Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Detecção de Texto Gerado por Modelos de Linguagem: Avaliação de Desempenho e Análise Comparativa no Contexto da Língua Portuguesa

Lucas Tassi Facciolla

JUIZ DE FORA
SETEMBRO, 2024

Detecção de Texto Gerado por Modelos de Linguagem: Avaliação de Desempenho e Análise Comparativa no Contexto da Língua Portuguesa

Lucas Tassi Facciolla

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Jairo Francisco de Souza

JUIZ DE FORA

SETEMBRO, 2024

Detecção de Texto Gerado por Modelos de Linguagem: Avaliação de Desempenho e Análise Comparativa no Contexto da Língua Portuguesa

Lucas Tassi Facciolla

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Jairo Francisco de Souza
Doutor em Informática

Heder Soares Bernardino
Doutor em Modelagem Computacional

Luciana Conceição Dias Campos
Doutora em Engenharia Elétrica

JUIZ DE FORA

16 DE SETEMBRO, 2024

Resumo

Modelos de Linguagem de Grande Escala, como o ChatGPT, estão transformando a forma como a sociedade interage com o mundo digital, gerando respostas difíceis de diferenciar das fornecidas por humanos. Estas ferramentas, sob posse de agentes maliciosos, podem ser utilizadas para disseminar desinformação, produzindo notícias falsas que buscam distorcer a visão política da população. Sendo assim, a tarefa de atribuição de autoria a esses conteúdos se torna crucial. Estudos têm sido realizados para desenvolver técnicas de detecção de textos gerados por *Large Language Models*, fortemente influenciados pelo campo de processamento de linguagem natural. Entretanto, grande parte dos trabalhos concentra-se no idioma inglês, carecendo de abordagens voltadas para outros idiomas, como o português. Portanto, este trabalho visa avaliar e realizar uma análise comparativa da eficácia de diversos métodos de detecção no contexto da língua portuguesa. Com isso, é possível viabilizar a utilização destas ferramentas como mecanismo de defesa da desinformação no Brasil.

Palavras-chave: ChatGPT, *Large Language Models*.

Abstract

Large Language Models, such as ChatGPT, are transforming the way society interacts with the digital world, generating responses that are hard to differentiate from those provided by humans. These tools, in the hands of malicious agents, can be used to spread disinformation, producing fake news that seeks to distort the political views of the population. Thus, the task of attributing authorship to these contents becomes crucial. Studies have been conducted to develop techniques for detecting texts generated by *Large Language Models*, strongly influenced by the field of natural language processing. However, much of the work focuses on the English language, lacking approaches tailored to other languages, such as Portuguese. Therefore, this study aims to evaluate and conduct a comparative analysis of the effectiveness of various detection methods in the context of the Portuguese language. This will help to enable the use of these tools as a mechanism to defend against disinformation in Brazil.

Keywords: ChatGPT, *Large Language Models*.

Agradecimentos

Agradeço primeiramente ao meu orientador Jairo, que ao longo da minha graduação e durante todo o processo de orientação me ofereceu não apenas o seu vasto conhecimento, mas também apoio, sendo essencial para o desenvolvimento deste trabalho. Aos meus pais, Donato e Maria Imaculada, sou imensamente grato por toda a ajuda e suporte ao longo da vida, sempre garantindo que eu tivesse tudo o que precisava para seguir em frente em minha trajetória acadêmica e pessoal. Ao meu irmão Matheus, por todo o companheirismo e pela constante motivação que me incentivaram a seguir em frente nos momentos mais desafiadores. À minha namorada Teresa, que foi minha companheira de jornada — inclusive nas madrugadas de escrita, ainda que dormindo ao meu lado —, fornecendo apoio, paciência e compreensão, que foram fundamentais para que eu pudesse me dedicar a este projeto. Muito obrigado a todos.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

LLM	Large Language Models
NLP	Natural Language Processing
IA	Inteligência Artificial
API	Application Programming Interface
BPTT	Backpropagation Through Time
EUA	Estados Unidos da América
GPU	Graphics Processing Unit
JSON	JavaScript Object Notation
LoRA	Low-Rank Adaptation
ML	Machine Learning
MTLD	Measure of Textual Lexical Diversity
RLHF	Reinforcement Learning with Human Feedback
RNA	Redes Neurais Artificiais
RNN	Recurrent Neural Network / Rede Neural Recorrente
SDK	Software Development Kit
SIR	Semantic Invariant Robust Watermarking
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
LLM	Large Language Models

1 Introdução

O campo da Inteligência artificial é estudado por décadas, com seus primeiros passos inaugurados por Alan Turing, em 1950, com seus trabalhos envolvendo o pensamento de máquinas computacionais (TURING, 1950). O uso de aplicações neste campo popularizam-se cada vez mais, se apresentando como agentes na facilitação e melhoria de velocidade na realização de tarefas (BENKO; LáNYI, 2009). Atualmente, com o surgimento de invenções como Modelos de Linguagem de Grande Escala (LLMs) (CHU et al., 2024), concebidos com o propósito de *chatbots*, grande atenção da sociedade está voltada para o tema de IA (GARCíA-PEñALVO; VáZQUEZ-INGELMO, 2023). Inicialmente, em 1990, LLMs baseavam-se em modelos estatísticos que estimavam a probabilidade de uma sentença ocorrer num texto. Posteriormente, a utilização de redes neurais com distribuições de vetores, modelos pré-treinados e afinados, qualificaram ainda mais esses modelos (CHU et al., 2024).

Atualmente, decorrente da substancial evolução na disponibilidade de recursos computacionais e diversidade de dados, modelos são treinados em base de dados extremamente volumosas, caracterizando-os como Modelos de Linguagem de Grande Escala. Os LLMs atuam como resolvedores de tarefas de propósito geral, capazes de gerar informações a partir de texto com conhecimento pontual, comumente chamados de *prompt* (WHITE et al., 2023).

Os LLMs são capazes de proporcionar aumento de eficiência, decisões baseadas em dados e redução de custos. Portanto, modelos de linguagem estão cada dia mais transformando a indústria, ameaçando derrubar práticas de trabalho padrão em todos os setores (JERMAKOWICZ, 2023). No entanto, estes modelos de linguagem, atualmente, são capazes de gerar texto quase que indistinguível a seres humanos, tornando a determinação de autoria de conteúdo disponível uma tarefa importante.

Em (ELOUNDOU et al., 2023), os autores apontam que cerca de 80% da força de trabalho nos EUA poderia ter pelo menos 10% de suas tarefas afetadas pela introdução dos Modelos de Linguagem de Grande Escala (LLMs). Essas descobertas sugerem potenciais

implicações significativas nos âmbitos econômico, social e político. Tais ferramentas potencializam o surgimento de consequências maliciosas devido a sua capacidade de geração de texto. Das diversas problemáticas, observam-se plágio, disseminação de notícias e textos falsos, parcialidade, entre outros (SADASIVAN et al., 2024).

Modelos de Linguagem de Grande Escala apresentam diversas vulnerabilidades, que quando atacadas, colocam a ferramenta em um estado de alucinação, provendo diversas informações não factuais. Desta forma, considerando um cenário político conturbado, a utilização dessas ferramentas para a disseminação de notícias falsas torna-se uma atividade corriqueira. Diversos usuários maliciosos, em posse de um modelo de linguagem de propósito geral, podem manipular a ferramenta para gerar diversos artigos falsos de acordo com seus propósitos, buscando persuadir um leitor específico. Sendo assim, o uso não regulamentado deste recurso causa uma diminuição na credibilidade da informação disponível, fortalecendo-se contribuidor de desinformação na sociedade (KERTYSOVA, 2018).

1.1 Descrição do problema

LLMs, atualmente, são capazes de compor textos ao nível de escrita humana, tornando-se imperceptível para um leitor discernir se o verdadeiro autor do conteúdo é um sistema computacional (UCHENDU et al., 2020). Com a habilidade de LLMs em produzir textos que soam credíveis e indistinguíveis como notícias falsas contendo citações e evidências fabricadas, estas ferramentas podem também ser utilizadas voltadas para o contexto político (BARMAN; GUO; CONLAN, 2024). Em (KREPS; MCCAIN; BRUNDAGE, 2022), os autores realizam uma análise mostrando que LLMs são capazes de produzir narrativas falsas com mais credibilidade do que humanos, e assim, contribuir para ambientes políticos proliferados por *fake news* e desinformação.

A desinformação, definida como informação intencionalmente falsa (JIANG et al., 2024), pode ser gerada por uma LLM conduzida por um indivíduo mal-intencionado (PAN et al., 2023) que busca atingir influência da opinião pública (BARMAN; GUO; CONLAN, 2024).

Tratando-se especificamente da desinformação política alimentada através de

notícias falsas, diversos estudos (ZHOU et al., 2023) (HANLEY; DURUMERIC, 2024) levantaram preocupações relacionadas a atores utilizando modelos poderosos como ChatGPT, GPT-3 para gerar *fake news*, evidenciando o panorama citado.

Neste contexto, pessoas podem ser influenciadas pela leitura de notícias falsas, criadas por IA contendo diversos elementos que fornecem credibilidade a notícia, achando que estão lendo notícias verdadeiras (SU; CARDIE; NAKOV, 2024). Portanto, tendo sua percepção política distorcida e assim, tornando a definição de autoria destes conteúdos uma importante tarefa para influenciar na sua absorção do conteúdo (UCHENDU et al., 2020).

Embora estudos voltados para a detecção de textos gerados por modelos de linguagem tenham avançado (GHOSAL et al., 2023), inclusive voltados para a detecção de *fake news* (LI; ZHANG; MALTHOUSE, 2024), ainda se encontra como uma tarefa complexa (SADASIVAN et al., 2024). Ademais, esses trabalhos são desenvolvidos sobre o idioma inglês, sendo assim, dentro de um contexto brasileiro, esta atividade se encontra mais desafiadora.

Dito isso, tendo em vista o ambiente informacional brasileiro politicamente conturbado, recheado de notícias falsas disseminadas por sistemas alternativos de comunicação (MIGUEL, 2019) e, considerando o impacto atual de LLMs na informação, a elaboração de maneiras de se precaver da desinformação politica brasileira se torna ainda mais importante.

1.2 Justificativa

Tendo vista o panorama citado anteriormente, onde a detecção de textos gerados por IAs no cenário brasileiro ainda é incipiente e, considerando os recentes trabalhos avançados no campo de ML sobre detectores de textos gerados por modelos de linguagem massivo, o desenvolvimento de estudos que relacionem esses dois fatores podem contribuir positivamente para a melhora da comunicação e ambiente politico brasileiro.

Sendo assim, surge a necessidade de realizar uma avaliação comparativa dessas metodologias sobre o idioma português, para garantir a qualidade da sua aplicação. Este trabalho busca lançar mão destas abordagens de detecção para que possam ser usadas

no hábito da leitura e assim, ajudar na mitigação dos efeitos da desinformação política veiculada por notícias falsas no contexto brasileiro (SADASIVAN et al., 2024). A adoção dessas metodologias durante a leitura possibilita a atribuição de autor aos textos, desempenhando um papel crucial na leitura de notícias falsas.

Estas abordagens permitem ao leitor, quando se tratando de notícias geradas por maquinas inteligentes, desenvolver um olhar mais crítico durante a leitura. Neste caso, uma pessoa pode, fazendo uso de algumas dessas ferramentas e com uma notícia política em mãos, aplicar os detectores e, quando se tratando de uma noticia confeccionada por uma LLM, não levar o que foi constatado como uma verdade, indo atrás de validar o conteúdo por meio de fontes mais confiáveis.

1.3 Objetivos

Diante disso, o presente trabalho visa avaliar o desempenho e realizar uma análise comparativa sobre as diversas abordagens de detecção de texto gerados por LLMs no contexto da língua portuguesa. Busca-se verificar que as atuais metodologias de detecção são capazes de gerar acurácia semelhante quando aplicadas ao idioma português. Deseja-se contribuir para o desenvolvimento de técnicas confiáveis e versáteis (sob o ponto de vista de idiomas) de detecção de texto gerados por inteligências artificiais. Com isso, almeja-se assegurar a qualidade destas ferramentas para a utilização de mais um mecanismo de defesa contra a desinformação política veiculada através de notícias falsas.

1.4 Organização

O presente trabalho está organizado em seis capítulos. No Capítulo 2 é tratado a fundamentação teórica necessária para a compreensão do tema, abordando os conceitos de aprendizado de máquina, redes neurais, Modelos de Linguagem de Grande Escala (LLMs), a arquitetura *Transformer*, bem como métricas linguísticas e a categorização de detectores utilizada na literatura. No Capítulo 3, são apresentados os principais trabalhos relacionados, com ênfase nas abordagens existentes para a detecção de textos gerados por LLMs e suas limitações encontradas. O Capítulo 4 descreve os materiais e métodos

utilizados, incluindo a construção do conjunto de dados e o detalhamento dos classificadores desenvolvidos. O Capítulo 5 apresenta os experimentos realizados e os resultados obtidos, acompanhados de uma análise comparativa entre os métodos avaliados. Por fim, o Capítulo 6 reúne as conclusões do estudo, discutindo suas contribuições, limitações e sugestões para trabalhos futuros.

