
Geração e Detecção de Textos Artificiais em Redes Sociais por meio de Modelos de Linguagem de Grande Escala

Gabriel Vinícius Ramos de Lacerda



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2025

Gabriel Vinícius Ramos de Lacerda

**Geração e Detecção de Textos Artificiais em
Redes Sociais por meio de Modelos de
Linguagem de Grande Escala**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau de
Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Fabíola Souza Fernandes Pereira

Monte Carmelo - MG

2025

Este trabalho é dedicado à minha mãe Terezinha, que sempre me apoiou e tornou essa etapa da minha vida possível, e a Deus, que sempre foi nosso guia nos momentos bons e, principalmente, nos ruins.

Agradecimentos

Gostaria de expressar minha profunda gratidão a todo o corpo docente da Universidade Federal de Uberlândia, que me proporcionou o conhecimento necessário para atuar na área de tecnologia. Em especial, agradeço à professora e orientadora Fabíola, que me acompanhou com dedicação do início ao fim deste trabalho.

Agradeço também a toda a minha família, cujo apoio foi essencial nesta trajetória. Em especial, à minha mãe, Terezinha Batista, que lutou para que eu tivesse as melhores condições antes e durante minha jornada universitária, e ao meu primo Marlon Brendo, que sempre me apoiou durante essa jornada.

*“Para examinar a verdade, é necessário, uma vez na vida, pôr todas as coisas em
dúvida tanto quanto possível.”
(René Descartes)*

Resumo

A crescente utilização de Modelos de Linguagem de Grande Escala (LLMs), trouxe a necessidade de discussões e análises sobre seu impacto em plataformas digitais, como redes sociais, devido à dificuldade na distinção de conteúdos gerados por tais tecnologias. Este trabalho explora o uso de LLMs na simulação de interações humanas no ambiente digital, mais especificamente na plataforma X (antigo *Twitter*), e analisa a eficácia de detectores presentes na literatura na distinção de textos gerados pelos modelos selecionados, além de investigar como a utilização de personas pré-definidas influencia a estrutura dos textos e sua detecção com base em métricas linguísticas. A metodologia deste trabalho envolve a coleta de *posts* reais extraídos do X a partir de temas selecionados. Foram desenvolvidas personas que foram adotadas pelos modelos de linguagem para gerar comentários artificiais relacionados a cada postagem. A partir destes dados gerados, diversos detectores da literatura foram utilizados, resultando em análises relacionadas à eficácia tanto da geração, quanto da detecção dos comentários. Além destes testes, foram aplicadas métricas linguísticas para descobrir as similaridades estruturais dos textos sintéticos, dos textos originais do X e como a não utilização das personas afeta o resultado das métricas e detecções. A fim de automatizar o fluxo de trabalho, foi desenvolvida uma ferramenta web e alguns *scripts* que integram os processos do experimento. Os resultados demonstraram que os LLMs conseguem, de forma satisfatória, replicar padrões de linguagem humana presentes em redes sociais, utilizando informalidade e variações linguísticas, tornando difícil a distinção entre texto real e gerado por IA. Detectores de IA se mostraram ferramentas relevantes para tal distinção. Além disso, foi possível observar que a utilização de personas e a variação dos parâmetros de geração influenciam de forma sutil a estrutura dos textos, assim como a complexidade de sua detecção. Tais resultados demonstram a evolução das técnicas de detecção a partir de diversos estilos de abordagem. Este trabalho contribui para a compreensão das capacidades dos modelos de linguagem atuais e das ferramentas de detecção, além de ressaltar a necessidade contínua de avanços tecnológicos na área de detecção de conteúdo artificial devido a seus impactos éticos, sociais e culturais.

A aplicação desenvolvida, bem como as bases de dados utilizadas nos experimentos,

estão publicamente disponíveis em (LACERDA, 2025).

Palavras-chave: Modelos de Linguagem de Grande Escala, Redes Sociais, Detectores de Texto de IA, Inteligência Artificial, Simulação de Interações Humanas.

Lista de ilustrações

Figura 1 – Exemplificação do processo de <i>Back-Translation</i> de uma frase. Figura retirada de (CHAUDHARY, 2020).	26
Figura 2 – Exemplificação do funcionamento do modelo Grover. Retirado de (ZELERS et al., 2020).	34
Figura 3 – Fluxo de interação entre <i>bots</i> e usuários na plataforma Mastodon para o experimento que simula a comunicação humana sobre eventos mundiais. Imagem retirada de (RADIVOJEVIC; CLARK; BRENNER, 2024). . .	45
Figura 4 – Fluxo de trabalho do experimento.	48
Figura 5 – Tela principal da aplicação, ilustrando as funcionalidades disponíveis. .	53
Figura 6 – Tela de geração de comentários.	53
Figura 7 – Ollama acessado a partir do terminal linux.	55
Figura 8 – Exemplo de comentários gerados para o tema de inteligência artificial utilizando o modelo LLaMA3.2.	58
Figura 9 – Funcionalidade de geração de comentários a partir de características escolhidas pelo usuário.	58
Figura 10 – Tela de detecção de texto gerado por IA.	61
Figura 11 – Interface web do ZeroGPT.	64
Figura 12 – Interface web do GPTZero.	64
Figura 13 – Tela de detecção personalizada de comentários.	66
Figura 14 – Tabela de resultados no formato de tabela (filtrados a partir de um comentário específico).	68
Figura 15 – Comparação da efetividade dos detectores por meio do gráfico.	68
Figura 16 – Comparação da efetividade dos detectores/LLMs de acordo com o tema dos comentários por meio de gráficos.	69
Figura 17 – Gerador de gráficos personalizados a partir de comentários próprios enviados para a aplicação.	69
Figura 18 – Comparação de efetividade geral dos LLMs na geração de comentários sintéticos contra os detectores.	77

Figura 19 – Comparação de efetividade dos LLMs em relação a cada tema.	78
Figura 20 – Comparação de efetividade dos LLMs utilizando apenas os detectores mais expressivos conforme resultados analisados na Seção 4.3.	78
Figura 21 – Efetividade dos LLMs na geração de texto sem utilização de personas.	79
Figura 22 – Efetividade dos LLMs na geração de texto sem utilização de personas na instrução (em relação a cada tema).	79
Figura 23 – Efetividade dos LLMs na geração de texto sem utilização de personas na instrução (detectores mais expressivos).	80
Figura 24 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs.	81
Figura 25 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs divididos pelo tema.	82
Figura 26 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs sem a utilização de personas pré-definidas.	82
Figura 27 – Detecção do RoBERTa e HuggingFace Detector a partir de comentários reais extraídos do X.	83

Lista de tabelas

Tabela 1	– Tabela adaptada de (BALDWIN et al., 2013), que apresenta os padrões de comprimento de textos em redes sociais.	20
Tabela 2	– Tabela gerada a partir de dados referenciados em (BROWN et al., 2020; VASWANI et al., 2023; DEVLIN et al., 2019; CHANG et al., 2023) sobre características de LLMs.	23
Tabela 3	– Tabela adaptada de (WU et al., 2024) e (YANG et al., 2023) sobre datasets de treinamento de detectores de texto gerados por IA.	29
Tabela 4	– Ferramentas para Detecção de Conteúdo Gerado por IA.	37
Tabela 5	– <i>Posts</i> referentes aos temas escolhidos.	49
Tabela 6	– Semelhanças entre os detectores GPTZero e ZeroGPT. Adaptado de informações disponíveis nos sites próprios de cada ferramenta.	65
Tabela 7	– Combinação de comentários por tema e LLM.	71
Tabela 8	– Tempo médio gasto para gerar cada comentário de acordo com cada LLM.	71
Tabela 9	– Métricas referentes aos textos dos comentários gerados pelos LLMs.	72
Tabela 10	– Tamanho médio e formalidade dos comentários em posts do X.	73
Tabela 11	– Métricas linguísticas dos comentários em posts do X.	73
Tabela 12	– Métricas resultantes dos comentários gerados pelos LLMs sem utilização de personas.	76
Tabela 13	– Tempo médio gasto para detecção de cada comentário.	81

Lista de siglas

AUROC Área Sob a Curva ROC (*Area Under the Receiver Operating Characteristic Curve*)

CNN Redes Neurais Convolucionais (*Convolutional Neural Networks*)

GLTR Giant Language Model Test Room

IA Inteligência Artificial

LLM Modelo de Linguagem de Grande Escala (*Large Language Model*)

MBTI Myers-Briggs Type Indicator

MLM Masked Language Modeling

MMD-MP Maximum Mean Discrepancy with Moment-based Projection

NLP Processamento de Linguagem Natural (*Natural Language Processing*)

OCEAN Big Five Personality Traits

PPO Otimização de Política (*Policy Optimization*)

RADAR Robust AI-text Detection And Recognition

RNN Redes Neurais Recorrentes (*Recurrent Neural Networks*)

Sumário

1	INTRODUÇÃO	14
1.1	Motivação	15
1.2	Problema	16
1.3	Hipótese	16
1.4	Objetivos	17
1.5	Contribuições	17
1.6	Organização da Monografia	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Textos de Redes Sociais	19
2.2	Modelos de Linguagem de Grande Escala	20
2.2.1	História e Evolução	21
2.2.2	Características Gerais dos LLMs	22
2.2.3	Treinamento	23
2.2.4	Arquitetura Transformers	27
2.2.5	Aplicações dos LLMs	27
2.3	Detectores de Textos Gerados por IA	28
2.3.1	Conjuntos de dados	28
2.3.2	Métodos de detecção	30
2.3.3	Ferramentas	36
2.4	Trabalhos Correlatos	37
2.4.1	Binoculars: Detecção de Texto Gerado por IA com Modelos Pré-Treinados	38
2.4.2	RADAR: Detectando Texto Gerado por IA com Aprendizado Adversarial	39
2.4.3	GLTR: Detecção Estatística e Visualização de Texto Gerado por IA . .	41
2.4.4	RoBERTa: Fine-Tuning do BERT para Melhor Performance	42
2.4.5	MMD-MP: Detecção de Texto Gerado por IA com Otimização Multi- população	42
2.4.6	LLMs Among Us: A Influência dos Bots nas Redes Sociais	43

3	METODOLOGIA	47
3.1	Seleção da Rede Social e dos Temas	47
3.2	Seleção das Publicações	49
3.3	Criação de Personas	50
3.4	Definição das Ferramentas e Funcionalidades da Aplicação . . .	52
3.4.1	Tecnologias Utilizadas	52
3.4.2	Seleção de LLMs e Geração de Comentários	53
3.4.3	Métricas Linguísticas	59
3.4.4	Escolha dos Detectores e Processo de Detecção	60
3.5	Geração dos Resultados	66
4	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	70
4.1	Método para a Avaliação	70
4.2	Avaliação dos LLMs de Geração de Texto	71
4.2.1	Métricas dos Textos de IA vs Textos Originais do X	71
4.2.2	Influência das Personas nas Métricas	75
4.2.3	Comparação Visual de Efetividade dos LLMs de Geração de Texto . . .	77
4.3	Avaliação dos Detectores	80
4.3.1	Tempos de Detecção	80
4.3.2	Comparação Geral dos Detectores	81
4.3.3	Comparação Geral dos Detectores sem Personas	82
4.3.4	Comparação entre o Melhor e o Pior Detector	83
4.4	Limitações	84
5	CONCLUSÃO	86
5.1	Principais Contribuições	87
5.2	Trabalhos Futuros	87
	REFERÊNCIAS	88

APÊNDICES 94

APÊNDICE A	– DETECTORES DE TEXTOS ARTIFICIAIS: DETALHES DE IMPLEMENTAÇÃO	95
A.1	Binoculars	95
A.2	RADAR	96
A.3	RoBERTa Large Openai Detector	96
A.4	Sapling	97
A.5	DistilBERT	98
A.6	HuggingFace Detector	99

Introdução

A rápida evolução e popularização das ferramentas baseadas em Inteligência Artificial (IA) trouxeram diversos impactos. De um lado, tais tecnologias facilitam muitas atividades cotidianas, por outro lado, levantam questões relevantes sobre seu uso. As aplicações dessas ferramentas variam desde tarefas simples como assistentes de voz e mecanismos de busca como o Google e o Baidu, até funções avançadas. De acordo com o artigo (CARVALHO, 2021), a China pretende se tornar líder mundial de IA até 2030, investindo em pesquisas avançadas, como desenvolvimento de algoritmos para cidades inteligentes, carros autônomos, comércio eletrônico e *startups* de IA na área da saúde, mostrando o amplo potencial dessas tecnologias.

O uso da inteligência artificial está se expandindo cada vez mais, com ferramentas sendo constantemente aprimoradas a ponto de, em determinados contextos, imitarem de forma fiel características humanas, tornando-se extremamente valiosas para a simulação de interações entre pessoas. Essas interações podem buscar diversos objetivos, tais como pesquisas, promoção de produtos e serviços, ou até aumentar as chances de candidatos políticos. Diante disso, nada melhor para essa finalidade do que um ambiente altamente comunicativo, o qual reúne milhares de usuários diariamente, sendo também um dos maiores formadores de opinião, seja culturalmente, socialmente ou até mesmo politicamente, o X. Nos últimos anos, o X (antigo *Twitter*) tornou-se extremamente popular contando com uma fácil utilização, onde diversos temas são debatidos por meio de uma interface intuitiva, possibilitando o rápido entendimento de todo um contexto relacionado ao assunto discutido por meio das chamadas *threads*, sequências de postagens conectadas entre si.

Visto que a inteligência artificial é uma grande aliada na simulação de interações humanas, e as redes sociais são alvos extremamente atrativos para esse propósito devido ao seu impacto na sociedade, surge mais uma aplicabilidade dessa tecnologia: os Modelos de Linguagem de Grande Escala, capazes de gerar textos que replicam, de forma quase perfeita, tais interações. Um Modelo de Linguagem de Grande Escala (*Large Language Model*) (LLM) é um modelo de inteligência artificial baseado em redes neurais profundas,

desenvolvido para compreender e gerar linguagem natural. Esses modelos são treinados com grandes quantidades de informações, seja para conhecimento geral da linguagem ou para tarefas específicas. Tais modelos estão em constante evolução, aprendendo com seus próprios erros por meio de feedbacks sobre suas respostas (OUYANG et al., 2022).

O uso de modelos de linguagem para a produção de texto, especificamente para redes sociais, gera um paradigma complexo na distinção do que realmente é humano. Com tais ferramentas, é possível a automatização de praticamente qualquer ação dentro das plataformas, sendo algumas delas a produção de publicações sobre determinados temas com o objetivo de popularizar ou denegrir a imagem de algo ou alguém, criação de perfis automatizados para interagir com outros usuários e a coleta de informações. Com todas as possibilidades geradas por estes modelos, os mesmos podem ser usados para fins questionáveis, como o plágio de conteúdos, produção de material manipulado, disseminação de conteúdo de ódio, manipulação da opinião pública e produção de desinformação (VYKOPAL et al., 2024).

Com a similaridade apresentada em textos gerados pelos LLMs utilizando linguagem natural, surge a dificuldade em distinguir entre o que é real e o que é texto sintético. Nesse intuito, também se popularizaram os detectores de texto gerados por IA, os quais utilizam abordagens semelhantes aos modelos geradores de texto em seu treinamento, porém com o objetivo contrário, de detectar padrões textuais que possam ajudar na distinção da origem do conteúdo. O uso desses detectores já é amplamente utilizado para fins acadêmicos e comerciais, possuindo grande variedade de estudos que abordam as ferramentas, técnicas, vantagens e limitações.

Este trabalho investiga o comportamento de grandes modelos de linguagem na geração de comentários para a plataforma X, utilizando *posts* reais como base, bem como a eficácia de detectores na distinção entre esses comentários. Além disso, métricas linguísticas foram aplicadas aos comentários gerados e também aos comentários reais extraídos da plataforma, visando à análise de nuances presentes entre textos naturais e sintéticos. Também será analisado como a criação de personalidades para uma IA no processo de geração de texto pode alterar sua estrutura e sua influência na detecção

Como este trabalho está diretamente relacionado à automação de processos, foi desenvolvida uma aplicação responsável por executar cada etapa da geração, análise e comparação dos resultados, incluindo variações funcionais que ampliam seu escopo para uso por terceiros.

1.1 Motivação

Com a crescente integração de modelos de linguagem para geração de textos em redes sociais, tornou-se necessário analisar o quão avançados esses modelos estão na simulação de interações humanas, assim como análise da eficácia dos detectores presentes na literatura

para distinção destes textos.

A motivação do estudo, vem da compreensão do estado atual dessas tecnologias e a correlação das mesmas com textos reais, buscando analisar de forma comparativa a eficácia das ferramentas e de elementos textuais.

1.2 Problema

A crescente adoção de modelos de linguagem baseados em IA tem levantado questões relevantes acerca da autenticidade e confiabilidade das informações disseminadas em plataformas digitais, mais especificamente nas redes sociais devido à capacidade desses modelos em simular o uso da comunicação humana. A capacidade desses modelos de geração de texto utilizando linguagem humana torna cada vez mais difícil a distinção deste conteúdo.

Nesse contexto, surge o desafio de analisar a eficácia de modelos de linguagem, treinados ou não para geração de texto, assim como a dificuldade na detecção dos mesmos por meio de ferramentas especializadas. Tais ferramentas, apesar de já possuírem refinamentos precisos e ainda estarem em constante evolução, ainda enfrentam limitações de acordo com sua abordagem, seja na eficácia, ou na forma de uso para o usuário final de acordo com seu contexto de trabalho. Além disso, torna-se importante saber como a mudança de parâmetros de geração de texto influencia ou não na estrutura dos mesmos se comparados com textos supostamente humanos.

Esse desafio é especialmente relevante nas redes sociais, onde a confiança na origem de certo conteúdo é altamente questionável e pode ter implicações diretas para a disseminação da desinformação e manipulação de opiniões.

1.3 Hipótese

A Hipótese principal deste estudo é de que modelos de linguagem, como os LLMs, são capazes de gerar comentários para redes sociais, no caso o X, se utilizando de características próprias de textos destas plataformas conforme é referenciado na literatura, e que detectores também presentes na literatura, consigam ter uma boa eficácia na detecção destes textos.

Com base nesses questionamentos, foram elaboradas hipóteses que orientam a investigação.

- ❑ Os modelos de linguagem replicam de forma convincente padrões presentes em textos humanos;
- ❑ A utilização de personas na instrução de geração influencia a estrutura do texto gerado ou altera a eficácia da detecção posteriormente.

Outras hipóteses complementares foram formuladas, dentre elas:

- ❑ Há diferenças notáveis no desempenho entre as ferramentas de geração e detecção de acordo com o propósito inicial da ferramenta;
- ❑ Algumas ferramentas de detecção de textos podem apresentar limitações em relação à complexidade e ao seu uso;
- ❑ Textos gerados por IA e por humanos apresentam estruturas relativamente diferentes, principalmente devido à informalidade. É possível contornar isso por meio de instruções na geração.

1.4 Objetivos

O objetivo deste estudo é investigar a eficácia de grandes modelos de linguagem na geração de comentários para redes sociais, mais especificamente o X, e avaliar a capacidade de detectores de texto gerados por IA em identificar os mesmos.

Os seguintes objetivos específicos decorrem do objetivo principal:

- ❑ Desenvolver e avaliar um método para geração de comentários por meio de LLMs utilizando personas pré-definidas;
- ❑ Avaliar ferramentas de detecção de textos gerados artificialmente;
- ❑ Avaliar como características textuais impactam na capacidade do texto gerado ser detectado por uma ferramenta;
- ❑ Desenvolver uma aplicação que automatiza os processos de geração, detecção, análise de métricas e análise comparativa dos resultados.

1.5 Contribuições

Este estudo apresenta contribuições tanto práticas quanto teóricas, relacionadas à aplicação e análise de diversos modelos de linguagem em redes sociais como o X. As principais contribuições estão na avaliação da capacidade dos LLMs em simular interações humanas gerando textos com linguagem natural e na análise da eficácia de ferramentas de detecção de textos gerados por IA disponíveis na literatura.

Entre as demais contribuições deste trabalho, podem ser citadas:

- ❑ Desenvolvimento de uma aplicação capaz de automatizar o processo de geração, detecção e análise de resultados, a qual conta com funcionalidades modulares que podem ser utilizadas em outros contextos. A aplicação permite também a escalabilidade de novas ferramentas caso necessário;
- ❑ Avaliação da performance de diversos modelos de linguagem na produção de textos utilizando linguagem natural e análise da variação de parâmetros no resultado final;
- ❑ Comparação entre textos sintéticos e originais do X, permitindo a análise de diferenças e similaridades;
- ❑ Análise da efetividade de ferramentas de detecção acadêmicas e comerciais;
- ❑ Discussão sobre os impactos sociais do uso de modelos de linguagem.

Estas contribuições buscam enriquecer o debate acerca da linha tênue entre conteúdo humano e sintético em ambientes digitais, contribuindo com materiais importantes para trabalhos futuros.

1.6 Organização da Monografia

Este trabalho foi organizado seguindo a estrutura abaixo:

- Capítulo 2 – Fundamentação Teórica:** Aborda os principais contextos necessários para entendimento do trabalho, assim como um panorama geral sobre textos de redes sociais, modelos de linguagem, detectores e trabalhos correlatos que servirão de base para os experimentos realizados posteriormente.
- Capítulo 3 – Metodologia:** Demonstra o fluxo de trabalho utilizado para a pesquisa, incluindo os critérios de definição da rede social, dos temas, das ferramentas, *posts*, *personas*, bem como as métricas utilizadas.
- Capítulo 4 – Experimentos e Análise dos Resultados:** Demonstra os resultados e os *insights* obtidos a partir da análise dos textos gerados, da análise dos detectores e do uso das métricas, além de discutir as limitações do estudo.
- Capítulo 5 – Conclusão:** Aborda o resumo das contribuições do trabalho e apresenta as sugestões para trabalhos futuros que possam se apoiar no mesmo.
- Capítulo 6 – Apêndices:** Detalhes de implementação dos detectores.

Fundamentação Teórica

Neste capítulo, serão abordados os principais conceitos e fundamentos teóricos para sustentação deste trabalho, com ênfase em textos de redes sociais, modelos de linguagem de grande escala e detectores de textos gerados por IA. Através da revisão da literatura relevante, este capítulo busca proporcionar uma compreensão aprofundada dos elementos que influenciam a análise desses textos, além de explorar os desafios na geração e detecção dos mesmos.

2.1 Textos de Redes Sociais

Redes sociais são mecanismos de comunicação e interação onde o usuário não precisa necessariamente aderir às regras gramaticais. Geralmente, o propósito dos usuários é apenas transmitir uma mensagem, mesmo que essa não seja feita da maneira mais clara possível. Vários aspectos dos textos publicados em redes sociais reforçam a percepção de confusão ou aleatoriedade na forma como são elaborados, sendo que algumas dessas características são facilmente percebidas no uso cotidiano dessas plataformas.

O trabalho (BALDWIN et al., 2013) ilustra alguns desses aspectos, incluindo inconsistências nas pontuações, uso irregular de letras maiúsculas ou minúsculas, frases incompletas, trechos de difícil compreensão, lacunas lexicais incompletas, uso de *hashtags* e menções, além de variações no tamanho dos textos conforme Tabela 1.

As mídias sociais desempenham um papel fundamental na influência de diversos campos da sociedade, tais como política, formação de opinião pública, interesses culturais, comportamento social e até padrões de consumo (ARIAS, 2019). Através de plataformas como o X, a percepção sobre praticamente qualquer tema pode ser moldada a partir de textos direcionados, campanhas virais ou pela atuação de influenciadores digitais.

Um exemplo que ilustra bem essa questão, foi o caso da empresa Cambridge Analytica, conforme citado em (FORNASIER; BECK, 2020). Em março de 2018, a companhia foi denunciada pela coleta ilícita de dados pessoais e utilização de *microtargeting*, uma estratégia de marketing que segmenta mensagens personalizadas para públicos específicos

Fonte	Compr. Médio das Palavras ($\pm$)	Média da Quant. de Palavras ($\pm$)
TWITTER-1	3.8 ± 2.4	9.2 ± 6.4
TWITTER-2	3.8 ± 2.4	9.0 ± 6.3
COMMENTS	3.9 ± 3.2	10.5 ± 10.1
FORUMS	3.8 ± 2.3	14.2 ± 12.7
BLOGS	4.1 ± 2.8	18.5 ± 24.8
WIKIPEDIA	4.5 ± 2.8	21.9 ± 16.2
BNC	4.3 ± 2.8	19.8 ± 14.5

Tabela 1 – Tabela adaptada de (BALDWIN et al., 2013), que apresenta os padrões de comprimento de textos em redes sociais.

com base em características de indivíduos. Esse fato ocorreu durante as eleições para presidente no ano de 2016, onde a empresa acessou informações de milhões de usuários do *Facebook* sem consentimento. A Cambridge Analytica usava essas informações para elaborar perfis minuciosos de eleitores e direcionar publicidade personalizada, visando influenciar decisões de voto e manipular visões sobre temas políticos e sociais.

Outro caso recente, que demonstra esse uso de perfis personalizados, é o trabalho (HANLEY; DURUMERIC, 2024), que emprega o modelo deBERTa (HE et al., 2021), um modelo avançado de Processamento de Linguagem Natural (*Natural Language Processing*) (NLP) (CHOPRA; PRASHAR; SAIN, 2013), o qual classificou mais de 15,56 milhões de artigos de 3074 sites de notícias. Observou-se que, entre 2 de janeiro de 2022 e 1 de maio de 2023, a quantidade de artigos produzidos por IA em sites confiáveis teve um crescimento de 57,3%, enquanto em sites não confiáveis, o aumento foi de impressionantes 474%, muito por conta do lançamento do ChatGPT (WELSBY; CHEUNG, 2023).

Dessa forma, nota-se como características únicas dos textos de redes sociais, muitas vezes caóticas ou pouco claras, podem ser acentuadas pela atuação de perfis automatizados, os quais utilizam grandes modelos de linguagem para gerar textos, tornando ainda mais desafiadora a análise da veracidade dessas informações.

2.2 Modelos de Linguagem de Grande Escala

Os LLMs são modelos de aprendizado de máquina treinados para compreender e gerar textos que imitem como o máximo de precisão as interações humanas utilizando linguagem natural (CHOPRA; PRASHAR; SAIN, 2013), ou seja, a maneira que os humanos nos comunicamos. Os textos gerados por grandes modelos de linguagem diferem em diversos aspectos dos textos humanos. Textos gerados por humanos são caracterizados por uso de gírias, abreviações, textos incoerentes e de difícil complexidade, algo que geralmente

não ocorre em textos automatizados que passam por instruções específicas da linguagem. Textos sintéticos apresentam uma semântica e estrutura mais elaboradas, além de fornecerem respostas extensas e detalhadas. Diferentemente dos textos humanos, eles não expressam sentimentos, uma característica marcante da produção textual humana.

Atualmente, esses modelos desempenham papéis cruciais em diversas áreas do cotidiano, tais como assistentes virtuais, tradutores ou na geração de conteúdos para redes sociais.

2.2.1 História e Evolução

Os modelos de linguagem de grande escala não surgiram em sua forma atual, e sim, são fruto de uma evolução gradual por meio de fases, conforme descrito em (MINAEE et al., 2024).

Na primeira fase, os modelos surgiram de forma estática, os quais tratavam os textos como sequências de palavras, e assim, tentavam prever o termo seguinte com base nas anteriores usando contagens de combinação de palavras.

Na segunda fase, surgiram as primeiras *Redes Neurais* (KROGH, 2008), modelos computacionais que se inspiravam no cérebro humano, possuindo a capacidade de compreender e aprender padrões a partir de grandes volumes de dados. Esses modelos utilizam vetores de incorporação, técnica que mapeia as palavras em representações numéricas em um espaço vetorial. Isso permitia que as redes neurais compreendessem relações semânticas entre as palavras, posicionando termos que tinham significados semelhantes de maneira mais próxima no espaço.

