <https://www.civilsdaily.com/news/crafting-safe-generative-ai-systems/> (adaptado). Acessado em 25/05/2024

de detectar texto gerado por um modelo como detecção de modelos de caixa preta (ou *black-box*, em inglês) e de caixa branca (*white-box*).

Modelos de caixa preta são definidos como modelos em que o usuário final não possui acesso ao modelo em si e a troca de informações se dá por meio de *Application Programming Interfaces* (APIs). As APIs são conjuntos de regras e protocolos que definem a comunicação e trocas de dados entre diferentes aplicações. Os métodos de detecção se baseiam puramente em treinar classificadores que extraiam o máximo de

informações relevantes da saída de tal modelo. Devido à falta de acesso ao modelo em si, apenas aos dados de entrada e saída, métodos de detecção de caixa preta são altamente dependentes da qualidade do conjunto de dados utilizado como entrada do modelo em questão (TANG; CHUANG; HU, 2024).

Métodos de detecção de caixa branca, são os métodos nos quais o usuário final tem acesso total ao modelo, ou seja, aos seus pesos e arquitetura. Dessa forma, diferentes estratégias podem ser aplicadas na geração de textos e no comportamento do modelo para identificar se uma saída foi gerada por ele. Para tal, podem ser utilizados métodos estatísticos que diferem entre textos gerados pelo modelo e textos feitos por humanos. O método mais comum atualmente é o de *watermarking* (marca d'água, em português), que consiste em dividir o vocabulário do modelo em duas listas diferentes (A e B, por exemplo) e, durante a geração do texto, fazer com que o modelo gere o máximo de palavras da lista A possível (WANG et al., 2024).

As diferenças entre os métodos de detecção de texto gerados por IA são tantas que extrapolam o campo teórico. Quanto maior acesso ao modelo (métodos caixa branca) mais controle se tem sobre seus resultados, influenciando diretamente na qualidade da detecção (TANG; CHUANG; HU, 2024). Dessa forma, métodos caixa branca têm, em geral, resultados de maior confiabilidade que métodos caixa preta. A Figura 2.6 demonstra que quanto maior o acesso ao modelo, maior a capacidade da detecção.

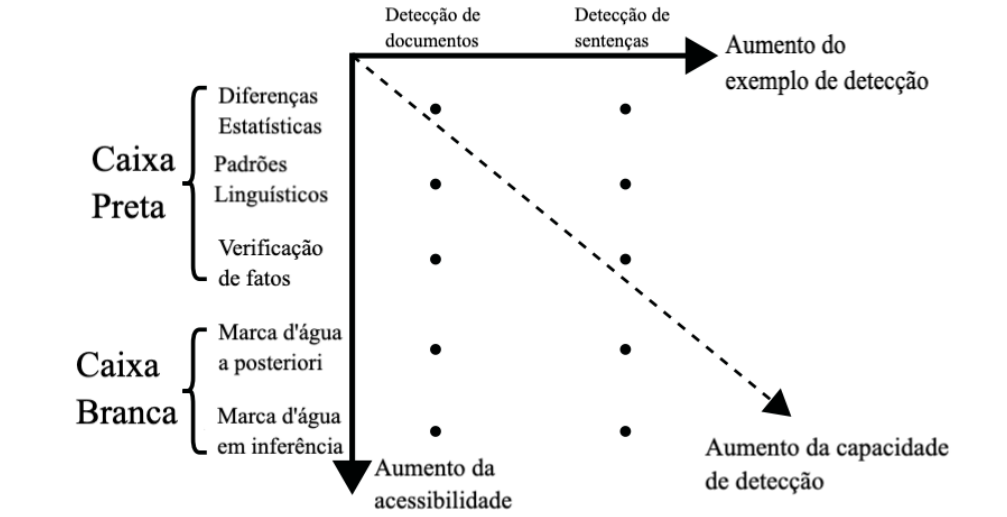
Assim, o nível de acesso ao modelo se mostra como um grande fator a ser considerado durante a detecção de textos gerados por LLMs, uma vez que dependendo do nível de limitação, diferentes métodos e estratégias devem ser aplicados.

2.1.5 *Low-Rank Adaptation* (LoRA)

Apesar de possuírem muitas vantagens no campo de PLN, os LLMs possuem, em geral, uma grande quantidade de parâmetros, que causam um alto custo computacional para operá-los. Modelos como o GPT3, por exemplo, possuem cerca de 170 bilhões de parâmetros treináveis (HU et al., 2021), gerando desafios para seu uso.

Considerando tal cenário, Hu et al. (2021) propõem a abordagem *Low-Rank Adaptation* (LoRA) para superar este desafio. Esta abordagem consiste na aplicação de camadas densas com poucos parâmetros que serão utilizadas juntamente com as camadas

Figura 2.6 – Disparidades na qualidade da detecção caixa preta em relação à caixa branca.



Fonte: (TANG; CHUANG; HU, 2024) (adaptado)

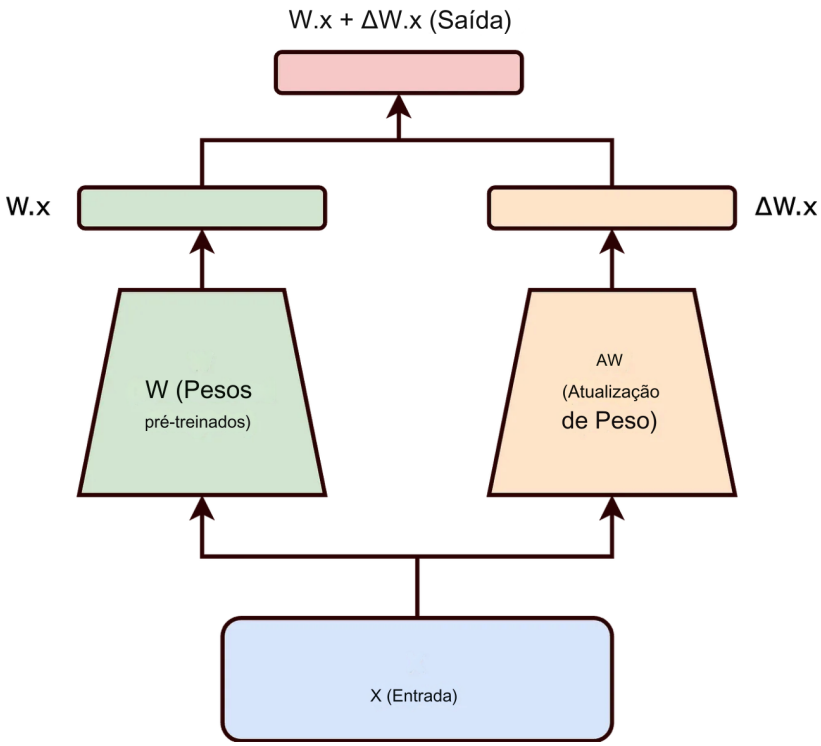
pré-treinadas e congeladas do modelo original, resultando em uma saída voltada para a aplicação específica. A Figura 2.7 apresenta um esquema simplificado de como a junção dos parâmetros pré-treinados (W) com os parâmetros LoRA (ΔW) podem ser combinados para gerar a nova saída. A aplicação deste método pode ser adaptada a partir dos seguintes parâmetros:

LoRA *Rank*: Este parâmetro define o grau (ou ordem) das matrizes LoRA aplicadas nas camadas adaptativas. Ou seja, quanto maior o *rank*, maior o número de pesos treináveis nestas camadas.

LoRA *Alpha*: Este parâmetro define um valor escalar que será aplicado à matriz final da camada adaptativa, antes de ser adicionada ao modelo. A alteração nesse valor define a influência dos pesos da camada adaptativa em relação à saída final do modelo.

Em modelos *transformers*, como é o caso dos LLMs, as camadas do LoRA são aplicadas adaptando as camadas de atenção do modelo, permitindo o uso deste método para um ajuste fino de tais modelos para tarefas específicas, como a classificação de textos.

Figura 2.7 – Esquema de funcionamento da estratégia LoRA.



Fonte: (JAWADE, 2023) (adaptado).

