Campus Vilhena

Coordenação do Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas

JAINA DADY EGLER CARDOSO

Análise de sentimentos no YouTube

VILHENA - RO

2026

Jaina Dady Egler Cardoso

Análise de sentimentos no YouTube

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO, Campus Vilhena, como requisito parcial para a obtenção do título de Tecnóloga em Análise e Desenvolvimento de Sistemas.

Orientador: José Lucas Brandão Montes

Vilhena - RO

2026

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Cardoso, Jaina Dady Egler.
Análise de sentimentos no YouTube / Jaina Dady Egler Cardoso. - Vilhena, 2026.
56 f. : il.

Orientador(a): Prof. Esp. Jose Lucas Brandao Montes.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Vilhena, 2026.

1. Análise de sentimentos. 2. YouTube. 3. BERTimbau. 4. Linguagem natural. 5. PLN. I. Montes, Jose Lucas Brandao (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Rosilene Maria do Couto Marques, CRB-11/321

Jaina Dady Egler Cardoso

Análise de sentimentos no YouTube

Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO, Campus Vilhena, como requisito parcial para a obtenção do título de Tecnóloga em Análise e Desenvolvimento de Sistemas.

Trabalho aprovado. Vilhena - RO, 09 de abril de 2026

Documento assinado digitalmente
gov.br
JOSE LUCAS BRANDAO MONTES
Data: 14/05/2026 16:42:37-0300
Verifique em https://validar.iti.gov.br

José Lucas Brandão Montes
Orientador

Documento assinado digitalmente
gov.br
GILBERTO PEREIRA DA SILVA
Data: 14/05/2026 16:48:15-0300
Verifique em https://validar.iti.gov.br

Gilberto Pereira da Silva
Convidado 1

Documento assinado digitalmente
gov.br
WESLEY JHONNES RAMOS ROLIM
Data: 14/05/2026 16:53:23-0300
Verifique em https://validar.iti.gov.br

Wesley Jhonnes Ramos Rolim
Convidado 2

Vilhena - RO
2026

Dedico este trabalho a todos aqueles que me fizeram não desistir dele.

Agradecimentos

Agradeço, antes de tudo, a minha mãe e minha irmã — por estarem comigo em todos os momentos, nos bons e, principalmente, nos difíceis. Pela fé que tiveram em mim mesmo quando eu mesma hesitava.

Aos colegas que caminharam ao meu lado e, com muita paciência e bom humor, suportaram minhas brincadeiras.

Aos professores, meu sincero agradecimento pela generosidade em ensinar e pela paciência nos tropeços. Foram fundamentais para que eu chegasse aqui.

E por fim, agradeço a mim mesma por não ter desistido, por continuar, por me perdoar nas falhas e seguir em frente, passo a passo. Chegar até aqui foi também um ato de coragem.

Resumo

A análise de sentimentos é uma das aplicações mais úteis da inteligência artificial no mundo digital atual. Com o grande volume de comentários gerados em plataformas como o YouTube, torna-se inviável fazer essa leitura de forma manual. Este trabalho teve como objetivo personalizar um modelo de linguagem pré-treinado, o BERTimbau, para interpretar sentimentos em comentários de vídeos em português. Foram utilizados três conjuntos de dados: IMDB PT-BR, Comentários Tóxicos e Tweets Ekman. Após treinamento e testes, o modelo demonstrou bons resultados na identificação de sentimentos negativos, especialmente quando treinado com dados mais próximos da linguagem informal das redes sociais. O processo envolveu coleta automatizada via API do YouTube, limpeza dos dados, tokenização e aplicação de técnicas de aprendizado profundo. O trabalho também discutiu limitações como o tempo elevado de treinamento e os desafios na interpretação de ironia e ambiguidade. Ainda assim, os resultados mostram que é possível aplicar inteligência artificial de forma prática para entender a opinião pública em larga escala.

Palavras-chave: análise de sentimentos, YouTube, BERTimbau, linguagem natural, PLN.

