Artigo Completo

Impacto do Tamanho da Amostra de Treinamento e do Vocabulário Para Tarefas de Classificação de Texto

Impact of Training Sample and Vocabulary Size on Text Classification Tasks

Ítalo Alves Tenório de Almeida [Universidade Federal do Vale do São Francisco | *italo.tenorio@discente.univasf.edu.br*]
Jorge Luís Cavalcanti Ramos [Universidade Federal do Vale do São Francisco | *jorge.cavalcanti@univasf.edu.br*]
Brauliro Gonçalves Leal [Universidade Federal do Vale do São Francisco | *brauliro.leal@univasf.edu.br*]
Rosalvo Ferreira de Oliveira Neto [Universidade Federal do Vale do São Francisco | *rosalvo.oliveira@univasf.edu.br*]

Universidade Federal do Vale do São Francisco, Av. Antônio C. Magalhães, 510 - Santo Antônio, Juazeiro, Bahia, 48902-300, Brasil.

Resumo. O crescimento dos modelos de linguagem baseados em arquiteturas Transformer tem ampliado significativamente as possibilidades de aplicação do Processamento de Linguagem Natural em tarefas de classificação de texto. Entretanto, aspectos fundamentais do treinamento desses modelos, como o tamanho do vocabulário e a quantidade de amostras utilizadas, ainda contam com poucas orientações práticas para sua configuração em tarefas de classificação de texto. Este trabalho analisa o impacto desses dois fatores no desempenho de modelos de classificação de texto utilizando o DistilRoBERTa. Foram realizados experimentos com dois conjuntos de dados públicos disponibilizados pela plataforma Hugging Face: um conjunto de resumos de artigos médicos, binarizado para neoplasmas contra as demais classes, e um conjunto de avaliações de filmes do IMDb. O estudo empregou validação cruzada *k-fold* (k = 5), planejamento fatorial 2^k e a métrica AUC-ROC para avaliar o desempenho dos modelos. Comparou-se o DistilRoBERTa com seu vocabulário original (genérico) e com um vocabulário reduzido e especializado, gerado a partir do corpus de treinamento, além de um modelo de propósito geral baseado em LLM, o GPT-3.5-turbo. Em amostras de 1000 a 5000 exemplos, o vocabulário original apresentou o melhor desempenho global entre as variantes do DistilRoBERTa em ambas as bases, e o vocabulário reduzido teve desempenho inferior, com maior variabilidade entre os *folds* em amostras menores. O vocabulário reduzido, porém, foi o que mais se beneficiou do aumento da amostra e, na base médica, superou o GPT-3.5-turbo para 2500 e 5000 amostras. A influência relativa do vocabulário e da amostra depende do domínio e do conjunto de dados: o vocabulário explicou a maior parte da variação total, em maior grau no IMDb do que na base médica. Os resultados contribuem para uma melhor compreensão dos fatores que influenciam o desempenho de modelos de classificação de texto e fornecem subsídios para o planejamento mais eficiente de experimentos em Processamento de Linguagem Natural.

Abstract. The rapid development of Transformer-based language models has significantly expanded the applicability of Natural Language Processing techniques to text classification tasks. However, fundamental training parameters, such as vocabulary size and training sample size, still have few practical guidelines for their configuration in text classification tasks. This work analyzes the impact of these two factors on the performance of text classification models based on DistilRoBERTa. Experiments were conducted using two publicly available datasets from the Hugging Face platform: a collection of medical research abstracts, binarized as neoplasms versus other classes, and the IMDb movie review dataset. The study employed 5-fold cross-validation, a 2^k factorial design, and the AUC-ROC metric to evaluate model performance. DistilRoBERTa with its original (generic) vocabulary was compared with a reduced, domain-specific vocabulary generated from the training corpus, as well as with the general-purpose large language model GPT-3.5-turbo. With samples of 1,000 to 5,000 examples, the original vocabulary achieved the best overall performance among the DistilRoBERTa variants on both datasets, whereas the reduced vocabulary performed worse, with higher variability across folds for smaller samples. The reduced vocabulary, however, benefited the most from larger samples and, on the medical dataset, outperformed GPT-3.5-turbo with 2,500 and 5,000 samples. The relative influence of vocabulary and sample size depends on the domain and dataset: vocabulary accounted for most of the total variation, to a greater extent on IMDb than on the medical dataset. These findings contribute to a better understanding of the factors that influence text classification performance and provide guidance for the design of more efficient Natural Language Processing experiments.

Palavras-chave: Processamento de Linguagem Natural, Classificação de Texto, Modelos de Linguagem, Tokenização, DistilRoBERTa

Keywords: Natural Language Processing, Text Classification, Language Models, Tokenization, DistilRoBERTa

Recebido/Received: 13 08 2026 • **Aceito/Accepted:** 07 10 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

O Processamento de Linguagem Natural (do inglês *Natural Language Processing*, NLP) constitui um ramo da ciência da computação, da inteligência artificial e da linguística, dedicado ao estudo da interação complexa entre computadores e a linguagem humana. O campo abrange a modelagem matemática e computacional dos diversos aspectos da linguagem, visando desenvolver uma ampla gama de sistemas. O objetivo central do NLP é permitir que máquinas compreendam e manipulem textos ou fala em linguagem natural de forma útil, o que envolve a extração de estrutura gramatical e significado a partir dos dados de entrada. A relevância do NLP é atestada pela vasta aplicação de suas técnicas em diversas áreas, incluindo a tradução automática, a sumarização de textos, interfaces de usuário mais interativas, sistemas de recuperação de informação multilingues e, fundamentalmente, como componente chave na construção de sistemas de IA (Inteligência Artificial) [Liddy, 2001].

Nos últimos anos, a evolução do NLP foi acelerada pelo surgimento dos LLMs (do inglês *Large language Models*), que têm capturado grande atenção devido ao seu desempenho robusto em uma vasta gama de tarefas de linguagem natural. Os LLMs são modelos estatísticos pré-treinados em grande escala, baseados em arquiteturas de redes neurais, notavelmente os Transformers. Eles se distinguem por conterem dezenas a centenas de bilhões de parâmetros, treinados em quantidades massivas de dados de texto web. Esta escalabilidade não apenas melhorou a compreensão e a geração de linguagem, mas também resultou em habilidades emergentes que não se manifestavam em modelos menores, como o aprendizado em contexto (*in-context learning*), o seguimento de instruções e a capacidade de raciocínio em múltiplas etapas. Essa capacidade de resolver tarefas de propósito geral posiciona os LLMs como o bloco construtor básico para o desenvolvimento de agentes de IA mais sofisticados. Dentro deste cenário, a classificação de texto consolidou-se como uma tarefa fundamental e perene do NLP, consistindo na categorização automática de documentos em rótulos ou classes predefinidas com base em seu conteúdo, tarefa crucial para a descoberta de conhecimento e para a tomada de decisões em um contexto de crescimento exponencial de dados digitais. Com o advento da era dos LLMs, a performance destes sistemas deixou de depender exclusivamente de grandes volumes de dados rotulados, abrindo espaço para abordagens de aprendizado *zero-shot* e *few-shot*, além da utilização de modelos para a geração de dados sintéticos. No entanto, a escolha da arquitetura permanece crítica; enquanto modelos de grande porte como o Llama3 e o GPT-4 mostram excelência em tarefas complexas, modelos menores e otimizados como o RoBERTa oferecem um equilíbrio superior entre desempenho e tempo de inferência, mantendo-se competitivos especialmente em cenários de classificação binária ou quando a eficiência computacional é uma prioridade [Kostina *et al.*, 2025; Vajjala and Shimangaud, 2025].

No entanto, a definição do tamanho ideal do vocabulário (o número de tokens) é uma questão aberta, dependendo da tarefa, do domínio e da linguagem, e afeta diretamente a eficiência computacional (o comprimento das sequências) e a capacidade de generalização do modelo. Similarmente, o tamanho das amostras de treinamento do modelo, bem como seu impacto no desempenho do mesmo, permanece um problema com solução altamente dependente da tarefa que o modelo irá executar. Embora existam trabalhos sobre tokenização [Mielke *et al.*, 2021] e sobre classificação de texto com modelos de linguagem [Kostina *et al.*, 2025; Vajjala and Shimangaud, 2025], na literatura consultada pelos autores são escassas as orientações práticas que relacionem conjuntamente o tamanho do vocabulário e o tamanho da amostra de treinamento ao desempenho de modelos de menor porte em classificação de texto.

Diante disso, este trabalho se propõe a investigar o impacto do tamanho do vocabulário e da amostra de treinamento na eficácia e nas características de desempenho em tarefas de classificação de texto.

2 Referencial Teórico

2.1 Processamento de Linguagem Natural

O NLP, também conhecido como Linguística Computacional, constitui um subcampo fundamental da Ciência da Computação. Sua essência reside na aplicação de técnicas computacionais para aprender, compreender e produzir conteúdo em linguagem humana. O NLP surgiu na década de 1950, situando-se na intersecção entre a inteligência artificial e a linguística. O objetivo principal do NLP é atingir o "processamento de linguagem semelhante ao humano", sendo uma disciplina enquadrada dentro da IA. [Liddy, 2001; Reshamwala *et al.*, 2013; Hirschberg and Manning, 2015].