2 Fundamentação teórica

Neste capítulo são apresentados os conceitos teóricos fundamentais para a compreensão do trabalho proposto. Inicia-se com uma introdução ao aprendizado de máquina, com ênfase nos métodos de classificação baseados em extração de características, os quais compõem uma das abordagens analisadas neste estudo.

Em seguida, é explorado o funcionamento das redes neurais artificiais, desde os princípios básicos até as arquiteturas sequenciais, que fornecem a base para os atuais Modelos de Linguagem de Grande Escala (LLMs) e são essenciais para a tarefa de geração de texto.

Na sequência, discorre-se sobre os LLMs em si, seu propósito e funcionamento, com foco especial na sua arquitetura *Transformer* revolucionária. São também introduzidos os conceitos de *fine-tuning* e *Reinforcement Learning with Human Feedback* (RLHF), explicando como essas técnicas permitem aos modelos alcançar níveis de proficiência comparáveis à escrita humana, tornando difícil a tarefa de diferenciar conteúdos autênticos de produções artificiais. Neste momento, é feita a importante distinção entre LLMs e *chatbots* que fazem uso dessas tecnologias, dado que uma das estratégias adotadas neste trabalho envolve a utilização de modelos especializados em instruções. Após isso, na seção 2.5 é explicado sobre características textuais de LLMs e suas métricas relevantes, como perplexidade e diversidade lexical, buscando já introduzir o leitor a algumas observações nas produções destes modelos.

Complementando essa base teórica, é introduzido o método *Low-Rank Adaptation* (LoRA), uma técnica eficiente para realizar *fine-tuning* de forma otimizada, utilizada na implementação do detector com LLM empregado neste trabalho. Por fim, aborda-se a categorização dos métodos de detecção de texto presentes na literatura, distinguindo entre abordagens *white-box* e *black-box*.

2.1 Aprendizado de máquina

Aprendizado de máquina é a ciência do desenvolvimento de algoritmos e modelos estatísticos que os sistemas de computador usam para realizar tarefas sem instruções explícitas, confiando em padrões e inferências. O aprendizado de máquina busca uma relação matemática existente entre qualquer combinação de dados de entrada e saída no qual se concentra na construção de sistemas que aprendem, e melhoram o desempenho, com base nos dados que consomem (ORACLE, 2024). Portanto, modelos de aprendizado de máquina compõem a classe de algoritmos que se baseiam em exemplos de um determinado fenômeno.

Normalmente, algoritmos de aprendizado buscam encontrar valores ótimos para parâmetros presentes em funções f. A solução para este problema de otimização, encontrado em f, é chamado de modelo estatístico e o processo de encontrar os parâmetros desta função é chamado de treinamento (BURKOV, 2019). Estes modelos possuem limites, chamados de limites de decisão, que separam as diferentes classes presentes em aprendizados supervisionados.

No aprendizado supervisionado, o conjunto de dados utilizado para treinar modelos supervisionados possui uma coleção de itens rotulados $(x_i,y_i)n_i$, onde:

- $\mathbf{X_i}$ é o vetor de *features* para o i-ésimo exemplo, e y_i é o rótulo correspondente.
- Um vetor de *features* $\mathbf{x_i} = (x_1, x_2, \ldots, x_D)$ é um vetor de tamanho D, onde cada elemento x_j é uma *feature* que descreve alguma característica do exemplo.
- O rótulo y_i representa a saída alvo. Em problemas de classificação, o objetivo é atribuir uma classe a um determinado dado observado, e y_i pertence a uma classe finita denotada por $\{1, 2, \ldots, C\}$.

Resumidamente, o principal objetivo de um algoritmo de aprendizado supervisionado é usar este conjunto de dados para produzir um modelo que recebe um vetor de *features* arbitrário e produz informação de saída que permite deduzir o rótulo deste respectivo vetor de *features*. O problema de transformar informação em *features* é chamado de engenharia de *features*.

2.2 Redes neurais artificiais

Redes neurais artificiais (RNA) compõem a gama de técnicas de Aprendizado de Máquina. O objetivo de redes neurais é possibilitar a modelagem de problemas através da abstração do funcionamento do cérebro humano. As redes neurais computacionais são compostas por nós, chamados de neurônios, os quais são conectados entre si em diversas camadas. Estes nós formulam as unidades de processamento, trafegando e propagando informação ao longo da rede. Os sinais passados entre os neurônios de uma rede neural recebem pesos, que determinam as decisões da rede (WU; FENG, 2018). Mais precisamente, uma RNA é descrita como um conjunto de n entradas (*features* do problema), as quais são multiplicadas e somadas pelos pesos, culminando em um resultado que é comparado a uma função de perda. O principal objetivo de uma rede neural é minimizar essa função. Sendo assim, a rede neural realiza processos iterativos de ajuste dos pesos em busca de reduzir ao máximo este valor. Esta etapa consiste no treinamento da rede e faz uso de um algoritmo de retropropagação para atualização dos parâmetros ao longo da mesma, chamado de *backpropagation* (FLECK et al., 2016).

Após o cálculo da soma ponderada das entradas de um neurônio, aplica-se uma função de ativação. Essa etapa é fundamental para permitir que a rede neural represente relações complexas nos dados, possibilitando o entendimento de problemas não-lineares (RASAMOELINA; ADJAILIA; SINčáK, 2020).

A Figura 2.1 mostra a estrutura de uma rede neural *feedfoward*, com ligações unidirecionais entre os neurônios (MACêDO, 2017). Este é o modelo mais simples de rede neural, composta por quatro camadas principais. Em vermelho, a camada de entrada, que é responsável por receber os dados de entrada. Em seguida, as duas camadas ocultas internas, com neurônios conectados entre si de forma densa, ou seja, cada neurônio de uma camada está ligado a todos os neurônios da camada seguinte. Por fim, à direita, está a camada de saída que fornece o resultado final do processamento.

A viabilidade e o sucesso das redes neurais foram impulsionados por avanços tecnológicos recentes, como o desenvolvimento de Unidades de Processamento Gráfico (GPUs), e pela disponibilidade de grandes volumes de dados. Devido a essa combinação, as RNAs alcançaram um desempenho bom em diversas tarefas, como processamento de

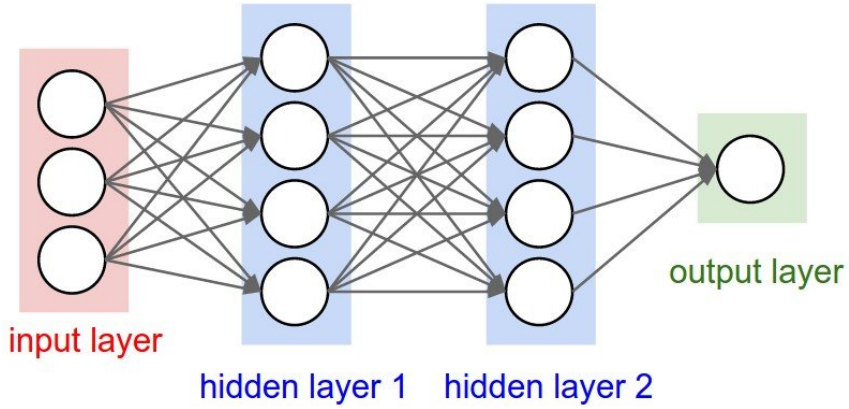

Figura 2.1: Representação de uma rede neural *feedfoward*

linguagem natural, reconhecimento de imagens e áudio. (LECUN; BENGIO; HINTON, 2015)

2.2.1 Redes neurais recorrentes

As Redes Neurais Recorrentes (RNNs, do inglês *Recurrent Neural Networks*) correspondem a uma arquitetura de redes neurais artificiais que permitem a utilização de saídas anteriores reutilizadas como entradas em etapas subsequentes, mantendo estados ocultos. Isso significa que as saídas atuais são influenciadas pelas informações adquiridas por meio de entradas anteriores, considerando o histórico da sequência de dados analisada (JURAFSKY; MARTIN, 2025).

Devido a esse comportamento, as RNNs, diferentemente de RNAs *feedforward*, são especialmente eficazes para resolver problemas que envolvem dados sequenciais, como a previsão de palavras em um texto, o qual é o mecanismo principal de LLMs. É essencial para o efetivo processamento de texto considerar o contexto e a ordem das informações (LECUN; BENGIO; HINTON, 2015). Nestas tarefas, este tipo de rede consegue capturar dependências posicionais entre os elementos da sequência, gerando uma distribuição de probabilidades de palavras coerentes com a sentença lida.

Em relação às redes neurais tradicionais (*feedforward*), nas RNNs, a função de perda considera o agregado dos erros ao longo de todos os estados no tempo e o algoritmo de retropropagação padrão é adaptado para o método conhecido como *Backpropagation Through Time* (BPTT), que realiza o ajuste dos pesos através dos pontos na sequência. Entretanto, RNNs clássicas sofrem com dependências sequenciais longas devido ao grande

número de parametros considerados nos cálculos. Este comportamento é conhecido como *exploding gradients* (STÉRIN; FARRUGIA; GRIPON, 2017).

2.3 Large Language Models

Os LLMs (Modelos de Linguagem de Grande Escala) são modelos matemáticos generativos da distribuição estatística de *tokens* em um vasto corpus de textos gerados por humanos, onde os *tokens* em questão incluem palavras, sendo compostos por uma grande quantidade de parâmetros. Estes modelos são chamados de generativos por possibilitarem a geração de outras amostras a partir deles (SHANAHAN, 2023). Isso significa que, dada a distribuição estatística das palavras na coleção de dados, é possível prever quais palavras têm maior probabilidade de seguir uma determinada sequência. Por exemplo, a palavra "cachorro" possui maior probabilidade de aparecer após a frase "latido do".

Os modelos de linguagem estatísticos são a alternativa mais simples para se modelar o texto em termos probabilísticos (MINAEE et al., 2024). Eles usam frequências de ocorrência de palavras ou sequências de palavras (*n-grams*) para prever a próxima palavra, com base em distribuições de probabilidade. Exemplos de n-gramas são ilustrados na Figura 2.2.

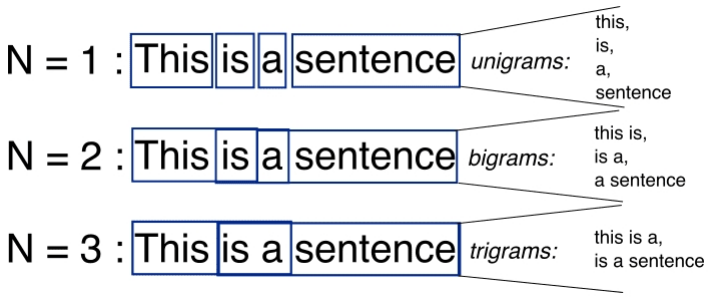

Figura 2.2: Modelo *uni-gram*, *bi-gram*, *tri-gram*

Um dos problemas que surgem em modelos que se baseiam em *n-gram* é a esparsidade de dados. Como esses modelos dependem da contagem de coocorrências de palavras ou n-gramas, eles enfrentam dificuldades em lidar com sequências de palavras raras ou novas, porque nem todas as combinações possíveis estão presentes nos dados de treinamento (DASGUPTA et al., 2024). Devido a limitações como estas, desenvolveram-

se modelos de linguagem neurais, que possuem capacidade de representar distribuições de probabilidade mais próximas da realidade.

2.3.1 Arquitetura *Transformer*

A arquitetura *Transformer*, proposta em (VASWANI et al., 2023), representa um marco na área de Processamento de Linguagem Natural e é a base para as LLMs modernas. Embora concebida inicialmente para tarefas de tradução de texto, sua aplicação foi ampliada para a geração de texto, áudio e imagens (LIN et al., 2021). O *Transformer* é uma arquitetura de redes neurais que processa os dados de entrada de maneira paralela, resultando em modelos com tempo de treinamento menor e qualidade superior às redes de processamento sequencial conhecidas.

O mecanismo central da arquitetura *Transformer* é o *self-attention*, que permite ao modelo capturar relações contextuais entre palavras de uma sequência, independentemente da distância entre elas. Esse mecanismo é implementado por meio da transformação de cada *token* de entrada utilizando três vetores treinados: Q (*query*), K (*key*) e V (*value*) (VASWANI et al., 2023). O modelo calcula a similaridade entre *queries* e *keys* para gerar vetores de atenção, que determinam a relevância dos elementos no contexto. Na prática, o *Transformer* utiliza múltiplas cabeças de atenção, conhecidas como *multi-head attention*, permitindo que o modelo aprenda diferentes representações semânticas simultaneamente.

A arquitetura *Transformer* é composta por duas partes principais: o *encoder* e o *decoder*, ambos formados por um empilhamento de N blocos idênticos. O *encoder* tem a função de mapear uma sequência de entrada de representações simbólicas $(x_1, ..., x_n)$ em uma sequência de representações $(z_1, ..., z_n)$. Por sua vez, o *decoder* utiliza essa representação para gerar uma sequência de saída $(y_1, ..., y_m)$, um símbolo de cada vez. Para isso, cada *token* de entrada é representado vetorialmente e enriquecido com um vetor de posição, sendo fornecido ao *encoder*, onde ocorrem as transformações utilizando o mecanismo de *self-attention*. A etapa do *decoder* ocorre de maneira similar, porém, a entrada é mascarada para que as próximas palavras sejam geradas com base na saída do *encoding* e da frase corrente, resultando em uma camada final que fornece a distribuição de probabilidade da próxima palavra.