Abstract

Sentiment analysis is one of the most useful applications of artificial intelligence in today's digital world. With the massive amount of comments posted on platforms like YouTube, manual analysis becomes unfeasible. This study aimed to fine-tune a pre-trained language model, BERTimbau, to interpret sentiments in Portuguese video comments. Three datasets were used: IMDB PT-BR, Toxic Comments PT-BR, and Tweets Ekman. After training and testing, the model showed good results in detecting negative sentiments, especially when using data closer to the informal language found on social media. The process involved automated comment collection through the YouTube API, data cleaning, tokenization, and deep learning techniques. The study also addressed limitations such as long training time and challenges in detecting irony and ambiguity. Nevertheless, the results demonstrate that artificial intelligence can be effectively applied to understand public opinion at scale.

Keywords: sentiment analysis, YouTube, BERTimbau, natural language processing, NLP.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	Application Programming Interface
PLN	Processamento de linguagem natural
BERT	Bidirectional Encoder Representations from Transformers
CLS	Class Token
IA	Inteligência Artificial
RPA	***Robotic Process Automation***.
Precisão	Proporção de acertos entre os itens classificados como positivos pelo modelo, indicando o quanto ele evita falsos positivos
Recall	Proporção de itens positivos corretamente identificados pelo modelo, indicando o quanto ele consegue encontrar todos os casos relevantes
F1-score	Métrica que combina precisão e recall, avaliando de forma equilibrada a capacidade do modelo em classificar corretamente os dados.

Sumário

1 Introdução

1.1 Contexto e problema

A internet ampliou não apenas o acesso à informação, mas também as formas de expressão. Plataformas como o YouTube se tornaram espaços em que milhões de pessoas compartilham opiniões diariamente sobre vídeos, campanhas, acontecimentos públicos, marcas, produtos e figuras públicas. Nesse contexto, realizar manualmente a análise das reações de um canal no YouTube pode ser uma tarefa complexa e demorada, devido ao grande volume de conteúdo publicado e à quantidade de comentários gerados. Um único vídeo pode receber milhares de comentários, e um canal pode reunir centenas ou milhares de vídeos, tornando inviável uma leitura individual e manual de todas as opiniões. Além do volume de dados, há também desafios relacionados à própria linguagem utilizada nas redes sociais. Comentários publicados na internet costumam apresentar abreviações, gírias, erros de escrita, ironias, emojis e expressões que mudam conforme o contexto social e temporal. Uma expressão que em determinado momento funciona como elogio pode, em outro contexto, ser usada como crítica ou deboche. Essas características tornam a interpretação automática de sentimentos uma tarefa mais difícil. A análise desses comentários pode ser realizada com o apoio de técnicas de Inteligência Artificial, especialmente por meio de modelos de aprendizado profundo voltados ao Processamento de Linguagem Natural. No entanto, encontrar bases de dados adequadas para treinar esses modelos ainda é um desafio, principalmente quando se trata da linguagem informal utilizada em redes sociais e plataformas de vídeo. Diante disso, este trabalho propõe avaliar a aplicação de um modelo de linguagem baseado no BERTimbau para a interpretação de sentimentos em comentários do YouTube, considerando tanto suas possibilidades de uso quanto suas limitações técnicas e linguísticas.

1.2 Objetivos

1.2.1 Objetivo geral

Aprimorar um modelo de linguagem existente para que seja possível analisar os sentimentos em relação aos vídeos de um canal no YouTube.

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Analisar as ferramentas disponíveis para o desenvolvimento do trabalho;
- Desenvolver um *bot* para capturar os comentários de um canal no YouTube;
- Iniciar testes com examinadores de sentimentos;
- Refinar a análise de sentimentos feita pelo modelo;
- Validar as métricas de avaliação do modelo de deep learning.

1.3 Justificativa

O uso estratégico de IA (Inteligência Artificial) por empresas de médio a grande porte já é realidade. De acordo com um estudo feito pela (IBM, 2023) 41% das empresas brasileiras usam IA. Entre os motivos para o uso, se destaca a falta de mão de obra ou de habilidades. Ao automatizar tarefas, os trabalhadores podem se dedicar a outras atividades mais produtivas.

A análise de sentimentos é o processo de utilização do Processamento de Linguagem Natural (PLN) para identificar emoções ou sentimentos em frases, por meio de técnicas de inteligência artificial (DIAS, 2021). Esse processo pode ser útil para entender os sentimentos do cliente em relação a produtos e serviços.