Embora o termo ”processamento” seja escolhido deliberadamente, e não ”compreensão” (*Natural Language Understanding* – NLU), a NLU continua sendo a meta final do NLP. Um sistema de compreensão da linguagem humana completo seria capaz de parafrasear um texto de entrada, traduzir o texto para outra língua, responder a perguntas sobre seu conteúdo e tirar inferências a partir dele [Liddy, 2001].

O NLP é a base para uma vasta gama de aplicações, muitas das quais já fazem parte do cotidiano, como os assistentes de diálogo falado [Hirschberg and Manning, 2015].

2.2 Large Language Models

Os LLMs representam a quarta e mais recente geração de Modelos de Linguagem (LMs), evoluindo a partir dos Modelos de Linguagem Pré-treinados (do inglês *Pre-trained Language Models*, PLM) que, por sua vez, sucederam os modelos de linguagem neurais e estatísticos [Zhao *et al.*, 2023]. O desenvolvimento da modelagem de linguagem, que data da aplicação da teoria da informação por Shannon na década de 1950 [Minaee *et al.*, 2024], atingiu um ponto de inflexão com o surgimento dos LLMs. Tecnicamente, os LLMs são modelos de linguagem baseados na arquitetura Transformer que contêm dezenas a centenas de bilhões (ou mais) de parâmetros e são pré-treinados em dados de texto massivos [Zhao *et al.*, 2023; Minaee *et al.*, 2024]. Eles exibem fortes capacidades de compreensão da linguagem natural e são capazes de resolver tarefas complexas através da geração de texto [Zhao *et al.*, 2023].

2.2.1 LLMs Representativos

A pesquisa em LLMs tem sido impulsionada por modelos chave, que são frequentemente agrupados em famílias. A Figura 1 reúne algumas das principais famílias de LLMs atualmente, descritas a seguir, da esquerda para a direita:

- Família GPT (OpenAI): Os Generative Pre-trained Transformers são modelos *decoder-only*. O GPT-3 (175B parâmetros) é notável por formalmente introduzir o ICL (do inglês *In-Context Learning*) [Brown *et al*., 2020; apud Minaee *et al*., 2023]. O ChatGPT, um chatbot de IA poderoso, é desenvolvido com base em LLMs da série GPT [Zhao et al., 2023]. Mais recentemente, o GPT-4 (1.76T parâmetros) elevou o nível de desempenho, incluindo capacidades multimodais [Minaee *et al*., 2024].
- Família LLaMA (Meta AI): Esta família, que inclui LLaMA (até 65B parâmetros) e LLaMA 2 (até 70B parâmetros), atraiu atenção significativa por ser de código aberto (open-source). O LLaMA 2, em particular, explorou o RLHF (do inglês *Reinforcement Learning From Human Feedback*) e possui uma versão otimizada para chat (LLaMA-chat) [Touvron *et al*., 2023; apud Zhao *et al*., 2023].
- Família PaLM (Google): O PaLM (até 540B parâmetros) foi treinado em um *dataset* de 780B tokens, e o PaLM 2 utilizou 3.6T tokens e incorpora o objetivo de pré-treinamento unificado MoD (*Mixture-of-Denoisers*) [Minaee *et al*., 2024].

2.3 Tokenização e Vocabulário

A tokenização é uma etapa fundamental e crítica no pré-processamento de texto, essencial para preparar os dados de entrada para modelos de linguagem profunda, como os LLMs. Essencialmente, a tokenização lida com a questão fundamental de quais são as unidades de texto que queremos modelar. A premissa básica é que o texto é fatiado em pequenas peças que são alimentadas no computador, eventualmente como sequências de números inteiros, e essas peças são chamadas de tokens [Mielke *et al*., 2021]. O vocabulário, por sua vez, consiste no conjunto de tokens reconhecidos pelo modelo, onde cada token e representado concretamente na forma de um vetor de alta dimensão [Nakash *et al*., 2025].

2.3.1 Metodologias de Segmentação de Subpalavras

As estratégias de tokenização moderna se dividem em abordagens de subpalavra (impulsionadas por dados), abordagens linguísticas/morfológicas, e abordagens de decomposição máxima (caractere/byte).

Os algoritmos de segmentação de subpalavra Dirigida por Dados (*Data-Driven*) buscam unidades de subpalavra que sejam frequentes ou que maximizem a verossimilhança do modelo de linguagem. BPE e WordPiece são os métodos de facto empregados pelos LLMs mais importantes, como BERT e GPT [Mielke *et al*., 2021].

2.4 Arquitetura Transformer

A arquitetura Transformer representa um marco fundamental na área de modelagem de sequências e transdução, sendo introduzida em 2017 com a proposta de que "atenção é tudo o que você precisa" . O modelo Transformer foi proposto como uma nova arquitetura de rede neural baseada somente em mecanismos de atenção, dispensando completamente a recorrência e as convoluções [Vaswani *et al*., 2017].

Historicamente, os modelos dominantes para problemas de transdução de sequência, como modelagem de linguagem e tradução automática, eram baseados em redes neurais recorrentes (do inglês *Recurrent Neural Networks*, RNNs), em particular LSTMs (*Long Short-Term Memory*) e redes neurais recorrentes com (GRUs) (*gated recurrent units*). Estes modelos recorrentes tipicamente levavam em conta a computação ao longo das posições dos símbolos na sequência de entrada e saída. Ao alinhar as posições com os passos no tempo de computação, geravam uma sequência de estados ocultos, uma função do estado oculto anterior e da entrada da posição atual . Essa natureza inerentemente sequencial impedia a paralelização dentro dos exemplos de treinamento, um obstáculo crítico para lidar com sequências mais longas [Vaswani *et al*., 2017].

O Transformer superou essa restrição fundamental ao adotar mecanismos de atenção, permitindo um grau significativamente maior de paralelização e alcançando novos resultados de ponta em tarefas de tradução com tempos de treinamento reduzidos . O mecanismo de atenção permite modelar dependências sem considerar sua distância nas sequências de entrada ou saída [Vaswani *et al*., 2017].

2.4.1 Estrutura Geral do Modelo

O Transformer segue a arquitetura geral de *encoder-decoder*, ilustrada na Figura 2. O *encoder* mapeia uma sequência de entrada de representações de símbolos

$$x = (x_1, ..., x_n)$$

para uma sequência de representações contínuas

$$z = (z_1, ..., z_n).$$

Dado z, o *decoder* então gera uma sequência de saída

$$y = (y_1, ..., y_m),$$

um elemento de cada vez, operando de forma autoregressiva e consumindo os símbolos gerados previamente como entrada adicional para gerar o próximo [Vaswani *et al*., 2017].

2.4.2 Atenção

A atenção é a base da arquitetura Transformer. Uma função de atenção pode ser descrita como o mapeamento de uma *query* e um conjunto de pares *key-value* para uma saída [Vaswani *et al*., 2017]. A saída é calculada como uma soma ponderada dos valores, onde o peso atribuído a cada valor é determinado por uma função de compatibilidade da *query* com a *key* correspondente [Vaswani *et al*., 2017].

O tipo específico de atenção utilizado no Transformer é denominado *Scaled Dot-Product Attention* (Atenção por Produto Escalar Escalonado). O processo envolve [Vaswani *et al*., 2017]:

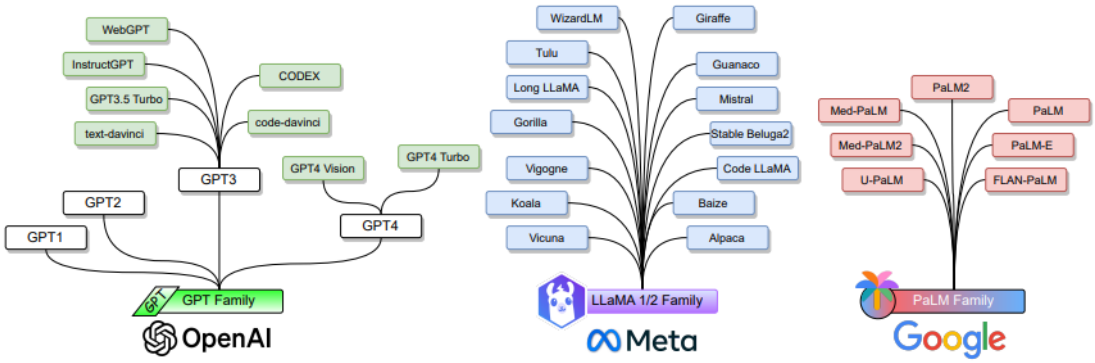


Figura 1. Famílias de LLMs [Minaee *et al.*, 2024].