2.2 Trabalhos Relacionados

Esta seção apresenta uma revisão sobre o contexto atual da literatura em relação aos temas relacionados ao problema de detecção de textos gerados por *Large Language Model*. A Seção 2.2.1 apresenta a conjuntura atual da área de Processamento de Linguagem Natural (PLN) para o português. Além disso, na Seção 2.2.2 são apresentadas as diferenças notáveis entre os textos escritos por humanos e os gerados por LLMs, enquanto que a Seção 2.2.3 apresenta uma análise do cenário atual dos detectores e do método mais promissor.

2.2.1 Processamento de Linguagem Natural para o português

Embora seja uma área de amplo estudo por pesquisadores de IA, o desenvolvimento de trabalhos relativos à língua portuguesa é uma lacuna a ser preenchida nos

estudos de PLN. Apesar da ampla variedade de falantes da língua portuguesa, com cerca de 260 milhões de pessoas em 2022, sendo a quarta língua mais falada do mundo (CAMOES, 2022), não há um amplo desenvolvimento de estratégias e pesquisas que englobam as particularidades da língua.

Tradicionalmente, devido a razões sócio-econômicas, a maior parte da pesquisa e desenvolvimento de tarefas de PLN são voltadas à língua inglesa (CASELI; NUNES, 2023). Além disso, o estudo do processamento do português possui, naturalmente, alguns obstáculos particulares. A alta complexidade da língua, por exemplo, se mostra como um complicador para o processamento de dados em português. O alto número de regras e diferentes estruturas (como regras de concordância, utilização de pronomes, etc.) faz com que o português seja não apenas uma língua de difícil aprendizado para não-falantes (PINTO, 2012), mas também para a captura de valores semânticos para uma máquina. Além disso, fatores como o alto número de diferentes dialetos do português existentes (CASTRO, 2006) também se mostram como um obstáculo para a generalização dos modelos.

Apesar destes desafios, existem atualmente exemplos de modelos de PLN que foram especificamente treinados para o português brasileiro, como o BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) (baseado no BERT) e o Sabiá (ALMEIDA et al., 2024). Tais modelos utilizam como base os modelos abertos e os treinam para condicioná-los ao uso específico em português. A importância destes modelos pode ser evidenciada pelo fato de serem os modelos com maior performance (OLIVEIRA et al., 2024) em detecção de discurso de ódio em relação ao conjunto *Toxic Language Dataset for Brazilian Poruguese*(LEITE et al., 2020). A melhora da performance destes modelos em relação aos modelos originais generalistas demonstra a eficácia da abordagem de estratégias específicas para diferentes línguas.

2.2.2 Disparidades entre textos gerados por LLM e por humanos

Os *Large Language Model*s são notáveis por possuírem alta performance em diferentes áreas como geração de códigos (LIU et al., 2024), classificação (ZHANG et al., 2024) e, principalmente, geração de textos (CHAKRABORTY et al., 2023), porém tais modelos ainda não são capazes de mimetizar a escrita humana com perfeição. Apesar de serem capazes de escrever textos longos e coerentes, que geralmente até mesmo humanos

não são capazes de discernir (SONI; WADE, 2023), os LLMs ainda possuem certas disparidades com textos de fato escritos por pessoas reais. Isso ocorre devido ao fato de estes modelos serem treinados com uma grande quantidade de dados, inviabilizando a limitação do treinamento a apenas dados que possuem características semelhantes entre si, gerando textos que buscam a generalização e se diferenciam da escrita de um humano.

Apesar de ainda não haver estudos que abordem a língua portuguesa até o momento, foi proposto por Guo et al. (2023) o dataset HC3, que define cerca de quarenta mil exemplos de perguntas e respostas nas línguas inglesa e chinesa geradas por humanos e pela ferramenta ChatGPT. Além da simples proposição do conjunto, Guo et al. (2023) também apresentam uma análise de padrões compartilhados pelos dados das duas línguas, dessa forma, é possível assumir que tais padrões também estariam presentes em dados do português. Os autores mostram que pode ser feita uma definição clara de disparidades entre os textos gerados por humanos e por IA. Mais notavelmente, destaca-se um padrão de respostas mais longas do ChatGPT, porém com uma diversificação menos extensa do vocabulário, ou seja, humanos em geral escrevem textos menores, porém utilizam um maior número de palavras diferentes. Além disso, os textos gerados pela ferramenta da *OpenAI* também são marcados pelo maior uso proporcional de algumas estruturas específicas, como substantivos e verbos. Porém, além da diferença nas estruturas linguísticas, os textos também se diferem a nível sentimental. Uma vez que o ChatGPT é especificamente treinado para eliminar viés e toxicidade (WU et al., 2023), os textos gerados por ele são, em geral, de cunho mais neutro e lógico que os de humanos.

Assim, apesar de serem capazes de produzir textos de alta complexidade e difícil diferenciação, os *Large Language Models* ainda são marcados por características específicas e compartilhadas entre diferentes línguas que podem ser exploradas para uma classificação de textos gerados por estes modelos.

2.2.3 Aplicação de *Large Language Models* como detectores

Devido ao recente sucesso de ferramentas que utilizam LLMs para automatizar tarefas e responder perguntas, como o ChatGPT da *OpenAI*, houve, em anos recentes, um grande aumento no número de pesquisas que têm como objetivo detectar textos gerados por tais modelos (WU et al., 2023). O crescente interesse nas pesquisas é acompanhado de diversas aplicações de técnicas como *watermarking* (marca d'água), análises estatísticas e

aplicação dos próprios LLMs para gerar classificadores. Apesar deste aumento, a grande maioria de tais estudos aplica as estratégias apenas para as línguas chinesa e inglesa, gerando uma escassez de resultados que demonstrem a performance de detectores para a língua portuguesa.

Devido à alta capacidade de compreensão e adaptabilidade dos LLMs, descritas na Seção 2.1.3, a aplicação do *finetuning* destes modelos para gerar classificadores tem demonstrado resultados promissores (WU et al., 2023). O estudo apresentado por Li et al. (2023) demonstra que a aplicação de *Large Language Model*s como RoBERTa (LIU et al., 2019) e Longformer (BELTAGY; PETERS; COHAN, 2020) para gerar classificadores apresenta desempenho superior a outros métodos como a aplicação de *watermaking*, classificadores *zero-shot* e classificadores baseados em estatísticas linguísticas. Além dos resultados superiores em uma classificação de textos simples, o classificador baseado no modelo Longformer apresentou um reconhecimento médio de aproximadamente 84% em textos fora de domínio, ou seja, com contextos que não estavam presentes nos dados de treinamento, demonstrando um aumento significativo na robustez do classificador. Apesar dos resultados promissores, o modelo ainda apresenta a mesma dificuldade dos outros métodos em lidar com ataques de paráfrase, ou seja, substituir palavras ou frases sem perda de semântica.

Desta forma, a utilização de *Large Language Model*s como base de classificadores de textos gerados por IA demonstra resultados promissores em relação a outros métodos atualmente aplicados. Além disso, falta na literatura uma análise do desempenho desta estratégia na aplicação para o português.

3 Metodologia proposta

Este capítulo apresenta a metodologia seguida para o desenvolvimento do trabalho. A Seção 3.1 aborda o método utilizado para desenvolvimento do *dataset* proposto, a partir do uso de um *dataset* externo de textos jornalísticos escritos por humanos. Além disso, a Seção 3.2 descreve os métodos aplicados para construção, treinamento e avaliação dos modelos classificadores.

3.1 Proposição do *dataset*

Para o desenvolvimento do conjunto de dados proposto, foi usado como base o conjunto de notícias *News of the brazilian newspaper* (MARLESSON, 2024). Este conjunto apresenta cerca de 160 mil exemplos de notícias retiradas do *website* do jornal brasileiro Folha de São Paulo, escritas entre janeiro de 2015 e setembro de 2017. A escolha deste conjunto foi feita pelo alto número de exemplos oferecidos e da alta variedade de categorias de notícias, permitindo a construção de um *dataset* diverso para este trabalho. Além disso, as datas de coletas das notícias representam um período prévio à proposição de LLMs e à popularização de ferramentas de IA como o *ChatGPT*, garantindo a coerência da categorização do conjunto proposto por este trabalho entre dados gerados por humanos e dados gerados por LLMs.