A terceira fase trouxe os modelos pré treinados, que representaram um avanço significativo aos modelos de linguagem neural utilizados até então. Diferentemente de seus antecessores, estes modelos não eram treinados inicialmente para uma tarefa específica, mas sim, treinados com grandes conjuntos de dados de diversos segmentos para aprenderem representações gerais da linguagem, como o idioma ou a semântica. Apenas após esse pré treinamento eles passam por um processo chamado *fine-tuning* ou ajuste fino (HOWARD; RUDER, 2018), o qual consiste no ajuste do modelo para tarefas mais específicas utilizando dados personalizados para determinadas tarefas.

Por fim, a última fase deste ciclo são os LLMs, modelos neurais que possuem de dezenas a centenas de bilhões de parâmetros, os quais são ajustados para aprender padrões complexos. Esta última fase trouxe consigo modelos muito mais robustos e com habilidades de compreensão e geração de dados de forma mais eficiente, assemelhando-se muito com a linguagem humana.

Além dessas mudanças, o trabalho (DUAN et al., 2024) traz alguns aspectos que tornaram os LLMs um avanço tão significativo em relação aos modelos anteriores, sendo estes:

❑ **Arquitetura de Modelo Homogêneo**

Anteriormente, o aprendizado de máquina utilizava diversas arquiteturas de modelos, já os LLMs utilizam predominantemente a arquitetura *Transformers* (DUAN et al., 2024) o que revolucionou o modo de trabalho otimizando significativamente seu desempenho.

❑ **Escala e Duração do Treinamento**

Os LLMs necessitam de treinamentos de grande escala com milhares de dados e parâmetros divididos em *clusters* de processamento. O treinamento pode durar dias, ou até meses de acordo com a infraestrutura utilizada.

❑ **Otimização de Software Especializado**

Com o tamanho destes modelos de linguagem, surgiram desafios no treinamento e execução dos mesmos. Para isso, algumas técnicas são utilizadas, tais como o paralelismo híbrido, que distribui o treinamento por vários *clusters*, combinando paralelismo de dados para acelerar o treinamento de grandes modelos. Uma ferramenta que ajuda nisso é o *Megatron*, que implementa paralelismo de modelo para lidar com bilhões de parâmetros de forma eficiente (SHOEYBI et al., 2020). Outra técnica utilizada, é o *DeepSpeed*, que otimiza a memória e acelera o treinamento ao fragmentar o modelo, reduzindo o consumo de memória.

❑ **Mudança no Paradigma de Treinamento**

Anteriormente, os modelos de *Deep Learning* eram treinados para tarefas específicas com os dados direcionados a essas tarefas. Os LLMs por sua vez, possuem treinamento mais extensivo, sendo treinados com grandes conjuntos de dados de maneira mais geral, criando assim modelos básicos, os quais posteriormente podem ser refinados para tarefas específicas utilizando técnicas como o *fine-tuning* (HOWARD; RUDER, 2018).

2.2.2 Características Gerais dos LLMs

Na literatura, características de grandes modelos de linguagem são discutidas em diversos trabalhos (BROWN et al., 2020; VASWANI et al., 2023; DEVLIN et al., 2019; CHANG et al., 2023), sendo algumas delas descritas na Tabela 2.

Característica	Descrição	Exemplos	Impacto e Aplicações
Escalabilidade	Lida com grandes volumes de dados e milhões de parâmetros, melhorando o desempenho.	GPT-3	Melhor eficiência e aprendizado contínuo.
Transferência de Aprendizado	Modelos pré-treinados podem ser adaptados a novas tarefas com pouco re-treinamento.	BERT, RoBERTa	Reduz tempo e recursos para treinar novos modelos.
Capacidade de Generalização	Aplicação do conhecimento para resolver novas tarefas sem exemplos diretos.	GPT-3 (few-shot learning)	Eficiência em múltiplas tarefas sem necessidade de ajuste fino.
Compreensão de Contexto	Interpreta texto considerando relações entre palavras e frases.	Transformer, BERT	Melhor entendimento de textos complexos.
Análise de Textos Longos	Identifica relações de longo alcance entre palavras em textos extensos.	Transformers (atenção)	Melhor desempenho em textos acadêmicos e técnicos.
Interpretação e Explicabilidade	Desafios para entender como os modelos tomam decisões e geram respostas.	Modelos Transformer	Necessidade de maior transparência, especialmente em saúde e direito.
Geração de Texto Coerente	Modelos criam textos com estrutura lógica e continuidade.	GPT-3, T5, BERT	Uso em chatbots, redação assistida e assistentes virtuais.

Tabela 2 – Tabela gerada a partir de dados referenciados em (BROWN et al., 2020; VASWANI et al., 2023; DEVLIN et al., 2019; CHANG et al., 2023) sobre características de LLMs.

2.2.3 Treinamento

Os modelos de linguagem de grande escala, são sistemas que, assim como os humanos, precisam de treinamento contínuo para realização e aprimoramento de suas tarefas. Nos últimos anos, com o crescimento exponencial da escalabilidade desses modelos, o processo de treinamento tornou-se cada vez mais complexo e desafiador, envolvendo uma série de técnicas, as quais exigem inovações tanto em infraestrutura, quanto em relação a otimização.

A abordagem utilizada para o treinamento depende das necessidades e recursos dis-

poníveis. O treinamento pode ser realizado de diversas formas, incluindo:

- ❑ **Aprendizado Supervisionado:** O modelo é treinado usando dados rotulados, ou seja, dados nos quais a resposta correta é conhecida.
- ❑ **Aprendizado Auto Supervisionado:** O modelo cria seus próprios rótulos a partir dos dados, sem receber respostas corretas.
- ❑ **Aprendizagem por Reforço:** O modelo aprende com base em retornos sobre a qualidade de suas respostas, sendo capaz de ajustar seu comportamento conforme as recompensas ou punições recebidas.

Essas técnicas podem ser combinadas conforme necessário, otimizando o desempenho do modelo em tarefas específicas. Além das técnicas de treinamento, existem também estratégias de aprimoramento que visam ajustar o modelo já treinado. Algumas dessas estratégias incluem:

- ❑ **Otimização de Memória:** Devido ao alto consumo de recursos na preparação dos modelos, técnicas como a *quantização* e o *checkpointing* são necessárias. A quantização, reduz a precisão dos dados e parâmetros, já o *checkpointing* armazena o estado do modelo em certos pontos para poder retomar de onde parou, minimizando o uso de memória e recálculo dos mesmos parâmetros que já foram realizados anteriormente. Essas técnicas são usadas para reduzir o uso de memória sem perder precisão.
- ❑ **Aceleração de IA:** Após o pré-treinamento de um modelo com um grande conjunto de dados, realiza-se seu ajuste para tarefas mais específicas. Isso pode envolver técnicas como a destilação, onde um modelo maior e mais complexo ensina um modelo menor, além da quantização e da otimização da arquitetura, com o objetivo de melhorar o desempenho e reduzir o tempo de execução em dispositivos com recursos limitados.
- ❑ **Transferência de Aprendizagem:** Essa técnica adapta um modelo pré-treinado de forma geral para novas tarefas, aproveitando o conhecimento anterior para melhorar o desempenho em uma nova tarefa. O modelo pode ser utilizado com ajustes menores ou até mesmo sem alterações significativas dependendo da tarefa em questão.

2.2.3.1 Paralelismo

Uma das principais técnicas de treinamento para LLMs é o Paralelismo, que fragmenta a carga computacional em várias unidades de processamento distribuídas, buscando uma maior eficiência no processamento.

Conforme exemplificado em (DUAN et al., 2024), existem três tipos de paralelismo no treinamento de LLMs:

- ❑ **Paralelismo de Dados:** Divide o conjunto de dados em várias GPUs, fazendo com que cada uma trabalhe com um subconjunto de dados.
- ❑ **Paralelismo de Modelo:** Divide o próprio modelo em GPUs diferentes, sendo utilizada quando o modelo é muito grande para ser carregado de uma vez.
- ❑ **Paralelismo de Pipeline:** Distribui o treinamento de forma sequencial entre GPUs, onde cada uma processa uma etapa e passa a informação para a seguinte.

2.2.3.2 Pré-Treinamento

Outra estratégia de treinamento de modelos de IA é o pré-treinamento. Modelos como o BERT (DEVLIN et al., 2019), um dos primeiros modelos pré-treinados para tarefas de NLP (CHOPRA; PRASHAR; SAIN, 2013) e o RoBERTa (LIU et al., 2019), um aperfeiçoamento do BERT que se destaca na detecção de textos sintéticos, passam por um pré-treino extenso, no qual são utilizados grandes volumes de dados de treinamento antes que estes passem por um processo de modelagem para tarefas específicas. Esse processo inicial, proporciona ao modelo um aprendizado geral de diversos temas e contextos, os quais facilitam o refinamento sem a necessidade de um treinamento feito do zero.

Além disso, apesar de um refinamento ser útil para atividades específicas, o uso destes modelos pode ser feito mesmo com o pré-treinamento inicial, pois com a quantidade de dados de aprendizado, o modelo já consegue ter bons resultados em tarefas básicas como é mostrado no artigo (DEVLIN et al., 2019).

2.2.3.3 Data Augmentation

O aumento de dados ou *Data Augmentation* (WANG et al., 2024b) é outra técnica fundamental para melhorar o desempenho dos modelos de aprendizado de máquina. Essa abordagem aumenta o volume de dados de treinamento, gerando variações dos dados originais. Essa técnica é extremamente útil quando não há dados suficientes disponíveis ou os dados são desequilibrados. Existem quatro tipos comuns para o aumento de dados, sendo elas, o aumento para imagens, para áudio, para dados tabulares, e para texto, o qual é foco deste trabalho.

No aumento de texto, busca-se ampliar de forma artificial os conjuntos de treinamento, introduzindo maior diversidade e reduzindo o desequilíbrio nos dados. Dentre algumas abordagens nesse método está o *Back-Translation* que faz a tradução do texto para outro idioma e depois realiza a retradução para o idioma original, permitindo variações sintáticas sem comprometer o entendimento, conforme a Figura 1.

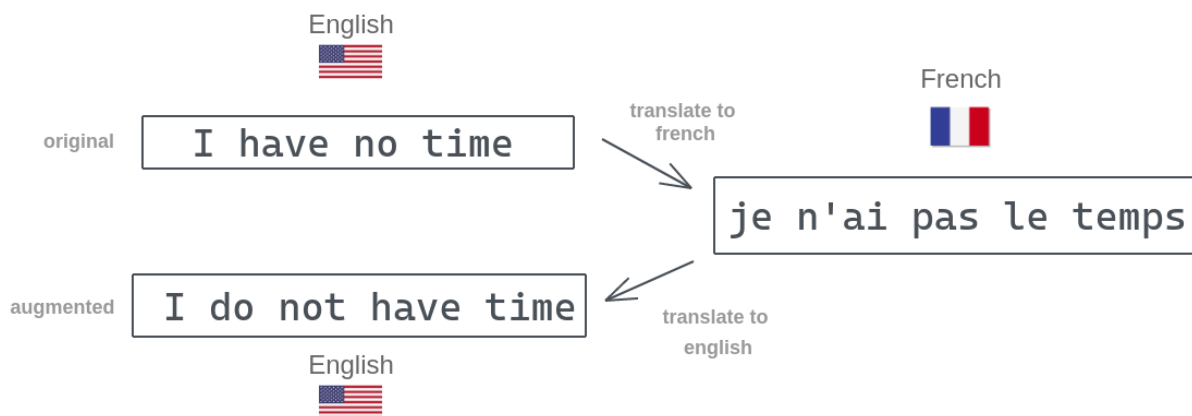


Figura 1 – Exemplificação do processo de *Back-Translation* de uma frase. Figura retirada de (CHAUDHARY, 2020).

Além dela, existe também a *Substituição de Sinônimos*, a qual realiza a substituição de textos sinônimos, visando preservar a semântica original e introduzir variações lexicais, ou seja, diferentes maneiras de expressar uma ideia, porém utilizando palavras diferentes. Por fim, a técnica de *Permutação de Palavras* reorganiza a estrutura da frase sem prejudicar sua compreensão, sendo muito útil para expor o modelo a diferentes formações sintáticas.

A utilização do aumento de dados traz solução para problemas como o *overfitting* que ocorre quando um modelo de aprendizado de máquina se acostuma com padrões específicos dos dados de treinamento ao invés de aprender padrões gerais. Ou seja, o modelo se sai muito bem com os dados de treino, porém quando submetido a novos exemplos, tem sua eficiência comprometida. A solução para esse problema é a criação de novas variações a partir dos dados já existentes, forçando o modelo a aprender padrões gerais ao invés de apenas memorizar exemplos de treinamento.

Outro problema que essa estratégia busca resolver, é o desbalanceamento dos dados. O método atua gerando exemplos adicionais para as classes minoritárias, que possuem menos representações no conjunto de dados. Dessa forma, equilibra-se a quantidade de exemplos entre as diferentes classes, o que contribui para uma aprendizagem mais eficiente do modelo.

O uso do aumento de dados atua como suporte para diversos problemas, porém também tem suas limitações, tais como a dependência da qualidade dos dados, ou seja, se os dados originais forem ruins, o aumento deles pode também não ser de alta qualidade. Outra limitação pode ser seu gerenciamento, já que, quanto mais se aumenta os dados, mais probabilidade eles têm de perder sua variabilidade, tornando o modelo menos robusto. Por fim, existe a limitação de custo computacional, já que algumas dessas técnicas podem exigir recursos significativos, gerando alto valor de demanda.

2.2.4 Arquitetura Transformers

Um dos principais avanços na criação de modelos de linguagem de grande escala, foi a aplicação da arquitetura *Transformers*, introduzida em (VASWANI et al., 2023). Essa arquitetura transformou a maneira como os modelos de NLP processam e analisam dados, substituindo métodos convencionais que se baseavam em Redes Neurais Recorrentes (*Recurrent Neural Networks*) (RNN) e Redes Neurais Convolucionais (*Convolutional Neural Networks*) (CNN).

Diferentemente de modelos anteriores, o *Transformers* não analisa a entrada palavra por palavra, mas possibilita que o modelo analise a sentença inteira de uma só vez. Isso ocorre devido ao *mecanismo de atenção*, que permite ao modelo reconhecer os elementos mais relevantes de uma entrada para entender seu contexto. O uso desse mecanismo de atenção, conforme descrito por (LUO et al., 2023), possibilita ao modelo, considerar simultaneamente diversas partes da entrada ao analisar uma palavra, assegurando uma análise mais contextualizada.

A geração de texto nos modelos que utilizam a arquitetura Transformers, ocorre de maneira sequencial, prevendo o próximo *token* com base no contexto anterior. Esses tokens podem ser palavras, partes de palavras ou até mesmo espaços em branco. O mecanismo de atenção permite a consideração de todo o contexto favorecendo a coerência e a fluidez na criação do texto (WU et al., 2024).

O *Transformer* também introduziu a atenção multi-cabeça. Ao invés de usar apenas uma cabeça para examinar as conexões entre as partes da entrada, o modelo usa várias cabeças de análise, possibilitando uma avaliação simultânea de diversos elementos do contexto, aprimorando a habilidade de identificação de dependências e sutilezas ao longo do texto.

2.2.5 Aplicações dos LLMs

Os LLMs não estão restritos apenas a grandes corporações ou a um grupo específico de pessoas, eles estão presentes em diversas áreas do cotidiano. Para ilustrar melhor seu impacto, abaixo estão algumas das aplicações dessas tecnologias.

- ❑ **Geração de Conteúdo:** Os LLMs são amplamente usados no auxílio à escrita, seja redigindo um e-mail sobre um assunto desconhecido, corrigindo erros gramaticais ou reformulando textos de forma mais clara e objetiva.
- ❑ **Tradução de Textos:** Ferramentas de tradução, como o Google Translate, utilizam modelos de inteligência artificial como o PaLM 2 da Google para aprimorar traduções, tornando-as mais precisas e naturais. O PaLM 2 expande o suporte a dialetos e variações linguísticas, gerando traduções mais fluidas e mantendo o contexto.

Outras aplicações às quais os LLMs têm se mostrado amplamente utilizados são:

- ❑ **Assistentes Virtuais:** Assistentes inteligentes facilitam diversas tarefas cotidianas, como realizar chamadas, encontrar músicas e até monitorar a saúde do usuário. Um exemplo disso é o PrISM-Q&A (ARAKAWA; LEHMAN; GOEL, 2024), um assistente desenvolvido para *smartwatches*, que ao integrar reconhecimento de atividades humanas, fornece respostas contextuais em tempo real durante a realização de tarefas físicas melhorando a experiência do usuário.
- ❑ **Detecção de Textos Gerados por IA:** Ferramentas como o Binoculars detectam textos gerados por inteligência artificial por meio do método *zero-shot*, que compara as probabilidades atribuídas por modelos semelhantes, identificando diferenças sutis a fim de distinguir entre textos humanos e sintéticos

2.3 Detectores de Textos Gerados por IA

Com o crescente avanço nas técnicas de criação, treinamento e refinamento de grandes modelos de linguagem, torna-se cada vez mais difícil distinguir textos gerados por humanos e por máquinas. Embora os LLMs sejam amplamente utilizados para tarefas cotidianas, como auxílio de clareza textual ou correção de erros em códigos, essas ferramentas também podem ser usadas para fins problemáticos.

Um destes usos é a utilização dos LLMs no meio acadêmico. O estudo realizado em (PUDASAINI et al., 2024) mostra os efeitos disso em universidades, onde a integridade acadêmica é questionada devido às facilidades proporcionadas por essas tecnologias, seja na criação de respostas ou na reescrita de conteúdos já existentes.

Trazendo o contexto das redes sociais, a manipulação de informações também pode ser facilmente realizada por meio de instruções fornecidas a esses modelos. A geração de texto tendencioso, apelativo ou até incorreto, torna-os suscetíveis a disseminação de *fake-news* em redes sociais (PAPAGEORGIOU et al., 2024), impactando esferas culturais, políticas e sociais.

Diante desses desafios, surgiram detectores de textos gerados por IA, cujo objetivo é identificar características específicas em textos, as quais possam diferir entre a maneira que um humano e uma máquina escrevem.

Nas subseções a seguir, serão exploradas as principais características, desafios, aplicações e utilidades desses detectores, bem como seus impactos e limitações.

2.3.1 Conjuntos de dados

Assim como os LLMs que são utilizados para a geração de texto, os quais são treinados com grandes volumes de dados, os modelos de detecção também passam por um processo

semelhante, porém utilizando dados direcionados para a tarefa de detecção. Esses conjuntos de treino possuem textos gerados por humanos, por máquinas, dados relacionados à desinformação, ou qualquer outro contexto que possa ser útil para a distinção do conteúdo. A Tabela 3 apresenta algumas bases de dados utilizadas com esse intuito.

Conjunto de Dados	Uso/Tipo	Tipo de Dados	LLMs
TuringBench (UCHENDU et al., 2021)	Treino, Teste	Textos de humanos e máquinas	GPT-3, GPT-2, GROVER, CTRL, XLM
HC3 (GUO et al., 2023)	Treino, Teste	Texto da web, QA, Mídias sociais	BERT, T5
CHEAT (YU et al., 2024)	Treino, Teste	Escrita científica	GPT-3, T5
Ghostbuster (VERMA et al., 2024)	Treino, Teste	Redações de estudantes, ficção criativa e notícias	GPT-3
GPT-Sentinel (CHEN et al., 2023a)	Treino, Teste	OpenWebText	GPT-2, GPT-3
M4 (WANG et al., 2024a)	Treino, Teste	Multidomínios	GPT-3, T5
MGTBench (HE et al., 2024)	Treino, Teste	Conjuntos de dados de perguntas e respostas	BERT, GPT-3
HC3 Plus (SU et al., 2024)	Treino, Teste	Escrita de notícias, Mídias sociais	GPT-3
MULTITuDE (MACKO et al., 2023)	Treino, Teste	MassiveSumm	T5, GPT-3
HumanEval (CHEN et al., 2021)	Teste	Exercício de código	Codex
OpenLLMText (CHEN et al., 2023b)	Treino, Teste	Texto da web	GPT-2, GPT-3.5, PaLM, LLaMA
GROVER Dataset (ZELLERS et al., 2020)	Treino, Teste	Escrita de notícias	GROVER
TweepFake (FAGNI et al., 2021)	Treino, Teste	Mídias sociais	GPT-3, BERT
GPT-2 Output Dataset6 (OPENAI, 2019)	Treino, Teste	Texto da web	GPT-2
ArguGPT (LIU et al., 2023)	Treino, Teste	Escrita científica	GPT-3, T5
MAGE (LI et al., 2024)	Treino, Teste	Mídias sociais, Escrita de notícias, QA, Geração de histórias, Compreensão e Raciocínio	GPT-3, BERT, T5

Tabela 3 – Tabela adaptada de (WU et al., 2024) e (YANG et al., 2023) sobre datasets de treinamento de detectores de texto gerados por IA.

A diversificação dos conjuntos de dados é fundamental para assegurar o desempenho eficaz do modelo de detecção. Um maior número de exemplos, contextos e fontes de dados, contribui para uma melhor capacidade de entendimento do detector, uma vez que os LLMs de geração de texto são aprimorados com as informações mais relevantes disponíveis, visando contornar essa detecção. No entanto, a elaboração de conjuntos de dados abrangentes apresenta um desafio significativo. Conforme mencionado em (WU et al., 2024), em determinadas situações, os pesquisadores utilizam dados relacionados a perguntas e respostas ou redações acadêmicas para treinar os modelos com textos produzidos por humanos, e, em seguida, geram textos com LLMs para o treinamento dos detectores. Isso é realizado por meio de métodos como engenharia direta, que consiste na criação ou modificação de características nos dados de treinamento, incluindo a geração de textos sintéticos ou ajustes estruturais, além da complementação com *bootstrap*, uma técnica que amplia a quantidade de dados ao criar múltiplos subconjuntos de treinamento a partir de amostras originais. Esses métodos têm como objetivo aprimorar a diversidade dos dados.

2.3.2 Métodos de detecção

Como demonstrado em (YANG et al., 2023), diversas táticas vêm sendo desenvolvidas para identificar textos produzidos por inteligência artificial. O processo de identificação desses textos pode variar dependendo do tipo de acesso que o detector possui ao modelo que gerou o texto. Nesse sentido, é necessário compreender dois conceitos: caixa branca e caixa preta. Essas duas estratégias se diferem na quantidade de informações disponíveis sobre o modelo que gerou o texto.

A detecção de *Caixa Preta* refere-se a situação onde o detector tem informações limitadas ou inexistentes sobre o modelo gerador do texto. Neste contexto, a meta é identificar características no texto que possam indicar sua origem artificial, sem ter acesso aos parâmetros internos do modelo. A identificação de caixa preta pode ser dividida em dois contextos:

- ❑ **Modelo Conhecido:** Neste caso, o modelo gerador é conhecido, mas o detector não tem acesso aos seus parâmetros internos. A detecção é feita a partir das propriedades superficiais do texto, como estrutura linguística e coerência semântica.
- ❑ **Modelo Desconhecido:** Nesse cenário, o detector não sabe qual modelo gerou o texto, o que torna a detecção mais desafiadora. O detector precisa identificar padrões gerais e características comuns a diversos modelos, baseando-se em aspectos estilísticos e estruturais.

Por outro lado, a detecção de *Caixa Branca* envolve uma situação onde o detector possui o acesso aos dados internos do modelo que produziu o texto. Esse nível de acesso

possibilita o uso de métodos mais precisos e eficazes, pois o detector tem a capacidade de investigar as propriedades e parâmetros do modelo. A detecção de caixa branca também pode ser dividida em duas situações com base no nível de acesso, sendo elas:

- ❑ **Acesso Completo aos Parâmetros:** Aqui o detector possui total acesso aos parâmetros do modelo, podendo acessar pesos, arquiteturas e outros elementos internos do modelo de geração. Assim, é possível aplicar técnicas avançadas como a marca d'água, que consiste na adição de atributos ocultos ao texto produzido, possibilitando sua identificação futuramente pelos modelos de detecção.
- ❑ **Acesso Parcial aos Parâmetros:** Neste caso o detector possui informações restritas sobre o modelo, tais como logs de saídas ou variáveis intermediárias. As técnicas de detecção são aplicadas com base nessas informações. Apesar da detecção não ser tão detalhada quanto no acesso completo, ainda é possível reconhecer certos comportamentos dos modelos e suas particularidades.

Outro conceito importante é o *Pré-Treinamento*, que tem papel crucial em várias dessas estratégias. O pré-treinamento envolve o processo de treinar um modelo usando grandes volumes de dados que podem ou não serem específicos para alguma tarefa, com a finalidade de proporcionar ao modelo uma compreensão completa da linguagem. Para detectores de texto gerado por IA como o BERT (DEVLIN et al., 2019), e sua versão aprimorada, o RoBERTa (LIU et al., 2019), o pré-treinamento permite com que o modelo adquira um entendimento profundo de diversos temas, contextos e padrões linguísticos, facilitando a diferenciação de textos humanos e sintéticos.

Apesar de todas as táticas de detecção de textos gerados por IA, de alguma forma se beneficiarem de algum tipo de treinamento prévio, as metodologias e propósitos desse treinamento podem variar bastante. Alguns modelos como o BERT, requerem um ajuste fino após o pré-treinamento para uso em tarefas específicas, como a identificação de texto gerado por LLMs. No entanto, alguns podem ser eficientes sem a necessidade de ajuste, apenas com o pré-treinamento, devido ao volume de dados gerais utilizados na fase inicial de treinamento. Conforme evidenciado em (DEVLIN et al., 2019), a simples utilização de modelos pré-treinados pode ser suficiente para obter resultados satisfatórios.

Entendidos esses conceitos, a seguir serão exploradas técnicas para detecção de textos sintéticos.

2.3.2.1 Classificadores Baseados em Marca d'água (*Watermarking*)

A técnica de marca d'água, inicialmente proposta em 1997 e posteriormente reformulada pelos mesmos autores no livro (COX et al., 2007), é uma tecnologia de prevenção contra o uso indevido de dados. Originalmente utilizada em imagens, vídeos e documentos digitais, seu objetivo é controlar o uso indevido ou replicação de conteúdos. Com o

tempo, essa técnica não se limitou apenas a essas mídias e passou a ser aplicada também no controle de conteúdos textuais. Atualmente, técnicas de marca d'água são empregadas antes, durante e depois da geração dos textos, possibilitando sua detecção posteriormente. O artigo (WU et al., 2024) demonstra algumas dessas técnicas e como os detectores se aproveitam das mesmas.

- **Marca D'água Baseada em Dados.** Essa técnica insere padrões ocultos nos próprios dados usados para treinar o modelo. Esses padrões, invisíveis ao olho humano, são conhecidos apenas por quem os inseriu. Entre as estratégias utilizadas estão padrões semânticos (como o uso de construções gramaticais específicas no final das sentenças), alterações na frequência de certas palavras e até sequências de entrada que disparam respostas específicas (gatilhos). Esses padrões permitem que posteriormente, detectores possam identificar textos gerados artificialmente por meio de análises estatísticas, probabilísticas ou baseadas em gatilhos.

- **Marca D'água Baseada em Modelo.** Diferentemente da abordagem baseada em dados, essa técnica modifica diretamente o funcionamento interno do modelo durante o treinamento. O foco está em ajustar como o modelo gera suas saídas, incorporando padrões sutis de geração. Isso pode incluir o uso de estruturas linguísticas específicas ou manipulação estatística para favorecer certas palavras ou estilos. Ainda podem ser usados gatilhos, mas com ênfase maior na modelagem do comportamento do modelo em si, em vez dos dados fornecidos a ele. A marca d'água torna-se assim parte do 'modo de pensar' do modelo, facilitando a detecção posterior das saídas geradas.