Alguns LLMs modernos fazem uso apenas da etapa de *decoder*, sendo chamadas de arquiteturas *decoder-only* (LIU et al., 2018). Estes modelos são especializados em prever a próxima palavra a partir de um *prompt*, executada até um critério ser atingido (geralmente o *token* de fim de sentença).

2.3.2 *Fine-tuning*, RLHF e *Chatbots* baseados em LLMs

Os modelos de linguagem pré-treinados possuem características particulares. Estes são treinados em grandes volumes de texto de forma não supervisionada para aprender representações gerais da linguagem e depois serem submetidos a conjunto de dados específicos para que sejam ajustados conforme a tarefa a ser realizada. Este processo é chamado de *fine-tuning*. Os modelos com ajuste fino configuram modelos pré-treinados submetidos a conjuntos de dados específicos, caracterizando-os como modelos com grandes quantidades de parâmetros modificados, os quais possuem maior capacidade de entendimento da tarefa e domínio particular.

Uma das grandes dificuldades de LLMs é alinhar suas saídas com as preferências de resposta dos humanos. Desta forma, uma das maneiras de atacar este problema é através de uma abordagem de *fine-tuning* conhecida como *Reinforcement Learning Through Human Feedback* (RLHF) (CHAUDHARI et al., 2024). RLHF realiza o ajuste de parâmetros baseado na avaliação das respostas por humanos. Inicialmente, seres humanos avaliam respostas geradas pelo modelo de linguagem para determinadas instruções, fornecendo *feedback* que permitem treinar um modelo de recompensa capaz de prever preferências humanas. Em seguida, o modelo principal é ajustado por meio de algoritmos de aprendizado por reforço para maximizar essas recompensas, produzindo respostas mais úteis e seguras (CASPER et al., 2023).

Como resultado da aplicação conjunta de *fine-tuning* e RLHF, surgiram os *chatbots* modernos baseados em LLMs, capazes de manter diálogos coerentes, gerar respostas fluentes e se adaptar ao contexto do usuário. Esses modelos são treinados em grandes conjuntos de dados conversacionais e ajustados para fornecer respostas instrutivas, relevantes e alinhadas com valores humanos, como é o caso do InstructGPT (OUYANG et al., 2022). Além disso, alguns modelos mais recentes incorporam mecanismos de busca

em tempo real, ampliando sua capacidade de fornecer informações atualizadas (HARIRI, 2025). Estes fatores são componentes chaves para a construção de LLMs do estado da arte como o GPT-4 (OPENAI et al., 2024).

2.4 Características e métricas para avaliação de LLMs

A perplexidade é uma métrica amplamente utilizada para avaliar a confiança de um modelo de linguagem na previsão de uma sequência de texto. Em termos práticos, ela mede o quão previsível é o próximo *token* de uma sentença, dado um *prompt* e os *tokens* anteriores. Considerando uma sequência de *tokens* $X = (x_0, x_1, ..., x_n)$, a perplexidade de X é definida como:

$$\mathrm{PPL}(X) = \exp\left(-\frac{1}{t}\sum_{i=1}^{t}\log p_\theta(x_i \mid x_{<i})\right) \tag{2.1}$$

Isto é, a média da log-verossimilhança, encontrado como p_θ na Equação 2.1, dos *tokens* presentes na sequência X. O cálculo da perplexidade é uma transformação da equação de *cross-entropy* (XU et al., 2025). Quanto maior a probabilidade atribuída ao próximo *token* pelo modelo, menor será a perplexidade associada à sequência gerada. Assim, a perplexidade é inversamente proporcional à probabilidade do texto (JURAFSKY; MARTIN, 2025). Em geral, textos gerados por LLMs tendem a apresentar valores de perplexidade mais baixos, indicando maior previsibilidade. Em contraste, textos produzidos por humanos frequentemente apresentam perplexidade mais elevada.

Além da perplexidade, outras características são conhecidas por distinguir textos gerados por máquinas daqueles escritos por humanos. Textos humanos costumam apresentar maior diversidade de vocabulário, comprimento mais variado e mais elementos emotivos. Por outro lado, textos produzidos por LLMs tendem a exibir maior repetição e uma variação lexical mais limitada, acompanhado de um teor mais neutro (MUÑOZ-ORTIZ; GÓMEZ-RODRÍGUEZ; VILARES, 2024). Uma das métricas utilizadas para mensurar a diversidade lexical é a *Measure of Textual Lexical Diversity* (MTLD) (MCCARTHY; JARVIS, 2010). A MTLD calcula a diversidade do texto através da média do tamanho de segmentos que possuem até 72% de palavras distintas, sendo estes segmentos

compostos por pelo menos 10 palavras.

É importante observar, contudo, que muitas dessas análises estilométricas foram desenvolvidas e validadas majoritariamente com textos em inglês. Devido a diferenças morfológicas, sintáticas e semânticas entre línguas, tais características podem não se manifestar da mesma forma em outros idiomas como o português (SU; WU, 2024). Além disso, essas características podem variar dependendo do domínio presente no texto, alterando o desempenho desses indicadores (THORAT; YANG, 2024). No âmbito de notícias, gênero textual deste trabalho, cenários como esse estão sujeitos a acontecer, porque notícias possuem um tipo de escrita particular (ABDULLAYEVA, 2021).

2.5 *Low-Rank Adaptation*

Apesar do notável desempenho de LLMs na tarefa de gerar texto artificial, e sua capacidade de ajuste a tarefas específicas, estes modelos são muito grandes, possuindo alta quantidade de parâmetros. Modelos como LLaMA 3.2 3B[1], por exemplo, contemplam aproximadamente 3 bilhões de parâmetros treináveis. Dessa forma, operações com LLMs causam um alto custo computacional, demandando muito tempo e energia para a realização de práticas como o *fine-tuning*.

O *fine-tuning* tradicional de LLMs requer a atualização de todos os parâmetros do modelo. Nesse contexto, (HU et al., 2021) propõe o método LoRA (*Low-Rank Adaptation*). Esta abordagem busca reduzir drasticamente o número de parâmetros necessários para o *fine-tuning* de modelos através do congelamento dos parâmetros originais e adição de pequenos pesos. Estes pesos, chamados de adaptadores, são responsáveis por aprender a nova informação específica da tarefa e formular como resultado o novo modelo ajustado. Considerando W como sendo os pesos atuais, ΔW como os novos pesos, temos o novo modelo como sendo:

$$y = W + \Delta W \tag{2.2}$$

Entre os parâmetros de configuração mais relevantes do LoRA, destacam-se:

[1]https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

- LoRA *Rank*: Define a dimensão dos novos pesos introduzidos. Quanto maior o *rank*, maior o número de parâmetros treináveis.

- LoRA *Alpha*: Fator de escala aplicado às atualizações dos pesos. Um valor de *alpha* mais elevado permite alterações mais expressivas nos pesos, enquanto valores menores resultam em atualizações mais suaves e conservadoras

Como destacado em (HU et al., 2021) pelos autores, a vantagem do LoRA está em sua eficiência computacional, possibilitando a realização de ajustes finos em grandes modelos sem a necessidade de reprocessar todos os seus parâmetros. Além disso, os adaptadores gerados ao final do treinamento ocupam menos espaço em disco, facilitando o armazenamento e no mesmo tempo, apresentando desempenho próximo ao método de *fine-tuning* padrão.

2.6 Métodos de detecção *white-box* e *black-box*

A detecção de textos gerados por Modelos de Linguagem de Grande Escala pode ser realizada por meio de duas abordagens principais, definidas de acordo com o nível de acesso que os classificadores possuem ao modelo suspeito de ter produzido o conteúdo. Essas abordagens são denominadas *white-box* e *black-box*.

Os métodos *black-box* caracterizam-se pela ausência de acesso direto à estrutura interna do modelo gerador (TANG; CHUANG; HU, 2023). Nessas abordagens, os classificadores interagem com o modelo apenas por meio de uma interface de programação de aplicações (API – *Application Programming Interface*). As APIs fornecem um canal para acesso ao LLM. Sendo assim, o detector possui informações limitadas ou inexistentes sobre o modelo gerador. Métodos *black-box* são tipicamente desenvolvidos para extrair e selecionar *features* relevantes a partir dos textos, se apoiando em características estilísticas e padrões linguísticos observados para realizar a diferenciação através de métodos de aprendizado de máquina (TANG; CHUANG; HU, 2023). Em contrapartida, os métodos *white-box* pressupõem acesso total ao modelo de linguagem em questão, incluindo seus parâmetros, arquitetura e distribuição de probabilidade nas saídas (TANG; CHUANG; HU, 2023). Dessa forma, métodos *white-box* podem permitir a alteração das palavras

escolhidas, assim como acesso a distribuição de probabilidades da rede, permitindo a inserção de marcas d'água e utilização de métodos estatísticos a partir de métricas extraídas do modelo, como exemplo da perplexidade (YANG et al., 2023).

A escolha entre métodos *white-box* e *black-box* depende de diversos fatores, especialmente do contexto de aplicação e da disponibilidade de acesso ao modelo gerador. Enquanto os métodos *white-box* oferecem maior precisão e controle, sua aplicação pode ser inviável em ambientes onde o modelo não é de código aberto ou acessível. Em contrapartida, métodos *black-box* são mais amplamente aplicáveis, porém exigem atenção especial à qualidade dos dados de treinamento e à seleção adequada de atributos e algoritmos de classificação (TANG; CHUANG; HU, 2023).

3 Trabalhos relacionados

Devido à capacidade de geração de texto por parte de Modelos de Linguagem de Grande Escala e suas possíveis consequências maliciosas, diversos métodos foram desenvolvidos para discernir conteúdos produzidos por humanos daqueles gerados por IAs. Esses métodos, em geral, se agrupam em cinco categorias principais: marca d'água (*watermarking*), detecção baseada em estatísticas, aprendizado de máquina com *features* linguísticas, abordagens com *fine-tuning* de modelos para a tarefa de detecção e o uso de LLMs como detectores.

Apesar de não exclusivos para LLMs, técnicas de *watermarking* são largamente exploradas como métodos de detecção, requerendo acesso ao ambiente de desenvolvimento do modelo para sua implementação. Inicialmente, (GU; GOEL; Ré, 2022) propõem a inserção de gatilhos com padrões especiais, como palavras ou frases raras, associados a saídas arbitrárias, formulando um conjunto de dados de treinamento envenenado. Sendo assim, o modelo cria uma forte correlação entre o gatilho e o rótulo especificado pelos donos, tornando-se possível identificar a autoria do texto. No entanto, a identificação dos gatilhos pode ser realizada por meio da análise de saída, tornando o modelo vulnerável a ataques (LUCAS; HAVENS, 2023). Em contrapartida, (KIRCHENBAUER et al., 2024) busca um método que possa ser teoricamente validado e que não necessite do uso de modelos de linguagem para decodificar a marca d'água. A estratégia consiste na formulação aleatória de conjuntos de *tokens* "verdes" e "vermelhos" que são utilizados na criação do texto. O *watermark* pode ser facilmente identificado através da análise dos *tokens* contidos no texto. Este método demonstra-se performático e resistente a ataques de alteração de texto, degradando a sua qualidade em caso de alteração. Seguindo a mesma linhagem de modelos de *watermarking*, (LIU et al., 2024) propõem o *Semantic Invariant Robust Watermarking* (SIR) para LLMs, buscando solucionar a limitação do método de (KIRCHENBAUER et al., 2024) contra ataques de parafraseamento. A técnica faz uso de um modelo de *embedding language* e um modelo de *watermark* treinado. Estes produzem *logits* do vocabulário encontrado no texto, somando-os e obtendo a lista de próximas pa-

lavras a serem geradas. A detecção considera que um trecho de texto deve possuir marca d'água se a média dos seus *logits* for maior que zero. Por outro lado, seguindo uma outra abordagem de *watermarking*, o DeepTextMark (MUNYER et al., 2024) explora a inserção de marca d'água após o processamento por meio da substituição de palavras por sinônimos via *embedding vectors* e detecção através de modelos baseados em *Transformers*.

Uma característica marcante de estratégias de *watermarking* é a necessidade de acesso ao modelo para implantação da marca d'água. Nesse sentido, o DetectGPT (MITCHELL et al., 2023) aborda um método *zero-shot*, realizando perturbações ao texto e, posteriormente, comparando a probabilidade log dos *tokens*. Se a média do log das perturbações for alta, a amostra provavelmente é de uma LLM. A presunção do trabalho é que textos gerados por máquinas tendem a ocupar regiões de curvatura negativa em funções de probabilidade e têm uma média baixa. Outros trabalhos, como o de (VASILATOS et al., 2025), utilizam a métrica de perplexidade como indicador para diferenciar textos acadêmicos gerados pelo ChatGPT e textos de humanos. É estabelecido um valor numérico limiar que determina a autoria. O processo de computar perplexidade demanda acesso ao LLM em questão, sendo assim, como o DetectGPT, ambos são abordagens *white-box* e baseadas em estatística.