Além dos textos humanos, o conjunto de dados proposto conta com exemplos de notícias humanas geradas por IA e exemplos de notícias que são de autoria humana porém reescritas por LLM. Ambos os tipos de textos artificiais, foram gerados por *prompts*, definidos como uma instrução fornecida ao modelo para orientar a geração da resposta esperada, com modelos caixa-preta. O *prompt* da geração de notícia foi definido para garantir que o modelo gere um texto a partir do título da notícia base e de tamanho similar ao tamanho da notícia original. Isso foi feito para que métodos de detecção não sejam baseados em aspectos como quantidade de caracteres. Já o *prompt* a reescrita de foi definido de forma a manter o contexto da notícia, porém utilizando diferentes palavras. A Tabela 3.1 apresenta os *prompts* utilizados para ambas tarefas. Para realizar

tais tarefas, foi utilizada a biblioteca de chamadas de API da *OpenAI*, que permite que sejam feitas requisições contendo parâmetros como *prompt*, aleatoriedade da resposta gerada (temperatura) e versão do modelo para os modelos e retorna a resposta gerada pela LLM. Este método apresenta vantagens como o acesso a modelos pesados sem o custo computacional de hospedá-los localmente. Além disso, esta biblioteca, permite acesso aos modelos Sabiá, treinados em português, desenvolvidos pela empresa *Maritalk* e que foram utilizados para gerar os textos do *dataset* proposto. A escolha deste modelo se deu visto que é o modelo treinado inteiramente em português mais avançado disponível atualmente.

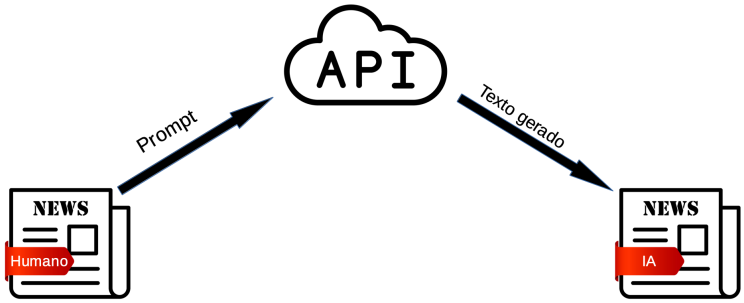
Tabela 3.1 – Descrição das tarefas relacionadas à geração e reescrita de notícias.

Tarefa	***Prompt* usado**
Geração de notícia	Crie uma notícia jornalística com exatamente *length*(noticia) caracteres, incluindo espaços, com o título: *titulo*. Assegure que o conteúdo seja informativo e esteja dentro do limite de caracteres especificado.
Reescrita de notícia	Reescreva a notícia jornalística abaixo, mantendo o mesmo contexto e informações, mas utilizando suas próprias palavras: *notícia*.

Fonte: elaborado pelo autor.

Foram escolhidos de forma aleatória um total de 1.008 pares de título e notícia do conjunto de dados original para gerar e reescrever notícias. A escolha aleatória se mostra importante para garantir a relevância estatística do *subset* utilizado. A Figura 3.1 apresenta o fluxograma da geração utilizando requisições para modelos caixa preta e a Tabela 3.2 apresenta os parâmetros utilizados nas requisições. O parâmetro *max_tokens*, que limita o número de *tokens* do texto gerado pela LLM, foi definido como um valor considerado suficientemente grande para que não limite o modelo a escrever textos menores que o tamanho do texto humano (10.000). Isso foi feito não apenas para evitar que o texto seja menor que o humano, mas também para evitar que os textos apresentem um padrão de serem cortados durante o desenvolvimento de uma palavra ou frase, o que poderia gerar viés no *dataset* resultante. Além disso, a temperatura foi definida como 0,7 para que os textos gerados apresentem diversidade de *tokens* gerados, porém ainda mantenham coerência entre si.

Figura 3.1 – Fluxograma de geração do *dataset*, por meio de API.



Fonte: Elaborado pelo autor

Tabela 3.2 – Configuração dos parâmetros utilizados.

Temperatura	***max_tokens***	**Modelo**
0.7	10000	sabia-3

Fonte: elaborado pelo autor.

Todos os trios de textos (humanos, artificiais e reescritos) do conjunto resultante foram formatados, para que mantenham um padrão em relação ao uso de caracteres especiais, como a substituição do '\n' para indicar uma nova linha por um simples espaço (' '). Além disso, os textos retornados pela API, gerados pelo modelo Sabiá-3, apresentaram um padrão de escrever o título enviado pelo *prompt* no início do texto gerado, normalmente entre pares de asteriscos (** Título **) ou *hashtags* (## Título ##). A Tabela 3.3 mostra um exemplo de texto gerado pelo modelo antes da formatação. Para evitar que este padrão gerasse algum viés nos modelos classificadores, todos os títulos, asteriscos e *hashtags* foram retirados dos trios de texto.

O *dataset* resultante de contendo 3024 exemplos foi manipulado passando por um processo de embaralhamento e divisão em *subsets* de treino e teste contendo 80% e 20% do total dos dados, respectivamente, para haver uma quantidade suficiente de exemplos para treino do modelo, ainda mantendo características suficientes presentes no conjunto de teste. Essa divisão foi feita de maneira aleatória, garantindo a validade dos *subsets* gerados. Além da divisão 80/20, foi extraído um conjunto de 10% dos dados de treino para fazer validação do treinamento dos modelos. A Tabela 3.4 apresenta como ficou definida a distribuição dos exemplos entre os conjuntos de treino e teste.

Tabela 3.3 – Exemplo de resposta do Sabiá sem formatação

Título original	**Resposta API**
Programa da nota fiscal eletrônica deve ser atualizado até terça	**Programa da nota fiscal eletrônica deve ser atualizado até terça**\n\nO governo anunciou que todos os usuários do programa de emissão da nota fiscal eletrônica precisam atualizar o sistema até a próxima terça-feira. A medida visa corrigir vulnerabilidades e garantir mais segurança nas transações comerciais. A nova versão do programa oferece melhorias significativas na performance e na estabilidade do sistema, além de incorporar novas funcionalidades que prometem facilitar o processo de emissão para os contribuintes.\n\nA atualização é obrigatória e os contribuintes que não atualizarem o programa até o prazo estipulado podem enfrentar dificuldades na emissão de notas fiscais eletrônicas e até mesmo penalidades. Para realizar a atualização, os usuários devem acessar o site oficial da Receita Federal e seguir as instruções disponibilizadas. A Receita Federal reforça a importância de manter o sistema atualizado para evitar problemas operacionais e garantir a conformidade fiscal.

Fonte: elaborado pelo autor.

Tabela 3.4 – Distribuição do exemplos humanos e artificiais entre os conjuntos de treino e teste.

Classe dos exemplos	**Treino**	**Validação**	**Teste**	**Total**
Número de exemplos humanos	727	79	202	1008
Número de exemplos artificiais	725	79	204	1008
Número de exemplos reescritos	730	79	199	1008

Fonte: elaborado pelo autor.

3.2 Construção e avaliação dos classificadores

Considerando os recentes sucessos da aplicação dos próprios *Large Language Models* como classificadores de textos gerados por este tipo de modelo, assim como apresentado na Seção 2.2.3, propõe-se a utilização de tal método como método de classificação.

Os modelos de LLM utilizados foram o BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020), BERTuguês (ZAGO; PEDOTTI, 2024), multi-BERT (DEVLIN et al., 2018) e modelos da família Llama (Llama Team Group, 2024). A escolha destes modelos foi feita para que possa ser feita uma comparação de desempenho entre modelos treinados especificamente para o português (BERTimbau e BERTuguês) e modelos multilinguísticos (multi-BERT e Llama 3). Além disso, essa escolha permite uma análise entre o desempenho de modelos mais simples baseados em BERT e modelos mais

complexos como o Llama 3. Todos os modelos utilizados estão disponíveis gratuitamente. A Tabela 3.5 apresenta uma comparação entre o número de parâmetros dos modelos listados e o tempo de inferência em GPU dos 605 exemplos de teste.

Tabela 3.5 – Comparação entre os modelos usados para classificação.

Modelo	**Número de parâmetros**	**Tempo para inferência em GPU (s)**
neuralmind/bert-large-portuguese-cased	335M	13,16
ricardoz/BERTugues-base-portuguese-cased	110M	4,45
google-bert/bert-base-multilingual-cased	179M	4,47
meta-llama/Llama-3.2-3B-Instruct	3,21B	229,21
meta-llama/Llama-3.1-8B-Instruct	8,03B	254,73
LSTM proposto	17,8M	2,6

Fonte: elaborado pelo autor.