- **Marca D'água Pós-Processamento.** Por fim, o método de pós-processamento, modifica o texto após sua criação. Após gerar o texto, procedimentos como a adição de caracteres Unicode ocultos, a troca de sinônimos e a mudança de estrutura são realizadas. Com isso, os detectores conseguem por meio de elementos sutis, identificar essas características inseridas posteriormente nos textos.

2.3.2.2 Métodos Baseados em Redes Neurais

Os métodos baseados em redes neurais têm se mostrado uma abordagem eficaz e amplamente utilizada na detecção de textos gerados por IA, conforme apresentado em (KROGH, 2008). Essas redes, inspiradas no funcionamento do cérebro humano, são compostas por camadas: a camada de entrada, que recebe os dados brutos, as camadas ocultas, que processam essas informações aplicando pesos e funções matemáticas e a camada de saída, que gera o resultado final, como uma previsão ou classificação.

Classificadores baseados em redes neurais buscam identificar padrões em textos para distinguir conteúdos gerados por inteligência artificial. Como citado em (WU et al., 2024), esses classificadores podem explorar diferentes tipos de informações, como recursos linguísticos ou características internas do modelo.

Classificadores que utilizam recursos linguísticos extraem características estatísticas e semânticas dos textos, empregando métodos de aprendizado de máquina para distinguir a fonte de geração. Apesar de apresentarem bons resultados em algumas situações, enfrentam limitações, como a ambiguidade linguística e dificuldades na detecção de textos produzidos para fins de desinformação, que podem ser sutilmente disfarçados.

No método de especificação do modelo, avaliam-se propriedades internas dos modelos de geração de texto, como perplexidade e log-probabilidade. Esses procedimentos buscam capturar particularidades da estrutura de geração de IA, reconhecendo padrões recorrentes em textos sintéticos que não são tão comuns em textos humanos. Essa estratégia tem se mostrado especialmente eficaz na detecção de inconsistências e características atípicas.

Como em quaisquer outras abordagens, há o risco de textos humanos serem confundidos com textos de IA e vice-versa, sobretudo porque os modelos de linguagem estão cada vez mais refinados, tornando sua distinção uma tarefa cada vez mais desafiadora. Ainda assim, modelos baseados em redes neurais continuam sendo fundamentais, inclusive para outras abordagens de detecção, contribuindo significativamente para o aprimoramento da capacidade e da eficácia dos sistemas de identificação.

2.3.2.3 Classificadores Baseados em Estatísticas

A análise estatística é outro método comum para detectar textos sintéticos utilizando a distribuição de palavras, frequência de termos e padrões linguísticos. Modelos baseados em n-gramas, como os de (AHMED; TRAORÉ; SAAD, 2017), e ferramentas como o Giant Language Model Test Room (GLTR) (GEHRMANN; STROBELT; RUSH, 2019), que comparam padrões textuais com os de modelos de linguagem, têm se mostrado eficazes. O GLTR por exemplo, aumentou a taxa de acerto de 54% para 72% sem ajustes no modelo. Algumas abordagens de detecção estatísticas são discutidas em (WU et al., 2024), sendo elas:

- ❑ **Estatísticas de Caixa Branca.** Avaliam as características internas do modelo gerador, tais como probabilidade atribuída a cada palavra gerada e sua coerência.
- ❑ **Estatísticas de Caixa Preta.** Avalia o texto de forma externa, sem conhecer o funcionamento interno do modelo gerador, examinando padrões de repetição, previsibilidade e fluidez do conteúdo.

Além dessas, há também a abordagem focada nos recursos linguísticos, a qual examina padrões como a distribuição de palavras, o número de palavras por frase, a estrutura sintática e semântica, procurando diferenças entre textos humanos e sintéticos.

Essas técnicas podem utilizar a técnica *Burstiness*, que avalia a diferença na distribuição de palavras raras e comuns. Textos humanos costumam ter uma distribuição mais

variada das palavras, enquanto textos sintéticos tendem a manter uma distribuição mais uniforme.

Outro método frequentemente empregado nesse contexto estatístico é a *Medição de Perplexidade*, que mede o grau de previsibilidade de um texto. Modelos de IA costumam produzir sequências mais previsíveis que textos humanos, resultando em menos perplexidade. Quanto menor a perplexidade, mais simples se torna a previsão da próxima palavra, indicando a possibilidade daquele texto ter sido gerado por inteligência artificial.

Os detectores estatísticos também têm a capacidade de operar no modelo *zero-shot* (PU et al., 2023), sem treinamento específico para cada modelo gerador. Esta metodologia possibilita que o sistema analise padrões gerais da linguagem e reconheça textos originados de modelos não conhecidos. Assim, detectores treinados em modelos mais básicos são capazes de identificar textos produzidos por modelos mais sofisticados sem a necessidade de retreinamento, o que torna essa tática eficiente até mesmo em cenários com limitações computacionais.

2.3.2.4 LLMs como Detectores

Outra estratégia interessante de detecção de textos gerados por LLMs é a utilização dos próprios modelos de grande escala para essa análise. Em (ZELLERS et al., 2020), é proposto o modelo GROVER, o qual pode criar artigos completos a partir de um título específico. Este modelo foi criado com uma natureza controlável para a geração de texto, o que o tornava um alvo fácil para a produção de desinformação, conforme ilustrado na Figura 2, já que os humanos apresentam dificuldade em distinguir a veracidade de *fake-news* sem uma pesquisa mais aprofundada.



Figura 2 – Exemplificação do funcionamento do modelo Grover. Retirado de (ZELLERS et al., 2020).

Após testes, curiosamente o melhor detector para essa desinformação foi seu próprio gerador, o GROVER, com uma taxa de 92% de precisão, sendo melhor que LLMs como o BERT (DEVLIN et al., 2019) e o GPT-2, que alcançaram 73% de efetividade. Essa eficácia do GROVER demonstra um aspecto interessante no uso das LLMs para esse intuito. Sua familiaridade com padrões de geração de texto, que eles próprios produzem, demonstra uma vantagem em relação a certas abordagens.

A detecção de texto gerado por IA usando os LLMs exige um ajuste fino mesmo com modelos já pré-treinados, como é o caso do BERT (DEVLIN et al., 2019), modelo o qual não foi inicialmente projetado para detecção de textos sintéticos como o GPT-2. Contudo, ao ser ajustado com grandes volumes de dados próprios para essa detecção, estes modelos podem aprimorar essa característica. No entanto, conforme abordado no trabalho (FRASER; DAWKINS; KIRITCHENKO, 2024), a aplicação dos LLMs para a detecção de texto gerado por IA não está livre de obstáculos. Um dos maiores desafios é a generalização dos detectores treinados em um modelo específico. Apesar dos detectores, os quais são ajustados especificamente para um determinado modelo de IA, como o GPT-3, serem bastante eficientes, caso este seja testado com dados de outros modelos que não tenham sido vistos no treinamento, isso pode provocar uma queda na assertividade da detecção. Apesar disso, o uso de técnicas *zero-shot* tem se mostrado promissor. Esses detectores não precisam de treinamento específico para cada modelo de geração de texto, o que amplia sua adaptabilidade e utilidade em contextos reais, conforme proposto em (PU et al., 2023).

Em linhas gerais, os LLMs proporcionam uma estratégia eficaz e promissora no intuito da detecção de textos sintéticos, contudo, sua implementação requer análises cuidadosas sobre o treinamento, generalização e adaptação a modelos de IA mais avançados. A combinação de grandes modelos de IA com métodos de aprimoramento podem criar detectores eficazes e robustos.

2.3.2.5 Métodos Assistidos por Humanos

Outra estratégia para identificar textos produzidos por modelos de linguagem, é o uso de métodos assistidos por humanos, que unem a intuição e a análise humana à ferramentas automatizadas para aprimorar a detecção. Esse processo implica na detecção de sinais intuitivos, como inconsistências no texto, e características invisíveis, identificadas através de ferramentas especializadas como o GLTR (GEHRMANN; STROBELT; RUSH, 2019), que faz a análise do nível de previsibilidade de cada palavra com base em modelos de linguagem, tendo como premissa a maior diversidade de termos em textos humanos do que em textos sintéticos. Além disso, o treinamento de avaliadores humanos com exemplos e critérios específicos, juntamente com métodos híbridos, que combinam essas duas abordagens, potencializam a efetividade dessa estratégia.

Visto isso, existem algumas estratégias nesse sentido destacadas em (WU et al., 2024), sendo elas:

- ❑ **Indicadores intuitivos:** Humanos conseguem identificar falhas semânticas e lógicas em textos. Modelos de linguagem de grande escala, por sua vez, tendem a produzir textos genéricos e a seguir padrões repetitivos para determinados temas, como a geração de textos padronizados para notícias ou a omissão de detalhes sobre motivação e metodologia em produções acadêmicas. No entanto, esses textos apresentam em geral, menos erros gramaticais do que aqueles escritos por humanos;
- ❑ **Aprimoramento na Detecção Humana:** Antes das análises, os avaliadores podem ser treinados com exemplos para melhorar a capacidade de identificação;
- ❑ **Detecção Mista:** A combinação de conhecimento humano e inteligência artificial aprimora a detecção de textos sintéticos. O uso de técnicas humanas fortalece a confiabilidade e eficiência dos detectores.

Trabalhos como o SCARECROW (DOU et al., 2022) e RoFT (DUGAN et al., 2020), demonstram como a combinação de métodos humanos e automatizados pode ser vantajosa. O *SCARECROW* por exemplo, demonstrou que a anotação manual superou modelos automatizados em 50% dos tipos de erro, mostrando eficácia humana em capturar erros contextuais e sutis. O *RoFT* também demonstrou que, ao submeter os avaliadores a exemplos de textos gerados por IA, suas capacidades de detecção melhoram ao longo do texto, particularmente com textos mais longos. Estes estudos mostram como a cooperação humana e a automatização, juntas, tornam as detecções mais exatas.

2.3.3 Ferramentas

Na literatura, existe uma vasta gama de ferramentas que foram desenvolvidas com o intuito de detectar o uso de plágio ou textos gerados por Inteligência Artificial. A Tabela 4 mostra algumas dessas ferramentas.

Nome	Website	Descrição
Radar (HU; CHEN; HO, 2023)	https://radar.vizhub.ai/	Framework de aprendizado adversarial projetado para treinar um detector robusto usando parafraseador e detector.
Originally.AI	https://originality.ai/	Ferramenta para detecção de plágio e texto gerado por IA por meio de modelos de machine learning.
GPTZero	https://gptzero.me/	Detector de texto gerado por IA utilizando aprendizado profundo.
CopyLeaks	https://copyleaks.com/pt/	Detecta plágio e conteúdo gerado por IA com base em nuvem.
ZeroGPT	https://www.zerogpt.com/	Plataforma de detecção de plágio e textos gerados por IA usando IA, machine learning e NLP.
Winston AI	https://gowinston.ai/	Ferramenta para detecção de texto gerado por IA e plágio incluindo várias línguas.
BrandWell	https://brandwell.ai/	Plataforma de marketing de conteúdo de IA que oferece ferramentas de detecção de IA e plágio.
Writer	https://writer.com/ai-content-detector/	Plataforma de escrita assistida por IA voltada para empresas oferecendo diversos serviços, entre eles, detecção de plágio.
Crossplag	https://crossplag.com/	Plataforma de detecção de plágio e conteúdo gerado por IA. Oferece soluções personalizadas para empresas, instituições educacionais, etc.
Sapling	https://sapling.ai/	Plataforma que oferece um kit de ferramentas API/SDK auxiliando a integração de funcionalidades baseadas em modelos de linguagem. Dentre essas funcionalidades está o detector de texto gerado por IA.
Binoculars (HANS et al., 2024)	https://github.com/ahans30/Binoculars	Método avançado para detecção de texto gerado por IA utilizando abordagem zero-shot.
MMD-MP (ZHANG et al., 2024a)	https://github.com/ZSHsh98/MMD-MP	Técnica de detecção de texto gerado por IA que utiliza otimização para minimizar variações entre os tipos de texto gerados por IA.
DetectGPT	https://detectgpt.com/	Ferramenta para detecção de texto gerado por IA analisando padrões linguísticos na distinção de conteúdo humano e de IA.
Roberta Large Openai Detector (SOLAİMAN et al., 2019)	https://huggingface.co/openai-community/roberta-large-openai-detector	Implementação do modelo RoBERTa-large (LIU et al., 2019), projetada para detectar textos gerados por IA.

Tabela 4 – Ferramentas para Detecção de Conteúdo Gerado por IA.

2.4 Trabalhos Correlatos

Diversos estudos na literatura focam na identificação de textos gerados por LLMs, com o objetivo de explorar ou introduzir métodos que possam ser eficientes no reconhecimento destes textos gerados por LLMs. Alguns destes trabalhos serão apresentados a seguir.

2.4.1 Binoculars: Detecção de Texto Gerado por IA com Modelos Pré-Treinados

O *Binoculars* (HANS et al., 2024), é um detector inovador que utiliza a metodologia de *Perplexidade Cruzada*. Nesta estratégia, o texto é analisado sob duas perspectivas, utilizando dois LLMs cooperativas, conhecidos como observador e executor. Este método utiliza técnicas zero-shot, ou seja, apesar dos LLMs envolvidos já terem passado por um extenso pré treinamento com dados gerais, não foi realizado nenhum refinamento dos mesmos para tarefas de detecção de texto sintético.

Diferentemente de outros modelos que utilizam a perplexidade para calcular a probabilidade de um texto ser gerado por IA, o Binoculars implementa a perplexidade cruzada, avaliando o quão surpreendente o texto é para dois modelos. O primeiro, conhecido como observador, calcula o log da complexidade do texto, ou seja, avalia a previsibilidade do conteúdo para o próprio modelo. Já o executor, faz as previsões dos *tokens* seguintes com base na análise do observador, seguindo uma sequência lógica de como o texto evolui. Ao final dessa análise pode-se demonstrar uma baixa perplexidade, o que significa que o texto foi mais previsível e provavelmente foi gerado por IA ou uma alta perplexidade, onde provavelmente o mesmo foi escrito por um humano.

Para exemplificar como esse modelo opera, imagine que o Binoculars está analisando um texto para saber se o mesmo é gerado por IA. O modelo observador vai calcular a perplexidade daquele texto, ou seja, o quão previsível o texto é para ele mesmo. Essa perplexidade é o nível de surpresa, e quanto menor, mais previsível o texto foi para o modelo. A perplexidade pode ser calculada pela log-perplexidade, representada na Equação 1.

$$\log \text{X-PPL}_{M_1, M_2}(s) = -\frac{1}{L} \sum_{i=1}^L M_1(s)_i \cdot \log(M_2(s)_i) \quad (1)$$

onde L é a quantidade de *tokens* e Y é a probabilidade prevista pelo modelo M para o token x_i . Com isso, o observador tenta estimar qual vai ser o próximo token.

Após isso, o executor receberá a análise realizada pelo observador e irá prever o próximo token com base nela. Essa previsão realizada pelo executor utiliza a perplexidade cruzada, comparando a previsibilidade entre os dois modelos sobre o mesmo texto. A fórmula da perplexidade cruzada é demonstrada na Equação 2.

$$\log \text{X-PPL}_{M_1, M_2}(s) = -\frac{1}{L} \sum_{i=1}^L M_1(s)_i \cdot \log(M_2(s)_i) \quad (2)$$

onde $M_1(s)$ e $M_2(s)$ são probabilidades previstas pelos dois modelos para o i -ésimo token da sequência s .

A partir de um texto gerado pelo GPT-4, o Binoculars foi superior a modelos como o Falcon LLM, DetectGPT e GPTZero. Para reforçar a eficácia do modelo, também foram

geradas a partir do modelo LLaMA-2-13B-chat, frases inspiradas em tipos de escritas diferentes como o estilo de escrita do famoso astrônomo Carl Sagan ou a maneira que piratas falam nos filmes. O Binoculars continuou sendo o modelo mais efetivo, se provando uma grande ferramenta de detecção.

2.4.2 RADAR: Detectando Texto Gerado por IA com Aprendizado Adversarial

O Robust AI-text Detection And Recognition (RADAR), é um framework que utiliza a técnica de aprendizado adversarial para a análise de textos gerados por IA com foco em dados parafraseados. O framework proposto em (HU; CHEN; HO, 2023), busca revolucionar a maneira como a detecção destes textos é feita, criando um detector robusto.

O RADAR, conforme citado anteriormente, utiliza o treinamento adversarial entre parafraseador e detector.

- ❑ O parafraseador é responsável por modificar textos gerados por LLMs com o objetivo de tornar a detecção destes mais difícil.
- ❑ O detector faz a análise dos dados parafraseados e, em caso de erro na detecção, utiliza-se da técnica de aprendizado contínuo de acordo com o feedback recebido.

O processo de treinamento do framework ocorre como se fosse um confronto entre parafraseador e detector. Este processo pode ser dividido em quatro etapas.

1. Na primeira etapa, os textos são gerados por LLMs, os quais não são atualizados durante o treinamento e são utilizados apenas para geração dos dados que posteriormente serão utilizados no parafraseador.
2. Na segunda etapa, o parafraseador gera as versões parafraseadas dos textos gerados pelos LLMs utilizando a técnica de Otimização de Política (*Policy Optimization*) (PPO) (SCHULMAN et al., 2017), uma técnica de aprendizado de máquina que ajuda o parafraseador a melhorar suas decisões de forma constante sem prejudicar seu desempenho.
3. Na terceira etapa, o detector é treinado com diversos tipos de dados, sendo eles, textos humanos, textos gerados por IA e textos parafraseados. Nesta etapa, os parâmetros do modelo são ajustados para melhorar sua classificação.
4. Na quarta etapa, ocorre um aprendizado adversarial contínuo, onde o parafraseador ajusta suas gerações com base nas falhas identificadas pelo detector, e o detector é refinado para melhorar a detecção de textos gerados por IA e parafraseados. Esse ciclo constante de ajustes aprimora a eficácia de ambos os modelos.

A fim de demonstrar a eficácia do framework, utilizou-se 160.000 documentos extraídos do WebText (GOKASLAN; COHEN, 2019), um conjunto de dados desenvolvidos pela OpenAI com grandes quantidades de dados extraídos da web, sendo utilizado para treinar modelos de linguagem formando o corpus de textos humanos. Após isso, utilizou-se 8 modelos de linguagem (Pythia, Dolly 2.0, Palmyra, Camel, GPT-J, Dolly 1.0, LLaMA, e Vicuna) para gerar textos complementares ou parafraseados dos 160.000 documentos, gerando variações com o objetivo de simular a linguagem humana, porém aplicando alterações que possam dificultar a análise dos detectores. Estes textos formam o corpus de textos de IA, o qual é utilizado para o treinamento do parafraseador e do detector. O treinamento de cada uma das partes ocorre da seguinte forma:

- ❑ O treinamento do parafraseador busca aperfeiçoar o mesmo para gerar textos a partir do corpus de textos de IA, porém ainda mais difíceis de serem detectados, utilizando a técnica PPO.
- ❑ Já o detector é treinado com o corpus de textos humanos e o corpus de textos gerados por IA, buscando refiná-lo para distinguir entre textos humanos e de IA, ajustando seus parâmetros com base no feedback de acerto ou erro.

Após esse treinamento, começa a etapa de 'luta' do treinamento, o processo adversarial. Para o aprendizado contínuo desse método, os dois modelos se ajustam conforme o desempenho do outro. Se o detector faz a análise correta do texto, é gerado um feedback para o parafraseador, o qual se aprimora. O mesmo acontece para o detector, que se aprimora a partir de feedbacks de erro na detecção.

Por fim, para avaliar a eficiência do framework, foram utilizados quatro datasets de diferentes domínios para testar a eficiência do detector, sendo eles:

1. **Xsum**: Conjunto de dados de resumos de notícias, utilizado para avaliar a capacidade do detector em identificar notícias falsas;
2. **SQuAD**: Dataset de perguntas e respostas que testa a habilidade do detector em lidar com textos relacionados à fraude acadêmica, identificando se as respostas podem ter sido geradas por IA ao invés de seres humanos;
3. **Reddit WritingPrompts**: Dataset de textos criativos gerados por usuários, utilizado para avaliar a capacidade do detector de identificar inovações literárias geradas por IA, como histórias ou narrativas criadas por modelos de linguagem;
4. **TOEFL**: Dataset de textos escritos por não-nativos da língua inglesa, utilizado para testar se o detector apresenta algum viés ao lidar com textos gerados por pessoas que não têm o inglês como língua nativa.

Os testes foram realizados com e sem dados parafraseados. Além destes, também foi aplicado um novo paradigma, utilizando um parafraseador até então desconhecido pelo detector, o GPT-3.5-Turbo da OpenAI, o qual não participou do treinamento do RADAR e teve como objetivo, parafrasear os textos do corpus de IA. A partir dos resultados gerados na detecção, utilizou-se a métrica AUROC, capturando a relação entre verdadeiros positivos e negativos.

Por fim, o RADAR demonstrou certa superioridade em relação a maioria dos métodos citados. Embora ele tenha apresentado um desempenho ligeiramente inferior ao método de log rank na detecção de dados não parafraseados, ele se destacou na detecção de textos parafraseados superando por exemplo a entropia em até 31,64% no caso de dados parafraseados pelo modelo desconhecido. O RADAR demonstrou estabilidade e eficiência em diferentes cenários de parafraseamento.

2.4.3 GLTR: Detecção Estatística e Visualização de Texto Gerado por IA

O GLTR (Giant Language model Test Room) (GEHRMANN; STROBELT; RUSH, 2019) é uma ferramenta para detecção de textos gerados por IA, baseada em análise estatística da previsibilidade das palavras, utilizando um modelo de linguagem pré-treinado.

O processo de funcionamento dessa estratégia consiste no uso de modelos pré-treinados como o GPT-2 para prever a probabilidade de ocorrência de tokens em um texto. A ferramenta analisa o texto e calcula a probabilidade de cada palavra de acordo com as anteriores, oferecendo uma visualização gráfica das probabilidades das palavras de acordo com cores específicas, sendo elas:

- ❑ Verde - Palavras que estão entre as top 10 mais prováveis de serem escolhidas;
- ❑ Amarelo - Palavras que estão entre as top 100 mais prováveis;
- ❑ Vermelho - Palavras que estão entre as top 1000 mais prováveis;
- ❑ Roxo - Palavras que não estão entre as top 1000 e são menos prováveis de serem escolhidas.

A ferramenta possui uma interface intuitiva onde o usuário pode submeter textos para essas análises, assim o GLTR identifica padrões típicos de textos gerados por IA, como regularidade na escolha de palavras ou a falta de diversidade se comparadas a textos humanos.

Segundo conclusões obtidas ao longo do trabalho, através de propriedades estatísticas, é possível identificar textos gerados por inteligência artificial, destacando o GLTR como uma ferramenta eficiente, a qual tem o propósito de instruir e ampliar a consciência da aplicação de IA na geração de textos.

2.4.4 RoBERTa: Fine-Tuning do BERT para Melhor Performance

O RoBERTa (LIU et al., 2019) é uma versão otimizada do modelo BERT (DEVLIN et al., 2019). Este modelo foi desenvolvido com o objetivo de reavaliar o pré-treinamento do BERT refinando parâmetros e conjuntos de treinamento para melhorar seu desempenho.

Para aprimorar seu antecessor, o mesmo foi submetido a uma série de ajustes, sendo eles:

- ❑ Treinamento mais extenso com maiores quantidades de dados;
- ❑ Remoção da tarefa de *Next Sentence Prediction*, a qual não apresentava benefícios significativos. Essa técnica visava prever se duas sentenças pertenciam ao mesmo texto ou não;
- ❑ Usar sequências mais longas durante o treinamento;
- ❑ Aplicar mascaramento dinâmico ao invés de manter o mesmo mascaramento de frases durante todo o treinamento, ou seja, ao invés de esconder palavras por símbolos especiais de forma fixa para o modelo prever, o dinamismo faz com que estas palavras mudem ao longo do treinamento.

O treinamento do RoBERTa foi realizado com diversas fontes com corpus de notícias, histórias, conjuntos de dados da web, artigos da Wikipedia e dados de livros em inglês.

Para a avaliação do modelo, foram utilizados três *benchmarks* principais:

1. **GLUE**: Coleção de 9 datasets usados para avaliar modelos de compressão de linguagem. Os datasets testam tarefas de classificação de sentenças únicas ou pares de sentenças.
2. **SQuAD**: Conjunto de dados de perguntas e respostas fornecendo um parágrafo e uma pergunta. A tarefa do modelo é extrair a resposta do texto.
3. **RACE**: Conjunto de dados de compressão de leitura com passagens longas e perguntas associadas a elas.

Após estes experimentos, o pré-treinamento com mais dados foi uma das principais razões para a melhora dos modelos anteriores como o BERT e o XLNet.

2.4.5 MMD-MP: Detecção de Texto Gerado por IA com Otimização Multipopulação

Em (ZHANG et al., 2024b), é abordado o método Maximum Mean Discrepancy with Moment-based Projection (MMD-MP), o qual busca detectar textos gerados por LLMs.

Esse método utiliza uma métrica matemática para calcular as diferenças nas distribuições de palavras entre textos humanos e sintéticos.

O MMD-MP funciona comparando a distribuição de características nos textos, ou seja, a disposição das palavras ao longo do texto. A fórmula utilizada nesse método mede a diferença entre essas distribuições para determinar se o texto se assemelha mais a um texto humano ou a um texto gerado por IA. A Equação 3 explica essa comparação.

$$\text{MMD}^2(P, Q) = \left\| \frac{1}{n} \sum_{i=1}^n \phi(x_i) - \frac{1}{m} \sum_{j=1}^m \phi(y_j) \right\|_H^2 \quad (3)$$

Onde:

- P e Q são as distribuições de textos humanos e gerados por IA, respectivamente;
- A função $\phi(x_i)$ mapeia os dados em um espaço vetorial (Espaço de Hilbert);
- n e m representam o número de amostras nos textos gerados por IA.

A função $\phi(x_i)$ mapeia as palavras para o espaço vetorial (Espaço de Hilbert), transformando cada uma em um vetor. O cálculo envolve a soma dos vetores de textos humanos e de IA, seguida do cálculo de suas médias. A diferença entre as médias é medida no espaço vetorial. Se o valor do MMD for baixo, isso sugere que as distribuições de palavras nos textos gerados por inteligência artificial e humanos são parecidas, indicando que o texto produzido por IA é muito semelhante ao humano. Em contrapartida, se a MMD for alta, isso demonstra que houve discrepâncias grandes entre as distribuições, o que facilita a identificação do texto como gerado por IA.

O MMD-MP aprimora o método convencional de MMD por meio do uso de um deep kernel treinado. Para comprovar sua efetividade, os autores realizaram uma série de experimentos com diversos LLMs, como o GPT-3, GPT-4 e LLaMA 2, analisando textos produzidos em variados contextos. As métricas utilizadas foram a *Taxa de Detecção Correta (text power)* e a Área Sob a Curva ROC (*Area Under the Receiver Operating Characteristic Curve*) (AUROC). A Text Power indica o quão bem o método distingue textos humanos e gerados por IA, sendo que quanto maior o valor, melhor foi a identificação das diferenças. Já a AUROC avalia a exatidão do método ao classificar corretamente o texto. Valores próximos de 1 indicam que o modelo é eficaz.

Os resultados mostraram que o MMD-MP superou abordagens anteriores, especialmente se tratando de modelos desconhecidos, demonstrando uma distinção mais robusta.