A justificativa deste projeto está na necessidade de uma abordagem automatizada e eficiente para análise de sentimentos em redes sociais, diante do enorme volume de conteúdo gerado online. Para atender a essa demanda, propõe-se o desenvolvimento de uma ferramenta capaz de aplicar técnicas de inteligência artificial para analisar comentários de forma prática e escalável. Com isso, será possível extrair percepções e tendências de forma mais rápida, subsidiando decisões em áreas como marketing e pesquisa de mercado, e superando as limitações da análise manual.

Um dos casos de sucesso da análise de sentimentos ocorreu quando a Nike, empresa norte-americana de calçados, roupas e acessórios, apoiou o jogador de futebol americano Colin Kaepernick em 2018. Segundo (Hotjar, 2022), Kaepernick se ajoelhou durante o hino nacional dos Estados Unidos em protesto contra a injustiça racial. Embora inicialmente o ato tenha sido interpretado por parte do público como desrespeitoso, a Nike decidiu apoiar o jogador em uma campanha associada à hashtag #justdoit. Nesse contexto, a empresa precisou acompanhar as reações nas mídias

sociais, especialmente diante das manifestações negativas iniciais. Ao manter seu apoio ao jogador, observou-se uma mudança na percepção pública, que passou a considerar o gesto como uma manifestação contra a injustiça racial e a violência policial. Essa estratégia não apenas contribuiu para a imagem da marca, mas também resultou em impacto financeiro positivo, como demonstrado na repercussão apresentada na Figura 1. Assim, o caso evidencia que soluções baseadas em análise de sentimentos, assim como outras aplicações de inteligência artificial, podem auxiliar organizações na interpretação da opinião pública e na tomada de decisões estratégicas.

Figura 1 – Notícia sobre o impacto da análise de sentimentos em campanhas publicitárias da Nike


ECONOMIA

Nike 'ganha' US$ 43 milhões com repercussão de campanha com Colin Kaepernick

Esse é o valor que a empresa acumula em exposição na mídia, segundo o Apex Marketing Group

Bloomberg

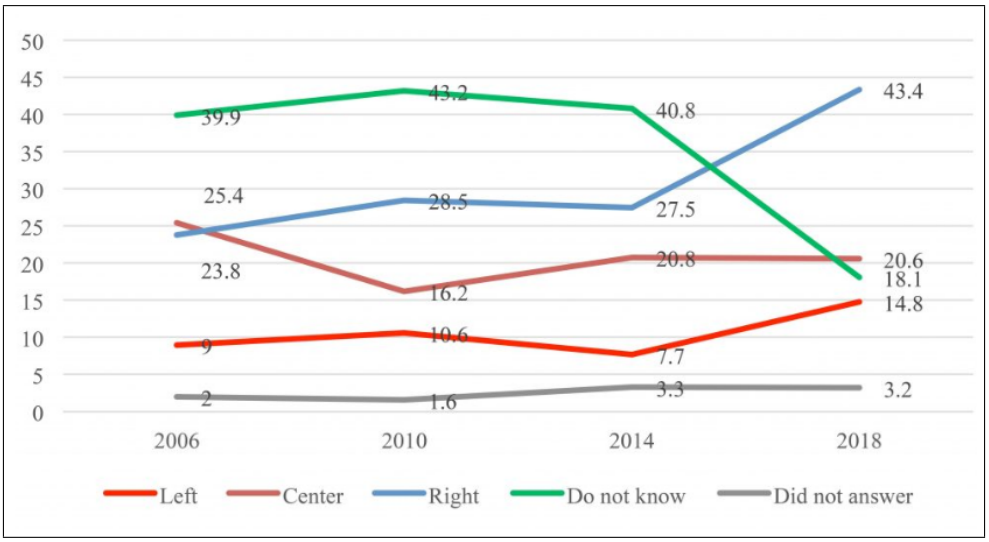
O ex-quarterback do San Francisco 49ers, Colin Kaepernick (centro), protagoniza a campanha da gigante esportiva Nike Foto: USA Today Sports / USA Today Sports

Fonte: (O GLOBO, 2018)