Todos os modelos classificadores passaram por dois tipos de classificação, uma classificação binária entre textos humanos e com interferência artificial (gerados + reescritos), além de uma classificação de 3 classes entre textos humanos, gerados por IA e reescritos pelo LLM.

Para a classificação utilizando modelos baseados em BERT (BERTimbau, BERTuguês e multi-BERT) foi feito o ajuste-fino supervisionado dos modelos, com pares de texto e *label* (0, 1 para classificação binária ou 0, 1, 2 para classificação em 3 classes). Os dados foram passados pelo tokenizador do próprio modelo e foi utilizado um *data collator* para organizar e preparar os lotes de treinamento. Todos os modelos foram treinados por 5 épocas, com *learning rate* de 0,00002 e decaimento de 0,01.

Também foi feito Ajuste Fino dos modelos Llama3.1-8B-Instruct e Llama3.2-3B-*Instruct*, que possuem 8 e 3 bilhões de parâmetros pré-treinados, respectivamente, (cerca de 170 bilhões a menos que o GPT3), garantindo alta eficiência em tarefas de PLN. A escolha de avaliar dois modelos da terceira geração da família Llama se deu para avaliar a diferença entre um modelo mais robusto porém mais antigo (Llama 3.1 8B) e um modelo mais novo porém com menos parâmetros (Llama 3.2 3B). Além dos treinamentos dos modelos, foi feito um Ajuste Fino pela própria *Meta AI* (organização autora dos modelos Llama) em tarefas de instrução ou *chat*. Devido ao alto número de parâmetros, e consequentemente, o custo computacional associado a estes, utilizou-se o método LoRA, que permite adaptar modelos grandes para tarefas aplicadas com baixo custo.

Durante o treino e teste dos modelos Llama, foi utilizado um método de Pergunta e Resposta, que corresponde à utilização de um *prompt* de pergunta (ou instrução) somado ao texto, além das saídas esperadas. Foram utilizados os seguintes *prompts*, para as classificações binária e de 3 classes, com o objetivo de manter a resposta gerada objetiva, permitindo sua avaliação:

***Prompt 1*:** *"Me diga, com sim ou não, se você considera o seguinte texto como sendo de autoria humana: `[TEXTO]`."*

***Prompt 2*:** *"Me diga, com 0, 1 ou 2, se você considera o seguinte texto como sendo de autoria humana, artificial ou reescrito artificialmente: `[TEXTO]`."*

Em ambos os *prompts*, `[TEXTO]` corresponde ao texto que será avaliado pelo modelo.

O modelo foi treinado de forma supervisionada a partir do conjunto de dados proposto na Seção 3.1. Devido ao poder de representação e captação de características de um modelo grande como o LLama, aliado ao relativamente baixo número de casos de treino, o modelo foi treinado passando apenas uma vez pelos dados, para evitar o sobreajuste. Tanto o LoRA *Rank* quanto o LoRA *Alpha* foram definidos com o valor 64, definindo camadas de adaptação com uma alta quantidade de pesos e alta influência na saída final do modelo.

Além disso, para fins de comparação entre métodos, propõe-se um modelo de RNN utilizando 2 camadas de *Long Short-Term Memory* (LSTM) bi-direcionais com 128 e 64 neurônios, respectivamente, alimentadas por uma camada inicial de *embedding*. Foram usadas camadas de LSTM bi-direcionais pois este método se mostra eficiente na captação do contexto de uma frase ao considerar tanto as palavras à frente quanto as anteriores. Em cada uma destas camadas foi feito *dropout* de 20% dos neurônios, a fim de evitar sobre-ajustes do modelo. Por fim, foi adicionada uma camada densa de 32 neurônios de ativação ReLU, além da camada densa final de classificação de um neurônio e ativação sigmoide, para a classificação de 2 classes e 3 neurônios e ativação softmax, para o segundo tipo de tarefa. Para treinamento e teste desta rede, os dados textuais foram passados por um tokenizador e foi feito o *padding* de zeros de acordo com a sentença tokenizada de maior tamanho. Os valores de entrada da camada de *embedding*

foram definidos de forma dinâmica de acordo com o tamanho do vocabulário do *dataset* e tamanho da maior sequência de *tokens*. A Figura 3.2 apresenta a arquitetura da rede proposta. A rede foi treinada passando pelo conjunto de dados 20 vezes e foram feitos *checkpoints* a cada 5 épocas. Após isso, foram analisados os gráficos de perda e validação para ser utilizado o *checkpoint* com os pesos que representassem a menor perda durante o treinamento, porém sem sobre-ajustes. Dessa forma, a RNN utilizada teve seus pesos ajustados ao conjunto de treino 10 vezes. Além disso, o modelo foi treinado com *learning rate* de 0,001, otimizador Adam e função de perda *binary crossentropy*.

3.3 Métricas de avaliação

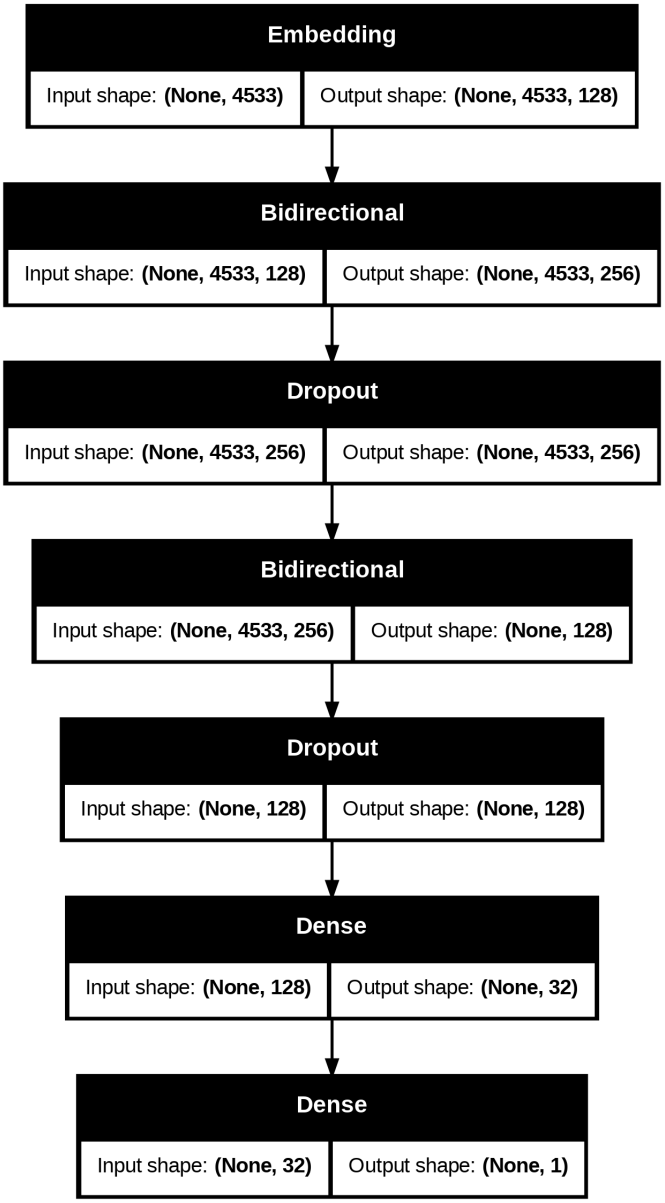
Para avaliar os modelos, foi utilizada a métrica de acurácia nos dados de teste com o intuito de validar a proposta deste trabalho. Foi feito o seu uso pois, além de amplamente aplicada na literatura para problemas de classificação, é uma métrica confiável em casos de dados balanceados, como o caso do conjunto proposto. O cálculo desta métrica se dá de acordo com a Equação 3.1 baseado nos seguintes conceitos:

- **Verdadeiro positivo (*true positive, TP*):** Exemplos corretamente classificados como positivos, neste caso, como de autoria humana.
- **Verdadeiro negativo (*true negative, TN*):** Exemplos corretamente classificados como negativos, neste caso, como de autoria artificial.
- **Falso positivo (*false positive, FP*):** Exemplos erroneamente classificados como positivos, neste caso, textos artificiais classificados como humanos.
- **Falso negativo (*false negative, FN*):** Exemplos erroneamente classificados como negativos, neste caso, textos humanos classificados como artificiais.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Além da acurácia, também serão utilizadas as métricas de precisão, revocação e F1-*score* do modelo (definidas de acordo com a Equação 3.2, Equação 3.3 e Equação 3.4, respectivamente). Tais métricas serão utilizadas pois permitem identificar possíveis

Figura 3.2 – Arquitetura da RNN proposta.