Outra linhagem de técnicas de detecção explora a análise de características inerentes ao texto por meio de métricas, sem requerer acesso ao LLM. Em (MAO et al., 2024) introduzem um método *black-box* para detecção de texto gerado por IAs. O trabalho assume que textos reescritos por LLMs tendem a variar pouco quando provindos de uma fonte que é um modelo de linguagem, diferentemente da reescrita de textos produzidos por humanos, que tendem a mudar bastante as palavras e sua organização. Sendo assim, a detecção de texto pode ser realizada pelo cálculo da similaridade do texto, utilizando medidas como similaridade de cossenos, sobre antes e depois da reescrita. Textos que são de LLMs, a princípio, terão seu score de similaridade maior do que com humanos. Tais abordagens requerem a troca de requisições entre a LLM e a aplicação de detecção, apresentando desafios operacionais como tempo de resposta e *rate-limit*. Ainda na linha de trabalhos voltados para o escopo de detecção *black-box*, outros estudos focam na utilização de aprendizado de máquina e redes neurais como classificadores.

Como citado anteriormente, textos produzidos por máquinas e humanos possuem inúmeras características linguísticas que os diferem, provendo uma sólida fonte para extração de *features* e produção de modelos de ML para classificação. Nesse sentido, (SHAH et al., 2023) constroem um classificador baseado em propriedades textuais como comprimento de palavras, contagem de sílabas, frequência de palavras e taxa de pontuação, sendo posteriormente submetido a diversos modelos de aprendizado de máquina como SVM, Decision Tree e Regressão Logística. Apesar de sua performance ser boa, encontrando 93% de precisão para regressão logística, alguns estudos apontam que classificadores baseados em ML não desempenham bem quando submetidos a contextos ambíguos (MINDNER; SCHLIPPE; SCHAAFF, 2023). Outra estratégia de detecção envolve o *fine-tuning* de modelos de linguagem utilizando bases de dados rotuladas que distinguem textos humanos e textos gerados por IA. O conjunto de dados HC3, desenvolvido por (GUO et al., 2023), reúne respostas de especialistas humanos e do ChatGPT em uma variedade de temas. Classificadores ajustados a partir desse conjunto apresentaram alto desempenho, especialmente quando considerados padrões de escrita específicos de cada domínio, apresentando resiliência a várias técnicas de ataque. Tal trabalho registrou resultados de precisão maiores que métodos populares na literatura como o *watermarking* de (KIRCHENBAUER et al., 2024) e o *zero-shot* de (MITCHELL et al., 2023). Por fim, dada a capacidade de LLMs para performar diferentes tarefas, alguns trabalhos, como o de (BHATTACHARJEE; LIU, 2023), utilizam as próprias LLMs para determinar a autoria do texto, somando-as à gama de técnicas de detecção de texto gerado por modelos de linguagem.

Embora todos esses métodos apresentem resultados promissores, a maioria deles foi desenvolvida e avaliada principalmente no idioma inglês. Ainda há uma lacuna quanto à eficácia dessas técnicas em outros idiomas, incluindo o português. Fatores como a morfologia rica, a variação sintática e o uso de expressões idiomáticas podem influenciar tanto a robustez das marcas d'água quanto a eficácia das abordagens linguísticas e estatísticas. Trabalhos como o de (CANDIDO et al., 2025) avaliam ferramentas comerciais na detecção sobre textos em português, apresentando resultados equiparáveis ao estado da arte. Porém, o conjunto de dados utilizado é pequeno, algumas ferramentas não res-

pondem corretamente a textos gerados artificialmente e o estudo não se aprofunda nos métodos utilizados pelas aplicações. Sendo assim, este trabalho propõe-se a preencher essa lacuna ao avaliar uma variedade de métodos existentes de detecção de textos gerados por LLMs no contexto da língua portuguesa, realizando uma analise comparativa sobre o desempenho.

Tabela 3.1: Resumo de abordagens de detecção de textos gerados por LLMs

Publicação	**Categoria**	**Tipo**	**Idioma**	**Estratégia**
Watermarking Pre-trained Language Models with Backdoor	Watermarking	White-box	Inglês	Insere gatilhos de padrões raros.
A Watermark for Large Language Models	Watermarking	White-box	Inglês	Conjuntos de *tokens* permitidos e não permitidos.
A Semantic Invariant Robust Watermark for LLMs	Watermarking	White-box	Inglês	Modifica *logits* através de *logits* semanticamente similares.
DeepTextMark	Watermarking	Black-box	Inglês	Substituição por sinônimos utilizando similaridade semântica.
HowkGPT	Statistical-based	White-box	Inglês	Análise de perplexidade de texto.
DetectGPT	Statistical-based	White-box	Inglês, Alemão	Medição da probabilidade log dos *tokens* presentes no texto.
RAIDAR: GENERATIVE AI DETECTION VIA REWRITING	Statistical-based	Black-box	Inglês	Similaridade de textos reescritos por LLMs.
Detecting and Unmasking AI-Generated Texts through Linguistic Features	Feature-based ML	Black-box	Inglês	Extração de *features* linguísticas para classificação.
How Close is ChatGPT to Human Experts? (HC3)	Fine-tuning	Black-box	Inglês, Chinês	Realiza *fine-tune* em modelos de linguagem em conjuntos de dados rotulados.
Fighting Fire with Fire: Can ChatGPT Detect AI-Generated Content?	LLM como detector	Black-box	Inglês	Utiliza os próprios LLMs para realizar a detecção.

4 Materiais e métodos

Este capítulo apresenta os materias e métodos para o desenvolvimento do trabalho. A Seção 4.1 aborda a formulação do conjunto de dados utilizado no trabalho, o qual é criado a partir do uso de um *dataset* externo de textos jornalísticos escritos por humanos. A Seção 4.2 detalha a construção dos modelos classificadores. A Figura 4.1 apresenta um resumo sobre o procedimento proposto.

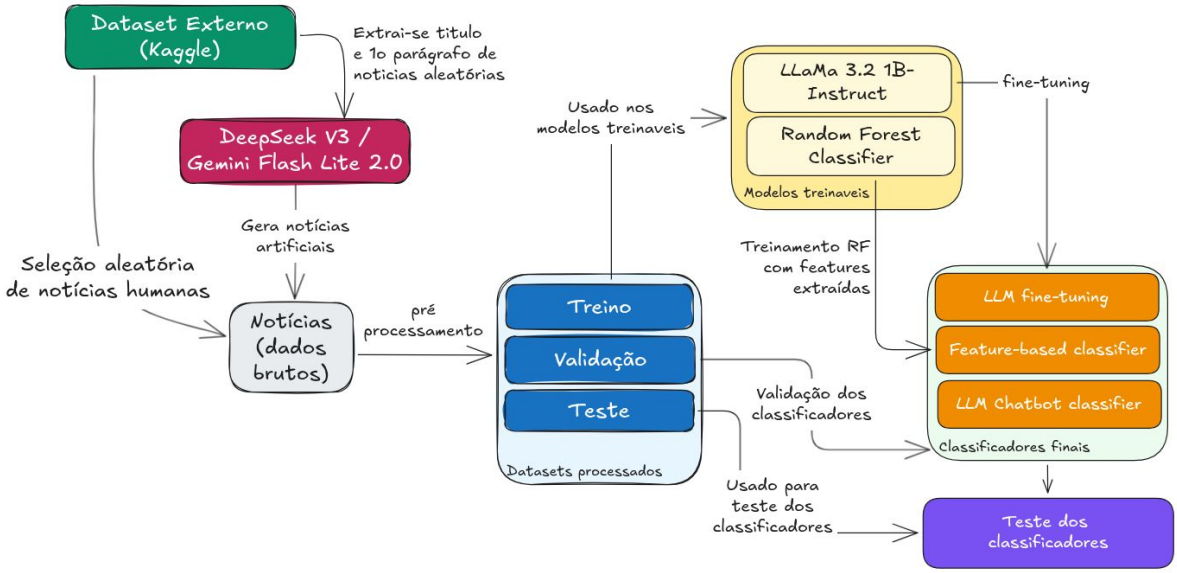

Figura 4.1: Pipeline que descreve os passos realizados desde a formulação do conjunto de dados aos experimentos com classificadores.

4.1 Descrição do conjunto de dados

Para o desenvolvimento deste trabalho, utilizou-se o *dataset* intitulado "News of the Brazilian Newspaper" [2], criado por Marlesson e disponibilizado na plataforma Kaggle. Esse conjunto de dados contém 167.053 exemplos de notícias, compostos por manchetes e conteúdos completos, oriundos do jornal Folha de São Paulo. As publicações abrangem o período entre janeiro de 2015 e setembro de 2017, sendo todos os textos escritos em língua portuguesa e obtidos via *web scraping*. O *dataset* apresenta uma grande diversidade

[2]https://www.kaggle.com/datasets/marlesson/news-of-the-site-folhauol

temática, contemplando desde notícias sobre política e economia até esportes e cultura, o que o torna uma base plural, interessante para análises linguísticas e experimentos com modelos de linguagem.

A escolha por esse *dataset* se deu principalmente por três fatores. Primeiramente, o fato de ter sido construído antes da popularização dos grandes modelos de linguagem (LLMs) aumenta a probabilidade que os textos ali presentes foram originalmente produzidos por humanos, sem a interferência de ferramentas modernas de geração automática. Esse aspecto é crucial para o objetivo deste trabalho, que é justamente diferenciar textos humanos de textos gerados por LLMs. Em segundo lugar, se trata de um conjunto de dados volumoso, possuindo pouco mais de 167 mil registros, o que oferece uma base grande para a criação de subconjuntos extensos. Por fim, o conteúdo ser composto por notícias jornalísticas é relevante para o trabalho em questão, visto que este é um domínio frequentemente utilizado para disseminação de desinformação nos dias atuais.

A partir dessa base original, foi construído três novos *datasets* compostos por 1700 notícias cada escritas por humanos (I), geradas pelo Gemini-Flash-Lite-2.0 (II) e geradas pelo modelo DeepSeek V3 (III). As notícias geradas pelos modelos foram criadas a partir dos mesmos registros presentes na base original, os quais foram selecionadas de maneira aleatória. Já para a porção compreendida por textos humanos, simplesmente extraiu-se os registros da base original, também de maneira aleatória. Para o subconjunto II, as notícias foram geradas por meio da API da Gemini (GenAI) e para o subconjunto III, por meio da API da OpenAI. Ambas através do módulo em Python. A escolha da participação de outro modelo possibilita comparar o desempenho observado entre os diferentes LLMs para a mesmas notícias. Além destes, foi gerado também um conjunto extra para teste com 100 notícias, 50 de humanos e 50 artificiais. A tabela 3.2 apresenta um resumo do tamanho e distribuições por categoria de notícia em cada subconjunto de dados.

Cada notícia gerada seguiu o mesmo formato das notícias humanas: uma manchete e um corpo textual. A geração foi realizada com base em uma *prompt* padronizada, que fornecia ao modelo uma manchete e um trecho do conteúdo da notícia original, com a seguinte instrução (em português):

```
Você está criando notícias.
```

Tabela 4.1: Distribuição de categorias entre Artificial e Humana

Tipo	**Artificial**	**Humana**
poder	361	313
mercado	276	267
mundo	236	254
esporte	218	213
cotidiano	177	199
ilustrada	124	146
colunas	87	101
paineldoleitor	37	35
saopaulo	30	35
educacao	24	18
tec	21	26
turismo	18	10
tv	17	21
ilustrissima	13	15
equilibrioesaude	13	12
sobretudo	15	15
empreendedorsocial	10	6
comida	7	4
ambiente	6	6
bbc	6	8
seminariosfolha	5	3
opiniao	5	10
ciencia	4	14
folhinha	4	0
guia-de-livros-discos-filmes	3	2
asmais	2	9
o-melhor-de-sao-paulo	2	2
banco-de-dados	1	1
especial	1	0
topofmind	1	2
rfi	1	0
serafina	0	1
Total	**1725**	**1750**

```
Considerando o título de notícia:

<manchete>

E o seguinte trecho que fornece contexto para criação da notícia:

<trecho da notícia>

Realize a criação de uma notícia inspirada no título e com auxílio

do trecho fornecido, de maneira clara, que tenha até 500 palavras.

Não insira caracteres especiais como '\n', '\t' no texto da sua resposta.

Ela deve ser em texto corrido sem quebra de linhas.
```

```
Use esse schema JSON:

News = {'title': str, 'corpus': str}
```

As respostas foram obtidas em formato JSON, permitindo um esquema estruturado e de fácil extração. O conteúdo gerado manteve-se todo em língua portuguesa. A temperatura de ambos modelos foi ajustada para 1.0, a fim de estabelecer um equilíbrio na geração das amostras, simulando textos mais realistas e diversos. O processo de geração foi automatizado com um script que submetia consultas à API da Gemini e da OpenAI. Para tratamento de erros oriundos de limites de requisições aos modelos (*rate-limit*), foi adotado 5 segundos de espera entre requisições para realizar o reenvio, guardando sempre o ponto de controle. A geração de 1700 notícias levou cerca de 1 hora no modelo Gemini e 8 horas para o modelo do DeepSeek. Os tempos foram fortemente influenciados pelas latências encontradas nas APIs utilizadas.