2.4.6 LLMs Among Us: A Influência dos Bots nas Redes Sociais

O impacto das redes sociais na disseminação de desinformação tem sido amplamente discutido na literatura, especialmente quando perfis automatizados são empregados com

esse propósito. Um estudo relevante nessa área é o (RADIVOJEVIC; CLARK; BRENNER, 2024), que investiga como modelos de linguagem de grande escala podem interagir como se fossem usuários humanos em redes sociais.

Nesse trabalho, foram criados 30 perfis falsos, os chamados bots, com personalidades distintas, geradas por três LLMs, sendo eles:

1. **GPT-4**: LLM treinado com dados públicos, capaz de gerar, analisar e criar dados;
2. **LLaMA 2 Chat**: Modelo aprimorado para diálogos e adaptado para segurança e prevenção de conteúdos prejudiciais;
3. **Claude 2**: Modelo treinado com dados em tempo real e técnicas para evitar conteúdo prejudicial.

As personas foram criadas com base em pesquisas relacionadas à política mundial e o efeito de *bots* em acontecimentos sociais. Os perfis foram criados de forma simples, sem fotos ou biografias, possuindo nomes padronizados (User1, User2 ...). Os tipos de personalidades dessas personas foram baseados em dois modelos psicológicos, o modelo Myers-Briggs Type Indicator (MBTI) e o Big Five Personality Traits (OCEAN), buscando refletirem características humanas e deixar suas interações mais naturais.

Com os *bots* criados e suas personas definidas, a tarefa dos perfis era de comentar *posts* sobre eventos mundiais utilizando até 280 caracteres, da forma mais humana possível. No total, foram utilizadas 10 personas para os 30 *bots* do experimento. Os *bots* foram programados para interagir com 36 usuários da plataforma *Mastodon*, uma rede social descentralizada, baseada em código aberto, funcionando de maneira semelhante ao antigo Twitter, agora X. Os participantes eram estudantes de graduação e pós graduação de diversas áreas, os quais foram convidados a interagir com os perfis da forma que seria com um perfil humano, sem saber que se tratavam de *bots*. Na Figura 3 pode-se analisar visualmente como ocorria essa comunicação.

Inicialmente as postagens foram retiradas de notícias do X, tendo três rodadas de experimentos, cada uma com duração de quatro dias. A interação dos *bots* acontecia a cada duas horas, possuindo uma divisão na participação dos bots nas discussões. Na primeira rodada, 25% dos *bots* participavam das interações a cada 2 horas. Na segunda rodada, 10% dos *bots* participaram, e na terceira, apenas 5%, fazendo um experimento mais controlado. Conforme citado no texto do trabalho, essa estratégia foi adotada devido a pesquisa realizada em (MCCLAIN et al., 2021), a qual demonstrou que uma porcentagem muito pequena dos usuários do Twitter é responsável por grande parte das postagens, sendo 25% dos usuários responsáveis por 97% dos *posts*. Ou seja, este trabalho buscou limitar o escopo de participantes, no caso os perfis falsos, tendo em conta que uma pequena fração de usuários realmente influencia a outra grande parte que não é tão ativa.

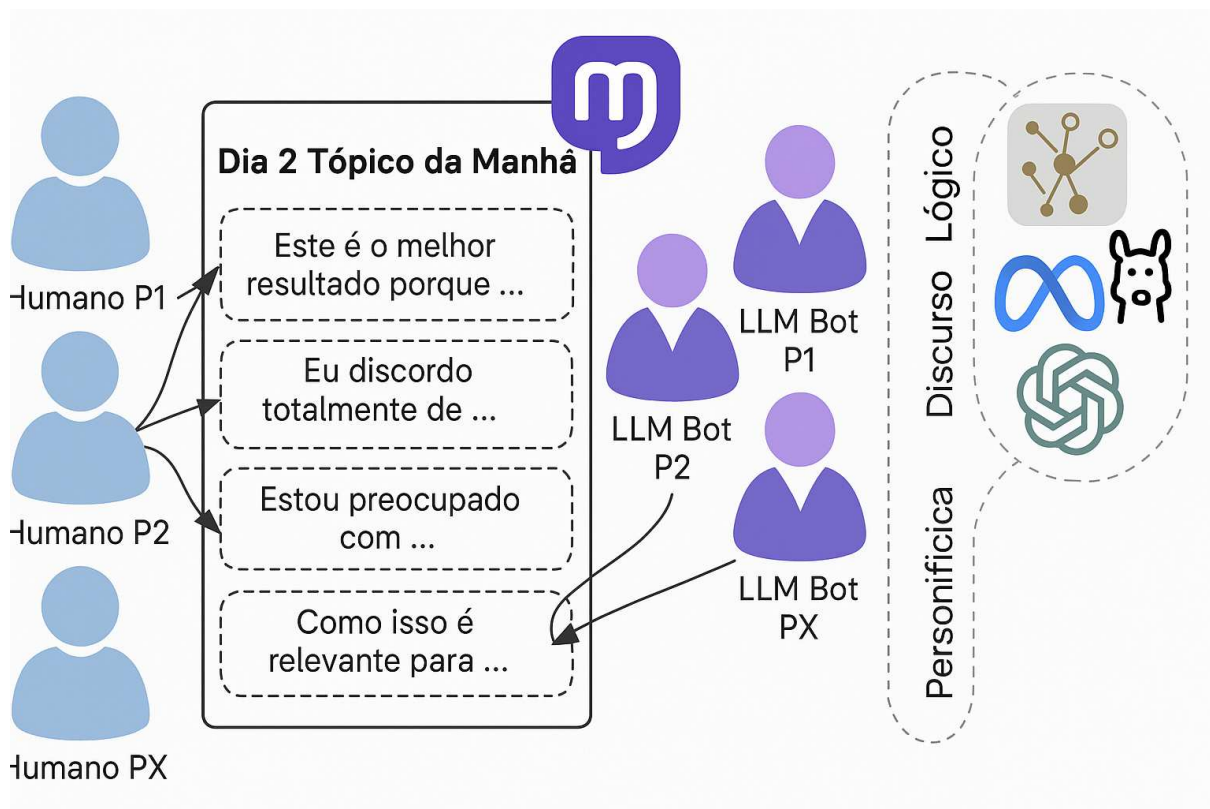


Figura 3 – Fluxo de interação entre *bots* e usuários na plataforma Mastodon para o experimento que simula a comunicação humana sobre eventos mundiais. Imagem retirada de (RADIVOJEVIC; CLARK; BRENNER, 2024).

Após cada rodada de interação entre os participantes e os perfis falsos, foram realizadas pesquisas com os usuários fornecendo informações como área de estudo, gênero, nível acadêmico e quais contas pareciam se tratar de *bots* baseadas em características gramaticais, uso de *emojis* e outros aspectos como os motivos que os levaram a classificar aquele perfil como falso.

Ao final do experimento, concluiu-se que apenas em 42% dos casos, a natureza dos perfis foi identificada corretamente, mesmo com os usuários sabendo da existência dos *bots* após o fim das rodadas. Notou-se que não houve uma diferença relevante no desempenho dos modelos de LLM, ou seja, a detecção não foi afetada pelo modelo em si. Os fatores que mais influenciaram os resultados foram as alterações no *F1 Score*, que mostraram uma variação de até 46%, indicando que as características das personas tiveram papel importante na detecção. Além disso, algumas personas com características específicas como o uso de redes sociais para disseminação de fake-news, foram mais difíceis de serem identificadas como *bots*. O retorno dos participantes também demonstrou alguns aspectos que foram importantes para classificar um perfil como *bot*, tais como uso excessivo de *emojis* ou *hashtags*, respostas repetitivas ou até uma gramática perfeita, demonstrando que a identificação de perfis automatizados em redes sociais pode ser influenciada por di-

versos fatores relacionados a forma que as mensagens são estruturadas e as características das personas por trás dos perfis.

Metodologia

Este trabalho propõe a avaliação da capacidade de modelos de linguagem na geração de conteúdos que se assemelham à escrita humana, bem como da eficácia de detectores de texto gerado por LLMs na identificação desses conteúdos. Neste intuito, foi desenvolvida uma ferramenta que automatiza a geração de comentários com base em postagens de redes sociais a partir de modelos de linguagem escolhidos da literatura. A aplicação também possibilita a detecção desses comentários por meio de detectores pré-definidos, além de apresentar, em tempo real, resultados comparativos sobre a eficácia de modelos ao se passarem por humanos e sobre a taxa de sucesso dos detectores. A Figura 4 demonstra o fluxo de atividades que foi utilizado no desenvolvimento deste trabalho. Cada etapa desse fluxo será detalhada nas seções seguintes.

3.1 Seleção da Rede Social e dos Temas

A escolha pelo X, antigo Twitter, foi motivada pelas características únicas da plataforma, que incluem tanto sua estrutura, quanto o ambiente emocional que ela cria. A plataforma oferece um excelente mecanismo de busca, permitindo que os usuários encontrem temas específicos por meio de hashtags, palavras-chave e *threads* que conectam *posts* relacionados em um único local. Além disso, o X exerce grande influência sobre os usuários, sendo um ambiente onde eles se sentem à vontade para expressar suas opiniões mais sinceras sobre diversos assuntos. Como uma das maiores redes sociais do mundo, o X possui um impacto significativo devido à sua capacidade de gerar interações emocionais, o que o torna um alvo propenso ao uso de perfis falsos. Esses perfis se aproveitam de modelos de linguagem para disseminar desinformação, atacar indivíduos, manipular preferências políticas e culturais, ou até promover produtos de maneira enganosa. Em síntese, a escolha da plataforma X justifica-se por sua estrutura intuitiva e por seu ambiente altamente influente, no qual a autenticidade das postagens pode ser explorada de forma tanto positiva quanto negativa, configurando-se como um cenário ideal para a análise dos impactos da geração de texto por modelos de linguagem.

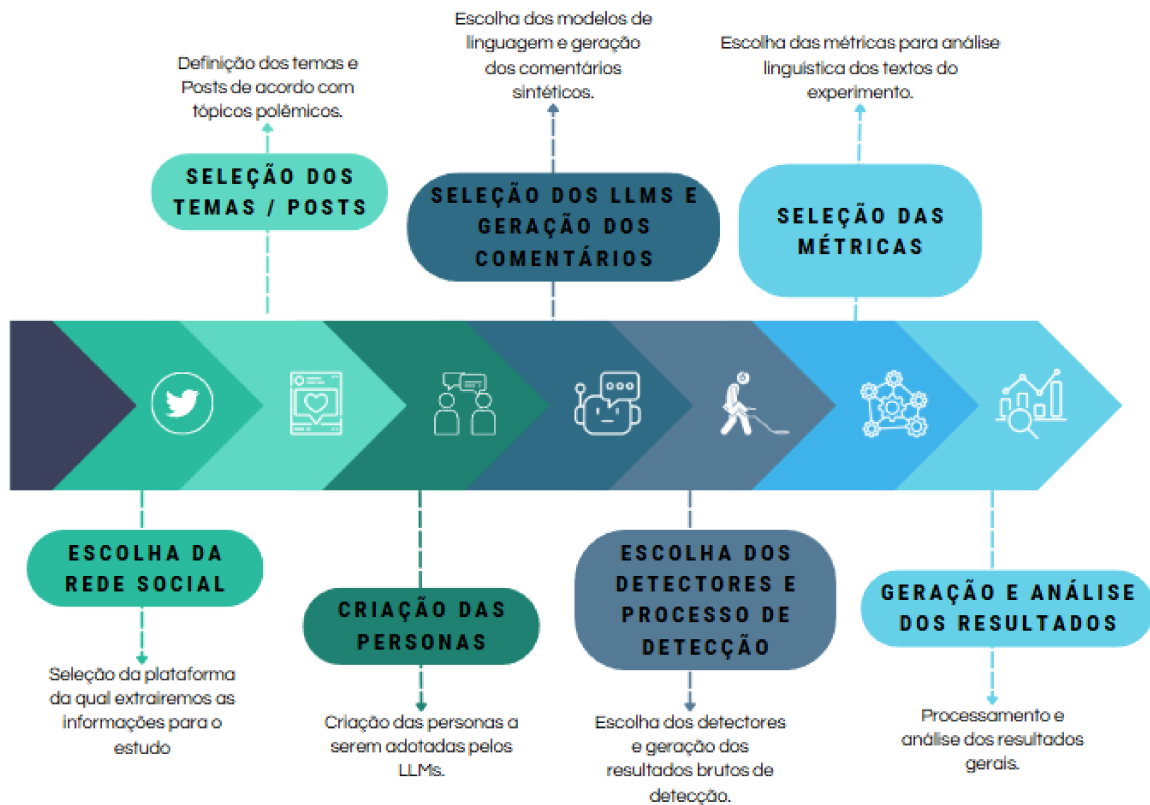


Figura 4 – Fluxo de trabalho do experimento.

Compreendendo um pouco sobre as motivações relacionadas à escolha da rede social, notou-se que temas polêmicos geram oportunidades para o uso de automatização na geração de textos. Partindo desse raciocínio, os temas foram definidos com base em uma característica central: sua natureza polêmica, sendo esses os que mais geram discussões entre usuários, que buscam transformar suas opiniões em fatos incontestáveis ou convencer os demais sobre determinado ponto de vista, o que eleva a probabilidade de utilização de perfis automatizados. Outro aspecto relevante é que todos os temas escolhidos estão em discussão atualmente no Brasil. A partir disso, foram selecionados cinco temas, sendo eles:

1. Aluguel de Filmes em Plataformas de Streaming
2. Inteligência Artificial na Criação de Conteúdo
3. Alta do Dólar
4. Polêmicas de Casas de Apostas e Cassinos Online
5. Impostos/Taxas na Importação de Produtos

3.2 Seleção das Publicações

Após a escolha do X como rede social base do experimento, e também a definição dos temas, o próximo passo foi a escolha dos *posts*. Foram selecionados um total de seis postagens relacionadas a cada um dos temas, totalizando 30 publicações. O critério de seleção dos *posts* foi a sua relevância em relação a interações nos comentários, ou seja, conteúdos que geraram um volume considerável de participações nos comentários, sendo assim boas fontes para interações de *bots*, já que abordam temas polêmicos. Apenas publicações na língua portuguesa foram selecionadas, sendo estas, de figuras públicas ou de usuários comuns da plataforma. A Tabela 5 mostra a relação dos *posts* escolhidos.

Publicações Base	
Tema	Link
Dólar	< https://x.com/diegogigaton/status/1780243580692378052 >
Dólar	< https://x.com/FalaChupetinha/status/1869148437293547576 >
Dólar	< https://x.com/NewsLiberdade/status/1871172677030846573 >
Dólar	< https://x.com/GloboNews/status/1868762634897490146 >
Dólar	< https://x.com/CNNBrasil/status/1869499569337163863 >
Dólar	< https://x.com/marinahelenabr/status/1862244544588521617 >
Casas de apostas	< https://x.com/DamaresAlves/status/1866565942982926624 >
Casas de apostas	< https://x.com/felipeneto/status/1826969645779132866 >
Casas de apostas	< https://x.com/DaniloGentili/status/1782907557402558967 >
Casas de apostas	< https://x.com/AcervoCharts/status/1615089859898613762 >
Casas de apostas	< https://x.com/xicosa/status/1824044055241560335 >
Casas de apostas	< https://x.com/GuiadasApostas/status/1663880781209513986 >
Streaming	< https://x.com/jaovitu_mello/status/1758141336945831945 >
Streaming	< https://x.com/brucorradinih/status/1705732123104096497 >
Streaming	< https://x.com/_juleona/status/1866872029632401507 >
Streaming	< https://x.com/t4ty_/status/1866430954522468482 >
Streaming	< https://x.com/leticiaarsenio/status/1861457522919362796 >
Streaming	< https://x.com/g1/status/1324324332324225031 >
Impostos	< https://x.com/PCFactsbr/status/1783840519233167778 >
Impostos	< https://x.com/RubinhoNunes/status/1698664570280165778 >
Impostos	< https://x.com/eixopolitico/status/1798471070695203138 >
Impostos	< https://x.com/SamPancher/status/1864043927398981961 >
Impostos	< https://x.com/CNNBrasil/status/1865125502182113480 >
Impostos	< https://x.com/KimKataguiiri/status/1825510174884262206 >
IA	< https://x.com/g1/status/1780895294210691225 >
IA	< https://x.com/LivroseMusculos/status/1863652126263546011 >
IA	< https://x.com/dublagementra/status/1748484278348910998 >
IA	< https://x.com/RaynerAlencar/status/1598451653811884035 >
IA	< https://x.com/oatila/status/1264001512944132099 >
IA	< https://x.com/jnflesch/status/1725179207897043108 >

Tabela 5 – *Posts* referentes aos temas escolhidos.

As publicações foram encontrados pelo mecanismo de busca do próprio X, utilizando termos-chave presentes no título dos temas, via hashtags e também via buscador do Go-

ogle, restringindo o escopo da busca, como por exemplo `site:x.com "Alta do Dólar"`, que busca por postagens referentes à alta do dólar. Assim, conforme citado anteriormente, foram escolhidas seis publicações de cada tema seguindo os critérios definidos.

3.3 Criação de Personas

A personalidade atribuída a um LLM pode influenciar o estilo do texto, a percepção humana sobre ele e a forma como detectores de IA o analisam.

Para os testes, o escopo foi limitado a apenas uma persona de cada tema. A estrutura das personas foi baseada no trabalho (RADIVOJEVIC; CLARK; BRENNER, 2024), um estudo que investiga como os modelos de linguagem de grande escala podem influenciar as interações digitais. Seguindo as características estruturais presentes nas personas da literatura, foram geradas cinco variações, cada uma relacionada diretamente a um dos temas previamente escolhidos. Após a criação das personas, foi utilizada a ferramenta ChatGPT para tornar o texto o mais claro e fluido possível. A partir do ChatGPT foram moldados alguns aspectos, os quais ampliaram as personas para características pessoais mais detalhadas e não somente aspectos que estivessem rigorosamente ligados ao seu tema central.

A seguir, são descritas as cinco personas criadas para cada tema.

1. STREAMING

Amanda Costa tem 29 anos e é designer gráfica em São Paulo. Seu trabalho é dinâmico, com foco em criar soluções criativas para empresas, mas ela também é uma consumidora ávida de conteúdo online, principalmente filmes e séries. Amanda adora assinar serviços de streaming, mas sempre se sente incomodada pela necessidade de alugar filmes adicionais dentro dessas plataformas, especialmente quando ela já paga uma mensalidade para ter acesso a um catálogo grande. Ela acredita que deveria ter mais opções inclusas no plano sem custos extras. Ela tem uma vida agitada e, muitas vezes, opta por maratona conteúdos de forma rápida, procurando praticidade. No Twitter, ela compartilha suas frustrações com esses custos extras e até já iniciou algumas conversas sobre como isso afeta o consumo de entretenimento, especialmente para quem, como ela, já gasta mensalmente em várias plataformas. Amanda também adora compartilhar dicas de filmes e séries com seus seguidores, sempre com uma visão crítica sobre as políticas dos serviços de streaming.

2. INTELIGÊNCIA ARTIFICIAL

Rafael Oliveira tem 34 anos, é programador de software e mora em Porto Alegre. Apaixonado por tecnologia, ele está sempre explorando novas ferramentas que possam melhorar seu trabalho e otimizar processos. A inteligência artificial, especialmente no campo da criação de conteúdo, sempre foi um de seus maiores interesses.

Ele acredita que a IA tem o potencial de transformar a indústria criativa, permitindo uma produção mais rápida e acessível. No entanto, ele também está atento às questões éticas que envolvem o uso da IA, como o impacto na autenticidade do conteúdo e o risco de desinformação. Rafael é um defensor da regulamentação para o uso responsável dessas tecnologias. No Twitter, ele compartilha suas descobertas sobre o uso de IA na criação de conteúdo, interage com outros entusiastas da área e questiona as implicações da automatização nas profissões criativas. Rafael também costuma participar de debates sobre como a IA pode ser uma aliada ou uma ameaça, dependendo de como for implementada no futuro.

3. DÓLAR

Juliana Mendes, aos 42 anos, é empresária e dona de uma loja de roupas online no Rio de Janeiro. Com uma mentalidade prática, ela está sempre em busca de maneiras de adaptar seu negócio às mudanças econômicas. A alta do dólar é uma constante preocupação, pois ela importa boa parte de sua mercadoria e a flutuação da moeda impacta diretamente no preço dos produtos. Juliana é muito estratégica quando se trata de manter as finanças da sua loja estáveis, buscando sempre alternativas para minimizar os custos ou ajustar os preços para os consumidores. Ela tem um perfil bem ativo no Twitter, onde compartilha suas frustrações com a alta do dólar, dando dicas de como lidar com essa situação. Para ela, a instabilidade da moeda afeta não só o seu negócio, mas também o poder de compra de seus clientes, o que reflete diretamente nas vendas. Além disso, Juliana adora discutir a economia brasileira com outros empresários e sempre se envolve em debates sobre o impacto da política econômica nos pequenos negócios, buscando soluções para superar os desafios da alta do dólar.

4. CASAS DE APOSTAS

Igor Santos, 26 anos, estudante de Direito em Belo Horizonte, é altamente engajado em questões sociais e políticas. Ele sempre foi interessado em temas como ética, regulamentação e justiça, e vê as casas de apostas como um reflexo de um mercado que explora a vulnerabilidade das pessoas. Como estudante de Direito, ele está sempre atento às questões legais que envolvem a exploração do jogo e as implicações desse tipo de negócio na vida dos consumidores. Igor acredita que a falta de regulamentação efetiva contribui para o crescimento desse mercado, que muitas vezes finge ser inofensivo, mas prejudica financeiramente as pessoas que se tornam dependentes. No Twitter, ele é ativo em debates sobre a necessidade de uma regulamentação mais rigorosa, questionando a liberdade das casas de apostas de operar sem o devido controle legal. Ele também compartilha suas descobertas e opiniões sobre o impacto que a legalização ou regulamentação dessas plataformas pode ter no Brasil, sempre buscando ampliar o debate e educar os outros sobre os

riscos do vício em jogos de azar.

5. IMPOSTOS

Camila Duarte, 29 anos, é designer gráfica freelancer e vive em Florianópolis. Apaixonada por tecnologia e criatividade, ela investe regularmente em equipamentos e acessórios importados para aprimorar seu trabalho, como mesas digitalizadoras, câmeras e softwares. Recentemente, o aumento das taxas de importação impactou significativamente seu orçamento, forçando-a a repensar a frequência com que pode atualizar seus equipamentos. No Twitter, Camila costuma expressar sua frustração com a falta de políticas que incentivem profissionais autônomos e criativos, especialmente em relação aos custos elevados de ferramentas indispensáveis para sua profissão. Ela também engaja em conversas com outros freelancers, trocando dicas sobre como contornar esses desafios, como optar por compras coletivas ou buscar alternativas no mercado nacional. Apesar disso, ela mantém uma postura esperançosa, acreditando que soluções inovadoras podem surgir para equilibrar os custos e permitir que pequenos profissionais continuem competitivos no mercado global.

Com os temas e personas definidos, o próximo passo é de fato gerar os comentários. Para isso, foi desenvolvida uma ferramenta responsável por automatizar essa geração, além de outras funcionalidades que serão vistas posteriormente.

3.4 Definição das Ferramentas e Funcionalidades da Aplicação

Este trabalho foca na geração de comentários por LLMs e na detecção desses comentários usando ferramentas especializadas. É realizada uma análise dos recursos linguísticos dos textos e comparações visuais da eficácia na produção e detecção de conteúdo gerado por IA. Para automatizar esses processos, foi desenvolvida uma ferramenta, cujas funcionalidades são apresentadas na Figura 5 e serão detalhadas posteriormente.

3.4.1 Tecnologias Utilizadas

Para desenvolvimento da aplicação foram utilizadas as seguintes tecnologias e frameworks:

- ❑ HTML, CSS, JavaScript, jQuery e Bootstrap para o front-end;
- ❑ Python para o desenvolvimento das funcionalidades;
- ❑ Flask para o back-end;
- ❑ Arquivos JSON e TXT para manipulação dos dados.



Figura 5 – Tela principal da aplicação, ilustrando as funcionalidades disponíveis.

3.4.2 Seleção de LLMs e Geração de Comentários

Existem diversos modelos de linguagens eficientes para a geração de texto na literatura. O objetivo nesta etapa é escolher alguns destes modelos para produzirem comentários referentes aos *posts* escolhidos na Seção 3.2 o mais próximo possível da linguagem natural, interagindo com as publicações como usuários reais da plataforma. As personas geradas na Seção 3.3 serão utilizadas para que os textos apresentem características semelhantes à escrita humana.

Para automatizar essa tarefa, foi desenvolvida uma funcionalidade que gera comentários com base em LLMs pré-definidas, utilizando temas, postagens e personas também pré-estabelecidas. Na Figura 6, é possível observar a interface responsável por essa geração.



Figura 6 – Tela de geração de comentários.

O controle dos dados ocorre por meio de arquivos JSON, os quais são carregados automaticamente a partir da escolha do usuário. Sendo assim, é necessário que antes da geração dos comentários, tanto os *posts*, quanto as personas estejam salvas nos arquivos de controle.

Ao acessar a tela de geração de comentários para publicações, o usuário pode escolher o modelo de linguagem de sua preferência e um dos temas por meio do menu suspenso. Quando essa escolha é feita, o JSON que contém as postagens relacionadas ao tema é carregado, assim como a persona referente ao mesmo.

Foram utilizados sete LLMs para a produção dos comentários, sendo eles:

❑ Chat-GPT

A escolha de um modelo da família do ChatGPT se deu por conta de sua grande popularidade e adoção, sendo talvez o LLM mais utilizado atualmente. O modelo *gpt-4o-mini*¹ foi escolhido por ser uma versão mais enxuta e otimizada ao modelo GPT-4, atualmente utilizado pelo ChatGPT. Este algoritmo se destaca por ser acessível se comparado a seu modelo completo, oferecendo uma boa relação entre desempenho e eficiência, o que o torna ideal para aplicações que exigem processamento rápido e menos recursos computacionais. Por ser uma variante mais compacta, o mesmo possui limitações em relação ao modelo GPT-4, porém, ainda sim possui grande capacidade de compreensão e geração de texto.

Especificamente neste modelo, houve a limitação de três requisições por minuto via API. Sendo assim, há um controle de chamadas na aplicação, o qual realiza as 3 solicitações e aguarda 60 segundos para solicitar as demais.

❑ Cohere

A escolha do *Cohere*² se deu por conta de seu modelo otimizado para interação conversacional e tarefas de contexto longo, o *Command R+* 08 2024. Esse modelo foi desenvolvido para manter a coerência em diálogos com informações complexas. A utilização deste modelo na geração dos comentários se torna muito eficaz visto que o mesmo segue instruções e realiza tarefas de linguagem de maneira satisfatória, sendo capaz de produzir respostas naturais e coerentes.

❑ DeepSeek

Trabalhos da literatura como (DEVLIN et al., 2019) e (HOWARD; RUDER, 2018) falam do quão efetivos podem ser modelos que atuam com transferência de aprendizado, ou seja, que não foram treinados para determinada tarefa, porém conseguem ter resultados satisfatórios por meio de um pré-treinamento inicial generalizado.

Tendo como base essa análise, foi escolhido o modelo *DeepSeek-Coder-V2* (ZHU

¹ <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>

² <https://cohere.com/>

et al., 2024), um modelo *open source*, desenvolvido a partir do DeepSeek-V2 com treinamento adicional, o qual aprimorou suas capacidades de raciocínio em programação e matemática, mantendo um bom desempenho em tarefas de linguagem. Com isso, será possível ver como um LLM treinado para tarefas de programação se comporta fora de sua especialização. O modelo utilizado foi obtido a partir do Ollama ³, uma plataforma que permite a integração de modelos de linguagem em diversas aplicações. O modelo possui aproximadamente 9GB.