Fonte: Elaborado pelo autor

tendências na classificação do modelo para uma determinada classe. Dessa forma, a interpretação destas métricas permite a análise do modelo sob diferentes perspectivas. As definições das métricas utilizadas são:

- **Acurácia:** Mede a proporção de previsões corretas em relação ao total de previsões feitas. Boa para se ter uma noção geral da taxa de acerto do modelo (já definida na Equação (3.1)).

- **Revocação:** Mede a taxa de casos positivos foram corretamente identificados. Útil para identificar a taxa de exemplos humanos que foram considerados artificiais. Ela é definida matematicamente por:

$$\text{Revocação} = \frac{TP}{TP + FN}. \tag{3.2}$$

- **Precisão:** Mede a taxa de casos identificados como positivos que são de fato positivos. Útil para identificar a taxa de exemplos artificiais que foram considerados humanos. Ela é definida matematicamente por:

$$\text{Precisão} = \frac{TP}{TP + FP}. \tag{3.3}$$

- **F1-*Score*:** Média harmônica entre precisão e revocação, definida por:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}}. \tag{3.4}$$

4 Experimentos e Resultados

Neste capítulo serão apresentados e analisados os resultados do trabalho de acordo com o que se propõe. A Seção 4.1 define a configuração e ambientação que os experimentos foram rodados, considerando fatores como linguagem, *framework* e *hardware*. Além disso, na Seção 4.2 são apresentados e interpretados os resultados tanto do conjunto de dados gerado, quanto das classificações dos modelos em cima destes dados.

4.1 Configuração Experimental

Toda a implementação do trabalho foi feita utilizando a linguagem *Python*, uma vez que esta propõe facilidades e soluções práticas para o desenvolvimento de modelos de aprendizado de máquina. Seguindo o mencionado na Seção 3.1, o conjunto de dados proposto foi desenvolvido a partir da API da empresa *OpenAi*.

A implementação dos classificadores aplicando o Ajuste Fino de LLM para a tarefa foi feita utilizando o *framework PyTorch*, além da biblioteca *LlamaFactory* para os modelos Llama e da biblioteca transformers para os modelos baseados em BERT. A avaliação dos modelos baseados em LLMs foi feita utilizando a biblioteca *scikit-learn*.

Já a implementação e avaliação do modelo de RNN foi feita inteiramente utilizando o *framework Tensorflow*. Ambos os modelos foram implementados e testados utilizando *notebooks* Jupyter ambientados no *Google Collab* utilizando a placa gráfica A100.

4.2 Análise dos resultados obtidos

A seguir serão apresentadas as análises e interpretações dos resultados obtidos. A Seção 4.2.1 traz uma análise e interpretação do *dataset* resultante de acordo com a metodologia proposta e suas características. Além disso, na Seção 4.2.1 são definidos os

resultados e análises do desempenhos do modelos de acordo com as métricas definidas, além da avaliação da relevância dos resultados de acordo com os dados.

4.2.1 Análise dos dados

O conjunto de dados resultante foi analisado principalmente sob a ótica da disparidade nos tamanhos dos textos gerados por IA e os textos de autoria humana dos dados originais. Apesar do *prompt* de geração ter sido definido para manter similaridade entre os tamanhos dos textos, quando analisados os pares de textos de mesmo título, 145 pares tiveram textos de autoria humana maiores, enquanto 55 pares tiveram textos gerados pelo modelo Sabiá 3 com maior número de caracteres. Além da quantidade de textos maiores, foram analisados os valores e os padrões dos tamanhos dos textos em cada um dos *subsets*, e do *dataset* total. A Tabela 4.1 dispõe as distribuições de cada classe nos conjuntos de treino e teste. A Figura 4.1 apresenta os valores médios, em caracteres, de cada classe enquanto as Figura 4.2 e Figura 4.3 apresentam os menores e maiores valores de cada *subset*, respectivamente. Os tamanhos dos textos de validação foram analisados juntamente com o conjunto de treino uma vez que no treinamento da RNN, a divisão entre treino e validação foi feita de maneira aleatória e os exemplos não puderam ser analisados separadamente.

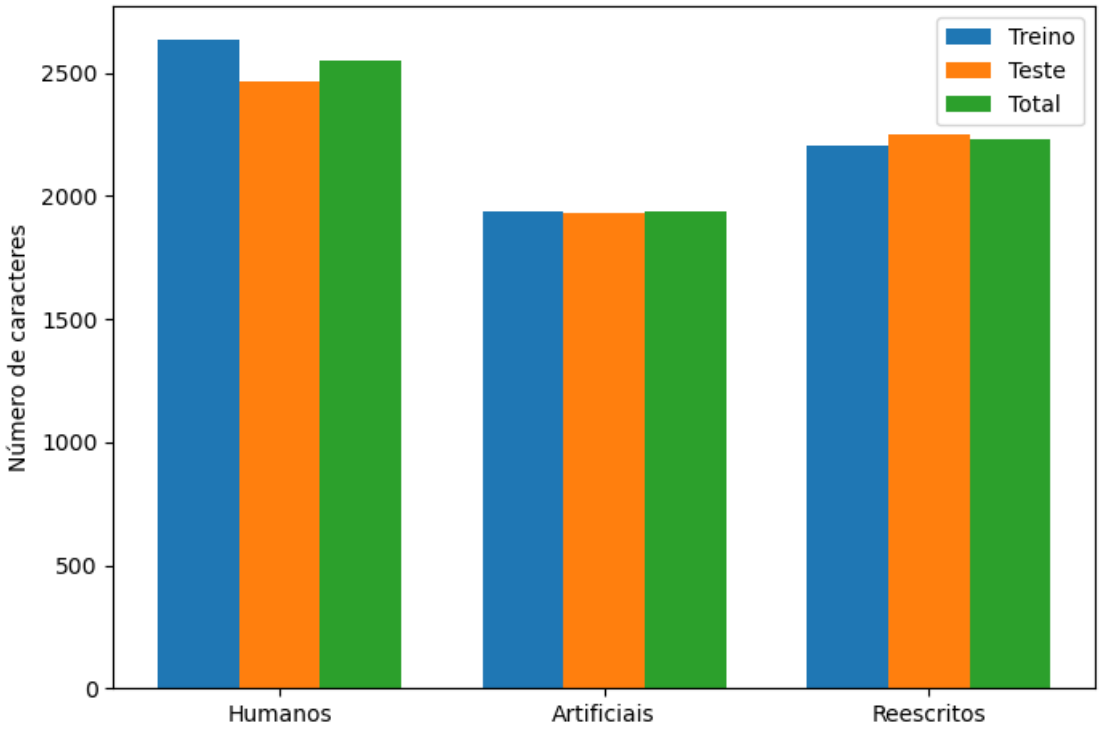
Tabela 4.1 – Tabela com distribuição de classes entre conjuntos de treino e teste

Métrica	**Treino**	**Teste**	**Total**
Número de exemplos humanos	806	202	1008
Número de exemplos artificiais	804	204	1008
Número de exemplos reescritos	809	199	1008

Fonte: elaborado pelo autor.

De acordo com a Figura 4.1, os textos humanos foram em geral 30% maiores que os artificiais e 15% maiores que os textos reescritos. Este padrão pode gerar um viés dos modelos classificadores durante a classificação dos textos, definindo como gerados por humanos textos maiores, ou gerados artificialmente em casos de textos menores. Uma vez que o tamanho médio de textos reescritos ficou entre os tamanhos de textos humanos e artificiais, não há razão para crer que um viés de tamanho possa ser extraído dessa classe. Apesar disso, durante a análise da Figura 4.2 é notável que os menores textos

Figura 4.1 – Médias dos tamanhos de texto de cada classe e do conjunto total.