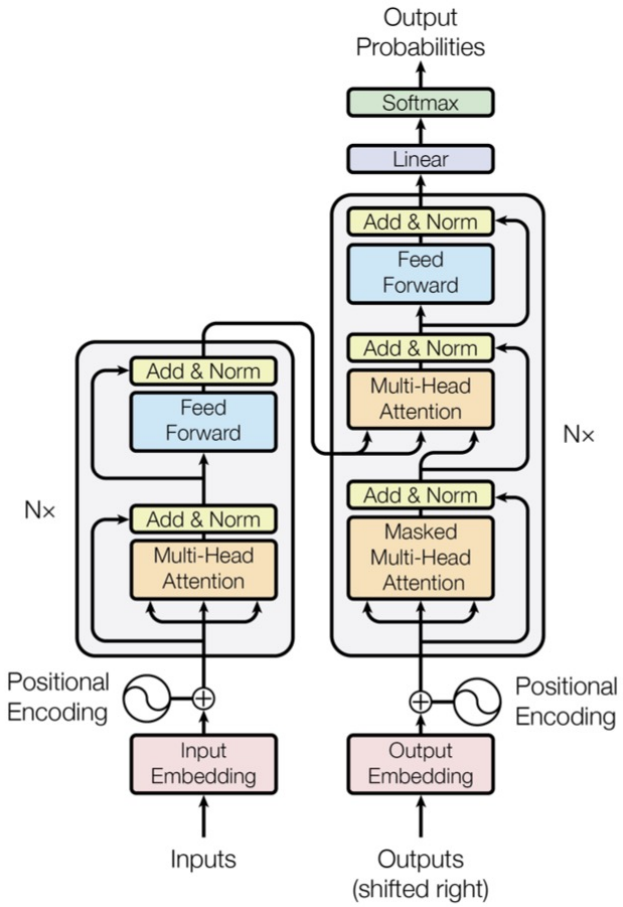


Figura 2. Diagrama da arquitetura Transformer [Vaswani *et al.*, 2017].

- Calcular os produtos escalares da *query* com todas as *keys*.
- Dividir cada resultado por d, onde d é a dimensão das *keys*.
- Aplicar a função softmax para obter os pesos sobre os *values* [Vaswani *et al.*, 2017].

Matematicamente, quando processado simultaneamente em um conjunto de *queries* Q, *keys* K e *values* V (empacotados em matrizes), a atenção é calculada como:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

Em vez de realizar uma única função de atenção com keys, values e queries de dimensão d_{model}, o Transformer utiliza MHA (do inglês *Multi-Head Attention*). O MHA projeta linearmente as queries, keys e values h vezes, usando projeções diferentes e aprendidas, para dimensões d, respectivamente. A função de atenção é então aplicada em paralelo a cada uma dessas versões projetadas, resultando em h saídas dimensionais. Estas são então concatenadas e projetadas novamente linearmente para produzir o valor final [Vaswani *et al.*, 2017]. O MHA permite que o modelo atenda conjuntamente a informações de diferentes subespaços de representação e em diferentes posições. A utilização de múltiplos *heads* (cabeças de atenção) mitiga o problema de que uma única cabeça de atenção, ao calcular a média, inibiria essa capacidade . Na arquitetura base, $h = 8$ cabeças paralelas são empregadas, com $d = 64$. Apesar de usar múltiplas cabeças, o custo computacional total permanece similar ao de uma única cabeça com dimensionalidade total, devido à redução das dimensões em cada head [Vaswani *et al.*, 2017].

2.5 Classificação de Texto

A classificação de texto é uma tarefa fundamental e perene no campo do NLP, consistindo na categorização automática de documentos ou fragmentos de texto em rótulos ou classes predefinidas com base em seu conteúdo. Este processo é crucial para a gestão da informação e descoberta de conhecimento, permitindo a organização eficiente de grandes volumes de dados digitais. Tradicionalmente, o fluxo de trabalho de classificação segue etapas padronizadas: coleta de dados, pré-processamento,

extração de características, redução de dimensionalidade e, por fim, a aplicação de técnicas de classificação seguidas de avaliação de desempenho [Kadhim, 2019].

2.5.1 Análise de Sentimento

A análise de sentimento, frequentemente referida como mineração de opinião, é uma subárea do NLP e da Classificação de Texto, dedicada à identificação e extração de informações subjetivas a partir de textos. O objetivo central desta tarefa é processar e analisar computacionalmente opiniões, pensamentos e impressões humanas sobre diversos tópicos, produtos ou serviços, permitindo que empresas e indivíduos tomem decisões fundamentadas com base na polaridade expressa pelos usuários [Wankhade *et al*., 2022]. A análise de sentimento é geralmente classificada em diferentes níveis de granularidade: nível de documento (atribui uma polaridade única ao texto completo), nível de frase (analisa cada sentença individualmente), nível de expressão (foca em unidades linguísticas básicas) e o nível de aspecto. Este último é considerado um dos ramos mais promissores, pois permite identificar sentimentos específicos sobre atributos distintos de um produto, como o "sabor" de uma comida ou a "lentidão" de um serviço.

3 Metodologia

3.1 Bases de Dados

Foram utilizadas duas bases de dados no experimento, ambas disponíveis publicamente no site HuggingFace (`https://huggingface.co/`).

3.1.1 Resumos de Artigos Médicos

Essa base, com tamanho de 14443 linhas, consiste em resumos de diversos artigos da área medica. Foi criada originalmente durante a escrita do trabalho de Schopf *et al*. [2022]. A base possui duas colunas: uma contendo o texto do resumo do artigo, e a outra contendo o rótulo da classe associado à área de pesquisa abrangida. São 5 rótulos: Neoplasmas, doenças cardiovasculares, doenças do sistema nervoso, doenças do sistema digestivo e condições patológicas gerais [Schopf *et al*., 2022].
Para os propósitos do experimento, a coluna de rótulos foi alterada de forma a representar um problema de classificação binária, ao invés de um problema multi-classe. Devido à quantidade de exemplos, a classe Neoplasmas foi escolhida como referência. Assim, os abstratos passaram a ser classificados em "neoplasmas" (rótulo 1), caso essa seja a área de pesquisa abrangida, e não-neoplasmas (rótulo 0) caso não seja. Essa binarização simplifica o cenário clínico original, que é multiclasse, e foi adotada para viabilizar o uso do AUC-ROC. Os resultados, portanto, não se estendem diretamente ao problema multiclasse.
As amostras de cada tamanho foram construídas com proporção equilibrada entre as classes, selecionando um número igual de exemplos positivos e negativos (proporção 50/50).

3.1.2 IMDb

Essa base, construída para classificação de texto (especificamente análise de sentimento), possui 100000 linhas e consiste em resenhas críticas de diversos filmes, extraídas do site IMDb.com. Possui duas colunas : uma contendo o texto da resenha, e a outra contendo um rótulo que classifica a resenha como positiva (1) ou negativa (0). Foi escolhida devido à grande quantidade de exemplos e ao fato de já estar estruturada como um problema de classificação binária, além de possuir distribuição 50/50 entre classes positiva e negativa [Maas *et al*., 2011].

3.2 Modelo de Linguagem

3.2.1 DistilRoBERTa

O modelo de linguagem escolhido para o treinamento foi o DistilRoBERTa, devido ao seu tamanho leve, disponibilidade gratuita e alto desempenho em tarefas de classificação de texto, além do fato de poder ser usado para *fine-tuning* em *datasets* específicos sem o requisito de recursos computacionais extensos [Shah *et al*., 2024]. O DistilRoBERTa destaca-se como uma excelente escolha para classificação de texto e análise de sentimento devido ao seu desempenho superior em comparação com outros modelos da mesma categoria. Além do alto desempenho, a principal vantagem técnica do DistilRoBERTa é a sua eficiência e tamanho reduzido. Como uma versão simplificada da arquitetura RoBERTa, ele foi projetado para utilizar menos camadas e parâmetros do que o modelo BERT original, visando diminuir significativamente o peso do modelo. Essa redução é complementada por um tokenizador especializado que utiliza um vocabulário menor, resultando em representações de tokens mais compactas que otimizam o processamento de texto sem comprometer a qualidade dos resultados. Dessa forma, o DistilRoBERTa oferece uma solução poderosa e rápida para o processamento de grandes volumes de dados, mesmo em ambientes com recursos computacionais limitados [Kyritsis *et al*., 2024].

3.2.2 GPT-3.5 turbo

Desde o lançamento inicial do ChatGPT em novembro de 2022, a OpenAI estabeleceu os padrões para modelos de linguagem de última geração. O GPT-3.5-turbo foi desenvolvido para equilibrar desempenho e eficiência, estima-se que possua cerca de 20 bilhões de parâmetros e o comprimento máximo de suas saídas atinge 4096 tokens. O modelo está disponível para ajuste fino por meio da API da OpenAI e, atualmente, seu tamanho relativamente menor é provavelmente o que o torna uma solução tão acessível em comparação com os muito maiores GPT-4 e GPT-4-turbo [Mazzullo *et al*., 2025].
O modelo GPT-3.5 turbo será utilizado para realizar uma classificação do conjunto de dados de validação nos seus rótulos correspondentes, por meio de um *prompt*.
Para realizar a classificação no conjunto de resumos médicos, o seguinte *prompt* foi utilizado: "*Você é um classificador altamente preciso de abstratos médicos. Você receberá um abstrato de um artigo medico e deve classificá-lo como 1, caso o mesmo trate da área de neoplasmas, ou 0 caso contrário. Responda apenas com*

'0' ou '1'."
Já para a classificação dos *reviews* de filmes contidos na base do IMDb, utilizou-se o seguinte *prompt*: "*Você é um classificador altamente preciso de reviews de filmes. Você receberá uma review e deve classificá-la como 1, caso a mesma seja positiva, ou 0 caso seja negativa. Responda apenas com '0' ou '1'.*"
O objetivo é comparar o desempenho, na classificação de texto, de um modelo de propósito geral amplamente usado com o de um modelo ajustado para a tarefa específica com um *fine-tuning* (o DistilRoBERTa, no caso deste experimento). O modelo foi escolhido levando em consideração as limitações computacionais e financeiras inerentes a este experimento. A conexão com o modelo será feita a partir da API da OpenAI e realiza um total de 3400 requisições (1700 para cada *dataset*).