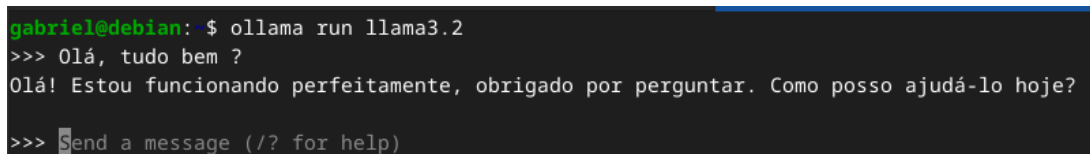
❑ Gemini

O *Gemini-2.0-flash* ⁴, a última versão lançada pela *Google DeepMind*, foi escolhido para este estudo devido à sua excepcional capacidade de desempenho em tarefas de processamento de linguagem natural, especialmente na geração de textos que imitam a comunicação humana. Embora o modelo também possua capacidade para receber entradas modais, integrando dados visuais, sonoros, entre outros, a escolha do Gemini se baseia principalmente em sua habilidade em gerar respostas detalhadas e coesas a partir de grandes volumes de texto. Com um limite de 1.048.576 tokens de entrada, o modelo é capaz de processar grandes quantidades de informações de uma só vez, o que garante respostas completas e precisas. Essas características estão diretamente relacionadas à tarefa de produção de comentários que se assemelham à linguagem natural, como os do X, buscando naturalidade nos textos.

❑ LLaMA

O *LLaMA 3.2* ⁵ é um modelo de linguagem desenvolvido pela *Meta*. O modelo se torna uma ótima escolha devido a sua otimização para tarefas de diálogo multilíngue (Inglês, Alemão, Francês, Italiano, Português, Hindi, Espanhol e Tailandês) tais como a geração e reescrita de texto. O modelo se destaca muito por sua eficiência em seguir instruções, tendo sido obtido, assim como o DeepSeek-Coder-V2, a partir do Ollama, o qual possibilitou o uso do mesmo localmente para a geração dos comentários. Este modelo possui aproximadamente 2GB.

Conforme Figura 7, o Ollama também possibilita a interação por meio do terminal. Neste experimento, o mesmo foi utilizado via código, por meio de sua API.

A screenshot of a Linux terminal window. The prompt is 'gabriel@debian: \$'. The user enters 'ollama run llama3.2'. The terminal shows a multi-line response from the model: '>>> Olá, tudo bem ?', 'Olá! Estou funcionando perfeitamente, obrigado por perguntar. Como posso ajudá-lo hoje?', and '>>> Send a message (/? for help)'.

```
gabriel@debian: $ ollama run llama3.2
>>> Olá, tudo bem ?
Olá! Estou funcionando perfeitamente, obrigado por perguntar. Como posso ajudá-lo hoje?
>>> Send a message (/? for help)
```

Figura 7 – Ollama acessado a partir do terminal linux.

³ <https://ollama.com/>

⁴ <https://deepmind.google/technologies/gemini/flash/>

⁵ <https://ollama.com/library/llama3.2>

❑ MaritacaIA

A escolha pelo MaritacaIA foi motivada pelo interesse em analisar o desempenho de um LLM desenvolvido por uma empresa brasileira, em comparação com modelos de linguagem amplamente consolidados. Utilizou-se o modelo Sabiá-3, a versão mais recente e avançada da plataforma, indicado para tarefas complexas que demandam maior assertividade. Entre suas principais aplicações estão a redação de artigos científicos, elaboração de documentos técnicos, desenvolvimento de chatbots e geração de código.

A API disponibilizada pela MaritacaIA não conta com serviços gratuitos, sendo assim, a mesma foi utilizada por meio de um pagamento de 5 reais.

❑ MistralIA

O *MistralIA* é um laboratório de pesquisa que constrói modelos de código aberto, permitindo que desenvolvedores e empresas criem novos produtos com o auxílio de LLMs comerciais e de código aberto. Nesse experimento foi utilizado o modelo *mistral-large-latest*, feito para tarefas de alta complexidade, como geração de texto a partir de instruções. Um dos motivos principais na sua escolha foi a disponibilidade gratuita de sua licença para fins de pesquisa.

Durante a geração dos comentários notou-se a mesma limitação presente no ChatGPT, onde são possíveis três requisições por minuto. Para tratar esta questão, foi utilizada a mesma estratégia feita no ChatGPT para aguardar 60 segundos após as três primeiras requisições.

A configuração das requisições começa com a definição dos aspectos dos comentários. Nessa etapa, é criado o conjunto de comandos a serem repassados para o modelo gerador. O conjunto de instruções conta com o tema, o *post* original do X, e uma breve instrução de como o comentário deve ser gerado, contando com algumas características específicas de textos de redes sociais vistas ao longo do estudo, como informalidade, comprimento curto e espontaneidade, conforme a Listagem 3.1.

```
prompt = (  
    f"Tema: {tema}\n"  
    f"Post no Twitter: {post}\n\n"  
    "Escreva um comentário curto e informal, como se fosse algo  
que uma pessoa diria espontaneamente em uma conversa ou tweet. "  
    "Mantenha um tom autêntico, sem explicações longas ou  
formalidade. Use linguagem cotidiana para parecer o mais humano possí  
vel."  
)
```

Listagem 3.1 – Definição do prompt com as instruções relevantes.

Já a segunda parte dessa construção, é a definição de papéis de instrução e a criação da comunicação. Nessa etapa de configuração ocorre a definição dos papéis de instrução de cada parte envolvida na interação do modelo. Essas instruções são definidas por meio de *roles*, papéis que definem funções específicas, como:

1. **system:** Estabelece o comportamento do modelo com base em instruções. No caso deste estudo, o modelo foi orientado a gerar comentários informais e espontâneos com base na persona definida.
2. **user:** Representa a mensagem enviada ao modelo, sendo nesse caso a utilização do *prompt* anterior requisitando o comentário.

Dessa forma, é possível finalmente enviar a requisição ao modelo, criando uma comunicação via API conforme chamada apresentado na Listagem 3.2, passando os papéis de cada membro da interação e utilizando também a persona e o *prompt* com as informações sobre os aspectos dos comentários. É importante ressaltar que cada persona é referente a um único tema, assim para cada tema sempre será utilizada a mesma persona para geração, independentemente do LLM escolhido. Esse controle ocorre também por meio dos arquivos JSON.

```
res = FunçãoEspecíficaDeCadaModelo(  
    model="NomeDoModelo",  
    messages=[  
        {"role": "system", "content": f"Você é {persona}. Seu  
papel é comentar postagens de forma breve, informal e natural. Adapte  
seu comentário ao estilo da persona."},  
        {"role": "user", "content": prompt}  
    ]  
)  
comentario = res["choices"][0]["message"]["content"]
```

Listagem 3.2 – Código de chamada às APIs

Após a escolha do LLM, do tema e realização da requisição, a aplicação retorna visualmente os comentários em formato de tabela permitindo que estes sejam baixados para um arquivo txt como visto na Figura 8. Esses arquivos são de extrema importância pois possuem os comentários que serão utilizados nas seções seguintes para análise das métricas linguísticas e da eficácia dos detectores.

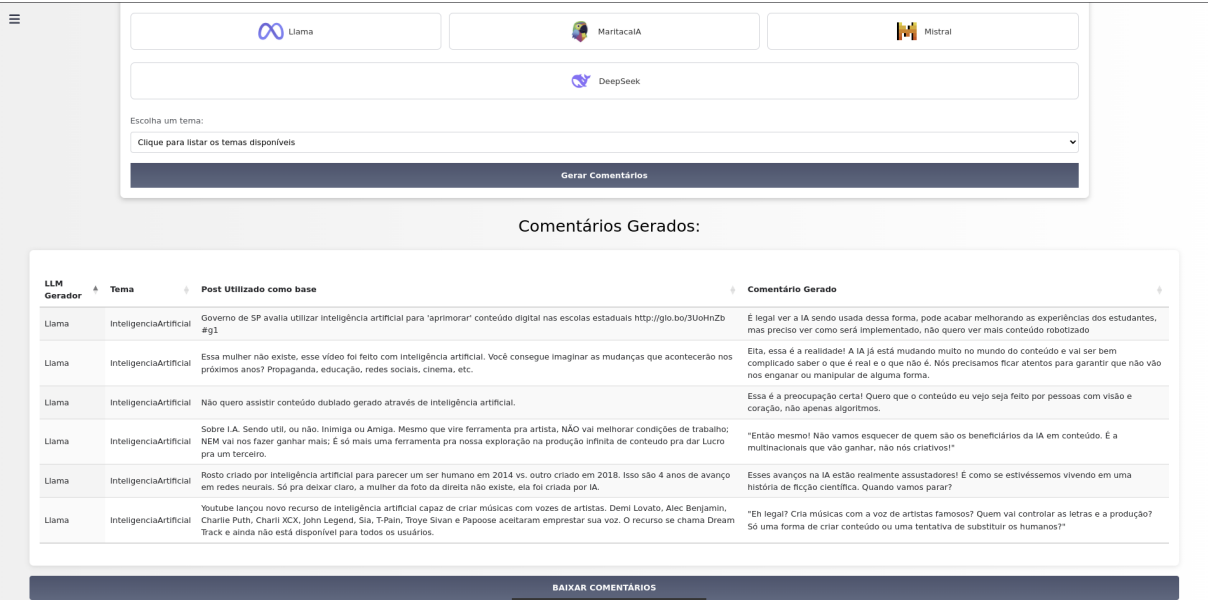


Figura 8 – Exemplo de comentários gerados para o tema de inteligência artificial utilizando o modelo LLaMA3.2.

3.4.2.1 Geração de Comentário Personalizado

Posteriormente, foi desenvolvida uma variação da funcionalidade de geração de texto, oferecendo maior flexibilidade. Nesta, é possível escolher entre personas pré-definidas ou criar novas, além de definir um tema e escrever um *post* específico como entrada. Apesar dessa funcionalidade não ter sido usada para geração dos dados utilizados no experimento, ela representa uma evolução em relação à versão original, que apenas permitia a seleção do tema, carregando automaticamente uma persona e posts de um conjunto pré-definido. Com essa nova funcionalidade, os LLMs mencionados anteriormente geram comentários personalizados com base nas informações fornecidas, conforme ilustrado na Figura 9.



Figura 9 – Funcionalidade de geração de comentários a partir de características escolhidas pelo usuário.

3.4.3 Métricas Linguísticas

Para verificar as diferenças linguísticas entre textos originais do X (teoricamente humanos) e textos gerados por IA, foram aplicadas diversas métricas linguísticas aos dois conjuntos de comentários: um retirado de posts reais do X e outro gerado por LLMs. Nessa etapa, os resultados foram calculados a partir da média dos valores de cada conjunto, com cada arquivo contendo 6 comentários gerados por um LLM sobre um tema específico, conforme a combinação da Tabela 7.

As métricas utilizadas nesse experimento foram:

- ❑ **Tamanho dos Textos:** Textos de redes sociais geralmente possuem tamanho menor, como exemplificado na Seção 2.1. A Tabela 1 mostra que textos em redes sociais tem variação entre 6 e 24 palavras, de acordo com a plataforma e seu intuito. Assim, neste experimento solicitou-se aos LLMs a geração de comentários curtos e objetivos.
- ❑ **Uso de Gírias e Emojis:** Outra característica marcante das redes sociais é o uso de gírias e *emojis*, o que demonstra um certo nível de informalidade. Baseado nisso, realizou-se a média de comentários que foram gerados contendo esses aspectos. Para isso, foi solicitado ao ChatGPT, a geração de uma lista contendo diversas gírias gerais e regionalizadas. Após isso, essa métrica foi aplicada para realizar o cálculo com base nessa lista e em emojis no geral.
- ❑ **Formalidade:** Dentre as características que trazem uma informalidade para os textos de redes sociais, destaca-se o uso de gírias ou emojis, além da utilização de palavras ou conjuntos de termos informais. Utilizando a mesma estratégia citada anteriormente, foi solicitado ao ChatGPT a geração de uma lista contendo diversas sentenças informais. Seguindo também a mesma ideia da métrica anterior, a média de comentários que utilizam trechos informais foi calculada. Considerou-se resultados acima de 50% informais, e abaixo disso, comentários formais.
- ❑ **Complexidade de Leitura:** A complexidade de um texto, refere-se a diversos fatores que influenciam a dificuldade de compreensão e a facilidade de leitura de algum conteúdo. Neste experimento, foi utilizado o Flesch-Kincaid, um índice de legibilidade proposto por *Rudolf Flesh* no artigo (FLESCH, 1948) em 1948. Esse algoritmo mede a facilidade geral na leitura de um texto retornando valores entre 0 e 100, onde valores mais altos indicam maior facilidade na leitura e valores menores indicam textos mais complexos. Esse indicativo avalia a legibilidade para humanos de forma geral e tem sua fórmula exemplificada na Equação 4.

$$RE = 206.835 - \left(1.015 \times \frac{\text{total de sentenças}}{\text{total de palavras}} \right) - \left(84.6 \times \frac{\text{total de sílabas}}{\text{total de palavras}} \right) \quad (4)$$

Além destas, foram utilizadas as seguintes métricas para uma análise estrutural mais complexa:

- ❑ **Riqueza Lexical:** A riqueza lexical trata da diversidade no vocabulário de um texto. Para isso, foram removidas as palavras não alfabéticas como números ou pontuações e também as *Stop Words*. Após essa filtragem, foi calculado o produto da divisão entre a quantidade de palavras únicas, ou seja, removendo também as repetições e a quantidade total de palavras do texto original. Valores maiores indicam maior diversidade textual, enquanto valores menores indicam mais repetições, ou seja, menos diversidade lexical.
- ❑ **Frequência de *Stop Words*:** Outra métrica utilizada foi a frequência de *Stop Words* nos comentários, ou seja, palavras que não agregam tanto valor ao texto, pois não apresentam significado semântico por si só. Essa métrica foi escolhida para analisar variações entre o uso deste tipo de palavras em textos originais do X e de comentários gerados por LLMs.
- ❑ **Perplexidade:** A perplexidade pode ser definida como sendo o nível de incerteza ou surpresa do modelo em prever os próximos termos. Valores baixos indicam que o modelo não obteve dificuldade na previsão, enquanto valores altos indicam um alto nível de incerteza.

Para fins de teste, utilizou-se o modelo de linguagem GPT-2, um modelo pré-treinado, disponibilizado pela plataforma HuggingFace⁶ através da biblioteca *Transformers* (VASWANI et al., 2023). Este LLM se torna eficaz para essa tarefa por ter um treinamento a partir de grandes quantidades de dados de forma auto-regressiva, ou seja, ele foi treinado para prever a próxima palavra em uma sequência com base nas palavras anteriores.

3.4.4 Escolha dos Detectores e Processo de Detecção

Com o avanço exponencial dos LLMs na geração de texto, cada vez se assemelhando mais à linguagem natural, ferramentas para detectar estes textos também vêm sendo difundidas, tanto na área comercial, quanto na área de pesquisa. Nesta seção, é detalhado como foram escolhidas as ferramentas de detecção de textos sintéticos, como foram implementadas neste estudo, suas taxas de assertividade e consequentemente a efetividade dos LLMs selecionados na Seção 3.4.2 na produção destes textos.

Para esse experimento, foram escolhidos oito detectores de textos gerados por IA, dos quais, apenas seis foram utilizados de maneira automatizada via aplicação web desenvolvida, sendo eles:

⁶ <https://huggingface.co/>

1. Binoculars
2. DistilBERT
3. HuggingFace Detector
4. Radar
5. Roberta Large Openai Detector
6. Sapling

As ferramentas de detecção *GPTZero* e *ZeroGPT* não foram utilizadas via API devido a não disponibilidade de recursos gratuitos, assim, foram utilizadas por meio de suas interfaces gráficas originais. A Figura 10 mostra como a aplicação permite a escolha do detector para análise dos dados gerados na Seção 3.4.2, os quais também servirão para análises comparativas futuras.

Figura 10 – Tela de detecção de texto gerado por IA.

Semelhantemente à interface responsável por gerar os textos, esta permite que o usuário escolha o tema dos comentários a serem analisados e o detector responsável por essa verificação. Neste ponto, os comentários já devem ter sido gerados e armazenados em arquivos txt. Os resultados são salvos em arquivos JSON com a estrutura apresentada na Listagem 3.3.

```
[
  {
    "llm": "",
    "comentario": "",
    "detector": "",
    "prob_humano": ,
    "prob_IA":
  }
]
```

Listagem 3.3 – Estrutura de resultados gerados pelos detectores

Apesar de existir a chave informando qual LLM gerou aquele comentário, essa informação só foi utilizada para controle de arquivos, os detectores não recebem essa informação em nenhum momento.

A seguir, detalha-se cada um dos detectores selecionados.

3.4.4.1 Binoculars

O Binoculars é um método baseado em técnica *zero-shot*, ou seja, não precisa de treinamento prévio para uma tarefa específica. De acordo com (HANS et al., 2024), testes indicam que essa ferramenta é capaz de detectar mais de 90% dos textos gerados por IA, além de apresentar uma taxa muito baixa de falsos positivos. Esses fatores aliados ao fato do Binoculars ser uma ferramenta de código aberto, o tornam uma ótima opção para esse estudo.

3.4.4.2 RADAR

Seguindo a ideia de detectores de código aberto, utilizou-se nesse experimento o RADAR, um framework de aprendizado adversarial projetado para treinar um detector robusto usando parafraseador e detector. Neste método, o parafraseador busca gerar conteúdos mais realistas possíveis enquanto o detector se aprimora a partir destes dados.

Para automatização, foi utilizado o código base do modelo proposto em (HU; CHEN; HO, 2023) com adaptações para detectar os textos gerados. O modelo utilizado no algoritmo é o TrustSafeAI/RADAR-Vicuna-7B, projetado para distinguir textos humanos e textos sintéticos.

3.4.4.3 Roberta Large Openai Detector

O RoBERTa é uma versão otimizada do modelo BERT (DEVLIN et al., 2019), a qual foi refinada e apresentou resultados melhores em tarefas de linguagem natural em relação ao seu modelo base. Já o *RoBERTa Large*, é um refinamento do RoBERTa contando com mais parâmetros e mais dados de treinamento.

Nesse experimento, foi utilizado o roberta-large-openai-detector (COMMUNITY, 2024), um refinamento do modelo RoBERTa Large ⁷. Essa implementação, que é o refinamento de outro refinamento, traz uma versão do RoBERTa Large com mais parâmetros e treinado com quantidades de dados maiores, utilizando saídas de texto do modelo GPT-2 1.5B, que possui 1.5 bilhões de parâmetros com o objetivo de identificar textos gerados por modelos de linguagem. O modelo foi treinado com exemplos de textos do WebText (mesmo conjunto de dados usado para treinar o GPT-2) e exemplos gerados pelo próprio GPT-2.

⁷ <https://huggingface.co/facebookAI/roberta-large>

3.4.4.4 Sapling

O Sapling é um conjunto de ferramentas (API/SDK) que ajuda a integração de funcionalidades baseadas em modelos de linguagem. Dentre essas funcionalidades, está seu detector de texto gerado por IA, o qual utiliza um modelo de aprendizado de máquina semelhante aos modelos geradores de texto. No entanto, ele gera a probabilidade de cada token ter sido gerado por IA.

3.4.4.5 DistilBERT

O DistilBert (SANH et al., 2020) é uma versão compacta e otimizada do modelo BERT, sendo 40% menor que o modelo original.

Em (HOWARD; RUDER, 2018), é realizado o *fine-tuning*, um processo para refinamento de modelos com o objetivo de o aprimorar para tarefas específicas. Nesse intuito, foi utilizado o modelo *DistilBERT base model (uncased)*, disponibilizado no HuggingFace, o qual foi treinado com uma base de dados presente no trabalho (KING et al., 2023). O trabalho propõe uma competição para criar um modelo que consiga detectar se uma redação foi feita por um estudante ou por uma IA.

3.4.4.6 HuggingFace Detector

O AI-Content-Detector, desenvolvido pelo usuário PirateXX na plataforma HuggingFace ⁸, utiliza o modelo RoBERTa Large, uma versão aprimorada do BERT, para classificar textos como originados por humanos ou por IA. A ferramenta aplica a técnica Masked Language Modeling (MLM), na qual palavras são mascaradas e o modelo faz previsões com base no contexto. Após o processamento, uma camada de classificação binária determina a origem do texto, levando em consideração as probabilidades dos tokens.

3.4.4.7 GPTZero e ZeroGPT

Como alternativas comerciais para a detecção de texto gerado por IA, foram escolhidas as ferramentas GPTZero e ZeroGPT, duas ferramentas amplamente citadas na literatura, contando com características semelhantes.

O ZeroGPT (ZEROGPT, 2024) é uma marca de serviços online que conta com diversos serviços, entre eles, o detector de texto gerado por IA, especializado em detecções para GPT-4. Diferentemente dos detectores citados até agora, este não conta com um código aberto ou uma API gratuita, porém permite sua utilização por meio de sua interface web conforme Figura 11.

Semelhantemente ao ZeroGPT, o GPTZero é um serviço que oferece várias funcionalidades, incluindo o detector de textos sintéticos, o qual incorpora as pesquisas mais recentes na detecção de GPT4, Google-Gemini, LLaMa e novos modelos de IA. Devido a

⁸ <https://huggingface.co/PirateXX/AI-Content-Detector>



Figura 11 – Interface web do ZeroGPT.

não disponibilidade de uma API gratuita, para esse experimento também foi utilizada a versão web, conforme Figura 12.

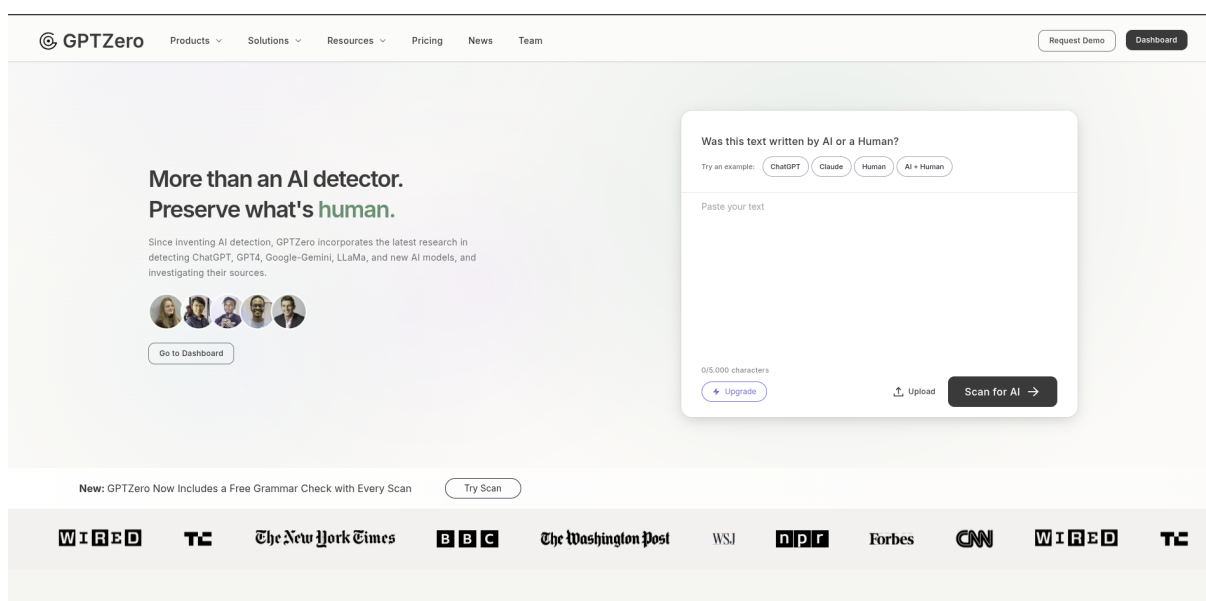


Figura 12 – Interface web do GPTZero.

Como citado anteriormente, as duas ferramentas possuem características semelhantes. Para exemplificar melhor isso, a Tabela 6 demonstra aspectos que tornam esses serviços semelhantes.

Característica	Descrição	Impacto e Aplicações
Detecção de Texto de IA	Ambos identificam se um texto foi gerado por IA como ChatGPT e GPT-4.	Uso acadêmico, prevenção de plágio e integridade textual.
Suporte a Múltiplos Modelos	Detectam textos de diferentes modelos, como GPT, Gemini e LLaMa.	Abrangência maior na análise de conteúdos gerados por IA.
Aplicação em Educação	Utilizados para verificar autoria de trabalhos acadêmicos.	Professores podem validar a originalidade de redações e ensaios.
Algoritmos Baseados em IA	Utilizam machine learning para identificar padrões textuais.	Modelos treinados continuamente para maior precisão.
Interface WEB	Ferramentas disponíveis via web para uso imediato.	Facilidade de acesso para estudantes, professores e empresas.
Planos Pagos	Oferecem versões gratuitas limitadas e planos pagos para uso avançado.	Mais verificações diárias, maior precisão e uso de API's

Tabela 6 – Semelhanças entre os detectores GPTZero e ZeroGPT. Adaptado de informações disponíveis nos sites próprios de cada ferramenta.

Apesar das semelhanças, a utilização dessas ferramentas demonstra algumas características distintas, sendo elas:

- ❑ No ZeroGPT, o texto a ser analisado precisa ter no mínimo 250 caracteres. Como neste estudo o foco são comentários curtos, os textos foram preenchidos com caracteres vazios ao final, o que pode ter afetado a detecção.
- ❑ Para o GPTZero, foi necessária a utilização do período de demonstração gratuita, devido a limitação na quantidade de análises.

Dessa maneira, todos os comentários gerados durante o experimento, foram inseridos manualmente nas ferramentas de detecção e seus resultados foram registrados no mesmo formato dos valores retornados pelos modelos automatizados.

3.4.4.8 Detecção Personalizada de Comentários

Da mesma forma que foi feita para a funcionalidade de geração de comentários, foi desenvolvida uma *feature* variante. Esta funcionalidade permite que o usuário entre com uma frase própria para a detecção, conforme Figura 13.

Diferentemente da funcionalidade original, esta, além de permitir a entrada de frases diretamente pela interface sem necessidade de textos pré-definidos, também permite que o usuário gere o resultado de detecção para todos os detectores de uma só vez ao não marcar nenhum LLM via interface.

Figura 13 – Tela de detecção personalizada de comentários.

3.5 Geração dos Resultados

A última etapa do desenvolvimento da aplicação foi a automatização de exibição das taxas de sucesso na geração e detecção dos comentários para fins comparativos.

No intuito de demonstrar visualmente essa comparação de resultados, de maneira que seja fácil visualizar o nível de sucesso de cada ferramenta, foi desenvolvida uma funcionalidade responsável por carregar os arquivos gerados na Seção 3.4.4, onde os detectores salvam as informações em arquivos JSON no formato demonstrado na Listagem 3.3.

Onde:

- ❑ **llm**: Modelo de linguagem responsável por gerar o comentário em questão;
- ❑ **comentario**: Comentário gerado a partir das informações passadas para o llm;
- ❑ **detector**: Detector responsável pela detecção do comentário;
- ❑ **prob_humano**: Probabilidade do comentário ter sido gerado por um humano segundo o detector;
- ❑ **prob_IA**: Probabilidade do comentário ter sido gerado por um LLM segundo o detector.

Com os arquivos carregados, os dados são agrupados por meio de funções, as quais recebem a lista dos LLMs utilizados e também o diretório contendo os resultados em arquivos JSON organizados por detector em subdiretórios específicos. A partir disso, foram executados os seguintes passos para calcular a média da efetividade de cada ferramenta:

1. Cada diretório de resultados é percorrido e as informações relevantes são extraídas dos arquivos JSON.
2. Um dicionário é criado para armazenar os dados extraídos no formato demonstrado na Listagem 3.4.

```
[
  {
    "detector": "",
    "llm": "",
    "prob_humano": ,
    "prob_IA": ,
    "tema": "",
    "comentário": "",
    "acerto":
  }
]
```

Listagem 3.4 – Formato do dicionário contendo as efetividades das ferramentas.