A escolha por comentários provenientes de um canal de conteúdo político justifica-se pelo fato de que esse tipo de ambiente tende a concentrar manifestações discursivas com maior carga opinativa, especialmente em contextos marcados por polarização ideológica, identificação partidária e rejeição a determinados grupos políticos. No caso brasileiro, estudos sobre comportamento eleitoral apontam a relevância do petismo e do antipetismo na organização das preferências políticas dos eleitores, além de indicarem maior explicitação do autoposicionamento ideológico ao longo dos anos (AMARAL, 2020). Conforme apresentado na Figura 2, observa-se uma mudança relevante na distribuição do autoposicionamento ideológico entre 2006 e 2018, com aumento da identificação à direita e redução expressiva das respostas "não sabe", o que evidencia um cenário de maior definição política entre os respondentes. Além disso, a comunicação política passou a utilizar com frequência as plataformas digitais como espaço de mobilização, disputa de narrativas e expressão pública

de opiniões (SILVA, 2019). Assim, optou-se pela análise de comentários do canal do presidente Luiz Inácio Lula da Silva, não com o objetivo de avaliar atores ou posicionamentos políticos, mas por se tratar de um ambiente propício à ocorrência de comentários com polaridades mais evidentes, favorecendo a aplicação e avaliação de modelos de análise de sentimentos.

Figura 2 – Autoposicionamento ideológico dos eleitores brasileiros entre 2006 e 2018



Fonte: (AMARAL, 2020).

Pode-se concluir, então, que a análise de sentimentos, embora seja um campo relativamente recente, já demonstrou seu valor em aplicações práticas. A aplicação dessa técnica tem se mostrado eficaz na compreensão das emoções e opiniões expressas em grandes volumes de dados, como nas redes sociais e em plataformas de vídeo, como o YouTube.

Apesar dos avanços apresentados na literatura e dos resultados obtidos em aplicações práticas, a análise de sentimentos em plataformas como o YouTube ainda apresenta desafios específicos, principalmente devido às características da linguagem utilizada nos comentários, como informalidade, ironia e uso frequente de símbolos, abreviações e gírias.

Este trabalho visa aprimorar um modelo de linguagem existente para analisar sentimentos em comentários de vídeos no YouTube. Para isso, são apresentados conceitos fundamentais de Processamento de Linguagem Natural, redes neurais, modelos Transformers e BERT, além das etapas práticas de coleta, preparação, treinamento e avaliação do modelo. A proposta busca verificar as possibilidades e

limitações da aplicação de modelos de linguagem em textos informais publicados na plataforma.

2 Fundamentação teórica

Antes de iniciar a pesquisa em si, é importante que se tenha uma compreensão de alguns conceitos fundamentais da área de Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN). Esses conceitos fornecem a base necessária para um entendimento mais aprofundado dos métodos e técnicas que serão explorados ao longo do estudo, além de possibilitar uma análise crítica e embasada dos resultados obtidos.

2.1 Como funciona a PLN

Desde a primeira revolução industrial, máquinas vem sendo usadas para substituir trabalhos executados por humanos, e os motivos vão desde a rapidez das máquinas para trabalhos repetitivos a execução de trabalhos complexos. De acordo com (MAGALHãES; VENDRAMINI, 2018) pode-se dizer que o uso de máquinas para realização de trabalho humano começou na primeira revolução industrial com a máquina a vapor, e agora, uma das principais novidades é a IA, que além de muitas outras tarefas, poderá lidar com uma das mais complexas tecnologias humanas: a fala.

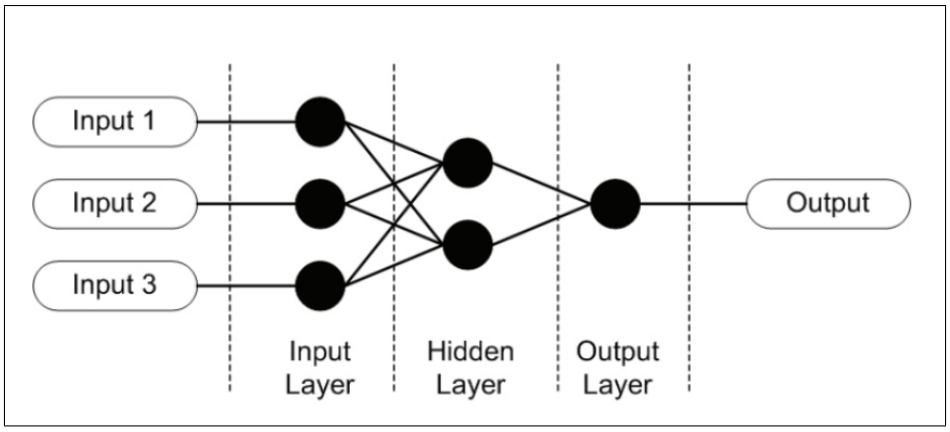
> "A PLN ou Processamento de Linguagem Natural é um dos diversos campos da Inteligência Artificial no qual os computadores são responsáveis por analisar, entender e derivar significados da linguagem humana de uma maneira inteligente e útil."(DIAS, 2021). Ou seja, para realizar os processos linguísticos normalmente feitos por humanos, a IA necessita imitar a construção da inteligência humana. Isso é feito por meio da utilização de neurônios artificiais, que são imitações de neurônios biológicos, porém, esses se tratam de modelos matemáticos.