3.3 Ferramentas

3.3.1 Python

A linguagem de programação Python (versão 3.11.9) será utilizada para a implementação computacional do experimento, junto com um conjunto de bibliotecas pertinentes às áreas abordadas no estudo. Python possui um ecossistema vasto e maduro de bibliotecas e frameworks, como NumPy, Pandas, TensorFlow, PyTorch e scikit-learn, que oferecem ferramentas otimizadas para processamento de dados, aprendizado de máquina e redes neurais profundas. Sua integração eficiente com linguagens de baixo nível (como C e C++) permite combinar facilidade de uso com alto desempenho computacional. Outro fator decisivo é a forte comunidade científica e de desenvolvedores, que garante ampla documentação, suporte contínuo e atualização constante das tecnologias, consolidando Python como a principal linguagem na pesquisa e no desenvolvimento de soluções baseadas em IA e aprendizado de máquina.

3.3.2 HuggingFace

A Hugging Face é uma plataforma e comunidade de código aberto que se tornou referência no campo do NLP e da inteligência artificial aplicada a textos. Seu principal destaque é a biblioteca Transformers, que disponibiliza uma ampla coleção de modelos pré-treinados baseados em arquiteturas modernas, como BERT, RoBERTa, GPT, T5 e DistilBERT, entre outros. Essa biblioteca simplifica a aplicação de técnicas avançadas de aprendizado profundo em tarefas como classificação de texto, tradução automática, sumarização, análise de sentimentos e reconhecimento de entidades nomeadas, com poucas linhas de código. Além disso, o ecossistema da Hugging Face inclui ferramentas para treinamento, avaliação e implantação de modelos, bem como uma comunidade ativa que compartilha pesos, *datasets* e experimentos. Assim, a Hugging Face e sua biblioteca Transformers democratizaram o acesso às tecnologias de ponta em NLP, permitindo que pesquisadores e desenvolvedores implementem soluções de IA sofisticadas de forma prática e eficiente. A biblioteca Transformers será utilizada para o carregamento dos modelos pré- treinados a serem utilizados.

3.3.3 SciKit-learn

A biblioteca scikit-learn será usada para a avaliação dos experimentos. A biblioteca Scikit-learn é uma das ferramentas mais utilizadas em projetos de aprendizado de máquina (machine learning) desenvolvidos em Python. Ela oferece uma ampla gama de algoritmos prontos para tarefas como classificação, regressão, clusterização, redução de dimensionalidade e validação de modelos. Além de sua facilidade de uso, o Scikit-learn é construído sobre bibliotecas robustas como NumPy, SciPy e Matplotlib, o que garante desempenho e integração com outras etapas do fluxo de ciência de dados. Sua API padronizada torna o processo de treinamento, teste e avaliação de modelos mais intuitivo, permitindo que tanto iniciantes quanto profissionais avancem rapidamente na implementação de soluções de IA.

3.3.4 Google Colab

O ambiente do Google Colab será utilizado para o treinamento em nuvem dos modelos. O Google Colab é uma plataforma gratuita baseada em nuvem que permite executar código Python diretamente em um ambiente interativo, sem a necessidade de instalação local. Amplamente utilizada na pesquisa e no ensino de inteligência artificial, o Colab oferece acesso gratuito a GPUs e TPUs, o que acelera significativamente o treinamento de modelos complexos. Além disso, sua integração com o Google Drive facilita o armazenamento e o compartilhamento de notebooks, promovendo a colaboração entre pesquisadores e desenvolvedores. Por ser baseado no Jupyter Notebook, o Colab combina código, texto explicativo, imagens e resultados em um único documento, tornando o processo de experimentação e aprendizado mais dinâmico e acessível.

3.4 Avaliação e Validação

3.4.1 K-fold Cross Validation

O *k-fold cross validation* é uma técnica de validação usada para avaliar o desempenho de um modelo de aprendizado de máquina de forma mais confiável. A estratégia consiste em dividir o conjunto de dados em k partes (ou "*folds*") de tamanhos iguais. O modelo é treinado k vezes, cada vez utilizando (k - 1) partes para treino e 1 parte diferente para teste, e em seguida as métricas de desempenho são calculadas em cada iteração. Finalmente, calcula-se a média dessas métricas, obtendo uma estimativa mais estável do desempenho do modelo. Nesse experimento, será utilizado k = 5, portanto o conjunto de dados será dividido em 5 partes iguais, onde a cada iteração uma das partes servirá como teste e as outras 4 como treinamento. Ao aproveitar esse método, maximizamos o uso dos dados disponíveis para treinamento e teste, evitando vazamento de dados entre os conjuntos, o que o torna particularmente adequado para cenários com conjuntos de dados anotados limitados [Rhoden, 2025]. Os *folds* foram estratificados, de modo a preservar em cada partição a proporção original entre as classes.

3.4.2 Planejamento fatorial 2^k

O planejamento fatorial 2^k é uma técnica experimental usada para estudar como k fatores (variáveis independentes) influenciam um resultado (variável dependente). Nesse método, cada variável é analisada em dois níveis, máximo e mínimo, e suas combinações são agrupadas em uma tabela na qual as colunas representam os fatores e combinações de fatores. Nesta, uma posição com entrada com valor igual a -1 denota o valor mínimo do parâmetro, enquanto 1, seu valor máximo. Assim, o experimento envolve 2^k combinações possíveis entre os níveis dos fatores [Prado and Leal, 2016]. Por exemplo, se houver 2 fatores ($k = 2$), serão realizadas 4 execuções experimentais, cobrindo todas as combinações possíveis. Essa metodologia permite Identificar efeitos principais (influência individual de cada fator), detectar interações entre fatores (como a combinação de dois fatores afeta o resultado) e planejar experimentos de forma eficiente, com o menor número possível de testes para obter informações relevantes. Neste experimento, serão analisados 2 fatores ($k = 2$) : o tamanho da amostra de treinamento e o tamanho do vocabulário do modelo.

3.4.3 Métricas de avaliação

A métrica utilizada nesse trabalho será a AUC-ROC (*Area Under the Receiver Operating Characteristic Curve*), uma das principais estratégias para avaliar modelos de classificação binária. Ela combina dois conceitos fundamentais:

- ROC Curve (*Receiver Operating Characteristic Curve*): é uma curva que mostra a relação entre a Taxa de Verdadeiros Positivos (TPR ou Sensibilidade) e a Taxa de Falsos Positivos (FPR = 1 − Especificidade), variando o limiar de decisão do classificador.
- AUC (*Area Under the Curve*): é a área sob a curva ROC, que resume em um único valor o desempenho do modelo para todos os limiares possíveis.

De acordo com a interpretação da métrica, um valor AUC = 1,0 implica um desempenho perfeito — o modelo separa completamente as classes positivas e negativas. um valor de AUC = 0,5 informa que o modelo não tem poder de discriminação — equivale a um palpite aleatório. Finalmente, um valor de AUC menor que 0,5 significa que o modelo está invertendo as classes (predizendo o oposto). Quanto maior o AUC, melhor o modelo em distinguir entre as duas classes.
A AUC tem sido tradicionalmente tratada como um critério importante para medir o desempenho de classificação de um modelo. Em geral, a maximização da AUC funciona bem quando existe uma distribuição de rótulos desbalanceada, ou a pontuação da AUC é a métrica padrão para avaliar e comparar diferentes métodos [Nguyen *et al*., 2023].

3.5 Parâmetros do Treinamento do modelo

Será analisado o impacto de dois fatores no desempenho do modelo escolhido : o tamanho do vocabulário e o tamanho da base de dados (em número de amostras). A base de dados terá três variações de tamanho : grande (5000) amostras, médio (2500 amostras) e pequeno (1000 amostras).
O vocabulário, por outro lado, será testado em duas modalidades:

- Vocabulário Original : vocabulário padrão usado pelo tokenizador que acompanha o modelo distilroberta-base. Possui 50265 tokens e foi gerado utilizando uma base de 160GB de texto, extraído de fontes como wikipedia, artigos de notícia em inglês e diversas outras páginas web [Liu *et al*., 2019]. Dessa forma, tem caráter genérico, contendo tokens relevantes para diversas áreas e assuntos.
- Vocabulário Reduzido : vocabulário criado para o experimento. Possui 5026 tokens (aproximadamente 10% do tamanho do vocabulário original) e foi gerado utilizando um corpus composto pelos extratos de texto contidos nas bases utilizadas. Dessa forma, o vocabulário utilizado no experimento com a base de abstratos médicos foi gerado usando esses mesmos abstratos como corpus. De forma análoga, o experimento com a base do IMDb usou um vocabulário gerado a partir dos textos de resenhas críticas de filmes, contidas na própria base. Dessa forma, temos um vocabulário reduzido e especializado, gerado a partir de textos relevantes para a área de classificação. Os *embeddings* do modelo foram reajustados para o novo tamanho do vocabulário.