Todos os subconjuntos de dados foram submetidos ao mesmo fluxo de processamento, garantindo que compartilhem as mesmas condições textuais e de estrutura. Esse processo de pré-processamento envolveu diversas etapas. Primeiramente, foram filtradas apenas as notícias cujo título tivesse mais de 40 caracteres, uma vez que a mediana do comprimento dos títulos no *dataset* original era de 66. Para os textos gerados por modelos, foi extraído o que chamamos de "primeiro parágrafo" das notícias, definido como o trecho até o primeiro ponto encontrado no corpo do texto. Notícias cujo primeiro ponto apareciam antes de 150 caracteres foram descartadas. Isso permite possuir notícias mais próximas a realidade e que possuam informação suficiente para serem gerados pelas IAs, evitando a inclusão de notícias raspadas incorretamente ou muito pequenas. Além disso, para evitar interferência nos modelos de detecção, foram removidos caracteres especiais, como quebras de linha (\n) e tabulações (\t). Aplicou-se também a função strip() do Python a todos os campos textuais, eliminando espaços em branco desnecessários no início ou fim dos textos. Por fim, o corpo das notícias utilizadas foram truncados para até 500 *tokens*, de modo a evitar textos muito longos que podem ser custosos computacionalmente para a etapa de inferência e treinamento, além de consumir balanço financeiro das APIs. O processo é ilustrado na Figura 4.2.

O conjunto de dados final foi separado entre treino, validação e teste. Na Tabela

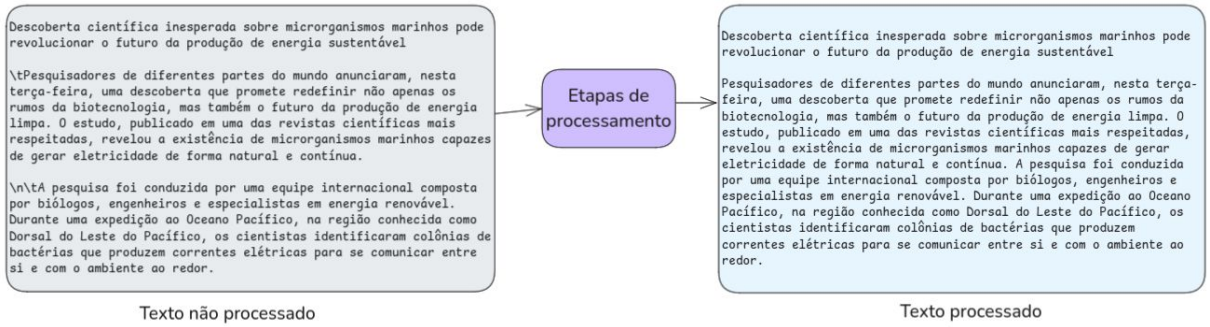

Figura 4.2: Texto obtido após a etapa de processamento

3.3 são apresentadas as distribuições.

Nome	Treino	Validação	Teste
Humano	1343	357	50
Artificial	2737	663	50

Tabela 4.2: Tabela com valores de treino e teste para os datasets criados.

4.2 Construção dos classificadores

Considerando as abordagens presentes na literatura para a detecção de texto gerado por Modelos de Linguagem de Grande Escala, desenvolveu-se 3 tipos de classificadores para avaliação comparativa no contexto da língua portuguesa. Dado a frequente aplicação e sucesso observado nas publicações, foram escolhidas as técnicas de LLM *fine-tuning*, *feature-based* e LLMs isoladas como detectores para aplicação no atual conjunto de dados. Todos os códigos foram realizados utilizando a linguagem Python na sua versão 3.12 e na plataforma do Kaggle, aproveitando seus recursos de GPUs para os treinamentos mais intensivos.

4.2.1 LLM *fine-tuning*

Para a estratégia de *fine-tuning* de LLMs, utilizou-se o modelo LLaMa 3.2 1B-Instruct fornecido pela Unsloth[3], obtido de maneira gratuita pela plataforma do HuggingFace[4]. Este modelo conta com 1 bilhão de parâmetros pré-treinados e é otimizado para fornecer respostas voltadas à instrução aos usuários. Sendo assim, é útil para a tarefa de clas-

[3]https://docs.unsloth.ai/
[4]https://huggingface.co/

sificação e torna possível experimentos com recursos computacionais e tempo acessível. Realizou-se o *fine-tuning* supervisionado do modelo pré-treinado a partir do dataset de notícias, fornecendo o seguinte *template* de *prompt*:

```
Below is a news article, written in Portuguese. Write a response that
appropriately completes the request

### Instruction:
Classify the text into 1 if Generated By Language Model or
0 if Generated By Human.
Your output should be only 0 or 1.

### Input:
News title: <title>
News corpus: <text>

### Response: <classe>
```

A parte relevante para o resultado é a "classe", a qual vem preenchida durante o treinamento com 1 ou 0, isto é, se o texto de notícia é, respectivamente, provindo de uma LLM ou não. Dessa forma, o objetivo é que o modelo seja capaz de se adaptar aos textos lidos e com o rótulo final observado, criando uma relação entre as palavras e o resultado final. Então, no momento da inferência, estimar a palavra com maior probabilidade de aparecer, neste caso, 1 ou 0. Os demais campos do *template* são preenchidos conforme a amostra a ser observada.

Dado que o modelo é grande, para viabilizar a solução, utilizou-se a GPU T4 fornecida gratuitamente na plataforma do Kaggle. Foi utilizado o modelo em sua forma original de 16-bits com um adaptador LoRA. O treinamento foi configurado para executar por 12 épocas com taxa de aprendizado de 0.0002, lotes de tamanho 16 e gradiente acumulado 4. Os parâmetros LoRA utilizados foram de 16 para *rank* e *alpha*. Esta parametrização gerou um tempo para treinamento de aproximadamente 7 horas e de inferência no *dataset* de teste de 6 segundos.

4.2.2 Classificador *feature-based*

A abordagem *feature-based* para classificação dos textos adota uma perspectiva mais tradicional de Aprendizado de Máquina, fundamentada na extração de características linguísticas e estatísticas do texto. A premissa é que textos gerados por LLMs possuem padrões mensuráveis que os diferenciam da escrita humana, como a fluidez, a diversidade lexical e a estrutura sintática.

Conforme levantado em (SHAH et al., 2023; MINDNER; SCHLIPPE; SCHAAFF, 2023) algumas dessas características, foram escolhidos os atributos de perplexidade, que mede quão previsível o texto é para um modelo de linguagem, esperando-se que textos de LLM possuam menor perplexidade. Também foi calculada a polaridade de sentimento para verificar tendências de neutralidade na escrita artificial, caracterizando-se como -1 para muito negativa e +1 para muito positiva. A hipótese é que LLMs tendem a gerar textos com sentimento mais neutro ou controlado. A estrutura e o estilo foram avaliados pela contagem de *stop words*, sinais de pontuação, número de sentenças e comprimento médio das palavras. Além disso, avaliou-se a diversidade léxica do texto através da métrica de MTLD, esperando-se que os textos humanos sejam mais variados. Um outro indicador foi a contagem de erros ortográficos, cuja ausência é uma forte característica de textos provindos de IAs. Por fim, a representação TF-IDF foi utilizada para capturar padrões de vocabulário e a importância relativa dos termos em cada documento. Com isso, totalizando um vetor de 8 atributos para alimentar o modelo.

Para implementação, a métrica de perplexidade foi fornecida pelo modelo com ajuste fino utilizado anteriormente, o indicador MTLD foi obtido através da implementação na biblioteca lexical-diversity[5] e o cálculo da polaridade de sentimento foi feito através da biblioteca TextBlob[6]. Para o restante das *features*, utilizou-se a biblioteca NLTK para processamento de linguagem natural em Português e por fim, a implementação de TF-IDF disponível no pacote scikit-learn. Foi utilizada a versão Python de todas as bibliotecas citadas.

A partir do vetor de atributos construído, foi treinado um classificador *Random Forest*, com a implementação também da biblioteca scikit-learn, utilizando os parâmetros

[5]https://pypi.org/project/lexical-diversity/
[6]https://textblob.readthedocs.io/en/dev/

do modelo de n_estimators como 200, e max_depth de 6. A busca pelos melhores hiperparâmetros foi feita através do conjunto de dados de validação afim de evitar sobreajuste.

4.2.3 LLMs como detectores

A terceira e mais direta abordagem investiga a capacidade inerente de LLMs de ponta de agirem como detectores sem qualquer treinamento específico. Para esta abordagem, foi utilizada a API da OpenAI, acessando o modelo GPT-4o, o qual é ajustado para produzir instruções. Para cada texto do conjunto de dados, uma requisição foi enviada à API contendo um *prompt* de sistema simples com o trecho *"You are a helpful assistant classifying texts into human generated (0) or LLM generated (1). Your response must be only 1 or 0."* que contextualiza o modelo sobre sua função e atividade a realizar, e como prompt de usuário utilizou-se o mesmo da abordagem de *fine-tuning*. Após a resposta da API, processamos a requisição com o valor retornado. A simplicidade desta técnica é sua maior vantagem, eliminando a necessidade de coleta de dados rotulados e de processos de treinamento, se apoiando ao vasto conhecimento e as capacidades textuais que esses modelos possuem.

5 Experimentos e Resultados

Neste capítulo, é detalhado a metodologia experimental e apresentado os resultados obtidos na tarefa de detecção de texto gerado por Modelos de Linguagem de Grande Escala (LLMs) no contexto da língua portuguesa bem como a interpretação dos resultados obtidos. A análise se desdobra nas três abordagens utilizadas. Adicionalmente, é realizada uma análise dos textos contidos no conjunto de dados, a fim de compreender potenciais vieses. Além disso, na Seção 5.1 é detalhado o ambiente dos experimentos, considerando fatores como linguagem de programação, *hardware* e métricas.

5.1 Configuração Experimental

Para a realização dos experimentos deste trabalho, foi utilizado o ambiente computacional da plataforma Kaggle, que oferece infraestrutura em nuvem com acesso gratuito a duas GPUs NVIDIA T4. A implementação dos experimentos foi realizada utilizando a linguagem de programação Python, devido à sua ampla adoção na comunidade de aprendizado de máquina e à compatibilidade com diversas bibliotecas especializadas. O processo de *fine-tuning* do modelo de linguagem foi conduzido com a biblioteca Unsloth, que fornece uma interface otimizada para ajuste fino de modelos da série LLaMA 3.2. Os modelos pré-treinados foram obtidos diretamente da plataforma HuggingFace, que também foi utilizada para a exportação e versionamento dos modelos ajustados. Para a implementação do classificador baseado em *features*, foi utilizado modelo *Random Forest* através da biblioteca scikit-learn. Já para o classificador com LLM isolada, utilizou-se o SDK da OpenAI[7] em Python para envio das mensagens ao modelo GPT-4o.

Por fim, as visualizações dos resultados foi empregada usando a biblioteca Matplotlib, e as métricas adotadas para avaliação dos modelos foram acurácia (*accuracy*), precisão (*precision*) e revocação (*recall*), de modo a fornecer uma análise mais abrangente do desempenho dos classificadores.

[7]https://platform.openai.com/docs/api-reference/introduction?lang=python

5.2 Análise dos dados

Antes de aplicar os modelos de detecção, realizou-se uma análise exploratória das distribuições dos textos nos conjuntos de dados utilizados. O objetivo foi entender as diferenças estruturais entre textos humanos e textos provindos de LLMs, de modo a encontrar possíveis vieses para os classificadores.

Foi observado que os textos gerados por LLMs possuíam diferentes distribuições quanto ao número de palavras. Apesar do *prompt* com as limitações ter sido especificado para os modelos geradores e o pré-processamento ter sido realizado, notou-se que os textos artificiais apresentavam uma distribuição mais normal, ocupando todo o intervalo (0 a 500 palavras). Já os textos de humanos apresentaram uma distribuição assimétrica à esquerda e uniforme, sugerindo que textos de humanos são mais longos. Na Figura 5.1 é possível observar as duas distribuições.

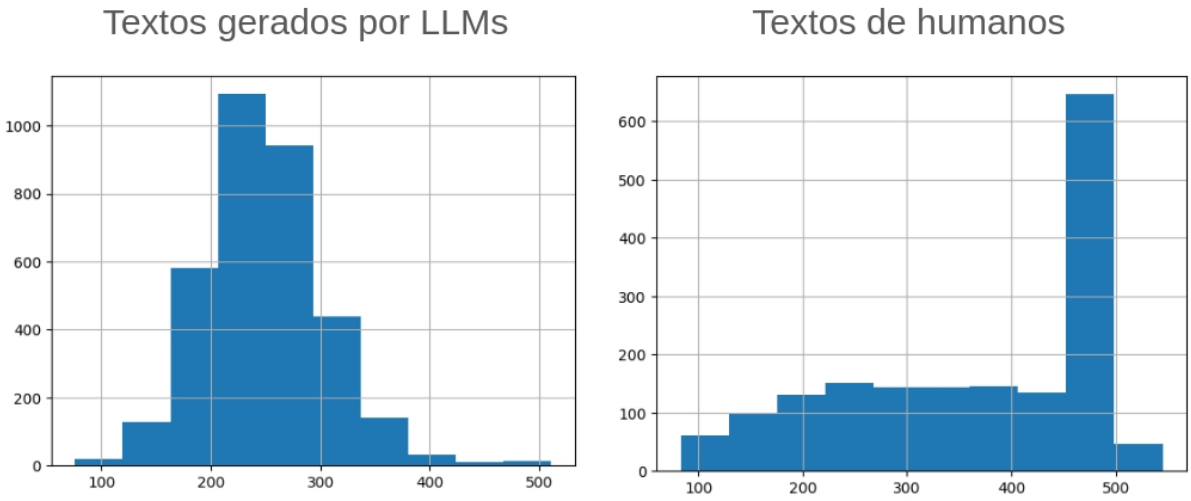

Figura 5.1: Distribuição do número de palavras nos textos de cada classe

Tal disposição dos dados pode trazer um viés de escolha para os classificadores, principalmente para o método baseado em *features* linguísticas, o qual leva em consideração questões como número de palavras e tamanho do texto para sua decisão.