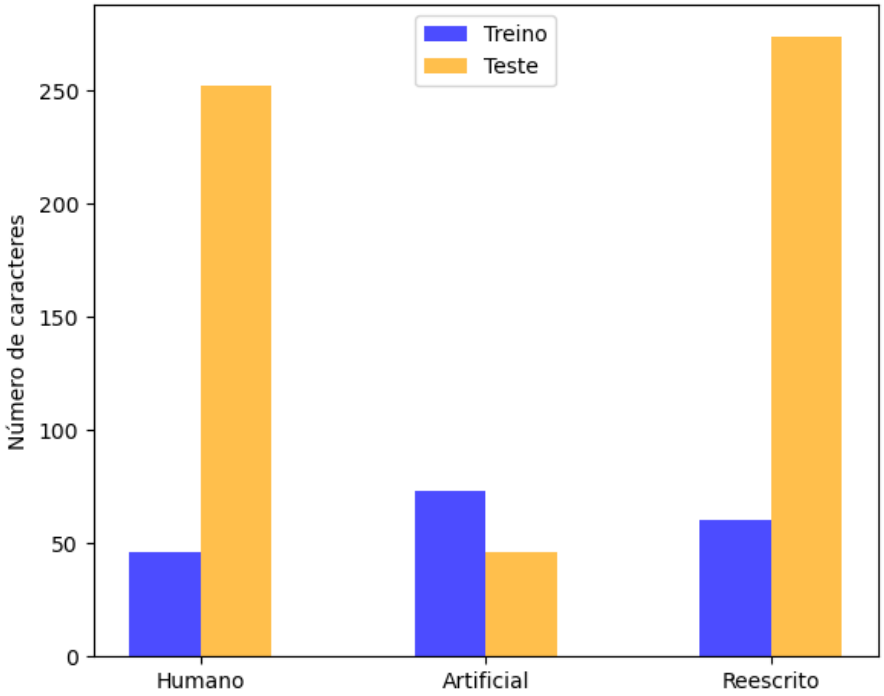


Fonte: elaborado pelo autor.

de cada tipo de exemplo (humano, reescrito e IA) são de tamanhos semelhantes, com o menor texto humano de treino humano sendo menor que os textos artificiais e reescritos. Tal comportamento permite a análise durante os resultados dos classificadores, para determinar se o tamanho dos textos é o único fator de identificação de padrões entre as classes. Essa análise se torna possível pois, apesar de textos humanos serem maiores em média, é possível verificar que as três classes possuem exemplos de textos que ocupam a maior parte do escopo dos tamanhos de texto.

Além da análise geral dos tamanhos dos textos, também podem ser feitas considerações sobre a divisão dos dados. A Tabela 4.1 permite notar que apesar dos exemplos humanos terem sidos aleatoriamente escolhidos e da divisão dos dados de treino e teste também ter sido feita de forma aleatória, o número de exemplos humanos artificiais e reescritos foi praticamente o mesmo nos *subsets* de treino e teste. É possível também notar a similaridade geral entre os valores dos *subsets* ao serem comparados com os conjuntos totais. Os valores de média, principalmente, demonstram padrão entre os dados

Figura 4.2 – Menores valores de cada classe em cada conjunto



Fonte: elaborado pelo autor.

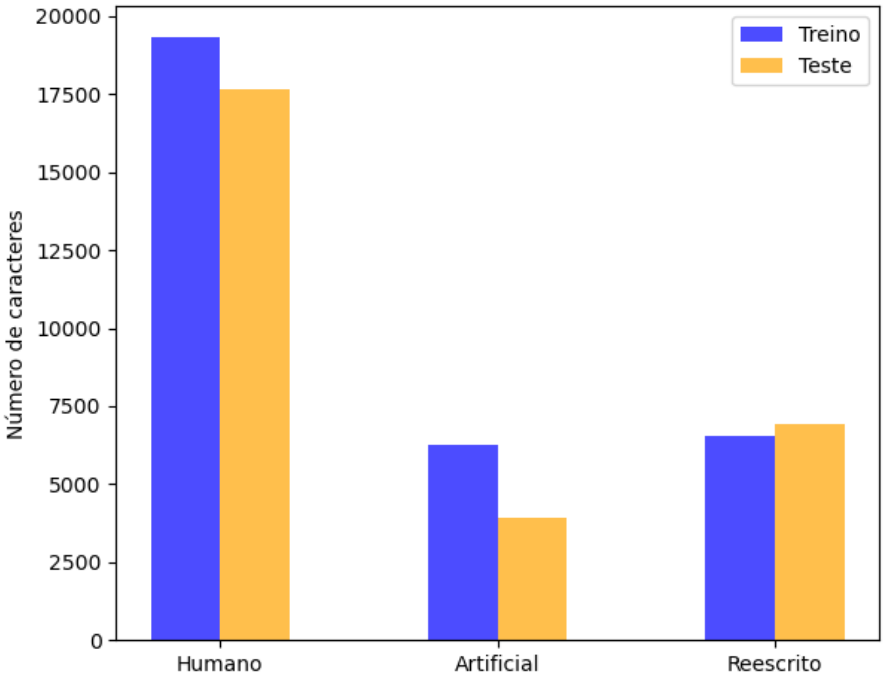
de treino, teste e geral. Desta forma, é possível afirmar o balanceamento do conjunto de forma geral e também da divisão dos dados.

4.2.2 Resultados dos classificadores

As Tabela 4.2 e Tabela 4.3 apresentam os resultados obtidos em classificações com e sem 0-*shot* em 3 e 2 classes, respectivamente.

A análise dos resultados das classificações 0-*shot* revelam informações úteis sobre ambos o conjunto de dados e a dificuldade inerente da tarefa em si. De acordo com a análise apresentada na Seção 4.2.1 pode-se notar um *dataset* de teste praticamente balanceado para a classificação de três classes e uma distribuição de aproximadamente duas para uma para a classificação binária, uma vez que as classes de notícias geradas por IA e reescrito foram unidas em uma única. Diante disso, as médias de acurácias utilizando 0-*shot* de 30-36% apresentadas na Tabela 4.3 e 44-65% presentes na Tabela 4.2 indicam que os modelos estão próximos do "chute" aleatório, ou seja, não estão extraindo

Figura 4.3 – Maiores valores de cada classe em cada conjunto



Fonte: elaborado pelo autor.

muitos padrões úteis dos dados. Assim, é possível afirmar que a tarefa não é trivial e o treinamento posterior pode ser atribuído efetivamente ao refinamento do modelo na captura das diferenças entre as classes, e não à correção de um viés pré-existente no conjunto.

Ao comparar os resultados dos modelos treinados, pode-se afirmar que o modelo Llama 3.1 8B é o mais robusto para esta tarefa, uma vez que teve os melhores resultados tanto para classificação em 3 classes quanto a binária, atingindo 98,18% e 97,7% de acurácia nessas respectivas tarefas. Os modelos BERTimbau, Llama 3.2 3B e BERTuguês mantiveram desempenhos promissores acima de 90% de acurácia em ambas as distribuições de rótulos, seguidos em ambos os casos pelo modelo Multi-BERT. A LSTM proposta apresentou um desempenho abaixo do satisfatório em ambas as rotulagens, revelando que essas tarefas requerem alto nível de abstração mais adequado aos mecanismos dos modelos baseados em *transformers* . Todos os modelos testados tiveram melhor desempenho na tarefa com apenas duas classes de dados em relação à classificação tripla, revelando que a adição de uma classe escalou a dificuldade da categorização. Apesar disso, os

Tabela 4.2 – Resultados da classificação em 2 classes com e sem *0-Shot*.

Métodos	***Zero-Shot***	**Acurácia (%)**	**Precisão (%)**	**Revocação (%)**	**F1-score**
LLM Multi-BERT	✓	66,4	44,3	66,4	0,532
LLM BERTugues	✓	61,0	61,6	61,0	0,613
LLM BERTimbau-Large	✓	60,5	57,8	60,5	0,591
RNN LSTM Bidirecional	✓	57,0	48,0	48,0	0,480
LLM Llama 3.1-8B	✓	49,6	46,6	49,6	0,480
LLM Llama 3.2-3B	✓	44,4	49,5	49,5	0,495
LLM Llama 3.1-8B	☒	**98,2**	97,5	**98,5**	**0,9800**
LLM BERTimbau-Large	☒	97,7	**97,7**	97,7	0,977
LLM Llama 3.2-3B	☒	97.7	97.4	97.4	0,974
LLM BERTugues	☒	94,7	94,9	94,7	0,9479
LLM Multi-BERT	☒	85,5	86,9	85,5	0,8619
RNN LSTM Bidirecional	☒	73,0	72,0	63,0	0,672

Fonte: elaborado pelo autor.