Já que o vocabulário foi analisado em dois níveis (original e reduzido), para propósitos de realização do experimento fatorial 2^k, serão considerados apenas os tamanhos pequeno e grande para cada uma das bases de dados, representando os níveis mínimos e máximos, respectivamente. Dessa forma, possibilita-se que cada fator seja analisado igualmente em dois níveis, como é próprio do experimento 2^k.
Os hiperparâmetros (número de épocas, taxa de aprendizagem) foram ajustados separadamente para cada conjunto de dados utilizando o otimizador Optuna (biblioteca em python), que ajusta os parâmetros utilizando otimização bayesiana, mais especificamente TPE (do inglês *Tree Structured Parzen Estimator*). A métrica alvo da otimização foi o erro de validação (*validation loss*), sendo este minimizado, garantindo assim que o AUC-ROC obtido pelo treinamento fosse representativo do modelo com o melhor poder de generalização. O ajuste foi feito separadamente em cada *fold*, com 5 testes (*trials*) em cada um. O intervalo de busca para a taxa de aprendizagem foi de $1 * 10^{-6}$ a $1 * 10^{-4}$, e o número de épocas foi testado entre 3 e 7.

3.6 Diagrama do experimento

O experimento pode ser decomposto em 3 subexperimentos independentes, cujos resultados serão comparados ao final do processo. O diagrama dos 3 subexperimentos está ilustrado na Figura 3.

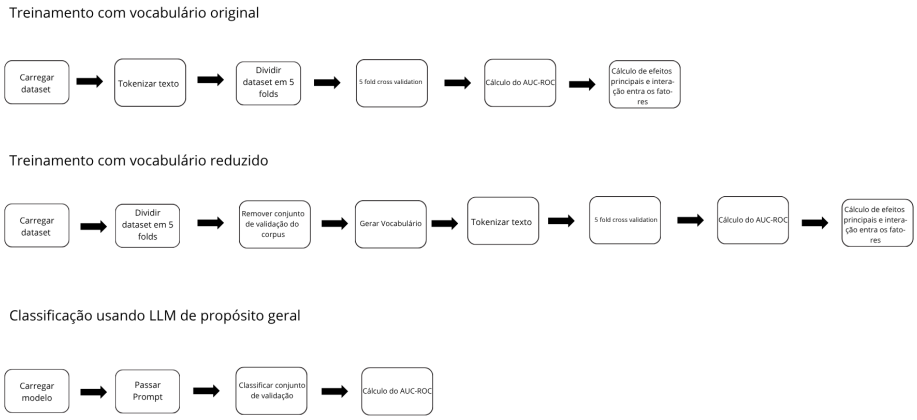


Figura 3. Diagrama do experimento.

Cada experimento é realizado para cada um dos 3 tamanhos de amostra selecionados, em cada um dos dois *datasets*. Os 3 sub-experimentos são explicados a seguir:

- Treinamento com vocabulário original : de acordo com o *k-fold* (k = 5), o texto de entrada é tokenizado com o vocabulário original do modelo e dividido em 5 partes. O treinamento é então realizado cinco vezes, alternando cada vez qual parte é usada como conjunto de validação, enquanto as outras 4 são usadas para o treinamento em si. Finalmente, os valores de AUC-ROC resultantes são usados para calcular os efeitos individuais e de interação entre os fatores (vocabulário e tamanho da base)
- Treinamento com vocabulário reduzido : o texto de entrada é dividido em 5 partes primeiro, e a tokenização é feita durante cada ciclo do *k-fold* (5 ciclos): durante cada ciclo, os textos pertencentes ao conjunto de validação são removidos do corpus que treina o tokenizador antes da geração do vocabulário, garantindo assim que não haja vazamento de dados. Feito isso, o vocabulário é gerado e os conjuntos de treinamento e validação são tokenizados, e o *k-fold* prossegue normalmente. Ao final, os valores de AUC-ROC resultantes são usados para calcular os efeitos individuais e de interação entre os fatores.
- Classificação usando LLM de propósito geral: usando uma LLM de propósito geral (GPT-3.5 turbo), classificamos o conjunto de validação, através de um *prompt* passado para o modelo. Dessa forma, comparamos o desempenho dos dois treinamentos realizados com o de um modelo popular genérico, sem qualquer *fine-tuning* para a tarefa de classificação desejada.

4 Resultados

4.1 Treinamento

As matrizes de confusão agregadas correspondentes para cada treinamento estão ilustradas nas Figuras 4 e 5 para o *dataset* médico, e nas Figuras 6 e 7 para o *dataset* do IMDb. Os quadrantes representam, da esquerda para a direita, de cima pra baixo: negativos-verdadeiros, falso-positivos, falso-negativos e positivos-verdadeiros. Com relação à acurácia (capacidade de classificar exemplos corretamente), precisão (capacidade de classificar exemplos positivos corretamente, evitando "alarmes falsos", ou falsos positivos) e *recall* (capacidade de encontrar todos os casos positivos), exibidos nas tabelas 1 (médico) e 2 (IMDb) observamos que as métricas precisão e acurácia tendem a aumentar de forma geral com o tamanho do *dataset*, independente da modalidade de vocabulário, embora os modelos com vocabulários reduzidos obtenham ganhos proporcionalmente mais expressivos que os vocabulários originais para a mesma base (ainda que o desempenho desses últimos seja maior). O recall, por outro lado, tem comportamento errático entre aumentos no tamanho da base de dados, tendo seu valor decrescido em alguns casos. Como as classes estão balanceadas e os *folds* são estratificados, esse comportamento não decorre de desbalanceamento, sendo na verdade causado apenas pelo aumento na fração de casos positivos que o modelo falha em classificar (falsos negativos).

O desempenho médio dos três modelos, para cada tamanho de *dataset*, pode ser visto na Figura 8. É possível observar que, para a classificação do *dataset* de abstratos médicos, o modelo DistilRoBERTa com o vocabulário original obteve o melhor desempenho, superando tanto o mesmo modelo com o vocabulário reduzido quanto até mesmo um modelo de propósito geral popular como o GPT-3.5 turbo. Além disso, nota-se que o DistilRoBERTa com vocabulário reduzido apresentou um desempenho inconsistente ao longo dos 5 *folds*, para os tamanhos de 1000 e 2500, com um desvio padrão (representado pelas linhas verticais) mais alto que as outras duas modalidades. Observa-se também que essa variante supera o GPT-3.5 turbo para os tamanhos médio e grande. Já na classificação do *dataset* de *reviews* do IMDb, o GPT-3.5 turbo e o modelo com vocabulário original apresentaram desempenhos próximos entre si (com o

GPT-3.5 turbo sendo ligeiramente melhor) e vastamente superiores ao modelo com vocabulário reduzido.

É interessante notar que a variante com vocabulário reduzido, apesar de apresentar desempenho inferior para todos os tamanhos de *dataset*, tem o seu próprio desempenho aumentado consistentemente com o aumento do tamanho do *dataset* para ambas as bases de dados, enquanto as outras duas modalidades mantêm seus desempenhos relativamente constantes para todos os três tamanhos escolhidos. É possível que, para uma quantidade suficientemente grande de dados, o vocabulário reduzido gere um desempenho próximo ou até mesmo superior às outras duas modalidades.

Os valores de AUC-ROC das 3 modalidades no *dataset* médico, ao longo dos 5 *folds*, são exibidos na Figura 9. Observa-se que, para todas as variações de tamanho da base, o vocabulário original prevalece em desempenho sobre o vocabulário reduzido. Para o *dataset* do IMDb, cujo desempenho por *fold* está exibido na Figura 10, o vocabulário genérico mostrou desempenho próximo ao GPT-3.5 turbo (equivalente ou ligeiramente inferior para o tamanho de 5000 (Figura 10c) e 2500 amostras (Figura 10b), e ambas as modalidades superam o vocabulário reduzido em todos os *folds*, para todos os tamanhos testados.

4.2 2^k fatorial e interação entre fatores

Os resultados para o experimento fatorial, de acordo com cada combinação de níveis para os dois fatores (vocabulário e tamanho da base), podem ser vistos para o *dataset* médico na Tabela 3, e para o *dataset* IMDb na Tabela 4. Com base nesses resultados, é possível calcular os efeitos individuais de cada fator e o efeito de interação entre os mesmos, mostrados na Tabela 5. Para o *dataset* médico, nota-se que há uma interação relativamente pequena entre os dois fatores, quando comparada aos seus efeitos individuais. O aumento no tamanho do *dataset* do nível mínimo para o nível máximo resulta em um aumento médio de 0.0668 no valor do AUC-ROC. Já a mudança do tamanho do vocabulário do mínimo (reduzido) para o máximo (genérico) resulta em um aumento médio de 0.1078 no valor do AUC-ROC.