3. Em seguida, os resultados de cada detector são agrupados. Todos os comentários associados a um mesmo detector ou LLM têm seus valores somados, e a média aritmética simples é aplicada para calcular a efetividade geral de cada ferramenta. Essa estratégia é usada tanto para avaliar os modelos de geração quanto os detectores. Para os gráficos com comparação por tema, os dados são agrupados por tema com o auxílio de funções específicas.

Para automatizar a exibição dos dados de forma comparativa, foram desenvolvidas as funcionalidades de exibição em formato de gráficos e tabela:

- ❑ **Tabela:** Buscando exibir todas as informações referentes às análises realizadas nos passos anteriores, foi utilizada a estratégia de exibição dos dados em tabela. Para isso, utilizou-se o *plugin DataTables* do *Jquery*, o qual possibilita de forma nativa, a busca de informações, conforme Figura 14, a qual mostra informações referentes a um comentário específico filtrado.
- ❑ **Gráficos:** Para demonstrar a efetividade de cada LLM na geração do texto, e também dos detectores na análise dos mesmos, possibilitando que seja feita uma análise visual comparativa, foi utilizada a estratégia dos gráficos de barra, conforme Figuras 15 e 16. Essas informações são coletadas a partir de funções específicas que recebem os dados retornados e realizam os cálculos de média referentes a cada gráfico.

10
Itens por página

Buscar

Detector	LLM Gerador	Tema	%Humano	%IA	Comentário
Binoculars	Cohere	Dolar	12.63%	87.37%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
DistBERT	Cohere	Dolar	100.00%	0.00%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
GPTZero	Cohere	Dolar	0.00%	100.00%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
HuggingFace	Cohere	Dolar	99.54%	0.46%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
Radar	Cohere	Dolar	97.10%	2.90%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
Roberta	Cohere	Dolar	0.06%	99.94%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
Sapling	Cohere	Dolar	0.00%	100.00%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.
ZeroGPT	Cohere	Dolar	100.00%	0.00%	É isso aí! A alta do dólar impacta todos nós, principalmente os mais pobres. O governo precisa ser responsável e pensar nas consequências das suas ações. Reduzir impostos é ótimo, mas não quando a conta não fecha e quem paga o pato é a população. O resultado é inflação, menos crescimento e um país mais pobre.

Primeira

Figura 14 – Tabela de resultados no formato de tabela (filtrados a partir de um comentário específico).

Efetividade geral dos detectores

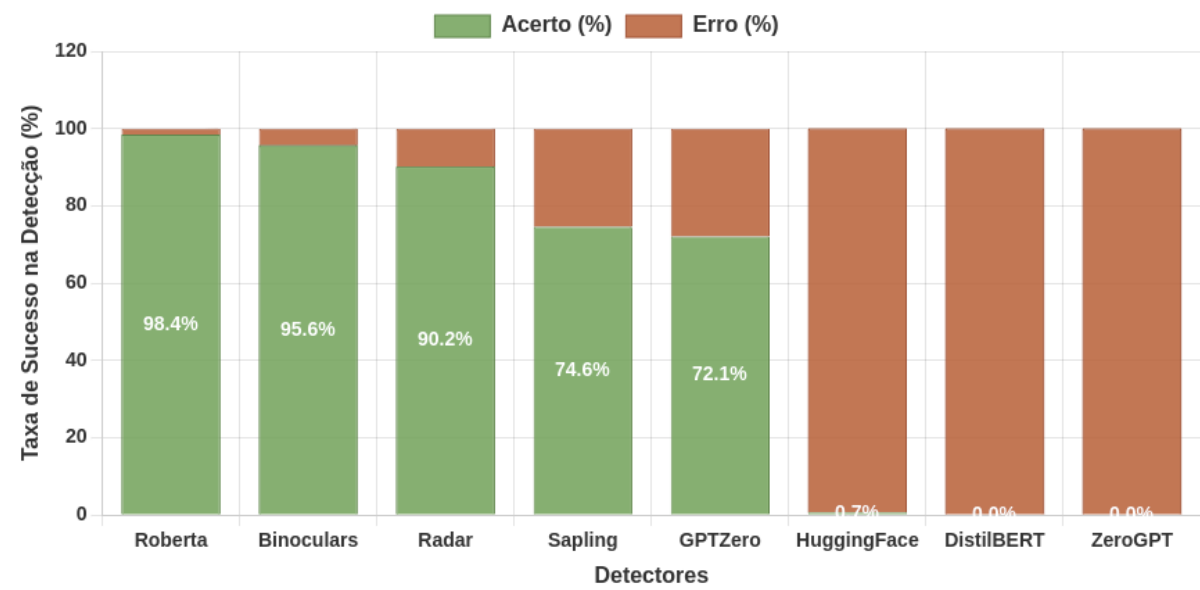


Figura 15 – Comparação da efetividade dos detectores por meio do gráfico.

A partir dessa funcionalidade, uma variante foi implementada, possibilitando que o usuário carregue um arquivo txt contendo comentários próprios e obtenha os resultados de detecção para todos os detectores do experimento. A partir disso, os resultados são demonstrados via gráfico comparativo, semelhante ao primeiro gráfico exibido. A interface responsável por receber esses dados é demonstrada na Figura 17.

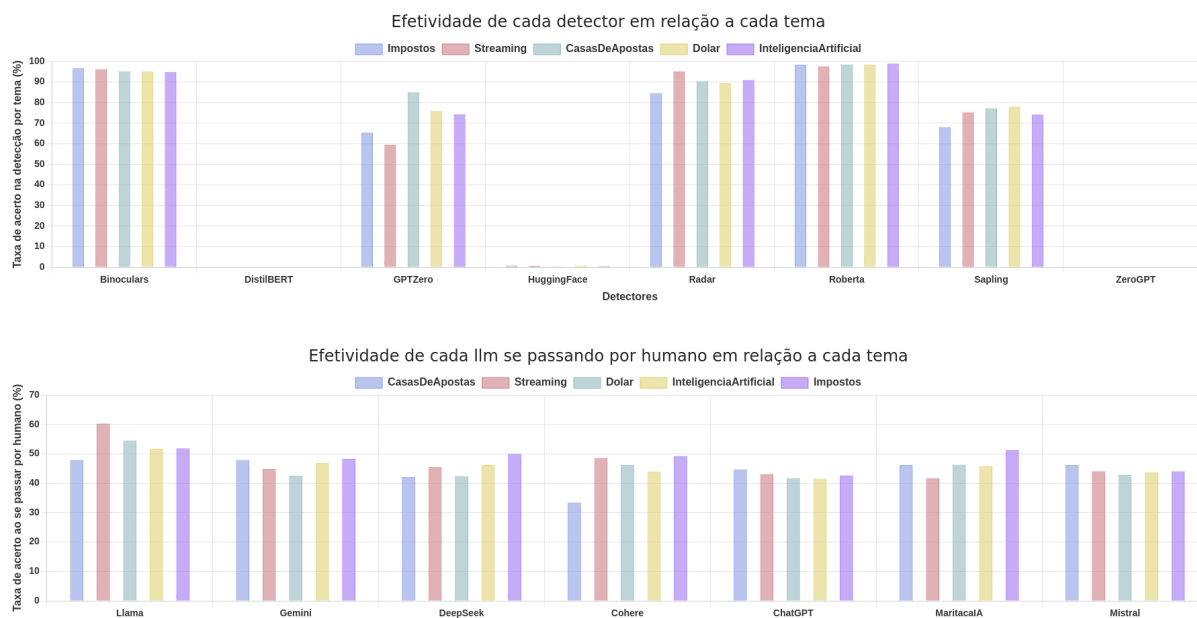


Figura 16 – Comparação da efetividade dos detectores/LLMs de acordo com o tema dos comentários por meio de gráficos.

Estatísticas com base em comentários próprios

Carregue um arquivo TXT com seus comentários (**separe os comentarios por linhas**):

Escolher arquivo
Nenhum arquivo escolhido

Enviar

Figura 17 – Gerador de gráficos personalizados a partir de comentários próprios enviados para a aplicação.

Experimentos e Análise dos Resultados

Neste capítulo, serão apresentados os procedimentos utilizados no experimento com a finalidade de evidenciar o potencial dos modelos de linguagem na simulação de interações humanas e a eficiência dos detectores de textos gerados por IA. Serão discutidos os resultados obtidos, destacando comparações de eficácia entre as ferramentas, as singularidades que podem ter afetado os resultados, e a análise crítica da metodologia proposta.

4.1 Método para a Avaliação

Esta seção descreve os métodos utilizados para validar a hipótese do experimento, incluindo:

- ❑ Aplicação de métricas para análise linguística e complexidade textual nos textos gerados por IA com e sem utilização de personas, além de comentários reais extraídos do X;
- ❑ Influência das personas nos resultados das métricas e na taxa de sucesso dos modelos de linguagem ao tentarem simular textos humanos, os quais foram analisados por diversos detectores da literatura;
- ❑ Comparação visual da efetividade dos LLMs por meio de gráficos;
- ❑ Avaliação dos detectores ao distinguirem textos humanos de textos gerados por IA;
- ❑ Correlação geral entre informações presentes na literatura e como este experimento corroborou ou discordou delas a partir dos resultados gerados, deixando claro as limitações do mesmo.

4.2 Avaliação dos LLMs de Geração de Texto

Nesta subseção, serão apresentados os critérios e métricas utilizadas para avaliar os modelos de linguagem quanto a sua capacidade de gerar comentários simulando interações humanas. Para isso, o foco será em métricas quantitativas, comparações com textos reais do X e comparação geral dos resultados de efetividade de cada LLM ao gerar comentários para publicações reais.

4.2.1 Métricas dos Textos de IA vs Textos Originais do X

Textos de redes sociais possuem características específicas em relação a tamanho de texto, utilização de emojis, informalidade, entre outras características. Para avaliar como os modelos de linguagem se comportaram em relação a essas características, uma série de métricas foram aplicadas aos comentários gerados, a fim de extrair dados relacionados aos aspectos dos textos em comparação com comentários reais do X. Para esse experimento foram utilizados 7 modelos de linguagem para gerar 6 comentários para cada um dos 5 temas, ou seja, um total de 35 arquivos, cada um com 6 comentários, conforme Tabela 7.

LLM	Temas	Quant. Comentários por Tema
ChatGPT	Dólar, Casas de Apostas, Streaming, Impostos, Inteligência Artificial	6
Cohere		6
DeepSeek		6
Gemini		6
LLaMA		6
MaritacaIA		6
Mistral		6

Tabela 7 – Combinação de comentários por tema e LLM.

Com base nesses arquivos, são analisadas uma série de métricas relacionadas aos textos gerados, conforme Tabela 8 e Tabela 9.

LLM	Tempo Médio (s)	Observação
Cohere	1.1689	-
Mistral	2.9870	Limite de 3 requisições por minuto
MaritacaIA	3.6123	-
Deepseek	8.5984	-
ChatGPT	1.3084	Limite de 3 requisições por minuto
Gemini	1.4698	-
LLaMA	5.1062	-

Tabela 8 – Tempo médio gasto para gerar cada comentário de acordo com cada LLM.

LLM	Tema	Tamanho Médio	Formalidade	Gíria (Média da %)	Emoji (Média da %)	Flesch-Kincaid	Riqueza Lexical	Freq. StopWords (%)	Perplexidade
ChatGPT	Dólar	30.33	Informal	66.67%	100.00%	50.79	0.992	44.20%	259.60
ChatGPT	CasasdeApostas	32.17	Formal	50.00%	100.00%	32.66	0.985	37.90%	261.55
ChatGPT	Impostos	28.33	Formal	66.67%	100.00%	48.58	0.991	37.60%	330.62
ChatGPT	Streaming	25.83	Formal	66.67%	100.00%	59.94	0.979	41.30%	424.45
ChatGPT	IA	31.17	Informal	66.67%	100.00%	47.43	0.980	40.40%	289.09
Cohere	Impostos	40.00	Formal	66.67%	0.00%	21.34	0.938	39.50%	213.71
Cohere	Streaming	33.00	Formal	66.67%	0.00%	42.93	0.994	41.70%	314.21
Cohere	CasasdeApostas	40.33	Formal	66.67%	0.00%	32.40	0.985	43.90%	343.52
Cohere	Dólar	43.00	Formal	66.67%	0.00%	39.03	0.994	43.20%	194.13
Cohere	IA	35.33	Formal	66.67%	0.00%	28.50	0.976	40.60%	416.52
DeepSeek	Impostos	67.50	Formal	100.00%	16.67%	30.82	0.917	42.70%	211.38
DeepSeek	Streaming	36.83	Formal	83.33%	0.00%	43.03	0.962	40.20%	520.86
DeepSeek	Dólar	54.00	Formal	83.33%	0.00%	27.49	0.965	40.70%	219.73
DeepSeek	CasasdeApostas	52.50	Formal	33.33%	0.00%	31.95	0.971	43.20%	268.27
DeepSeek	IA	57.00	Formal	83.33%	0.00%	13.11	0.969	41.60%	207.15
Gemini	Impostos	30.33	Formal	66.67%	100.00%	50.62	0.990	33.70%	460.16
Gemini	IA	30.17	Informal	83.33%	100.00%	57.83	0.958	36.20%	473.96
Gemini	Dólar	35.67	Formal	66.67%	100.00%	68.74	0.965	32.50%	378.38
Gemini	Streaming	31.50	Formal	100.00%	100.00%	51.12	0.975	28.70%	560.29
Gemini	CasasdeApostas	37.00	Informal	83.33%	83.33%	47.50	0.967	35.80%	473.82
LLaMA	CasasdeApostas	29.33	Formal	33.33%	0.00%	36.68	0.990	50.90%	274.74
LLaMA	IA	33.33	Formal	66.67%	0.00%	38.00	0.986	49.90%	317.33
LLaMA	Impostos	39.17	Formal	50.00%	0.00%	22.01	0.994	46.80%	254.63
LLaMA	Streaming	31.00	Formal	83.33%	0.00%	37.83	1.000	48.80%	427.89
LLaMA	Dólar	25.00	Formal	0.00%	0.00%	37.78	1.000	51.50%	425.98
MaritacaIA	CasasdeApostas	49.17	Informal	83.33%	16.67%	20.26	0.979	40.40%	266.10
MaritacaIA	Dólar	41.17	Formal	83.33%	50.00%	43.55	0.956	38.50%	264.46
MaritacaIA	IA	44.67	Informal	100.00%	16.67%	41.66	0.953	40.60%	255.84
MaritacaIA	Streaming	43.33	Informal	100.00%	33.33%	38.17	0.965	34.10%	337.30
MaritacaIA	Impostos	45.83	Informal	83.33%	50.00%	33.30	0.973	33.40%	333.07
Mistral	Streaming	19.83	Formal	66.67%	66.67%	57.64	0.990	34.10%	407.80
Mistral	IA	29.50	Formal	33.33%	0.00%	45.26	0.970	41.90%	404.08
Mistral	CasasdeApostas	28.67	Formal	16.67%	0.00%	32.72	0.973	37.30%	316.18
Mistral	Impostos	20.83	Formal	66.67%	50.00%	44.98	0.966	36.30%	286.85
Mistral	Dólar	28.17	Formal	50.00%	16.67%	39.83	0.991	44.40%	244.80

Tabela 9 – Métricas referentes aos textos dos comentários gerados pelos LLMs.

Já as Tabelas 10 e 11 mostram as mesmas métricas (à exceção do tempo de geração por se tratar de comentários já existentes) aplicadas a comentários reais do X, os quais foram extraídos de postagens utilizados na seção de geração de comentários como base para geração dos textos sintéticos.

Tema	<i>Post</i>	Tamanho Médio	Formalidade
CasasdeApostas	Link	11.97	Formal
Dólar	Link	11.96	Formal
Impostos	Link	34.15	Formal
IA	Link	12.30	Formal
Streaming	Link	10.65	Formal

Tabela 10 – Tamanho médio e formalidade dos comentários em posts do X.

Tema	Gíria (Média %)	Emoji (Média %)	Flesch-Kincaid	Riqueza Lexical	StopWords (%)	Perplexidade
CasasdeApostas	19.01%	15.49%	43.98	0.95	31.37%	1422.53
Dólar	17.00%	19.00%	69.53	0.93	21.74%	981.65
Impostos	36.70%	1.83%	11.08	0.96	42.28%	481.03
IA	17.48%	8.74%	32.09	0.97	31.91%	2959.14
Streaming	23.02%	10.07%	31.86	0.97	36.01%	5093.78

Tabela 11 – Métricas linguísticas dos comentários em posts do X.

Para os dados referentes aos comentários originais foram analisados:

- ❑ 142 comentários do tema Casas de Apostas;
- ❑ 100 comentários do tema Dólar;
- ❑ 109 comentários do tema Impostos;
- ❑ 103 comentários do tema Inteligência Artificial;
- ❑ 139 comentários do tema Streaming.

Realizando a análise das métricas, é possível observar algumas características relacionadas aos textos sintéticos e aos comentários originais do X, tais como o tamanho médio dos textos. Apesar dos modelos de linguagem terem sido instruídos a gerar comentários mais curtos, os modelos apresentaram textos mais extensos. O deepseek-coder-v2 por exemplo, chegou a gerar comentários com até 67 palavras, talvez por conta do propósito do modelo, que não é a geração de texto, e sim a geração de códigos. Por outro lado, conforme vimos na literatura, textos de redes sociais possuem tamanhos mais concisos, tendo apresentado uma média de 10 a 35 palavras.

Além disso, a aplicação das demais métricas permitiu destacar as seguintes observações:

❑ Formalidade (Uso de Gírias e Emojis)

A utilização de gírias e *emojis* variou bastante de acordo com cada LLM, porém em apenas um conjunto de dados (LLaMA para o tema Dólar), não foi utilizada nenhuma das gírias ou emojis pré-definidos para análise. Em contrapartida, o índice de utilização desses recursos nos textos originais é muito mais baixa em relação aos textos sintéticos gerados no experimento, tendo uma média de 17% a 23% na utilização de gírias e de 1,83% a 19% na utilização de emojis a partir de todos os comentários analisados. Embora a literatura aponte que textos em redes sociais costumam usar muitos emojis e gírias, os comentários reais do X analisados apresentaram baixa ocorrência desses elementos. Já os comentários gerados pelas LLMs usaram essas características com frequência, principalmente porque foram solicitadas nas instruções de geração. Ainda assim, os modelos LLaMA e Cohere não utilizaram emojis, o que os deixou mais próximos dos textos humanos reais nesse aspecto.

❑ Complexidade Linguística

Na análise realizada a partir dos resultados relacionados a Riqueza lexical, notou-se que tanto os comentários de IA, quanto os comentários originais do X apresentaram valores bem semelhantes, o que indica que ambos possuem boa variedade no vocabulário.

Os resultados obtidos a partir do algoritmo de Flesch-Kincaid apresentaram variações significativas conforme o tema abordado e, no caso dos textos gerados, também em função do LLM utilizado. A maior pontuação foi registrada no tema 'dólar', com comentários originais extraídos do X, enquanto a menor ocorreu no tema 'impostos', também com base em comentários reais da plataforma.

❑ Frequência de StopWords

Da mesma maneira que ocorreu nas métricas de complexidade, a frequência de *Stop Words* também apresentou-se variada, porém, nos comentários originais do X a frequência dessas palavras se manteve menor para todos os temas, mostrando que estes apresentaram uma comunicação mais focada.

❑ Perplexidade

Na análise de perplexidade observou-se que os textos gerados pelos modelos de linguagem apresentaram valores entre 194 e 424, o que indica variabilidade na previsibilidade do modelo. Já para os comentários sintéticos, a variação foi entre 481 e 5093, indicando uma maior imprevisibilidade nestes textos de acordo com o GPT-2.

4.2.2 Influência das Personas nas Métricas

Conforme é concluído no artigo (RADIVOJEVIC; CLARK; BRENNER, 2024), a persona por trás dos comentários pode ter mais influência na percepção humana do que propriamente o LLM que gera um texto. Visto isso, para analisar o quão diferente podem ser os textos gerados sem a utilização de personas pré-definidas, foram gerados novamente a mesma quantidade de comentários da primeira fase de testes, seguindo a estrutura de geração da Tabela 7, porém, sem utilizar as personas, apenas o *prompt* com o tema, o post, e a instrução base para comentários. Os resultados desse teste são apresentados na Tabela 12.

Correlacionando a Tabela 12 de métricas dos comentários sem personas, e a Tabela 9 com as personas, observa-se que nesse estudo essa alteração de paradigma na geração não obteve resultados muito diferentes, mantendo faixas relativamente semelhantes, porém alguns resultados podem ser destacados.

- ❑ O LLM MaritacaIA teve, em média, um aumento de 23.3 palavras no tamanho dos comentários gerados sem a utilização de personas.
- ❑ Os textos gerados sem utilização das personas apresentaram uma maior taxa de informalidade. Nos comentários da primeira etapa, 8 conjuntos de arquivos apresentaram maior grau de informalidade do que formalidade. Já nos arquivos gerados na segunda parte, ou seja, os originais do X, foram apresentados 14 conjuntos com maior informalidade.
- ❑ Ao aplicar o algoritmo de Flesh-Kindaic para avaliar a complexidade dos textos, houve um ligeiro aumento na dificuldade de leitura humana, principalmente nas frases do ChatGPT e do Cohere.
- ❑ Os comentários gerados sem as personas apresentaram um pequeno aumento na complexidade dos textos, estes analisados pelo modelo GPT-2. Apenas o modelo deepseek-coder-v2 manteve uma taxa de perplexidade maior em textos gerados com a utilização das personas.

LLM	Tema	Tamanho Médio	Formalidade	Gíria (Média da %)	Emoji (Média da %)	Complexidade de Leitura	Riqueza Lexical	Freq. StopWords	Perplexidade
ChatGPT	Dólar	29.5	Informal	66.67%	100.00%	54.898483	1.000000	37.2526%	243.168117
ChatGPT	CasasdeApostas	30.0	Informal	50.00%	100.00%	60.930857	0.988889	42.0013%	301.399567
ChatGPT	Impostos	29.333333	Formal	83.33%	83.33%	66.119875	0.981643	40.6861%	314.123620
ChatGPT	Streaming	25.666667	Formal	66.67%	100.00%	65.024774	0.958527	38.0299%	491.744333
ChatGPT	IA	25.166667	Informal	66.67%	100.00%	59.318441	0.990196	41.7399%	429.993376
Cohere	Impostos	41.333333	Formal	83.33%	0.00%	35.791833	0.962478	38.0535%	248.183175
Cohere	Streaming	35.833333	Formal	66.67%	0.00%	49.056107	0.938953	39.0912%	379.527473
Cohere	CasasdeApostas	40.333333	Informal	66.67%	0.00%	33.199704	0.966791	34.6761%	293.241011
Cohere	Dólar	35.666667	Formal	66.67%	0.00%	56.776859	0.989247	43.7893%	501.472576
Cohere	IA	41.333333	Formal	66.67%	0.00%	25.492404	0.953914	38.5974%	246.770878
DeepSeek	Impostos	57.833333	Formal	83.33%	16.67%	41.453126	0.945672	40.7355%	292.655548
DeepSeek	Streaming	45.666667	Formal	66.67%	0.00%	28.794482	0.957900	33.9085%	411.241505
DeepSeek	Dólar	69.666667	Informal	66.67%	0.00%	34.251644	0.941406	44.3861%	271.747488
DeepSeek	CasasdeApostas	48.666667	Formal	16.67%	0.00%	28.264066	0.974626	38.6298%	365.918666
DeepSeek	IA	46.166667	Formal	66.67%	16.67%	-5.090132	0.969210	41.3043%	238.746832
Gemini	Impostos	36.0	Informal	66.67%	100.00%	64.255262	0.971980	32.9401%	327.722286
Gemini	IA	29.333333	Formal	83.33%	100.00%	61.054446	0.964006	33.1035%	460.798814
Gemini	Dólar	38.333333	Informal	66.67%	100.00%	65.615708	0.921627	33.3844%	289.014378
Gemini	Streaming	32.333333	Informal	100.00%	100.00%	64.409273	0.962337	34.6784%	474.093724
Gemini	CasasdeApostas	30.0	Informal	83.33%	83.33%	45.626818	0.961111	39.1019%	481.614511
LLaMA	CasasdeApostas	30.833333	Formal	50.00%	0.00%	34.518914	0.982407	47.9852%	323.647893
LLaMA	IA	23.833333	Formal	33.33%	0.00%	46.283303	1.000000	50.8852%	453.831162
LLaMA	Impostos	23.166667	Formal	33.33%	0.00%	42.120538	1.000000	51.3438%	473.863867
LLaMA	Streaming	18.833333	Formal	66.67%	0.00%	38.628588	1.000000	45.7177%	445.583428
LLaMA	Dólar	20.0	Formal	16.67%	0.00%	33.749917	1.000000	46.2357%	518.436368
MaritacaIA	CasasdeApostas	78.666667	Informal	100.00%	66.67%	36.416096	0.960173	37.6018%	262.596441
MaritacaIA	Dólar	69.666667	Informal	100.00%	0.00%	35.708308	0.944053	36.0437%	309.561944
MaritacaIA	IA	66.666667	Informal	100.00%	16.67%	35.839034	0.930402	40.0823%	241.277357
MaritacaIA	Streaming	62.0	Informal	100.00%	0.00%	36.868695	0.920423	35.0230%	326.763002
MaritacaIA	Impostos	73.666667	Informal	83.33%	50.00%	40.851266	0.945546	37.7134%	285.283558
Mistral	Streaming	22.333333	Formal	50.00%	50.00%	50.219756	0.972426	33.9162%	510.754722
Mistral	IA	27.5	Formal	50.00%	33.33%	45.026574	0.975914	41.1311%	316.685244
Mistral	CasasdeApostas	18.166667	Formal	33.33%	33.33%	43.656096	1.000000	36.4242%	540.692780
Mistral	Impostos	26.333333	Formal	83.33%	50.00%	40.398816	0.970794	30.7866%	401.065261
Mistral	Dólar	20.000000	Formal	50.00%	16.67%	55.641040	1.000000	44.3649%	447.358701

Tabela 12 – Métricas resultantes dos comentários gerados pelos LLMs sem utilização de personas.

4.2.3 Comparação Visual de Efetividade dos LLMs de Geração de Texto

Além das métricas em relação aos textos gerados pelos modelos de linguagem, também foram utilizadas estratégias visuais para comparar a efetividade dos modelos ao simularem interações humanas. Nas próximas subseções serão demonstrados estes resultados de acordo com critérios importantes para diferenciar a fonte dos textos.

4.2.3.1 Efetividade Geral dos Modelos de Linguagem

A Figura 18 demonstra a média de sucesso e fracasso de cada LLM após análises de todos os detectores. Os modelos tiveram desempenhos semelhantes tendo uma variação de sucesso entre 42,77% e 53,27%. O modelo que teve o melhor desempenho foi o LLaMA (LLaMA3.2), enquanto o pior foi o ChatGPT (gpt-4o-mini).