Tabela 4.3 – Resultados da classificação de 3 classes com e sem *0-Shot*.

Métodos	***Zero-Shot***	**Acurácia (%)**	**Precisão (%)**	**Revocação (%)**	**F1-score**
LLM Llama 3.1-8B	✓	36,0	41,0	36,0	0,383
RNN LSTM Bidirecional	✓	35,0	28,0	35,0	0,311
LLM BERTimbau-Large	✓	34,2	25,5	34,2	0,292
LLM Multi-BERT	✓	32,7	20,4	32,7	0,211
LLM Llama 3.2-3B	✓	31,9	32,2	31,7	0.319
LLM BERTugues	✓	30,6	16,6	30,6	0.215
LLM Llama 3.1-8B	☒	**97,7**	**97,7**	**97,7**	**0,9770**
LLM BERTimbau-Large	☒	96,5	96,5	96,5	0,965
LLM Llama 3.2-3B	☒	93.0	93.0	93.0	0,930
LLM BERTugues	☒	91,9	92,1	91,9	0,919
LLM Multi-BERT	☒	82,0	85,8	82,0	0,8385
RNN LSTM Bidirecional	☒	56,0	58,0	56,0	0,560

Fonte: elaborado pelo autor.

resultados promissores revelam que a terceira classe (‘reescritos’) possui características que podem ser extraídas para sua identificação.

Apesar do modelo Llama 3.1 8B ter tido os melhores resultados, os modelos baseados em BERT e treinados em português obtiveram resultados superiores aos multilínguas, se mantendo competitivos em relação aos modelos generalistas, com o BERTimbau apresentando melhor performance geral que o modelo Llama 3.2, mesmo possuindo aproximadamente 2.9 bilhões de parâmetros a menos (de acordo com os conteúdos da Tabela 3.5). Além disso, quando comparados ao modelo MultiBERT, os modelos espe-

cíficos para a língua portuguesa apresentaram resultados consideravelmente superiores em relação ao modelo generalista que, por sua vez, não foi capaz de apresentar a mesma comparabilidade em relação aos modelos da família Llama. A análise em conjunto destas duas comparações reforça a hipótese de que a aplicação de modelos treinados sob um escopo de dados mais concentrado para uma determinada tarefa apresenta resultados mais satisfatórios que a aplicação de modelos generalistas, que apresentam uma necessidade de um número muito maior de parâmetros e, por consequência, pior desempenho computacional para se manterem superiores aos modelos mais simples, porém mais focalizados.

A partir das análises supracitadas, é possível afirmar que a tarefa de classificação de textos gerados por LLMs é possível e viável. Além disso, a classificação de textos parafraseados por IAs também deixa marcas definidas em sua estrutura que permitem a sua classificação. Desta forma, os resultados apresentados são considerados satisfatórios em relação à hipótese levantada.

5 Considerações Finais

Este capítulo apresenta as considerações finais e direções futuras do presente trabalho. A Seção 5.1 traz uma análise e conclusão do trabalho sob a ótica dos objetivos definidos. Enquanto a Seção 5.2, define os trabalhos futuros a serem realizados, propondo um cronograma para a realização de suas atividades.

5.1 Conclusão

Conclui-se, assim, o trabalho proposto atingindo de forma satisfatória os objetivos inicialmente definidos. Foi feita uma revisão literária sobre a conjuntura atual da detecção de textos gerados por LLMs de forma geral e, principalmente, no contexto do português. A partir desta revisão, foi definida não apenas a importância do tema, mas também a falta de abordagens que se proponham a solucioná-lo no escopo da língua portuguesa. Desta forma, foi proposto um conjunto de dados balanceado que permite o treinamento e teste de algoritmos de aprendizado de máquina que visam abordar o problema. Juntamente com o *dataset*, foram propostos um conjunto de modelos classificadores, baseados em diferentes tecnologias como LLMs e RNNs. Ainda, tais classificadores tiveram seus resultados analisados sob diferentes perspectivas, tais como eficiência, custo computacional e capacidade multi-linguística, alcançando resultados relevantes de até aproximadamente 98% e 97% nas duas tarefas de classificação propostas.

Dessa forma, é possível afirmar que o trabalho apresentou resultados considerados positivos tanto na proposição do *dataset* quanto no desenvolvimento do classificador. A análise dos resultados revela sua relevância, mesmo considerando problemas como viés de tamanho dos dados. Isso faz com que os bons resultados alcançados pelos modelos propostos revelem que há discrepâncias entre textos escritos por humanos e por máquinas em português, além de marcadores que tornam possível a identificação de textos reescritos por modelos artificiais. Além disso, a proximidade nos resultados de modelos mais simples treinados em português revela sua maior eficácia em tarefas específicas para a linguagem.

5.2 Trabalhos Futuros

Considerando os problemas encontrados nos dados propostos, fica como maior trabalho futuro, propor melhorias na metodologia apresentada (ou uma completamente nova), que eliminem o viés relativo ao tamanho dos dados, para se ter um *dataset* completamente balanceado e sem diferenças notáveis entre as três classes. Além disso, futuramente, a metodologia da criação do conjunto de dados proposto deve ser aplicada a todos os exemplos do *dataset News of the Brazilian Newspaper*. Para avaliar ainda melhor a nova metodologia, trabalhos futuros devem também se concentrar em aplicar diferentes perspectivas de testes sobre os algoritmos propostos.

Referências

ABEND, O.; RAPPOPORT, A. The state of the art in semantic representation. In: BARZILAY, R.; KAN, M.-Y. (Ed.). Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017. p. 77–89. Disponível em: <https://aclanthology.org/P17-1008>.

ALMEIDA, T. S.; ABONIZIO, H.; NOGUEIRA, R.; PIRES, R. Sabiá-2: A New Generation of Portuguese Large Language Models. 2024.

ALPAYDIN, E. Introduction to machine learning. [S.l.]: MIT press, 2020.

BELTAGY, I.; PETERS, M. E.; COHAN, A. Longformer: The Long-Document Transformer. 2020. Disponível em: <https://arxiv.org/abs/2004.05150>.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, IEEE, v. 35, n. 8, p. 1798–1828, 2013.

BLANK, I. A. What are large language models supposed to model? Trends in Cognitive Sciences, Elsevier, 2023.

BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A. et al. Language models are few-shot learners. Advances in neural information processing systems, v. 33, p. 1877–1901, 2020.

CAMOES instituto. Dia Mundial da Lingua portuguesa: 5 de maio de 2022. 2022. Acessado em 05 de junho de 2024. Disponível em: <https://www.instituto-camoes.pt/images/pdf_noticias/Dados_sobre_a_l%C3%ADngua_portuguesa_de_2022.pdf>.

CASELI, H. d. M.; NUNES, M. d. G. V. Processamento de linguagem natural: conceitos, técnicas e aplicações em português. 2023.

CASTRO, I. Introdução à história do português. Edições Colibri Lisboa, 2006.

CHAKRABORTY, S.; BEDI, A. S.; ZHU, S.; AN, B.; MANOCHA, D.; HUANG, F. On the Possibilities of AI-Generated Text Detection. 2023.

CHANG, Y.; WANG, X.; WANG, J.; WU, Y.; YANG, L.; ZHU, K.; CHEN, H.; YI, X.; WANG, C.; WANG, Y.; YE, W.; ZHANG, Y.; CHANG, Y.; YU, P. S.; YANG, Q.; XIE, X. A Survey on Evaluation of Large Language Models. 2023.

DEVLIN, J.; CHANG, M.; LEE, K.; TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805, 2018. Disponível em: <http://arxiv.org/abs/1810.04805>.

DONATO, H.; ESCADA, P.; VILLANUEVA, T. A transparência da ciência com o chatgpt e as ferramentas emergentes de inteligência artificial: como se devem posicionar as revistas científicas médicas. The Transparency of Science with ChatGpt and the Emerging Artificial Intelligence Language Models: Where Should Medical Journals Stand, 2023.

ELSE, H. Abstracts written by chatgpt fool scientists. Nature, Nature, v. 613, n. 7944, p. 423–423, 2023.