Quanto ao *dataset* do IMDb, notamos um impacto mais de duas vezes maior no desempenho ao mudar o tipo de vocabulário, sendo essa também a contribuição predominante. O efeito de interação entre os fatores é semelhante em valor absoluto, porém tem contribuição proporcionalmente menor que na base médica. A tabela 6 mostra as parcelas individuais de cada fator com relação à soma dos quadrados total, indicando assim o papel relativo de cada fator na variação total dos dados. Para o *dataset* médico, vemos que 56.03% da variação total pode ser atribuída ao tamanho do vocabulário, enquanto 21.2% decorre do tamanho do *dataset*, e 22.7% da interação entre os dois. Por outro lado, no *dataset* do IMDb o tamanho do vocabulário contribui 79.90%, enquanto o tamanho do *dataset* tem um impacto muito menor (12.32%), e a interação entre os dois fatores tem uma contribuição relativamente menor (7.78%).

Tabela 1. Métricas das matrizes de confusão (*dataset* médico)

Vocab. / Tamanho	Original			Reduzido		
	Acurácia	Precisão	Recall	Acurácia	Precisão	Recall
1000	0.8730	0.8414	0.9148	0.6890	0.6750	0.7120
2500	0.8760	0.8619	0.8938	0.7264	0.7855	0.6187
5000	0.8728	0.8687	0.8784	0.8312	0.8275	0.8368

Tabela 2. Métricas das matrizes de confusão (*dataset* IMDb)

Vocab. / Tamanho	Original			Reduzido		
	Acurácia	Precisão	Recall	Acurácia	Precisão	Recall
1000	0.8410	0.8163	0.8742	0.5750	0.5895	0.4544
2500	0.8528	0.8196	0.9027	0.6168	0.6670	0.4578
5000	0.8532	0.8538	0.8524	0.7012	0.7135	0.6724

Tabela 3. Níveis de parâmetros máximo (+) e mínimo (-) e resultados correspondentes para o *dataset* de abstratos médicos

Vocabulário	Tamanho da Base	AUC-ROC
-	-	0.7568
-	+	0.8926
+	-	0.9336
+	+	0.9315

Tabela 4. Níveis de parâmetros máximo (+) e mínimo (-) e resultados correspondentes para o *dataset* de *reviews* de filmes

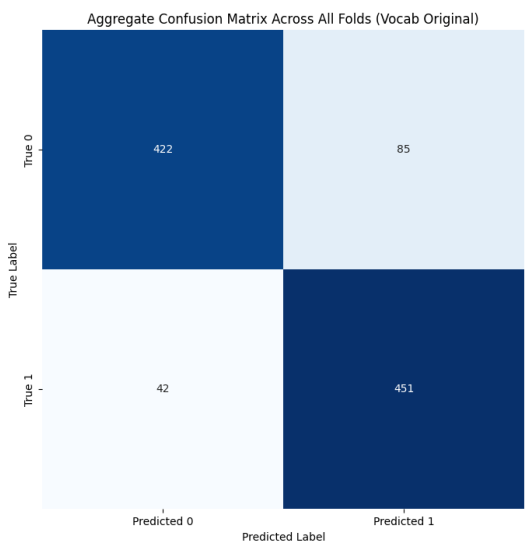
Vocabulário	Tamanho da Base	AUC-ROC
-	-	0.6193
-	+	0.7763
+	-	0.9109
+	+	0.9289

Tabela 5. Efeitos principais e interação entre os fatores

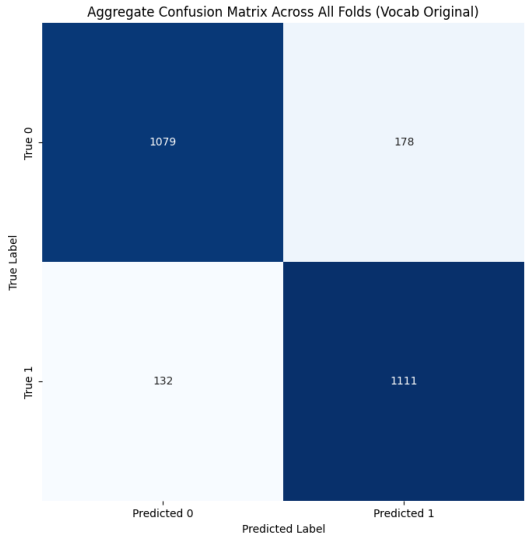
Base de Dados	Efeito Vocabulário	Efeito *Dataset*	Efeito de Interação
Base Médica	0.1078	0.0668	- 0.0689
Base IMDb	0.2221	0.0875	-0.0695

Tabela 6. parcelas dos efeitos (soma dos quadrados)

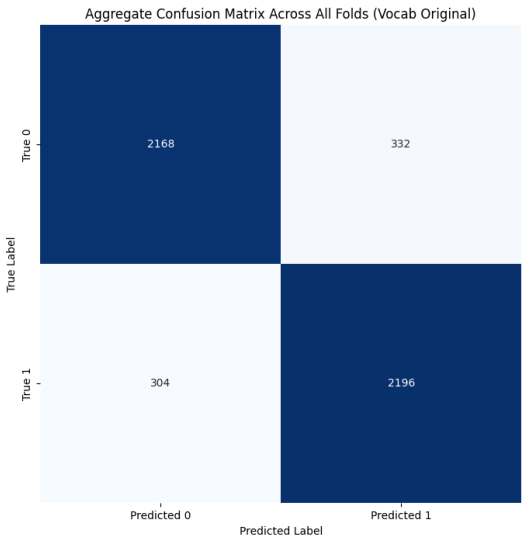
Base	Vocab.	*Dataset*	Interação	Total (soma)
Médica	0.0116	0.0044	0.0047	0.0207
IMDb	0.0493	0.0076	0.0048	0.0617



(a) matriz de confusão (1000 amostras)

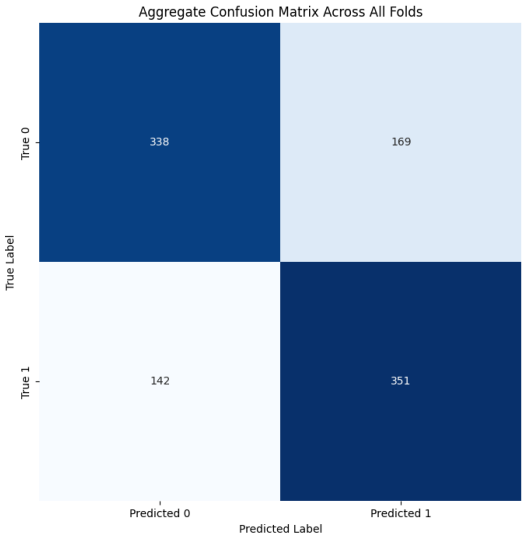


(b) matriz de confusão (2500 amostras)

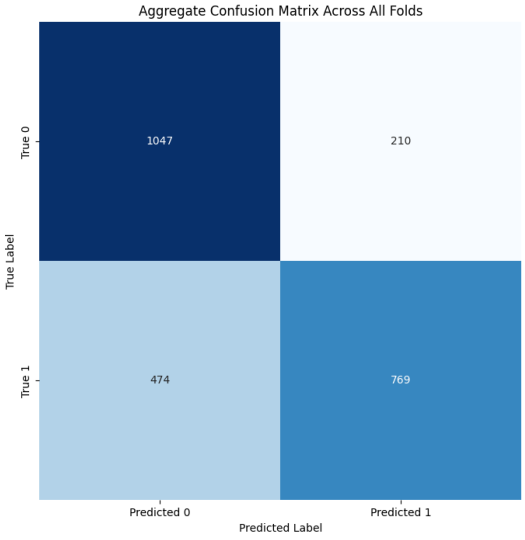


(c) matriz de confusão (5000 amostras)

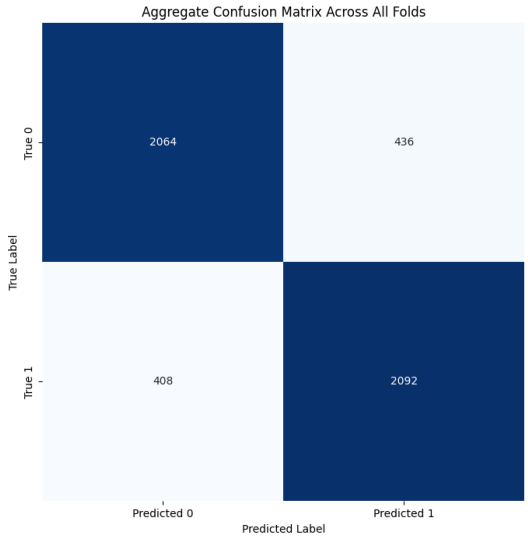
Figura 4. Matrizes de confusão agregadas (*dataset* medico c vocab original)



(a) matriz de confusão (1000 amostras)

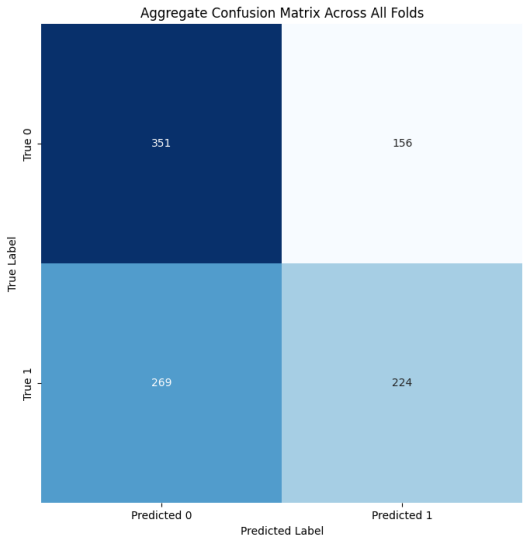


(b) matriz de confusão (2500 amostras)