Efetividade geral dos LLMs se passando por humano

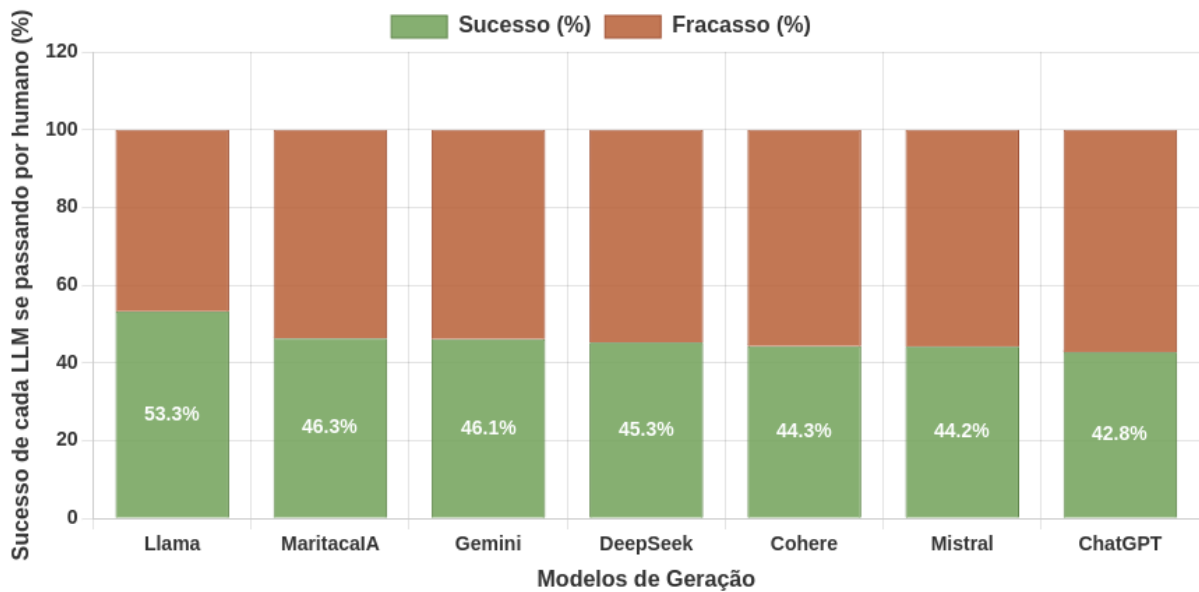


Figura 18 – Comparação de efetividade geral dos LLMs na geração de comentários sintéticos contra os detectores.

Especificando um pouco mais essa efetividade de cada algoritmo, a Figura 19 compara a eficácia de cada LLM de acordo com cada tema. Notou-se que a variabilidade na eficiência não foi tão comprometida pelo tema, porém dois resultados se destacaram.

1. O LLaMA apresentou resultado ligeiramente mais satisfatório nos comentários do tema 'Streaming'.
2. O Cohere apresentou desempenho ligeiramente menos satisfatório nos comentários do tema 'Casas de Apostas'.

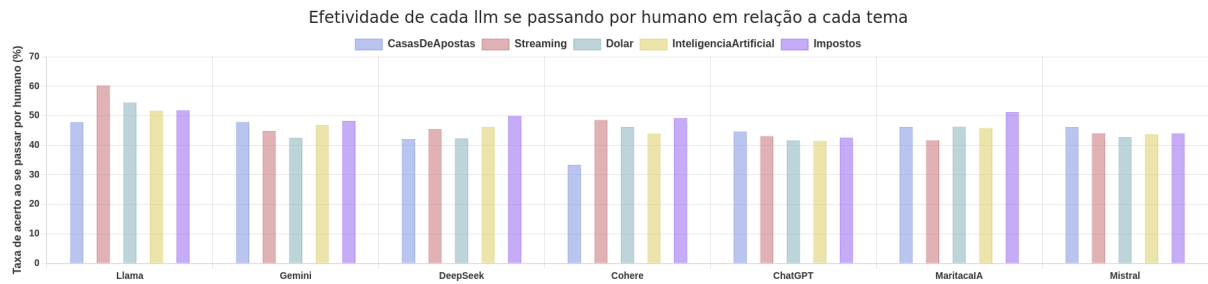


Figura 19 – Comparação de efetividade dos LLMs em relação a cada tema.

Com base nos resultados obtidos na Subseção 4.3, observou-se uma taxa de acerto muito baixa em relação à eficácia dos detectores *HuggingFace Detector*, *DistilBERT* e *ZeroGPT*. Como essa eficácia está sendo calculada por meio de médias feitas com base em todos os detectores, foram utilizadas nessa nova etapa, apenas resultados das ferramentas mais expressivas, realizando uma nova comparação, conforme Figura 20. Nesta, observa-se que a eficácia dos modelos cai drasticamente. Antes com média de sucesso de 53,27%, agora o modelo LLaMA continua apresentando o melhor desempenho, porém apresentando 25,52% de sucesso, enquanto o pior ranqueado, o ChatGPT apresentou apenas 8,58%.

Efetividade geral dos LLMs se passando por humano

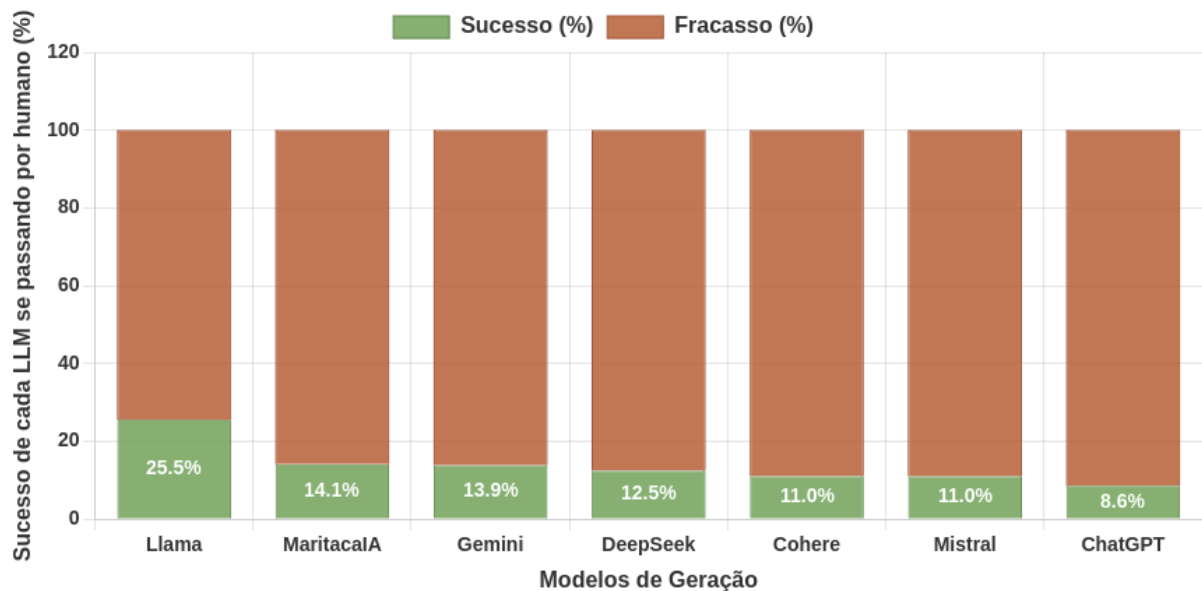


Figura 20 – Comparação de efetividade dos LLMs utilizando apenas os detectores mais expressivos conforme resultados analisados na Seção 4.3.

4.2.3.2 Efetividade na Geração de Texto sem Utilização de Personas

De forma semelhante à análise das métricas, os comentários gerados sem personas foram avaliados quanto à sua efetividade em simular linguagem humana utilizando os oito detectores mencionados na Subseção 3.4.4. As Figuras 21 e 22 apresentam as médias de sucesso de cada LLM, tanto de forma geral quanto por tema. Os resultados indicam que essa variação não gerou mudanças significativas na detecção.

Efetividade geral dos LLMs se passando por humano

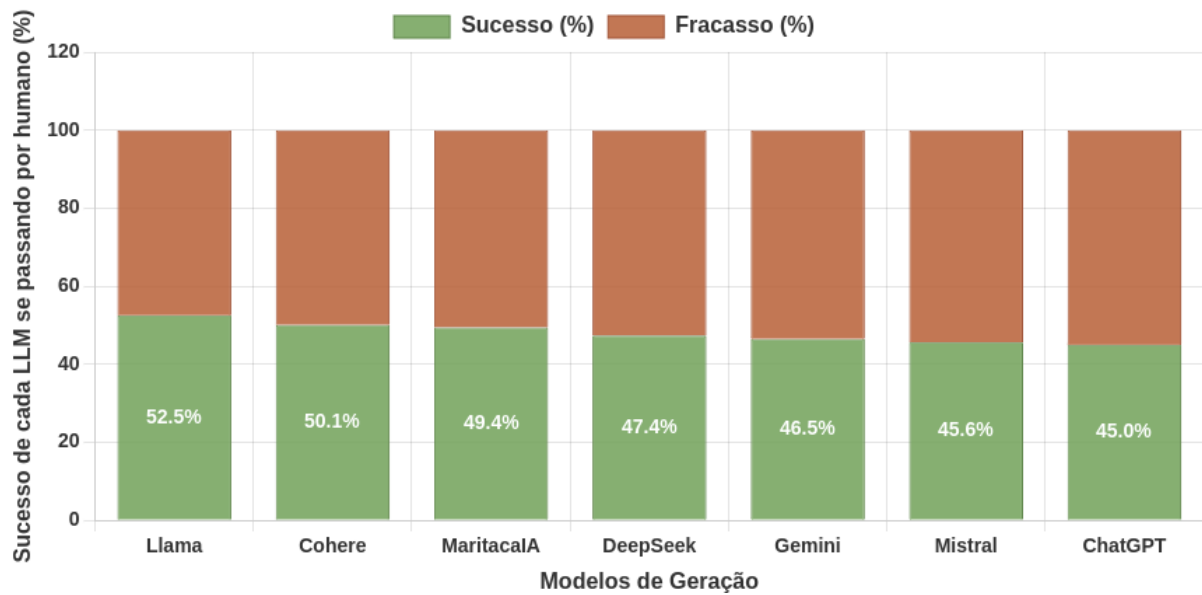


Figura 21 – Efetividade dos LLMs na geração de texto sem utilização de personas.

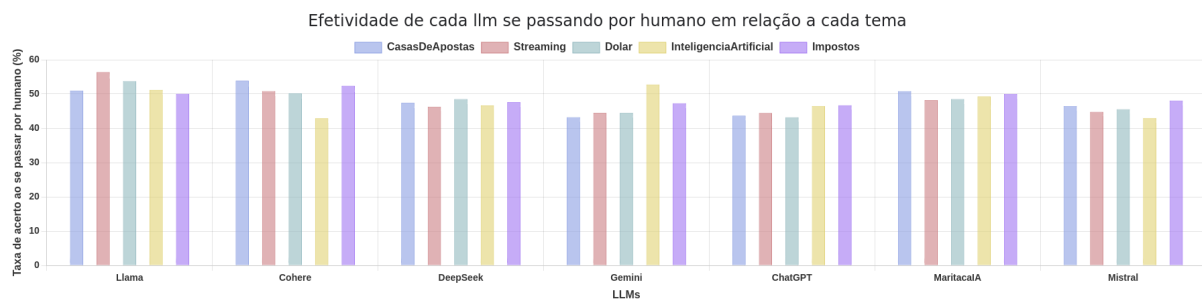


Figura 22 – Efetividade dos LLMs na geração de texto sem utilização de personas na instrução (em relação a cada tema).

Também seguindo os mesmos critérios utilizados nas métricas, os resultados dos três detectores de menor expressão (HuggingFace Detector, DistilBERT e ZeroGPT) foram ocultados e aplicou-se a média apenas nos detectores que obtiveram bons resultados. Assim, é possível observar que os resultados continuaram semelhantes aos da Seção 4.2.3.1,

onde as personas foram pré-definidas para os modelos de geração textual. A Figura 23 demonstra os resultados referentes a essa efetividade dos LLMs, onde a maior taxa de sucesso foi do LLaMA (LLaMA3.2) com 24,81%, enquanto a pior continuou sendo do ChatGPT com 12,04%.

Efetividade geral dos LLMs se passando por humano

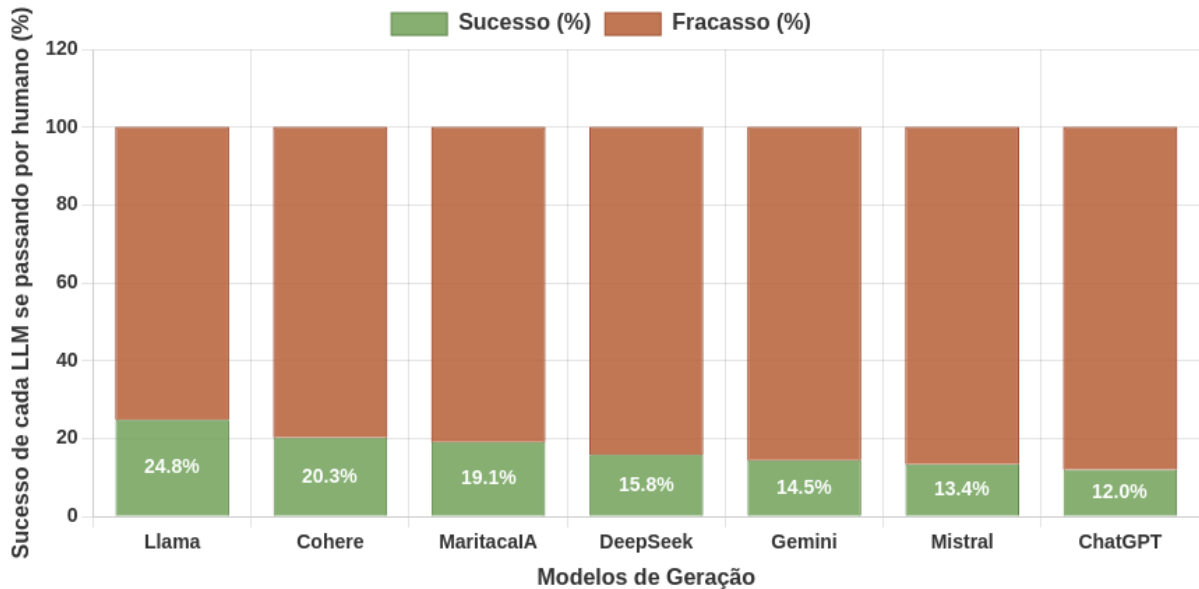


Figura 23 – Efetividade dos LLMs na geração de texto sem utilização de personas na instrução (detectores mais expressivos).

4.3 Avaliação dos Detectores

Nesta subseção, será abordada a avaliação dos detectores de textos gerados por IA, focando em sua capacidade de identificar textos gerados por modelos de linguagem além de abordar brevemente a comparação com textos originais do X. Paralelamente a isso, será discutido se a utilização de uma persona pré-definida na geração dos comentários influenciou na análise e resultado dos detectores. Por fim, será apresentada uma análise comparativa entre o modelo que apresentou o melhor resultado geral, contra o modelo que teve um desempenho pouco significativo.

4.3.1 Tempos de Detecção

Foram calculadas as médias de tempo para realização da detecção de cada comentário de acordo com cada ferramenta. Conforme Tabela 13, o modelo Binoculars apresentou um tempo bem maior em relação aos demais, sendo o único detector utilizado por meio do modelo baixado, totalizando aproximadamente 37GB. Os demais detectores apresentaram

deteções rápidas por meio das APIs. Vale ressaltar que, para os detectores GPTZero e ZeroGPT, os valores não são calculados de forma automática já que utilizou-se a interface das aplicações para a entrada de dados, acarretando em uma aferição de tempo enviesada.

LLM	Tempo Médio (s)	Observação
Binoculars	42.3447	-
Sapling	1.715836	-
ZeroGPT	-	Acessado via interface.
GPTZero	-	Acessado via interface.
DistilBERT	0.031112	-
RADAR	0.190618	-
RoBERTa	0.1788	-
HuggingFace	0.059584	-

Tabela 13 – Tempo médio gasto para detecção de cada comentário.

4.3.2 Comparação Geral dos Detectores

A Figura 24 mostra que os detectores roBERTa, Binoculars e Radar tiveram alto desempenho (90,2% a 98,4%), enquanto HuggingFace Detector, DistilBERT e ZeroGPT tiveram desempenho muito baixo (0% a 0,7%).

Efetividade geral dos detectores

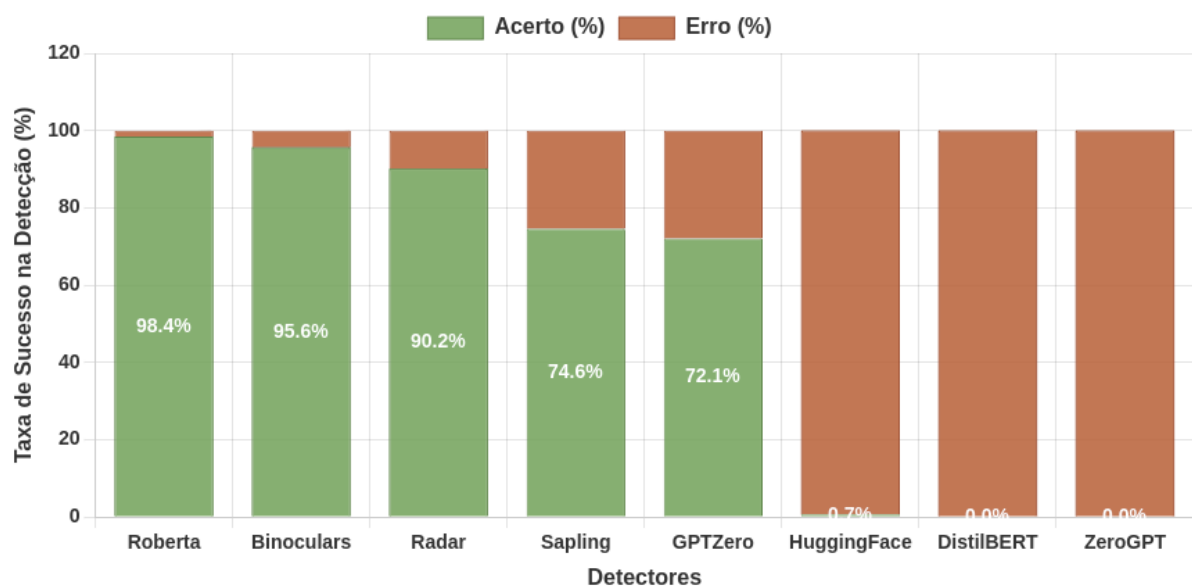


Figura 24 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs.

A Figura 25 apresenta os resultados classificados de acordo com o tema dos comentários.

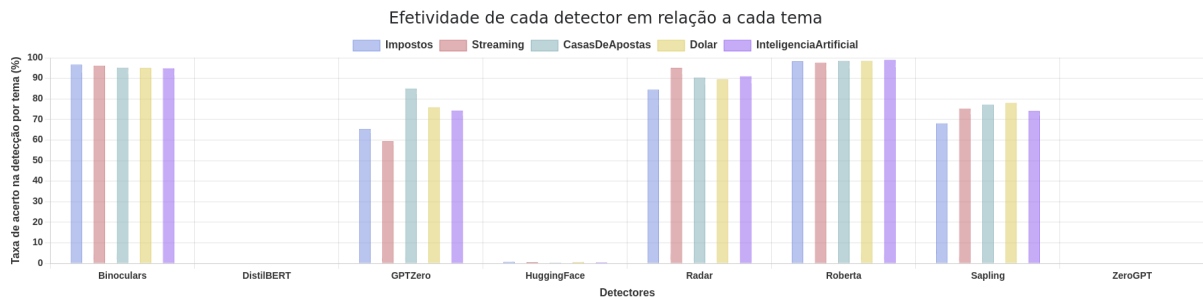


Figura 25 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs divididos pelo tema.

4.3.3 Comparação Geral dos Detectores sem Personas

Conforme análise realizada na Subseção 4.2.1, a geração de texto utilizando ou não personas pré-definidas não alterou significativamente as métricas dos comentários. Para analisar como os detectores se comportam com estes novos textos, estes foram processados em uma nova bateria de testes, novamente utilizando todos os oito detectores deste experimento. Os resultados são apresentados na Figura 26. Pode-se observar que a utilização de personas, assim como nos resultados de eficiência dos modelos de geração textual, também não alterou significativamente as taxas de detecção para cada uma das ferramentas, porém vale destacar que por menos relevantes que tenham sido essas diferenças, todos os detectores apresentaram uma taxa de acerto menor nos textos gerados sem utilização de personas, exceto o HuggingFace Detector, que melhorou sua detecção de 0,7% para 1,1%.

Efetividade geral dos detectores

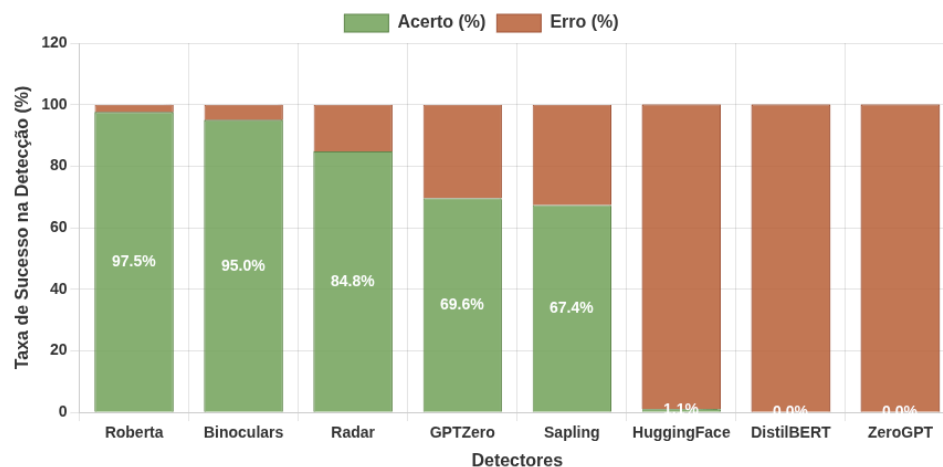


Figura 26 – Média de assertividade dos detectores na identificação dos comentários gerados pelos LLMs sem a utilização de personas pré-definidas.

4.3.4 Comparação entre o Melhor e o Pior Detector

Para comparar a taxa de possíveis falsos positivos do melhor e do pior detector, supomos que todos os comentários que extraímos dos *posts* reais do X são genuinamente gerados apenas por humanos sem a utilização de modelos de linguagem. A partir destes comentários, foram selecionados dois detectores para essa análise, o RoBERTa, sendo o detector com maior taxa de assertividade em textos gerados por IA, e o HuggingFace Detector, o qual apresentou resultados ruins, próximos a 100% de erro em textos de IA, mas que ainda sim obteve um resultado válido, diferente das ferramentas ZeroGPT e DistilBERT que, essas sim, apresentaram 100% de falha em suas análises.

A Figura 27 ilustra o desempenho das duas ferramentas. Diferentemente da abordagem utilizada durante esse experimento, para essa etapa, ao invés de calcular a média de efetividade dos detectores, utilizou-se uma classificação binária, onde resultados maiores ou iguais a 50% indicam textos gerados por IA, e valores abaixo de 50% indicam comentários analisados como humanos. Supondo que a maioria dos comentários dos *posts* sejam de humanos que não tenham utilizado IA para gerar os textos, o RoBERTa apresenta uma taxa muito alta de falsos positivos, detectando muitos textos humanos como textos gerados por IA.

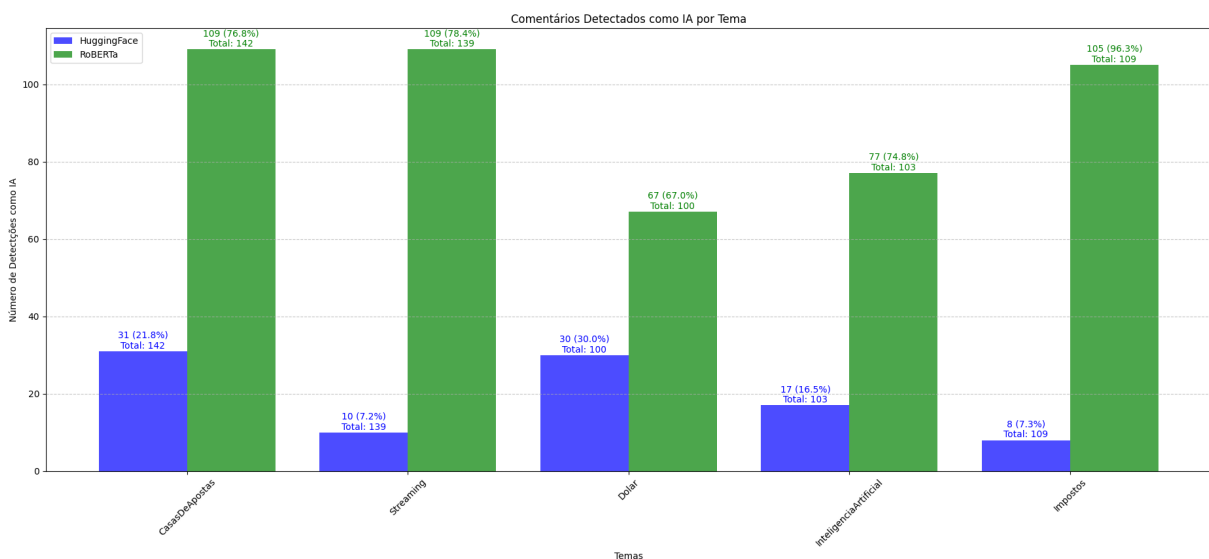


Figura 27 – Detecção do RoBERTa e HuggingFace Detector a partir de comentários reais extraídos do X.

4.4 Limitações

Durante os experimentos, diversas limitações moldaram a maneira com que as escolhas foram feitas e posteriormente implementadas. Dentre os aspectos que delimitaram o escopo das análises estão:

1. Limitações de Hardware

Devido à falta de um poder de processamento que gerasse os resultados de forma satisfatória no tempo necessário para conclusão dos experimentos, houve uma falta de aprofundamento em técnicas de treinamento de modelos de linguagem. Diversos outros modelos e datasets foram estudados a fim de desenvolver um modelo eficaz na detecção de texto gerado por IA, porém devido a escalabilidade do treinamento e as limitações anteriormente citadas, foi utilizado um dataset com poucas amostras para aprimoramento do DistilBERT, algo que, como apresentado nos resultados, não obteve desempenho significativo.

Também devido às limitações de hardware, modelos acessados via API tiveram prioridade de escolha, porém ferramentas como Binoculars para detecção e LLaMA (LLaMA3.2) ou DeepSeek (DeepSeek-Coder-v2) foram escolhas dentro do limite de recursos disponíveis.

2. Tamanho dos textos para a análise dos detectores

Conforme demonstrado na Tabela 4, diversas foram as ferramentas analisadas ao longo dos testes, porém devido ao tamanho mínimo dos textos solicitados pelas ferramentas para a análise, muitas foram descartadas, pois o foco do trabalho tinha o tamanho dos textos como um critério importante. Sendo assim, a maioria dos detectores citados foram descartados e, a fim de descobrir como um deles se comportaria, utilizou-se o GPTZero aplicando espaços vazios no final das frases até o comprimento de 250 caracteres, o que pode ter influenciado na detecção, aspecto que também pode ter condicionado outros modelos que não restringiram sua detecção de acordo com o tamanho dos textos.

3. Disponibilidade de Ferramentas Gratuitas

Diversas ferramentas tanto de detecção, quando de geração de texto, não disponibilizam períodos gratuitos para utilização. Durante esse experimento utilizou-se ferramentas que fossem gratuitas ou oferecessem recursos limitados de acordo com certa quantidade de utilização. Alguns modelos como o Cohere, Sapling ou ChatPGT, ao fim do experimento já não possuíam mais cotas para utilização das APIs. O GPTZero também possibilitou um período de duas semanas de uso gratuito a partir da assinatura de teste, o que foi suficiente para gerar os resultados.