GALASSI, A.; LIPPI, M.; TORRONI, P. Attention in natural language processing. IEEE Transactions on Neural Networks and Learning Systems, Institute of Electrical and Electronics Engineers (IEEE), v. 32, n. 10, p. 4291–4308, out. 2021. ISSN 2162-2388. Disponível em: <http://dx.doi.org/10.1109/TNNLS.2020.3019893>.

Gemini Team Group. Gemini: A Family of Highly Capable Multimodal Models. 2024.

GONZALEZ, M.; LIMA, V. L. S. Recuperação de informação e processamento da linguagem natural. In: SN. XXIII Congresso da Sociedade Brasileira de Computação. [S.l.], 2003. v. 3, p. 347–395.

GUO, B.; ZHANG, X.; WANG, Z.; JIANG, M.; NIE, J.; DING, Y.; YUE, J.; WU, Y. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. 2023.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. Science, American Association for the Advancement of Science, v. 349, n. 6245, p. 261–266, 2015.

HU, E. J.; SHEN, Y.; WALLIS, P.; ALLEN-ZHU, Z.; LI, Y.; WANG, S.; WANG, L.; CHEN, W. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.

IBRAHIM, H.; LIU, F.; ASIM, R.; BATTU, B.; BENABDERRAHMANE, S.; ALHAFNI, B.; ADNAN, W.; ALHANAI, T.; ALSHEBLI, B.; BAGHDADI, R.; BéLANGER, J. J.; BERETTA, E.; CELIK, K.; CHAQFEH, M.; DAQAQ, M. F.; BERNOUSSI, Z. E.; FOUGNIE, D.; SOTO, B. Garcia de; GANDOLFI, A.; GYORGY, A.; HABASH, N.; HARRIS, J. A.; KAUFMAN, A.; KIROUSIS, L.; KOCAK, K.; LEE, K.; LEE, S. S.; MALIK, S.; MANIATAKOS, M.; MELCHER, D.; MOURAD, A.; PARK, M.; RASRAS, M.; REUBEN, A.; ZANTOUT, D.; GLEASON, N. W.; MAKOVI, K.; RAHWAN, T.; ZAKI, Y. Perception, performance, and detectability of conversational

artificial intelligence across 32 university courses. Scientific Reports, Springer Science and Business Media LLC, v. 13, n. 1, ago. 2023. ISSN 2045-2322. Disponível em: <http://dx.doi.org/10.1038/s41598-023-38964-3>.

JACKSON, P.; MOULINIER, I. Natural language processing for online applications: Text retrieval, extraction and categorization. [S.l.]: John Benjamins Publishing, 2007. v. 5.

JAWADE, B. Understanding LoRA (Low-Rank Adaptation) for Fine-tuning Large Models. 2023. <https://towardsdatascience.com/understanding-lora-low-rank-adaptation-for-finetuning-large-models-936bce1a07c6>. Accessed: 2024-08-31.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. nature, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LEITE, J. A.; SILVA, D.; BONTCHEVA, K.; SCARTON, C. Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. In: WONG, K.-F.; KNIGHT, K.; WU, H. (Ed.). Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing. Suzhou, China: Association for Computational Linguistics, 2020. p. 914–924. Disponível em: <https://aclanthology.org/2020.aacl-main.91>.

LI, Y.; LI, Q.; CUI, L.; BI, W.; WANG, L.; YANG, L.; SHI, S.; ZHANG, Y. Deepfake text detection in the wild. arXiv preprint arXiv:2305.13242, 2023.

LIU, F.; LIU, Y.; SHI, L.; HUANG, H.; WANG, R.; YANG, Z.; ZHANG, L.; LI, Z.; MA, Y. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation. 2024.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

Llama Team Group. The Llama 3 Herd of Models. 2024. Disponível em: <https://arxiv.org/abs/2407.21783>.

MARLESSON. News of the Brazilian Newspaper. 2024. Kaggle. Accessado em 31-08-2024. Disponível em: <https://www.kaggle.com/datasets/marlesson/news-of-the-site-folhauol?resource=download>.

OLIVEIRA, A. da S.; CECOTE, T. de C.; ALVARENGA, J. P. R.; LUZ, E. J. da S. et al. Toxic speech detection in portuguese: A comparative study of large language models. In: Proceedings of the 16th International Conference on Computational Processing of Portuguese. [S.l.: s.n.], 2024. p. 108–116.

PINTO, J. A aquisição de português le por alunos marroquinos: dificuldades interlinguísticas. In: SEEPLU-CILEM-LEPOLL. Actas del II Congreso Internacional de la Sociedad Extremeña de Estudios Portugueses y la Lusofonía (SEEPLU). [S.l.], 2012. p. 217–239.

PINTO, S. C. S. Processamento de linguagem natural e extração de conhecimento. Dissertação (Mestrado) — Universidade de Coimbra, 2015.

ROSSONI, L.; CHAT, G. A inteligência artificial e eu: escrevendo o editorial juntamente com o chatgpt. Revista Eletrônica de Ciência Administrativa, v. 21, n. 3, p. 399–405, 2022.

SANT, F. P.; SANT, I. P.; SANT, C. de C. et al. Uma utilização do chat gpt no ensino. Com a Palavra, o Professor, v. 8, n. 20, p. 74–86, 2023.

SHANAHAN, M. Talking about large language models. Commun. ACM, Association for Computing Machinery, New York, NY, USA, v. 67, n. 2, p. 68–79, jan 2024. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3624724>.

SONI, M.; WADE, V. Comparing abstractive summaries generated by chatgpt to real summaries through blinded reviewers and text classification algorithms. arXiv preprint arXiv:2303.17650, 2023.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: pretrained bert models for brazilian portuguese. In: SPRINGER. Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9. [S.l.], 2020. p. 403–417.

TANG, R.; CHUANG, Y.-N.; HU, X. The science of detecting llm-generated text. Commun. ACM, Association for Computing Machinery, New York, NY, USA, v. 67, n. 4, p. 50–59, mar 2024. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3624725>.

VALLANCE, C. Chatgpt: New Ai chatbot has everyone talking to it. BBC, 2022. Acessado em 15 de abril de 2024. Disponível em: <https://www.bbc.com/news/technology-63861322>.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. Attention Is All You Need. 2017.

VIEIRA, R.; LOPES, L. Processamento de linguagem natural e o tratamento computacional de linguagens científicas. Em corpora, p. 183, 2010.

WANG, R.; CHEN, H.; ZHOU, R.; MA, H.; DUAN, Y.; KANG, Y.; YANG, S.; FAN, B.; TAN, T. LLM-Detector: Improving AI-Generated Chinese Text Detection with Open-Source LLM Instruction Tuning. 2024. Disponível em: <https://arxiv.org/abs/2402.01158>.

WU, J.; YANG, S.; ZHAN, R.; YUAN, Y.; WONG, D. F.; CHAO, L. S. A survey on llm-gernerated text detection: Necessity, methods, and future directions. arXiv preprint arXiv:2310.14724, 2023.

YUAN, L.; CHEN, Y.; CUI, G.; GAO, H.; ZOU, F.; CHENG, X.; JI, H.; LIU, Z.; SUN, M. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. In: OH, A.; NEUMANN, T.; GLOBERSON, A.; SAENKO, K.; HARDT, M.; LEVINE, S. (Ed.). Advances in Neural Information Processing Systems. Curran Associates, Inc., 2023. v. 36, p. 58478–58507. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2023/file/b6b5f50a2001ad1cbccca96e693c4ab4-Paper-Datasets_and_Benchmarks.pdf>.

ZAGO, R.; PEDOTTI, L. d. S. Bertugues: A novel bert transformer model pre-trained for brazilian portuguese. 2024.

ZHANG, Y.; WANG, M.; REN, C.; LI, Q.; TIWARI, P.; WANG, B.; QIN, J. Pushing The Limit of LLM Capacity for Text Classification. 2024.

ZHOU, C.; LI, Q.; LI, C.; YU, J.; LIU, Y.; WANG, G.; ZHANG, K.; JI, C.; YAN, Q.; HE, L.; PENG, H.; LI, J.; WU, J.; LIU, Z.; XIE, P.; XIONG, C.; PEI, J.; YU, P. S.; SUN, L. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT. 2023.