5.3 Análise dos classificadores

Esta seção apresenta os resultados obtidos pelas diferentes abordagens testadas para a detecção de textos gerados por LLMs em língua portuguesa. Foram avaliados três métodos:

um modelo LLM com ajuste fino para a tarefa, um classificador tradicional baseado em *features* linguísticas e, por fim, o uso direto de um LLM isolado (GPT-4o) como detector. Os resultados são apresentados na Tabela 5.1.

Classificador	**Precision**	**Recall**	**Accuracy**
LLM fine-tuning	94%	100%	90%
Baseado em features	89%	82%	86%
LLM isolada	100%	6%	53%

Tabela 5.1: Resultados dos classificadores

Cada uma dessas abordagens oferece uma perspectiva distinta sobre o problema. O modelo com ajuste fino explora o potencial de adaptação dos LLMs à tarefa específica de detecção. O classificador baseado em *features* linguísticas busca diferenciar textos artificiais e humanos a partir de métricas objetivas. Já o uso direto de um LLM isoladamente avalia a capacidade do próprio modelo, sem ajustes, de reconhecer textos gerados artificialmente.

Além disso, na Tabela 5.2 é apresentado os resultados das previsões dos detectores encontradas para cada modelo em cada classificador, através da sua taxa de acerto, permitindo comparar o desempenho dos modelos separadamente e verificar se existe alguma diferença durante a detecção. Isto é, se algum modelo é mais detectável que o outro.

Classificador	**Modelo**	**Taxa de acerto**
LLM fine-tuning	DeepSeek	100%
	Gemini	100%
Baseado em features	DeepSeek	92%
	Gemini	72%
LLM isolado	DeepSeek	4%
	Gemini	8%

Tabela 5.2: Resultados dos classificadores por modelo

A análise a seguir busca avaliar o desempenho e entender as características dos dados que mais influenciaram suas decisões, assim como padrões observados durante os experimentos e suas conclusões.

5.3.1 Método de LLM *fine-tuning*

A primeira abordagem testada foi a de LLM com ajuste fino. O modelo foi avaliado sobre sua inferência no conjunto de teste. Os resultados observados foram notavelmente altos.

A acurácia de 90% indica que o modelo classificou corretamente quase a totalidade das amostras de teste. A precisão de 94% para a classe "1" demonstra boa capacidade de encontrar textos gerados por LLMs. Por fim, o alto valor de *recall* implica que o modelo foi capaz de identificar todos os textos gerados por LLM presentes no conjunto de teste, não deixando passar nenhum como se fosse humano.

Realizando uma análise sobre as escolhas feitas pelo classificador com ajuste fino, observou-se que durante a inferência, a média de perplexidade dos textos gerados por LLMs era consistentemente menor que a dos textos humanos — 9.52 para a classe 0 e 6.09 para a classe 1. Isso demonstra que os textos gerados por LLM estão alinhados com as probabilidades do modelo, não ficando tão surpreso com as palavras encontradas, isto é, os textos seguem padrões comuns à sua base de treinamento. Dessa forma, a perplexidade é um atributo relevante para detecção destes textos. A distribuição do valor de perplexidade é apresentado na Figura 5.2.

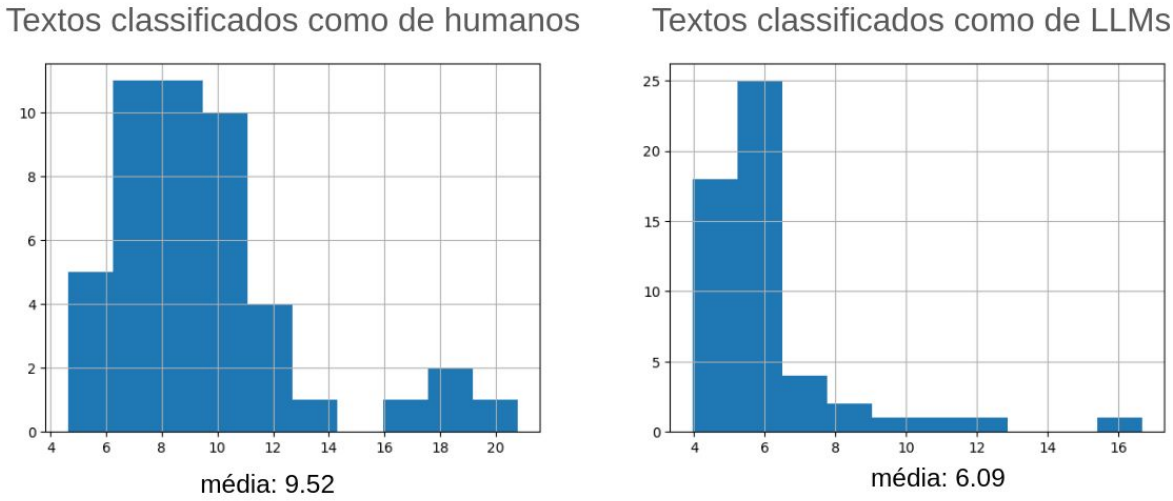

Figura 5.2: Distribuição da perplexidade nos textos classificados pelo método de fine-tuning

Além disso, para melhor entendimento das escolhas feitas pelo modelo, foi realizada a extração dos 50 bigramas mais frequentes por cada categoria predita e posteriormente, os bigramas foram categorizados manualmente dentro dos temas "Economia e Negócios", "Política e Justiça", "Cultura, Esportes e Mídia", "Tempo e Expressões Gerais", "Geografia e Lugares", "Social e Cotidiano". Veja as proporções nos gráficos da Figura 5.3.

Figura 5.3: Proporção da categoria dos bigramas para cada classe

A análise dos bigramas sugere a associação do modelo com ajuste fino com textos de vocabulário informativo e de caráter menos subjetivo à classe gerada por LLM. Por outro lado, bigramas que denotam linguagem coloquial e temas do cotidiano foram predominantemente associados à escrita humana. Este resultado revela padrões de escrita produzidos pelos modelos, os quais embora treinados para gerar texto "natural", tendem a produzir conteúdo com caráter mais informativo e estruturado, enquanto a produção humana é mais espontânea e variada.

Por fim, os experimentos realizados com a abordagem de LLM *fine-tuning* apresentaram resultados perfeitos para ambos os modelos testados, alcançando 100% em taxa de acerto. Esse desempenho indica que o *fine-tuning* permite identificar de maneira eficiente ambos os modelos, não apresentando uma diferença no seu desempenho. Isto é, desempenho semelhante na tarefa tanto para textos gerados pelo modelo da DeepSeek quanto para o da Google.

5.3.2 Método de *features* linguísticas

A segunda abordagem, através do uso do classificador *Random Forest*, obteve-os resultados apresentados na Tabela 5.1 para as 8 *features* extraídas. O método de classificação baseada em *features* linguísticas demonstrou-se como um método robusto. Esses valores indicam que o modelo foi capaz de distinguir com razoável eficácia textos humanos de textos gerados por LLMs, com destaque para sua precisão de 89%, reforçando sua confiabilidade ao apontar um texto como artificial. Esses resultados evidenciam o potencial de abordagens baseadas em métricas linguísticas para tarefas de detecção textual no contexto

da língua portuguesa.

Através do inspecionamento da importância de features no modelo, conforme apresentado no gráfico da Figura 5.4, é possível verificar que, como esperado, a perplexidade está posicionada entre os atributos mais significativos, juntamente do número de *stop-words* e frases.

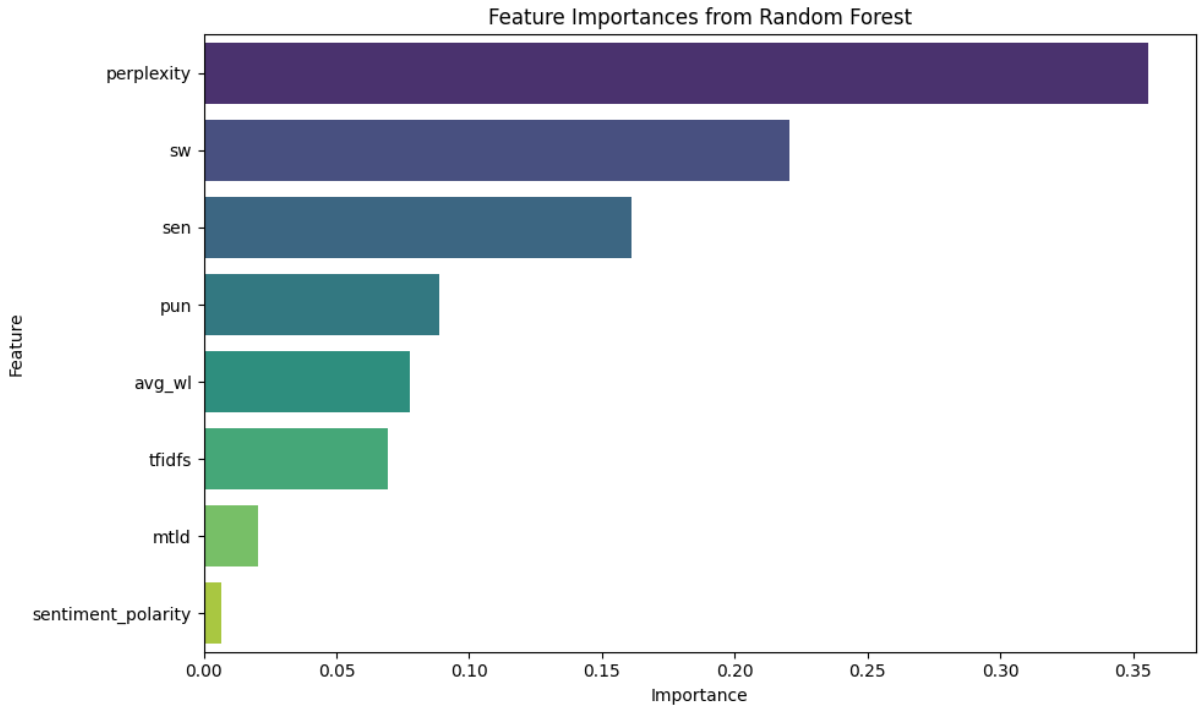

Figura 5.4: Relevância de cada *feature* para o modelo

A relevância de tais *features* é justificável dado que os valores médios desses atributos se diferenciam para cada classe. Dessa forma, o modelo é capaz de segmentar as amostras de maneira eficiente. Observe na Tabela 5.3 os números para o conjunto de treino.

Tabela 5.3: Médias das *features* linguísticas por classe de texto

Classe	**sw**	**pun**	**sen**	**avg_wl**	**mtld**	**perplexity**	**sentiment_polarity**	**tfidfs**
0 (humano)	151.5	53.1	16.9	3.58	112.4	6.36	0.028	11.54
1 (LLM)	99.9	34.8	9.99	3.73	112.5	4.17	0.018	10.11

A análise das *features* demonstrou que textos humanos se distinguem por uma maior imprevisibilidade (perplexidade) e por uma estrutura mais verbosa, com um número significativamente maior de sentenças, *stop words* e pontuação. Em contraste, a riqueza lexical não se mostrou um diferenciador eficaz, indicando que a detecção depende mais da

estrutura geral do texto do que da escolha de vocabulário. Além disso, textos produzidos por humanos tendem a ser ligeiramente menos neutros que de máquinas, mas que essa característica não contribuiu tanto para a previsão do modelo. Tais observações relacionadas a número de sentenças, *stop words* e pontuação podem ser reflexo do desbalanceamento na quantidade de palavras do texto de cada classe citado na Seção 5.2.

Para as diferenças encontrados para os dois modelos, o modelo DeepSeek apresentou taxa de acerto de 92%, enquanto o Gemini obteve um valor de 72% para a mesma métrica. Isso indica que ambos os detectores conseguem identificar corretamente os casos positivos, mas apresentam diferenças na capacidade de detecção. O desempenho superior no DeepSeek sugere que esse modelo esteve mais adequado às *features* extraídas, em comparação ao modelo Gemini.