Como um dos critérios dos experimentos era mostrar como uma ferramenta brasileira se comportaria, a ferramenta MaritacaIA foi utilizada via API de maneira paga.

4. Limitações na Geração de Texto pelos LLMs

Alguns LLMs escolhidos, por possuírem um limite no tamanho da geração de textos, cortavam os conteúdos abruptamente, sem concluir o raciocínio. Após alguns testes, essa limitação foi removida por meio dos parâmetros do modelo, e passou-se a solicitar, via *prompt*, a produção de textos curtos que se aproximassem ao máximo da linguagem natural.

5. Veracidade de Comentários do X

Nos testes realizados com comentários reais obtidos diretamente dos *posts* do X, supõe-se, de forma hipotética, que todos os comentários foram escritos por usuários reais sem o uso de modelos de linguagem, algo que pode não ser totalmente verdade.

Conclusão

Neste trabalho, foi realizada uma análise aprofundada do comportamento de grandes modelos de linguagem (LLMs) na geração de comentários para redes sociais, em especial para a plataforma X, assim como da eficácia de detectores de texto sintético. Por meio da geração de textos com diferentes parâmetros, como a criação de personas, e da aplicação de detectores reconhecidos na literatura, foi possível analisar as nuances entre textos sintéticos e humanos.

Os resultados finais indicam que os LLMs conseguem gerar textos altamente semelhantes à linguagem humana, pelo menos vistos a olho nu sem a utilização de uma ferramenta própria para análise. A partir da análise automatizada utilizando as ferramentas da literatura, observou-se que os textos gerados por IA divergem um pouco em relação a textos reais, principalmente no tamanho dos comentários, na quantidade exagerada de *emojis* e também na informalidade. No geral, a complexidade linguística dos textos reais e sintéticos se manteve próxima. Entre os modelos analisados, o Llama3.2 destacou-se com uma média de 25.5% de sucesso contra os detectores ao se passar por um humano, enquanto o ChatGPT se destacou negativamente com uma taxa de apenas 8.6%. Notou-se também, que a não utilização de personas pré-definidas na instrução de produção de texto não interferiu significativamente nas métricas estruturais dos textos e nem na detecção. Para os resultados de detecção, os valores foram mais discrepantes, tendo o *RoBERTa Large OpenAI Detector* como o detector mais eficaz, tendo uma média de 98.4% de sucesso, enquanto modelos como o DistilBERT refinado e o ZeroGPT apresentaram 0% de sucesso. Outros detectores como o Binoculars e o Radar também apresentam ótimos resultados. Além disso, análises de comentários extraídos diretamente do X mostraram que, embora o *RoBERTa Large OpenAI Detector* tenha se destacado na detecção de textos gerados por IA, ele também apresentou uma alta taxa de possíveis falsos positivos, considerando que os comentários analisados são de fato humanos.

Este estudo contribui para a compreensão das capacidades dos modelos de linguagem atuais e das ferramentas de detecção, além de ressaltar a necessidade contínua de avanços tecnológicos na área de detecção de conteúdo artificial devido a seus impactos.

5.1 Principais Contribuições

Dentre as principais contribuições deste trabalho, destacam-se:

- ❑ Desenvolvimento de uma aplicação automatizada, responsável por todo o processo de geração, detecção e análise comparativa dos resultados, além de scripts responsáveis por realizar métricas linguísticas em conjuntos de dados.
- ❑ Avaliação comparativa entre comentários reais e textos gerados por IA, destacando diferenças e semelhanças linguísticas e estruturais.
- ❑ Análise da influência de personas pré-definidas no processo de geração de texto e seus impactos na detecção.
- ❑ Estudo prático da eficácia de diversos detectores da literatura limitados a um escopo específico, as redes sociais.
- ❑ Reflexão crítica sobre o uso dos LLMs nas redes sociais e seus impactos na sociedade.

5.2 Trabalhos Futuros

Esta pesquisa abre caminho para diversas oportunidades de pesquisas futuras, tais como:

- ❑ Utilização de personas neutras em relação ao tema de geração, permitindo que os textos apresentem maior perplexidade e dificultem a detecção.
- ❑ Ampliação das bases de dados utilizadas no estudo, selecionando redes sociais que não possuem um foco 100% voltado para interações humanas para análise do comportamento das ferramentas e dos textos em escopos alternativos.
- ❑ Estudo de ferramentas que surgirão ou terão novas versões disponibilizadas.
- ❑ Utilização de detectores com maior nível de refinamento, utilizando bases de dados citados neste e em outros trabalhos.
- ❑ Análise sentimental da reação de conteúdos gerados por IA em redes sociais por meio de experimentos direcionados.
- ❑ Ampliação da ferramenta desenvolvida para a implementação de quaisquer expansões do trabalho.

Referências

AHMED, H.; TRAORÉ, I.; SAAD, S. Detection of online fake news using n-gram analysis and machine learning techniques. In: **International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments**. [s.n.], 2017. Disponível em: <<https://api.semanticscholar.org/CorpusID:9670573>>. Citado na página 33.

ARAKAWA, R.; LEHMAN, J. F.; GOEL, M. Prism-qa: Step-aware voice assistant on a smartwatch enabled by multimodal procedure tracking and large language models. **Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.**, Association for Computing Machinery, New York, NY, USA, v. 8, n. 4, nov. 2024. Disponível em: <<https://doi.org/10.1145/3699759>>. Citado na página 28.

ARIAS, E. How does media influence social norms? experimental evidence on the role of common knowledge. **Political Science Research and Methods**, v. 7, n. 3, 2019. Citado na página 19.

BALDWIN, T. et al. How noisy social media text, how different social media sources? In: **International Joint Conference on Natural Language Processing**. [s.n.], 2013. Disponível em: <<https://api.semanticscholar.org/CorpusID:14652667>>. Citado 3 vezes nas páginas 9, 19 e 20.

BROWN, T. B. et al. **Language Models are Few-Shot Learners**. 2020. Disponível em: <<https://arxiv.org/abs/2005.14165>>. Citado 3 vezes nas páginas 9, 22 e 23.

CARVALHO, A. C. P. d. L. F. d. Inteligência artificial: riscos, benefícios e uso responsável. **Estudos Avançados**, 2021. Citado na página 14.

CHANG, Y. et al. **A Survey on Evaluation of Large Language Models**. 2023. Disponível em: <<https://arxiv.org/abs/2307.03109>>. Citado 3 vezes nas páginas 9, 22 e 23.

CHAUDHARY, A. **Back Translation for Text Augmentation with Google Sheets**. 2020. Acesso em: 29 abr. 2025. Disponível em: <<https://amitness.com/posts/back-translation-in-google-sheets#introduction-to-back-translation>>. Citado 2 vezes nas páginas 7 e 26.

CHEN, M. et al. **Evaluating Large Language Models Trained on Code**. 2021. Disponível em: <<https://arxiv.org/abs/2107.03374>>. Citado na página 29.

CHEN, Y. et al. **GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content**. 2023. Disponível em: <<https://arxiv.org/abs/2305.07969>>. Citado na página 29.

_____. Token prediction as implicit classification to identify LLM-generated text. In: BOUAMOR, H.; PINO, J.; BALI, K. (Ed.). **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**. Singapore: Association for Computational Linguistics, 2023. p. 13112–13120. Disponível em: <<https://aclanthology.org/2023.emnlp-main.810/>>. Citado na página 29.

CHOPRA, A.; PRASHAR, A.; SAIN, C. Natural language processing. **International Journal of Technology Enhancements and Emerging Engineering Research**, v. 1, p. 131–134, 2013. Disponível em: <<https://api.semanticscholar.org/CorpusID:196052257>>. Citado 2 vezes nas páginas 20 e 25.

COMMUNITY, O. **RoBERTa Large OpenAI Detector**. 2024. Hugging Face. Original RoBERTa paper: arXiv:1907.11692. Disponível em: <<https://huggingface.co/openai-community/roberta-large-openai-detector>>. Citado na página 62.

COX, I. et al. **Digital Watermarking and Steganography**. 2. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. ISBN 9780080555805. Citado na página 31.

DEVLIN, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: BURSTEIN, J.; DORAN, C.; SOLORIO, T. (Ed.). **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <<https://aclanthology.org/N19-1423/>>. Citado 10 vezes nas páginas 9, 22, 23, 25, 31, 34, 35, 42, 54 e 62.

DOU, Y. et al. **Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text**. 2022. Disponível em: <<https://arxiv.org/abs/2107.01294>>. Citado na página 36.

DUAN, J. et al. **Efficient Training of Large Language Models on Distributed Infrastructures: A Survey**. 2024. Disponível em: <<https://arxiv.org/abs/2407.20018>>. Citado 2 vezes nas páginas 22 e 25.

DUGAN, L. et al. **RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text**. 2020. Disponível em: <<https://arxiv.org/abs/2010.03070>>. Citado na página 36.

FAGNI, T. et al. Tweepfake: About detecting deepfake tweets. **PLOS ONE**, Public Library of Science (PLoS), v. 16, n. 5, p. e0251415, maio 2021. ISSN 1932-6203. Disponível em: <<http://dx.doi.org/10.1371/journal.pone.0251415>>. Citado na página 29.

FLESCH, R. A new readability yardstick. **Journal of Applied Psychology**, v. 32, n. 3, p. 221–233, 1948. Citado na página 59.

- FORNASIER, M. d. O.; BECK, C. Cambridge analytica: Escândalo, legado e possíveis futuros para a democracia. **Revista Direito em Debate**, v. 29, n. 53, p. 182–195, maio 2020. Disponível em: <<https://www.revistas.unijui.edu.br/index.php/revistadireitoemdebate/article/view/10033>>. Citado na página 19.
- FRASER, K. C.; DAWKINS, H.; KIRITCHENKO, S. **Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods**. 2024. Disponível em: <<https://arxiv.org/abs/2406.15583>>. Citado na página 35.
- GEHRMANN, S.; STROBELT, H.; RUSH, A. GLTR: Statistical detection and visualization of generated text. In: COSTA-JUSSÀ, M. R.; ALFONSECA, E. (Ed.). **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations**. Florence, Italy: Association for Computational Linguistics, 2019. p. 111–116. Disponível em: <<https://aclanthology.org/P19-3019/>>. Citado 3 vezes nas páginas 33, 35 e 41.
- GOKASLAN, A.; COHEN, V. **OpenWebText Corpus**. 2019. <<http://Skylion007.github.io/OpenWebTextCorpus>>. Citado na página 40.
- GUO, B. et al. **How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection**. 2023. Disponível em: <<https://arxiv.org/abs/2301.07597>>. Citado na página 29.
- 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>>. Citado na página 20.
- HANS, A. et al. **Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text**. 2024. Citado 3 vezes nas páginas 37, 38 e 62.
- HE, P. et al. **DeBERTa: Decoding-enhanced BERT with Disentangled Attention**. 2021. Disponível em: <<https://arxiv.org/abs/2006.03654>>. Citado na página 20.
- HE, X. et al. **MGTBench: Benchmarking Machine-Generated Text Detection**. 2024. Disponível em: <<https://arxiv.org/abs/2303.14822>>. Citado na página 29.
- HOWARD, J.; RUDER, S. **Universal Language Model Fine-tuning for Text Classification**. 2018. Disponível em: <<https://arxiv.org/abs/1801.06146>>. Citado 4 vezes nas páginas 21, 22, 54 e 63.
- HU, X.; CHEN, P.-Y.; HO, T.-Y. **RADAR: Robust AI-Text Detection via Adversarial Learning**. 2023. Disponível em: <<https://arxiv.org/abs/2307.03838>>. Citado 3 vezes nas páginas 37, 39 e 62.
- KING, J. et al. **LLM - Detect AI Generated Text**. 2023. <<https://kaggle.com/competitions/llm-detect-ai-generated-text>>. Kaggle. Citado na página 63.
- KROGH, A. What are artificial neural networks? **Nature Biotechnology**, nature publishing group, v. 26, n. 2, p. 195–7, 2008. ISSN 1087-0156. Citado 2 vezes nas páginas 21 e 32.

- LACERDA, G. **AI text in social networks**. 2025. <<https://github.com/GabrielLaacerda/ai-text-in-social-networks>>. Repositório no GitHub. Acesso em: 29 abr. 2025. Citado na página 6.
- LI, Y. et al. **MAGE: Machine-generated Text Detection in the Wild**. 2024. Disponível em: <<https://arxiv.org/abs/2305.13242>>. Citado na página 29.
- LIU, Y. et al. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. 2019. Disponível em: <<https://arxiv.org/abs/1907.11692>>. Citado 4 vezes nas páginas 25, 31, 37 e 42.
- _____. **ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models**. 2023. Disponível em: <<https://arxiv.org/abs/2304.07666>>. Citado na página 29.
- LUO, Q. et al. Self-attention and transformers: Driving the evolution of large language models. In: **2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)**. [S.l.: s.n.], 2023. p. 401–405. Citado na página 27.
- MACKO, D. et al. Multitude: Large-scale multilingual machine-generated text detection benchmark. In: **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2023. p. 9960–9987. Disponível em: <<http://dx.doi.org/10.18653/v1/2023.emnlp-main.616>>. Citado na página 29.
- MCCLAIN, C. et al. **The Behaviors and Attitudes of U.S. Adults on Twitter: A minority of Twitter users produce a majority of tweets from U.S. adults, and the most active tweeters are less likely to view the tone or civility of discussions as a major problem on the site**. [S.l.], 2021. Disponível em: <<http://www.jstor.org/stable/resrep63496>>. Citado na página 44.
- MINAEE, S. et al. **Large Language Models: A Survey**. 2024. Disponível em: <<https://arxiv.org/abs/2402.06196>>. Citado na página 21.
- OPENAI. **GPT-2 Output Dataset**. 2019. Disponível em: <<https://github.com/openai/gpt-2-output-dataset>>. Citado na página 29.
- OUYANG, L. et al. **Training language models to follow instructions with human feedback**. 2022. Disponível em: <<https://arxiv.org/abs/2203.02155>>. Citado na página 15.
- PAPAGEORGIOU, E. et al. A survey on the use of large language models (llms) in fake news. **Future Internet**, v. 16, n. 8, 2024. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/16/8/298>>. Citado na página 28.
- PU, X. et al. **On the Zero-Shot Generalization of Machine-Generated Text Detectors**. 2023. Disponível em: <<https://arxiv.org/abs/2310.05165>>. Citado 2 vezes nas páginas 34 e 35.
- PUDASAINI, S. et al. **Survey on Plagiarism Detection in Large Language Models: The Impact of ChatGPT and Gemini on Academic Integrity**. 2024. Disponível em: <<https://arxiv.org/abs/2407.13105>>. Citado na página 28.

- RADIOVOJEVIC, K.; CLARK, N.; BRENNER, P. **LLMs Among Us: Generative AI Participating in Digital Discourse**. 2024. Disponível em: <<https://arxiv.org/abs/2402.07940>>. Citado 5 vezes nas páginas 7, 44, 45, 50 e 75.
- SANH, V. et al. **DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter**. 2020. Disponível em: <<https://arxiv.org/abs/1910.01108>>. Citado na página 63.
- SCHULMAN, J. et al. **Proximal Policy Optimization Algorithms**. 2017. Disponível em: <<https://arxiv.org/abs/1707.06347>>. Citado na página 39.
- SHOEYBI, M. et al. **Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism**. 2020. Disponível em: <<https://arxiv.org/abs/1909.08053>>. Citado na página 22.
- SOLAIMAN, I. et al. Release strategies and the social impacts of language models. **arXiv preprint arXiv:1908.09203**, 2019. Citado na página 37.
- SU, Z. et al. **HC3 Plus: A Semantic-Invariant Human ChatGPT Comparison Corpus**. 2024. Disponível em: <<https://arxiv.org/abs/2309.02731>>. Citado na página 29.
- UCHENDU, A. et al. **TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation**. 2021. Disponível em: <<https://arxiv.org/abs/2109.13296>>. Citado na página 29.
- VASWANI, A. et al. **Attention Is All You Need**. 2023. Disponível em: <<https://arxiv.org/abs/1706.03762>>. Citado 5 vezes nas páginas 9, 22, 23, 27 e 60.
- VERMA, V. et al. **Ghostbuster: Detecting Text Ghostwritten by Large Language Models**. 2024. Disponível em: <<https://arxiv.org/abs/2305.15047>>. Citado na página 29.
- VYKOPAL, I. et al. Disinformation capabilities of large language models. In: **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Association for Computational Linguistics, 2024. p. 14830–14847. Disponível em: <<http://dx.doi.org/10.18653/v1/2024.acl-long.793>>. Citado na página 15.
- WANG, Y. et al. **M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection**. 2024. Disponível em: <<https://arxiv.org/abs/2305.14902>>. Citado na página 29.
- WANG, Z. et al. **A Comprehensive Survey on Data Augmentation**. 2024. Disponível em: <<https://arxiv.org/abs/2405.09591>>. Citado na página 25.
- WELSBY, P.; CHEUNG, B. M. Y. Chatgpt. **Postgraduate Medical Journal**, v. 99, n. 1176, p. 1047–1048, 07 2023. ISSN 0032-5473. Disponível em: <<https://doi.org/10.1093/postmj/qgad056>>. Citado na página 20.
- WU, J. et al. **A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions**. 2024. Disponível em: <<https://arxiv.org/abs/2310.14724>>. Citado 7 vezes nas páginas 9, 27, 29, 30, 32, 33 e 36.

- YANG, X. et al. **A Survey on Detection of LLMs-Generated Content**. 2023. Disponível em: <<https://arxiv.org/abs/2310.15654>>. Citado 3 vezes nas páginas 9, 29 e 30.
- YU, P. et al. **CHEAT: A Large-scale Dataset for Detecting ChatGPT-writtEn AbsTracts**. 2024. Disponível em: <<https://arxiv.org/abs/2304.12008>>. Citado na página 29.
- ZELLERS, R. et al. **Defending Against Neural Fake News**. 2020. Disponível em: <<https://arxiv.org/abs/1905.12616>>. Citado 3 vezes nas páginas 7, 29 e 34.
- ZEROGPT. **AI Text Detector**. 2024. Disponível em: <<https://www.zerogpt.com>>. Citado na página 63.
- ZHANG, S. et al. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. In: **International Conference on Learning Representations (ICLR)**. [S.l.: s.n.], 2024. Citado na página 37.
- _____. **Detecting Machine-Generated Texts by Multi-Population Aware Optimization for Maximum Mean Discrepancy**. 2024. Disponível em: <<https://arxiv.org/abs/2402.16041>>. Citado na página 42.
- ZHU, Q. et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. **arXiv preprint arXiv:2406.11931**, 2024. Citado na página 55.

Apêndices

Detectores de Textos Artificiais: Detalhes de Implementação

A.1 Binoculars

Para implementar a detecção de textos gerados por LLMs, foi desenvolvido um código de integração com o modelo proposto no artigo. A base do Binoculars utiliza os modelos Falcon-7B e Falcon-7B-Instruct. Conforme apresentado na Listagem A.1, o código realiza a tokenização da entrada e calcula métricas como perplexidade e entropia, que auxiliam na distinção entre textos escritos por humanos e aqueles gerados por inteligência artificial.

```
class Binoculars:
    def __init__(self, observer="tiiuae/falcon-7b", performer="tiiuae/
falcon-7b-instruct"):
        self.tokenizer = AutoTokenizer.from_pretrained(observer)
        self.observer_model = AutoModelForCausalLM.from_pretrained(
observer, device_map={"": "cpu"})
        self.performer_model = AutoModelForCausalLM.from_pretrained(
performer, device_map={"": "cpu"})
        self.threshold = 0.85
    def compute_score(self, text: str) -> float:
        encodings = self.tokenizer([text], return_tensors="pt",
truncation=True, max_length=512)
        obs_logits = self.observer_model(**encodings).logits
        perf_logits = self.performer_model(**encodings).logits
        return perplexity(encodings, perf_logits) / entropy(obs_logits,
perf_logits, encodings, self.tokenizer.pad_token_id)
    def predict(self, text: str) -> str:
        return "IA" if self.compute_score(text) < self.threshold else "
Humano"
```

Listagem A.1 – Código base do Binoculars simplificado.

Utilizando o modelo pronto do Binoculars, recebemos uma lista contendo os comentários a serem analisados e uma lista dos LLMs, a qual só é usada para organizar os resultados no final. Após receber esses parâmetros, usamos a biblioteca do Binoculars, resumida na Listagem A.1 para calcular as probabilidades de cada uma das frases. Os resultados são retornados por meio de um dicionário.

A.2 RADAR

Na implementação do modelo RADAR, a função apresentada na Listagem A.2 é responsável por receber os comentários, e a partir deles, realizar a tokenização, a qual posteriormente será convertida nos tensores. Nessa etapa, a frase é separada em tokens e convertidas em números, já que as redes neurais trabalham com valores numéricos. Após essa etapa, são gerados os *logits* do modelo, sendo estes, os valores da detecção em formato bruto. Como esses valores devem estar em formato de porcentagem, a função softmax é utilizada, a qual é responsável por converter esses valores entre 0 e 1, onde a soma dos dois seja igual a 1 (100%). A partir daí é definida a previsão do RADAR.

```
def calcular_probabilidade(frase, model, tokenizer, device):
    with torch.no_grad():
        inputs = tokenizer(frase, return_tensors="pt", truncation=True,
                           max_length=512)
        inputs = {k: v.to(device) for k, v in inputs.items()}
        logits = model(**inputs).logits
        probabilities = F.softmax(logits, dim=-1)
        prob_ia = probabilities[0][0].item() * 100
        prob_human = (1 - probabilities[0][0].item()) * 100
        return round(prob_human, 2), round(prob_ia, 2)
```

Listagem A.2 – Função para cálculo de probabilidade por meio do RADAR.

A.3 RoBERTa Large Openai Detector

No trecho de código da Listagem A.3, é possível ver como é realizada a detecção utilizando este modelo. O trecho é responsável por carregar o modelo roBERTa-large-openai-detector, aplicar a tokenização dos dados, fazer a detecção e normalizar os resultados retornados.

```
def probabilidade_IA(comentarios, modelos):
    warnings.filterwarnings("ignore")
    logging.set_verbosity_error()

    model_name = "roberta-large-openai-detector"
    model, tokenizer = AutoModelForSequenceClassification.
from_pretrained(model_name), AutoTokenizer.from_pretrained(model_name)

    def calcular_probabilidade(frase):
        inputs = tokenizer(frase, return_tensors="pt", truncation=True,
max_length=512)
        probs = F.softmax(model(**inputs).logits, dim=-1)[0]
        return round(probs[0].item() * 100, 2), round(probs[1].item() *
100, 2)

    resultados = []
    for item in comentarios:
        #Repassa cada comentário para a função calcular_probabilidade a
qual retorna as probabilidades

    return resultados
```

Listagem A.3 – Função simplificada de detecção do modelo RoBERTa Large Openai Detector

A.4 Sapling

A API do Sapling calcula a probabilidade dos termos terem sido gerados por IA ou humano e depois realiza a generalização desses resultados retornando a probabilidade geral. O código da Listagem A.4 demonstra como isso é feito.

```
def calcular_probabilidade_api(frase, api_key, url):

    api_key = os.getenv('KEY_SAPLING')
    url = "https://api.sapling.ai/api/v1/aidetect"

    response = requests.post(url, json={"key": api_key, "text": frase})
    if response.status_code == 200:
        result = response.json()
        return round((1 - result['score']) * 100, 2), round(result['
score'] * 100, 2)
    return None, None
```

Listagem A.4 – Função para cálculo de probabilidade por meio do Endpoint do Sapling.

Na função `calcular_probabilidade_api`, as frases são recebidas e enviadas para API via endpoint, o qual retorna as probabilidades que posteriormente são convertidas para porcentagens.

A.5 DistilBERT

Para treinamento do modelo foi utilizado um dataset com 10.000 redações, algumas escritas por humanos e outras por LLMs. O dataset conta com três arquivos, sendo eles:

1. `test|train_essays.csv`:

- ❑ **id**: Identificador único para cada redação.
- ❑ **prompt_id**: Identificador do prompt ao qual a redação responde.
- ❑ **text**: O texto da redação.
- ❑ **generated**: Indica se a redação foi escrita por um estudante (0) ou gerada por um modelo de IA (LLM) (1). Este campo **não está presente** no arquivo `test_essays.csv`.

2. `train_prompts.csv`:

- ❑ **prompt_id**: Identificador único para cada prompt.
- ❑ **prompt_name**: O título do prompt.
- ❑ **instructions**: As instruções dadas aos estudantes para escrever as redações.
- ❑ **source_text**: O texto fonte que os estudantes leram para escrever as redações, no formato Markdown. Pode incluir títulos, parágrafos numerados e subtítulos.

3. `sample_submission.csv`:

- ❑ Arquivo de exemplo para submissão, com o formato correto que os participantes devem seguir ao enviar suas previsões. Esse arquivo é fornecido para garantir que a submissão siga o formato adequado.

Após o treinamento do modelo, utiliza-se o DistilBERT para carregar os arquivos gerados e realizar a detecção dos comentários conforme a Listagem A.5.

```
def probabilidade_IA(comentarios, modelos):
    model = DistilBertForSequenceClassification.from_pretrained('/home/
gabriel/TCC_GabrielVnCS/trained_model_HC3')
    tokenizer = DistilBertTokenizer.from_pretrained('/home/gabriel/
TCC_GabrielVnCS/trained_model_HC3')

    resultados = []
    for item in comentarios:
        if isinstance(item, dict) and item.get('llm') in modelos and
isinstance(item.get('comentarios'), str):
            for frase in filter(bool, map(str.strip, item['comentarios']
].split("\n\n"))):
                probs = F.softmax(model(**tokenizer(frase,
return_tensors="pt", truncation=True, max_length=64, padding=True)).
logits, dim=-1)
                resultados.append({'llm': item['llm'], 'comentario':
frase, 'prob_humano': round(probs[0][0].item() * 100, 2), 'prob_IA':
round(probs[0][1].item() * 100, 2)})

    return resultados
```

Listagem A.5 – Função para cálculo de probabilidade do DistilBERT.

Com o modelo pré-treinado, carrega-se o treinamento salvo e realiza a validação e tratamento dos dados de entrada. Com isso, é feita a tokenização das frases e aplicação da função `model(**inputs).logits`, responsável por calcular os valores referentes a detecção. Com esses valores brutos, é aplicada a função `softmax` para normalizar esses dados entre 0 e 1, os quais posteriormente são convertidos para porcentagem.

A.6 HuggingFace Detector

A implementação utilizada nesse experimento adapta o modelo original obtido via HuggingFace, o qual recebe os comentários e realiza os procedimentos necessários para detecção conforme simplificado na Listagem A.6.

```
def probabilidade_IA(comentarios, modelos):

    # Carrega o tokenizer e o modelo pré-treinado
    tokenizer = AutoTokenizer.from_pretrained("PirateXX/AI-Content-
Detector", token=ACCESS_TOKEN)
    model = AutoModelForSequenceClassification.from_pretrained("PirateXX
/AI-Content-Detector", token=ACCESS_TOKEN)
    model.to(device)

    # Função que divide o texto em frases
    def text_to_sentences(text):
```

```
# Remove quebras de linha e divide o texto em frases

# Função que divide o texto em pedaços de 900 caracteres
def chunks_of_900(text, chunk_size=900):
    # Divide o texto em pedaços menores, com no máximo 900
    caracteres

# Função que faz a predição do modelo para uma consulta
def predict(query):
    # Codifica a consulta, aplica o modelo e retorna a probabilidade
    de ser real

# Função que calcula a probabilidade de o texto ser gerado por IA
def findRealProb(text):
    # Divide o texto em pedaços, faz a predição para cada pedaço e
    calcula a probabilidade média
```

Listagem A.6 – Código simplificado do detector HuggingFace Detector.