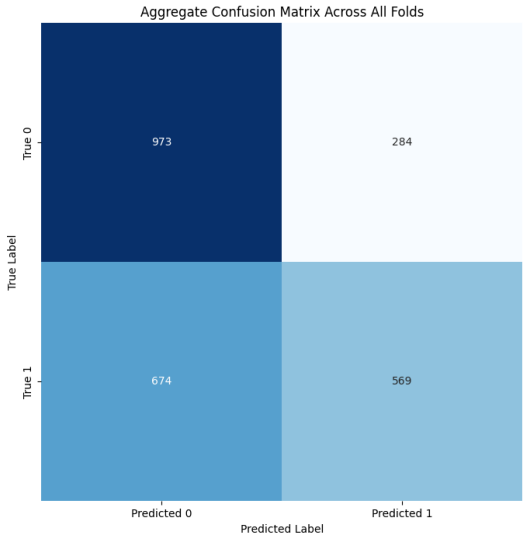


(c) matriz de confusão (5000 amostras)

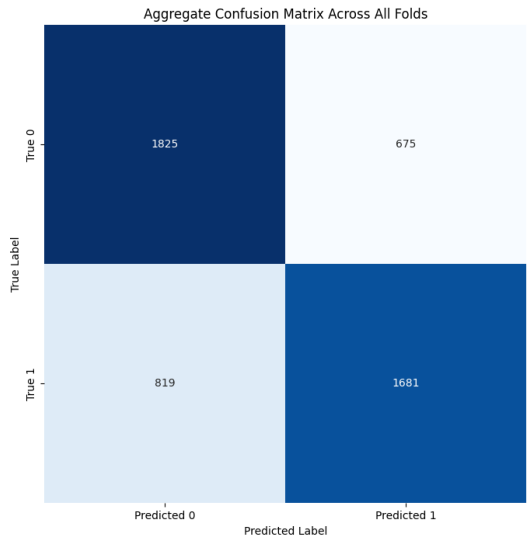
Figura 5. Matrizes de confusão agregadas (*dataset* médico c/ vocab. reduzido)



(a) matriz de confusão (1000 amostras)

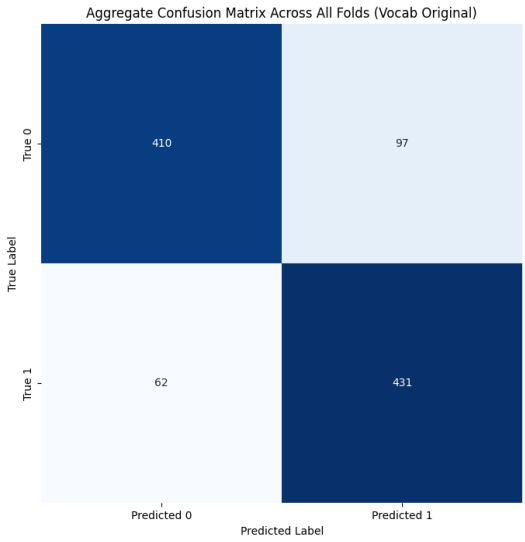


(b) matriz de confusão (2500 amostras)

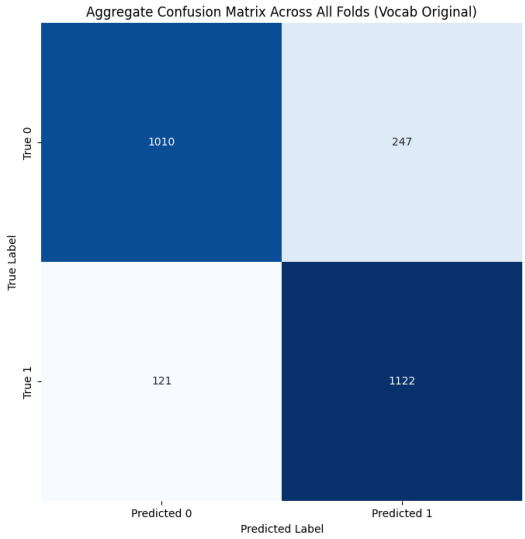


(c) matriz de confusão (5000 amostras)

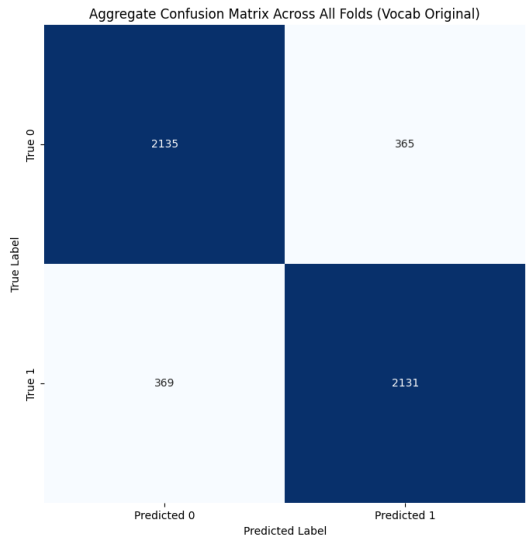
Figura 6. Matrizes de confusão agregadas (*dataset* IMDb c/ vocab. reduzido)



(a) matriz de confusão (1000 amostras)

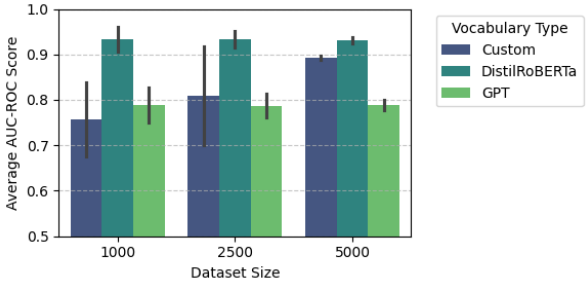


(b) matriz de confusão (2500 amostras)

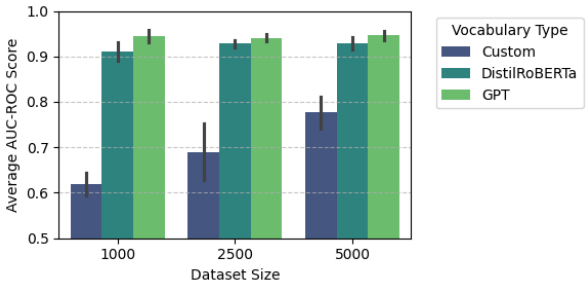


(c) matriz de confusão (5000 amostras)

Figura 7. Matrizes de confusão agregadas (*dataset* IMDb c/ vocab. original)

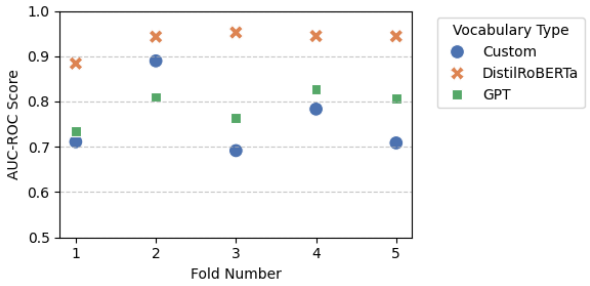


(a) Valor AUC-ROC médio (*dataset* médico)

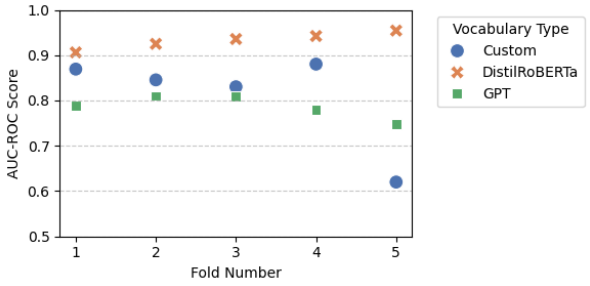


(b) Valor AUC-ROC médio (*dataset* do IMDb)

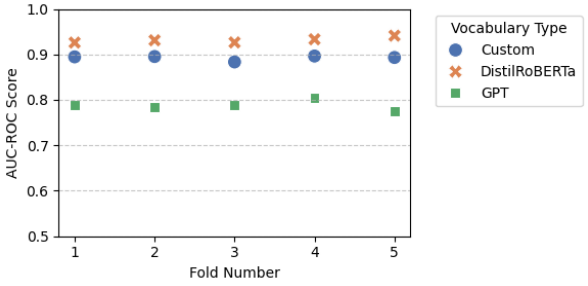
Figura 8. Valor AUC-ROC para ambos os *datasets* (média dos 5 *folds*)



(a) Valores AUC-ROC por *fold* (1000 amostras)