5.3.3 Método de LLMs isolados como detectores

Por fim, testamos a capacidade de LLMs, sem qualquer *fine-tuning* específico posterior, de identificar se um texto foi gerado por humanos ou por outra LLM. Neste caso, utilizou-se um LLM de instrução. Os resultados foram significantemente inferiores, com acurácia próxima a 53%, sugerindo uma escolha praticamente aleatória por parte do modelo. A maioria das respostas fornecidas pelo modelo classificava os textos como de humanos, mesmo nos casos em que haviam sido gerados por outras LLMs. Como o conjunto de teste é balanceado em 50% de notícias artificias e 50% de notícias humanas, a acurácia ficou próxima a 50%. A precisão de 100% sugere que o modelo de linguagem só aponta o texto como de LLM quando está muito certo disso. O valor de *recall* em 6% também evidência que o modelo é extremamente conservador para prever a classe positiva. Este resultado mostra que o modelo, em sua forma genérica, não possui capacidade confiável de auto reconhecimento de texto gerado artificialmente, e tende a privilegiar a resposta "humano" por padrão, provavelmente por uma falta de alinhamento explícito com a tarefa e evitar falsas acusações.

O modelo DeepSeek alcançou apenas 4% de taxa de acerto, enquanto o Gemini atingiu um valor de 8%. Portanto, é possível notar uma pequena diferença entre o desempenho dos dois modelos, com o Gemini sendo ligeiramente mais detectável. Dessa forma,

este método de classificação não apresenta um desempenho consistente para ser utilizado na tarefa de detecção de texto gerado por Modelos de Linguagem de Grande Escala no contexto da língua portuguesa.

6 Conclusão

Este trabalho teve como objetivo principal investigar a viabilidade de métodos de detecção de textos gerados por modelos de linguagem no contexto da língua portuguesa, com foco na análise comparativa. Para isso, foi realizado um levantamento dos métodos existentes e uma avaliação experimental da eficácia de três detectores distintos: (I) LLM *fine-tuning*, (II) baseado em *features* linguísticas e (III) LLM isolado. Um dos principais méritos deste estudo foi adaptar e aplicar essas técnicas ao idioma português, ainda pouco representado na literatura científica sobre o tema. Além disso, a construção de um conjunto de dados com textos artificiais gerados por diferentes modelos a partir de notícias jornalísticas reais permitiu uma avaliação prática e realista de um gênero textual com grande potencial para ser utilizado em desinformação.

Os resultados demonstram que a detecção de textos gerados por LLMs em português é uma tarefa viável e promissora. Dentre os métodos analisados, o detector I apresentou o melhor desempenho, alcançando valores superiores a 94% em precisão e acurácia. Esse classificador se mostrou eficiente na identificação de textos gerados por ambos os modelos utilizados, apresentando padrões detectáveis de estrutura e estilo associados à produção de estrutura mais informativa e menos subjetiva. O classificador II também obteve resultados satisfatórios, com precisão e acurácia em torno de 86%. Nesse caso, atributos como perplexidade e a frequência de *stop words* emergiram como atributos relevantes para a diferenciação entre textos humanos e artificiais. Já o método III apresentou desempenho inferior, com baixa precisão e alta *recall* — possivelmente em função da ausência de conhecimento especializado na tarefa e da tendência dos modelos a evitarem falsas acusações, comportamento coerente com observações feitas em estudos similares na literatura. As observações apresentadas nos experimentos indicam a presença de padrões de escrita diferenciadores entre humanos e LLMs, o que abre espaço para o desenvolvimento de abordagens que explorem essas características textuais. Por fim, identificou-se que os diferentes detectores podem ter uma queda no desempenho da classificação dependendo do modelo que esta sendo utilizado para gerar o texto artificial.

No entanto, o trabalho apresenta algumas limitações. O tamanho do conjunto de dados utilizado é pequeno e restrito ao domínio jornalístico com textos artificiais possuindo estruturas bastante específicas, podendo introduzir viés aos detectores. Este panorama pode limitar a generalização dos mesmos. Ademais, parâmetros como a temperatura e outros fatores de geração dos modelos não são conhecidos, o que pode ter influenciado a forma e a complexidade dos textos produzidos. Sendo assim, trabalhos futuros podem ir na linha de explorar diferentes experimentos, envolvendo diferentes conjuntos de dados com as abordagens utilizadas neste trabalho e outras técnicas de detecção.

Bibliografia

ABDULLAYEVA, U. R. Specific language of newspaper style through headlines. *ACADEMICIA: An International Multidisciplinary Research Journal*, South Asian Academic Research Journals, v. 11, n. 5, p. 1188–1192, 2021.

BARMAN, D.; GUO, Z.; CONLAN, O. The dark side of language models: Exploring the potential of llms in multimedia disinformation generation and dissemination. *Machine Learning with Applications*, v. 16, p. 100545, 2024. ISSN 2666-8270. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S2666827024000215⟩.

BENKO, A.; LáNYI, C. S. History of artificial intelligence. In: KHOSROW-POUR, D. M. (Ed.). *Encyclopedia of Information Science and Technology, Second Edition*. IGI Global, 2009. p. 1759–1762. Disponível em: ⟨https://doi.org/10.4018/978-1-60566-026-4.ch276⟩.

BHATTACHARJEE, A.; LIU, H. *Fighting Fire with Fire: Can ChatGPT Detect AI-generated Text?* 2023. Disponível em: ⟨https://arxiv.org/abs/2308.01284⟩.

BURKOV, A. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN 9781999579517. Disponível em: ⟨https://books.google.com.br/books?id=0jbxwQEACAAJ⟩.

CANDIDO, L. S.; BARBOSA, C. A. de M.; MARTINS, L. G.; COSTA, E. J. Análise de ferramentas de detecção de ia para textos científicos em português gerados por chatgpt, gemini e deepseek. In: SBC. *Workshop sobre as Implicações da Computação na Sociedade (WICS)*. [S.l.], 2025. p. 78–91.

CASPER, S.; DAVIES, X.; SHI, C.; GILBERT, T. K.; SCHEURER, J.; RANDO, J.; FREEDMAN, R.; KORBAK, T.; LINDNER, D.; FREIRE, P.; WANG, T.; MARKS, S.; SEGERIE, C.-R.; CARROLL, M.; PENG, A.; CHRISTOFFERSEN, P.; DAMANI, M.; SLOCUM, S.; ANWAR, U.; SITHTHARANJAN, A.; NADEAU, M.; MICHAUD, E. J.; PFAU, J.; KRASHENINNIKOV, D.; CHEN, X.; LANGOSCO, L.; HASE, P.; BıYıK, E.; DRAGAN, A.; KRUEGER, D.; SADIGH, D.; HADFIELD-MENELL, D. *Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback*. 2023. Disponível em: ⟨https://arxiv.org/abs/2307.15217⟩.

CHAUDHARI, S.; AGGARWAL, P.; MURAHARI, V.; RAJPUROHIT, T.; KALYAN, A.; NARASIMHAN, K.; DESHPANDE, A.; SILVA, B. C. da. *RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs*. 2024. Disponível em: ⟨https://arxiv.org/abs/2404.08555⟩.

CHU, Z.; NI, S.; WANG, Z.; FENG, X.; YANG, M.; ZHANG, W. *History, Development, and Principles of Large Language Models-An Introductory Survey*. 2024. Disponível em: ⟨https://arxiv.org/abs/2402.06853⟩.

DASGUPTA, S.; MAITY, C.; MUKHERJEE, S.; SINGH, R.; DUTTA, D.; JANA, D. *HITgram: A Platform for Experimenting with n-gram Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2412.10717⟩.

ELOUNDOU, T.; MANNING, S.; MISHKIN, P.; ROCK, D. *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models*. 2023. Disponível em: ⟨https://arxiv.org/abs/2303.10130⟩.

FLECK, L.; TAVARES, M. H. F.; EYNG, E.; HELMANN, A. C.; ANDRADE, M. d. M. Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13, p. 47–57, 2016.

GARCíA-PEñALVO, F.; VáZQUEZ-INGELMO, A. What do we mean by genai? a systematic mapping of the evolution, trends, and techniques involved in generative ai. *International Journal of Interactive Multimedia and Artificial Intelligence*, v. 8, n. 4, p. 7–16, 12/2023 2023. ISSN 1989-1660. Disponível em: ⟨https://www.ijimai.org/journal/sites/default/files/2023-11/ijimai8_4_1.pdf⟩.

GHOSAL, S. S.; CHAKRABORTY, S.; GEIPING, J.; HUANG, F.; MANOCHA, D.; BEDI, A. S. *Towards Possibilities and Impossibilities of AI-generated Text Detection: A Survey*. 2023. Disponível em: ⟨https://arxiv.org/abs/2310.15264⟩.

GU, A.; GOEL, K.; Ré, C. *Efficiently Modeling Long Sequences with Structured State Spaces*. 2022. Disponível em: ⟨https://arxiv.org/abs/2111.00396⟩.

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. Disponível em: ⟨https://arxiv.org/abs/2301.07597⟩.

HANLEY, H. W. A.; DURUMERIC, Z. *Machine-Made Media: Monitoring the Mobilization of Machine-Generated Articles on Misinformation and Mainstream News Websites*. 2024. Disponível em: ⟨https://arxiv.org/abs/2305.09820⟩.

HARIRI, W. *Unlocking the Potential of ChatGPT: A Comprehensive Exploration of its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing*. 2025. Disponível em: ⟨https://arxiv.org/abs/2304.02017⟩.

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*. 2021. Disponível em: ⟨https://arxiv.org/abs/2106.09685⟩.

JERMAKOWICZ, E. K. The coming transformative impact of large language models and artificial intelligence on global business and education. *Journal of Global Awareness*, v. 4, n. 2, p. Article 3, 2023. Disponível em: ⟨https://scholar.stjohns.edu/jga/vol4/iss2/3⟩.

JIANG, B.; ZHAO, C.; TAN, Z.; LIU, H. *Catching Chameleons: Detecting Evolving Disinformation Generated using Large Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2406.17992⟩.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. ed. [s.n.], 2025. Online manuscript released January 12, 2025. Disponível em: ⟨https://web.stanford.edu/~jurafsky/slp3/⟩.

KERTYSOVA, K. Artificial intelligence and disinformation: How ai changes the way disinformation is produced, disseminated, and can be countered. *Security and Human Rights*, Brill — Nijhoff, Leiden, The Netherlands, v. 29, n. 1-4, p. 55 – 81, 2018. Disponível em: ⟨https://brill.com/view/journals/shrs/29/1-4/article-p55_55.xml⟩.

KIRCHENBAUER, J.; GEIPING, J.; WEN, Y.; KATZ, J.; MIERS, I.; GOLDSTEIN, T. *A Watermark for Large Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2301.10226⟩.

KREPS, S.; MCCAIN, R. M.; BRUNDAGE, M. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, v. 9, n. 1, p. 104–117, 2022.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, May 2015. ISSN 1476-4687. Disponível em: ⟨https://doi.org/10.1038/nature14539⟩.

LI, X.; ZHANG, Y.; MALTHOUSE, E. C. *Large Language Model Agent for Fake News Detection*. 2024. Disponível em: ⟨https://arxiv.org/abs/2405.01593⟩.

LIN, T.; WANG, Y.; LIU, X.; QIU, X. *A Survey of Transformers*. 2021. Disponível em: ⟨https://arxiv.org/abs/2106.04554⟩.

LIU, A.; PAN, L.; HU, X.; MENG, S.; WEN, L. *A Semantic Invariant Robust Watermark for Large Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2310.06356⟩.

LIU, P. J.; SALEH, M.; POT, E.; GOODRICH, B.; SEPASSI, R.; KAISER, L.; SHAZEER, N. *Generating Wikipedia by Summarizing Long Sequences*. 2018. Disponível em: ⟨https://arxiv.org/abs/1801.10198⟩.

LUCAS, E.; HAVENS, T. GPTs don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models. In: OVALLE, A.; CHANG, K.-W.; MEHRABI, N.; PRUKSACHATKUN, Y.; GALYSTAN, A.; DHAMALA, J.; VERMA, A.; CAO, T.; KUMAR, A.; GUPTA, R. (Ed.). *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. Toronto, Canada: Association for Computational Linguistics, 2023. p. 242–248. Disponível em: ⟨https://aclanthology.org/2023.trustnlp-1.21/⟩.

MACêDO, D. L. d. *Enhancing Deep Learning Performance Using Displaced Rectifier Linear Unit*. Dissertação (Master's thesis) — Universidade Federal de Pernambuco, Recife, Brazil, jul. 2017. Advisor: Teresa Bernarda Ludermir; Co-advisor: Cleber Zanchettin. Disponível em: ⟨https://repositorio.ufpe.br/handle/123456789/28361⟩.

MAO, C.; VONDRICK, C.; WANG, H.; YANG, J. *Raidar: geneRative AI Detection viA Rewriting*. 2024. Disponível em: ⟨https://arxiv.org/abs/2401.12970⟩.

MCCARTHY, P. M.; JARVIS, S. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, v. 42, n. 2, p. 381–392, May 2010. ISSN 1554-3528. Disponível em: ⟨https://doi.org/10.3758/BRM.42.2.381⟩.

MIGUEL, L. F. Jornalismo, polarização política e a querela das fake news. *Estudos em Jornalismo e Mídia*, v. 16, n. 2, p. 46–58, 2019.

MINAEE, S.; MIKOLOV, T.; NIKZAD, N.; CHENAGHLU, M.; SOCHER, R.; AMATRIAIN, X.; GAO, J. *Large Language Models: A Survey*. 2024. Disponível em: ⟨https://arxiv.org/abs/2402.06196⟩.