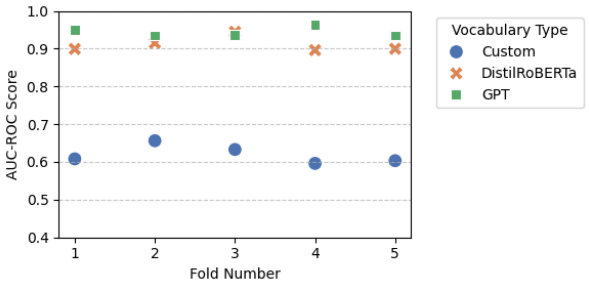


(b) Valores AUC-ROC por *fold* (2500 amostras)

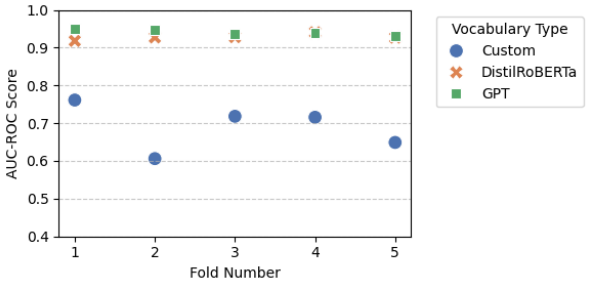


(c) Valores AUC-ROC por *fold* (5000 amostras)

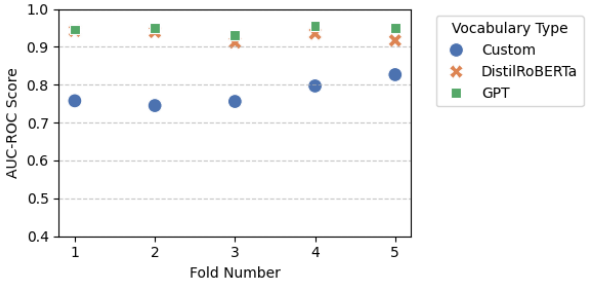
Figura 9. Valores AUC-ROC por *fold* para cada tamanho do *dataset* médico. O termo "Custom"(legenda) se refere ao vocabulário reduzido



(a) Valores AUC-ROC por *fold* (1000 amostras)



(b) Valores AUC-ROC por *fold* (2500 amostras)



(c) Valores AUC-ROC por *fold* (5000 amostras)

Figura 10. Valores AUC-ROC por *fold* para cada tamanho do *dataset* do IMDb. O termo "Custom"(legenda) se refere ao vocabulário reduzido

5 Conclusão

Este trabalho investigou o impacto do tamanho da base de treinamento e do tamanho do vocabulário no desempenho de modelos de classificação de texto, utilizando o DistilRoBERTa como modelo-base e comparando seus resultados com os obtidos por uma LLM de propósito geral, o GPT-3.5-turbo. Para isso, foram conduzidos experimentos em dois conjuntos de dados distintos, um composto por resumos de artigos médicos e outro por avaliações de filmes do IMDb, avaliados por meio da métrica AUC-ROC e analisados através de validação cruzada *k-fold* e planejamento fatorial 2^k. Os experimentos foram realizados para três tamanhos de *dataset* e duas modalidades de vocabulário.

Os resultados indicam que a generalização do vocabulário pode ter influência expressiva no desempenho em domínios específicos. No conjunto de resumos médicos, o DistilRoBERTa com vocabulário genérico obteve AUC-ROC médio em torno de 0.93 (em todos os tamanhos de base), superior ao obtido com o vocabulário reduzido (0.76 com 1000 amostras a 0.89 com 5000 amostras) e ao do GPT-3.5-turbo (aproximadamente 0.79), além de menor variabilidade entre os *folds*. Na análise fatorial (Seção 4.2), a troca do vocabulário reduzido pelo original elevou o AUC-ROC em 0.1078 em média, contra 0.0668 associado ao aumento da amostra de 1000 para 5000, e o vocabulário respondeu por 56.03% da variação total, contra 21.2% do tamanho da amostra e 22.7% da interação.

No IMDb, o tamanho do vocabulário teve influência muito maior do que a amostra: o aumento de 1000 para 5000 amostras elevou o AUC-ROC em 0.0875 em média, contra 0.2221 associado à troca para o vocabulário original, e o tamanho do vocabulário respondeu por 79.90% da variação total, contra 12.32% da amostra e 7.78% da interação (Tabela 6).

Os valores de AUC-ROC dos modelos ajustados mostram que o melhor desempenho foi o do modelo com o vocabulário genérico e maior, tendo este desempenho pouco se alterado com a alteração no tamanho da amostra e sido próximo ao modelo GPT-3.5 turbo para a base do IMDb. A modalidade com vocabulário reduzido obteve ganhos expressivos no AUC-ROC à medida que se aumentou o tamanho da amostra, tendo até mesmo superado eventualmente o GPT-3.5 turbo para a base médica, embora seu desempenho tenha se mantido menor do que os demais modelos na base do IMDb.

De forma geral, os resultados indicam que a simples utilização de vocabulários extensos e genéricos, quando em posse de um conjunto de dados relativamente pequeno, pode conduzir ao melhor desempenho em tarefas de classificação de texto em domínios específicos (como o de classificação médica). Os resultados também evidenciam que o impacto relativo dos fatores analisados depende fortemente das características do conjunto de dados, do domínio da aplicação e da otimização dos hiper-parâmetros para cada domínio.

Como limitações, a base médica foi binarizada (neoplasmas contra as demais classes), o que simplifica o problema clínico original, e a análise fatorial foi realizada sem teste estatístico.

Declarações complementares

Contribuições dos autores

IATA foi responsável pela realização integral do estudo, incluindo o desenho experimental, a execução dos experimentos, a análise dos resultados e a redação do manuscrito. Os demais autores participaram da concepção do trabalho e da revisão do texto. Todos os autores leram e aprovaram a versão final.

Conflitos de interesse

Os autores declaram que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

Os dados utilizados neste estudo foram disponibilizados em formato CSV sob o nome de arquivo ***rica-v1-n1-tenorio-et-al-2026-data.csv*** *juntamente com o código-fonte dos experimentos conduzidos, em formanto Interactive Python Notebook sob o nome de arquivo* ***rica-v1-n1-tenorio-et-al-2026-experimento.ipynb****. Ambos os arquivos encontram-se no repositório da revista.*

Uso de Inteligência Artificial

Ferramentas de inteligência artificial generativa (TeXGPT, através do editor LaTeX Overleaf) foram utilizadas apenas como apoio à revisão textual e à formatação do manuscrito. Os autores revisaram todo o conteúdo e assumem total responsabilidade pelo trabalho apresentado.

Referências

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.*, 52(1):273–292.

Kostina, A., Dikaiakos, M. D., Stefanidis, D., and Pallis, G. (2025). Large language models for text classification: Case study and comprehensive review.

Kyritsis, K., Spatiotis, N., Perikos, I., and Paraskevas, M. (2024). A comparative performance evaluation of algorithms for the analysis and recognition of emotional content. In *Artificial Intelligence*. IntechOpen.

Liddy, E. D. (2001). Natural language processing. In *Encyclopedia of Library and Information Science*, volume 2, pages 2126–2136. Marcel Dekker, Inc., New York, 2nd edition.

Liu, Y. *et al*. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mazzullo, E., Bulut, O., Walsh, C., Sitarenios, G., and MacIntosh, A. (2025). Fine-tuning GPT-3.5-turbo for automatic feedback generation. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 40–47, New York, NY, USA. ACM.

Mielke, S. J. *et al*. (2021). Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. arXiv preprint.

Minaee, S. *et al*. (2024). Large language models: A survey. arXiv preprint.

Nakash, I., Calderon, N., Ben David, E., Hoffer, E., and Reichart, R. (2025). AdaptiVocab: Enhancing LLM efficiency in focused domains through lightweight vocabulary adaptation.

Nguyen, N. D. *et al*. (2023). Low-resource named entity recognition: Can one-vs-all auc maximization help? In *2023 IEEE International Conference on Data Mining (ICDM). IEEE*, pages 1241–1246.

Prado, L. C. and Leal, B. G. (2016). Meteorological variables influence analysis over petrolina-juazeiro pole monthly evapotranspiration. *IEEE Latin America Transactions*, 14(10):4373–4376.

Reshamwala, A., Mishra, D., and Pawar, P. (2013). Review on natural language processing. *IRACST Engineering Science and Technology: An International Journal (ESTIJ)*, 3(1):113–116.

Rhoden, E. B. (2025). Exploring active sampling strategies for self-training in named entity recognition.

Schopf, T., Braun, D., and Matthes, F. (2022). Evaluating unsupervised text classification: Zero-shot and similarity-based approaches. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, pages 6–15, New York, NY, USA. ACM.

Shah, M. S. S., Abuaieta, A. M., and Almazrouei, S. S. (2024). Safeguarding online communications using DistilRoBERTa for detection of terrorism and offensive chats. *J. Inf. Secur. Cybercrimes Res.*, 7(1):93–107.

Vajjala, S. and Shimangaud, S. (2025). Text classification in the LLM era – where do we stand?

Vaswani, A. *et al*. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wankhade, M., Rao, A. C. S., and Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.*, 55(7):5731–5780.

Zhao, W. X. *et al*. (2023). A survey of large language models. arXiv preprint v. 1, n. 2.