MINDNER, L.; SCHLIPPE, T.; SCHAAFF, K. Classification of human- and ai-generated texts: Investigating features for chatgpt. In: _____. *Artificial Intelligence in Education Technologies: New Development and Innovative Practices*. Springer Nature Singapore, 2023. p. 152–170. ISBN 9789819979479. Disponível em: ⟨http://dx.doi.org/10.1007/978-981-99-7947-9_12⟩.

MITCHELL, E.; LEE, Y.; KHAZATSKY, A.; MANNING, C. D.; FINN, C. *DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature*. 2023. Disponível em: ⟨https://arxiv.org/abs/2301.11305⟩.

MUÑOZ-ORTIZ, A.; GÓMEZ-RODRÍGUEZ, C.; VILARES, D. Contrasting linguistic patterns in human and llm-generated news text. *Artificial Intelligence Review*, v. 57, n. 10, p. 265, Aug 2024. ISSN 1573-7462. Disponível em: ⟨https://doi.org/10.1007/s10462-024-10903-2⟩.

MUNYER, T.; TANVIR, A.; DAS, A.; ZHONG, X. Deeptextmark: A deep learning-driven text watermarking approach for identifying large language model generated text. *IEEE Access*, PP, p. 1–1, 01 2024.

OPENAI; ACHIAM, J.; ADLER, S.; AGARWAL, S.; AHMAD, L.; AKKAYA, I.; ALEMAN, F. L.; ALMEIDA, D.; ALTENSCHMIDT, J.; ALTMAN, S.; ANADKAT, S.; AVILA, R.; BABUSCHKIN, I.; BALAJI, S.; BALCOM, V.; BALTESCU, P.; BAO, H.; BAVARIAN, M.; BELGUM, J.; BELLO, I.; BERDINE, J.; BERNADETT-SHAPIRO, G.; BERNER, C.; BOGDONOFF, L.; BOIKO, O.; BOYD, M.; BRAKMAN, A.-L.; BROCKMAN, G.; BROOKS, T.; BRUNDAGE, M.; BUTTON, K.; CAI, T.; CAMPBELL, R.; CANN, A.; CAREY, B.; CARLSON, C.; CARMICHAEL, R.; CHAN, B.; CHANG, C.; CHANTZIS, F.; CHEN, D.; CHEN, S.; CHEN, R.; CHEN, J.; CHEN, M.; CHESS, B.; CHO, C.; CHU, C.; CHUNG, H. W.; CUMMINGS, D.; CURRIER, J.; DAI, Y.; DECAREAUX, C.; DEGRY, T.; DEUTSCH, N.; DEVILLE, D.; DHAR, A.; DOHAN, D.; DOWLING, S.; DUNNING, S.; ECOFFET, A.; ELETI, A.; ELOUNDOU, T.; FARHI, D.; FEDUS, L.; FELIX, N.; FISHMAN, S. P.; FORTE, J.; FULFORD, I.; GAO, L.; GEORGES, E.; GIBSON, C.; GOEL, V.; GOGINENI, T.; GOH, G.; GONTIJO-LOPES, R.; GORDON, J.; GRAFSTEIN, M.; GRAY, S.; GREENE, R.; GROSS, J.; GU, S. S.; GUO, Y.; HALLACY, C.; HAN, J.; HARRIS, J.; HE, Y.; HEATON, M.; HEIDECKE, J.; HESSE, C.; HICKEY, A.; HICKEY, W.; HOESCHELE, P.; HOUGHTON, B.; HSU, K.; HU, S.; HU, X.; HUIZINGA, J.; JAIN, S.; JAIN, S.; JANG, J.; JIANG, A.; JIANG, R.; JIN, H.; JIN, D.; JOMOTO, S.; JONN, B.; JUN, H.; KAFTAN, T.; KAISER Łukasz; KAMALI, A.; KANITSCHEIDER, I.; KESKAR, N. S.; KHAN, T.; KILPATRICK, L.; KIM, J. W.; KIM, C.; KIM, Y.; KIRCHNER, J. H.; KIROS, J.; KNIGHT, M.; KOKOTAJLO, D.; KONDRACIUK Łukasz; KONDRICH, A.; KONSTANTINIDIS, A.; KOSIC, K.; KRUEGER, G.; KUO, V.; LAMPE, M.; LAN, I.; LEE, T.; LEIKE, J.; LEUNG, J.; LEVY, D.; LI, C. M.; LIM, R.; LIN, M.; LIN, S.; LITWIN, M.; LOPEZ, T.; LOWE, R.; LUE, P.; MAKANJU, A.; MALFACINI, K.; MANNING, S.; MARKOV, T.; MARKOVSKI, Y.; MARTIN, B.; MAYER, K.; MAYNE, A.; MCGREW, B.; MCKINNEY, S. M.; MCLEAVEY, C.; MCMILLAN, P.; MCNEIL, J.; MEDINA, D.; MEHTA, A.; MENICK, J.; METZ, L.; MISHCHENKO, A.; MISHKIN, P.; MONACO, V.; MORIKAWA, E.; MOSSING, D.; MU, T.; MURATI, M.; MURK, O.; MéLY, D.; NAIR, A.; NAKANO, R.; NAYAK, R.; NEELAKANTAN, A.; NGO, R.; NOH, H.; OUYANG, L.; O'KEEFE, C.; PACHOCKI, J.; PAINO, A.; PALERMO, J.; PANTULIANO, A.; PARASCANDOLO, G.; PARISH, J.; PARPARITA, E.; PASSOS, A.; PAVLOV, M.; PENG, A.; PERELMAN, A.; PERES, F. de A. B.; PETROV, M.; PINTO, H. P. de O.; MICHAEL; POKORNY;

POKRASS, M.; PONG, V. H.; POWELL, T.; POWER, A.; POWER, B.; PROEHL, E.; PURI, R.; RADFORD, A.; RAE, J.; RAMESH, A.; RAYMOND, C.; REAL, F.; RIMBACH, K.; ROSS, C.; ROTSTED, B.; ROUSSEZ, H.; RYDER, N.; SALTARELLI, M.; SANDERS, T.; SANTURKAR, S.; SASTRY, G.; SCHMIDT, H.; SCHNURR, D.; SCHULMAN, J.; SELSAM, D.; SHEPPARD, K.; SHERBAKOV, T.; SHIEH, J.; SHOKER, S.; SHYAM, P.; SIDOR, S.; SIGLER, E.; SIMENS, M.; SITKIN, J.; SLAMA, K.; SOHL, I.; SOKOLOWSKY, B.; SONG, Y.; STAUDACHER, N.; SUCH, F. P.; SUMMERS, N.; SUTSKEVER, I.; TANG, J.; TEZAK, N.; THOMPSON, M. B.; TILLET, P.; TOOTOONCHIAN, A.; TSENG, E.; TUGGLE, P.; TURLEY, N.; TWOREK, J.; URIBE, J. F. C.; VALLONE, A.; VIJAYVERGIYA, A.; VOSS, C.; WAINWRIGHT, C.; WANG, J. J.; WANG, A.; WANG, B.; WARD, J.; WEI, J.; WEINMANN, C.; WELIHINDA, A.; WELINDER, P.; WENG, J.; WENG, L.; WIETHOFF, M.; WILLNER, D.; WINTER, C.; WOLRICH, S.; WONG, H.; WORKMAN, L.; WU, S.; WU, J.; WU, M.; XIAO, K.; XU, T.; YOO, S.; YU, K.; YUAN, Q.; ZAREMBA, W.; ZELLERS, R.; ZHANG, C.; ZHANG, M.; ZHAO, S.; ZHENG, T.; ZHUANG, J.; ZHUK, W.; ZOPH, B. *GPT-4 Technical Report*. 2024. Disponível em: ⟨https://arxiv.org/abs/2303.08774⟩.

ORACLE. *O que é Machine Learning?* 2024. Urlhttps://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/.

OUYANG, L.; WU, J.; JIANG, X.; ALMEIDA, D.; WAINWRIGHT, C. L.; MISHKIN, P.; ZHANG, C.; AGARWAL, S.; SLAMA, K.; RAY, A.; SCHULMAN, J.; HILTON, J.; KELTON, F.; MILLER, L.; SIMENS, M.; ASKELL, A.; WELINDER, P.; CHRISTIANO, P.; LEIKE, J.; LOWE, R. *Training language models to follow instructions with human feedback*. 2022. Disponível em: ⟨https://arxiv.org/abs/2203.02155⟩.

PAN, Y.; PAN, L.; CHEN, W.; NAKOV, P.; KAN, M.-Y.; WANG, W. Y. *On the Risk of Misinformation Pollution with Large Language Models*. 2023. Disponível em: ⟨https://arxiv.org/abs/2305.13661⟩.

RASAMOELINA, A. D.; ADJAILIA, F.; SINčáK, P. A review of activation function for artificial neural network. In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. [S.l.: s.n.], 2020. p. 281–286.

SADASIVAN, V. S.; KUMAR, A.; BALASUBRAMANIAN, S.; WANG, W.; FEIZI, S. *Can AI-Generated Text be Reliably Detected?* 2024. Disponível em: ⟨https://arxiv.org/abs/2303.11156⟩.

SHAH, A.; RANKA, P.; DEDHIA, U.; PRASAD, S.; MUNI, S.; BHOWMICK, K. Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features. *International Journal of Advanced Computer Science and Applications*, The Science and Information Organization, v. 14, n. 10, 2023. Disponível em: ⟨http://dx.doi.org/10.14569/IJACSA.2023.01410110⟩.

SHANAHAN, M. *Talking About Large Language Models*. 2023. Disponível em: ⟨https://arxiv.org/abs/2212.03551⟩.

STÉRIN, T.; FARRUGIA, N.; GRIPON, V. An intrinsic difference between vanilla rnns and gru models. *COGNTIVE*, v. 84, p. 2017, 2017.

SU, J.; CARDIE, C.; NAKOV, P. *Adapting Fake News Detection to the Era of Large Language Models*. 2024. Disponível em: ⟨https://arxiv.org/abs/2311.04917⟩.

SU, Y.; WU, Y. *Robust Detection of LLM-Generated Text: A Comparative Analysis*. 2024. Disponível em: ⟨https://arxiv.org/abs/2411.06248⟩.

TANG, R.; CHUANG, Y.-N.; HU, X. *The Science of Detecting LLM-Generated Texts*. 2023. Disponível em: ⟨https://arxiv.org/abs/2303.07205⟩.

THORAT, S.; YANG, T. *Which LLMs are Difficult to Detect? A Detailed Analysis of Potential Factors Contributing to Difficulties in LLM Text Detection*. 2024. Disponível em: ⟨https://arxiv.org/abs/2410.14875⟩.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: ⟨https://doi.org/10.1093/mind/LIX.236.433⟩.

UCHENDU, A.; LE, T.; SHU, K.; LEE, D. Authorship attribution for neural text generation. In: WEBBER, B.; COHN, T.; HE, Y.; LIU, Y. (Ed.). *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020. p. 8384–8395. Disponível em: ⟨https://aclanthology.org/2020.emnlp-main.673⟩.

VASILATOS, C.; ALAM, M.; RAHWAN, T.; ZAKI, Y.; MANIATAKOS, M. *HowkGPT: Investigating the Detection of ChatGPT-generated University Student Homework through Context-Aware Perplexity Analysis*. 2025. Disponível em: ⟨https://arxiv.org/abs/2305.18226⟩.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. *Attention Is All You Need*. 2023. Disponível em: ⟨https://arxiv.org/abs/1706.03762⟩.

WHITE, J.; FU, Q.; HAYS, S.; SANDBORN, M.; OLEA, C.; GILBERT, H.; ELNASHAR, A.; SPENCER-SMITH, J.; SCHMIDT, D. C. *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. 2023. Disponível em: ⟨https://arxiv.org/abs/2302.11382⟩.

WU, Y.-c.; FENG, J.-w. Development and application of artificial neural network. *Wireless Personal Communications*, v. 102, n. 2, p. 1645–1656, Sep 2018. ISSN 1572-834X. Disponível em: ⟨https://doi.org/10.1007/s11277-017-5224-x⟩.

XU, J.; ZHANG, H.; YANG, Y.; YANG, L.; CHENG, Z.; LYU, J.; LIU, B.; ZHOU, X.; BACCHELLI, A.; CHIAM, Y. K.; CHIEW, T. K. One size does not fit all: Investigating efficacy of perplexity in detecting llm-generated code. *ACM Transactions on Software Engineering and Methodology*, Association for Computing Machinery (ACM), jul. 2025. ISSN 1557-7392. Disponível em: ⟨http://dx.doi.org/10.1145/3748506⟩.

YANG, X.; PAN, L.; ZHAO, X.; CHEN, H.; PETZOLD, L.; WANG, W. Y.; CHENG, W. *A Survey on Detection of LLMs-Generated Content*. 2023. Disponível em: ⟨https://arxiv.org/abs/2310.15654⟩.

ZHOU, J.; ZHANG, Y.; LUO, Q.; PARKER, A. G.; CHOUDHURY, M. D. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2023. (CHI '23). ISBN 9781450394215. Disponível em: ⟨https://doi.org/10.1145/3544548.3581318⟩.