Na figura 3 é exemplificada a estrutura de uma rede neural simples, que possui 3 camadas:

- Input layer (ou camada de entrada) É a camada que recebe os dados;
- Hidden Layer (ou camada escondida): É a camada de processamento dos dados;
- Output Layer (ou camada de saída): É a camada que obtém o resultado final dos dados processados.

Também é possível notar a esquerda da imagem 3 inputs (ou entradas). Cada ponto preto na imagem representa um neurônio e as linhas são as conexões entre eles.

Figura 3 – Rede neural



Fonte: (SUZUKI, 2011)

Então, de acordo com a estrutura, existem três inputs (entradas) que passam por três camadas, cada uma composta por seus neurônios. Cada conexão entre neurônios possui um peso, que define a importância daquela entrada no resultado final. À medida que os dados passam pelas camadas, essas entradas são multiplicadas pelos respectivos pesos e somadas, gerando o output (saída). O treinamento do modelo ocorre justamente pelo ajuste desses pesos, até que a rede aprenda a produzir saídas mais próximas do esperado. Após analisar a imagem pode-se concluir que:

> Uma rede neural é composta por um elevado número de elementos processadores, os neurônios, amplamente interligados através de conexões com um determinado valor que estabelece o grau de conectividade entre estes, denominado peso da conexão ou sinapse (FURTADO, 2019)

2.1.1 Encoding

Para (CECCON, 2019) um dos desafio da PLN é o processamento de palavras pois computadores, em última análise trabalham com números. Ao receber um texto, os modelos de PLN realizam o processo de **encoding**, no qual as palavras são convertidas em representações numéricas que podem ser processadas pelo modelo.

2.1.2 One-hot encoding

O One-hot encoding é o processo de transformação de palavras de um corpus em vetores. Esses vetores são caracterizados por terem posições binárias (0,1), sendo que, apenas uma posição terá o 1. Esse processo já auxilia o computador a processar

as palavras, porém, as palavras vetorizadas não tem relação. Outro problema é que quanto maior a quantidade de palavras, maior será o vetor.

2.1.3 Word embedding

O Word Embedding, assim como o One-hot encoding, realiza a transformação de palavras de um corpus, ou seja, do conjunto de textos analisados pelo modelo, em vetores numéricos. Diferentemente do One-hot encoding, essa técnica utiliza números reais, permitindo a preservação das relações semânticas entre as palavras.

2.1.4 One-hot encoding x Word embedding

Abaixo, será explicado visualmente a diferença entre as duas maneiras de encoding de maneira simplificada. Será utilizada a frase "O robô consertou o computador".

Figura 4 – Exemplo de encoding com One-hot encoding

	o	robô	consertou	computador.
o	1	0	0	0
robô	0	1	0	0
consertou	0	0	1	0
computador.	0	0	0	1

Fonte: A autora, 2024

A imagem 4 acima exemplifica uma matriz em que a frase de exemplo foi disposta para fazer o one-hot encoding. As linhas e colunas contém as mesmas palavras: "o", "robô", "consertou"e "computador". É importante notar os seguintes pontos na imagem acima:

- Redução de artigos: A letra 'O' aparece duas vezes na frase, porém, caso fosse utilizada duas vezes na composição da matriz, o número 1 ficaria duplicado, então ela é considerada apenas uma vez.
- Tamanho da matriz: As colunas e linhas da matriz são feitas com base nas palavras únicas, o que significa que quanto mais palavras, maiores serão os vetores gerados.
- Após o processo, são obtidos os seguintes vetores:

o = [1,0,0,0]

robô = [0,1,0,0]

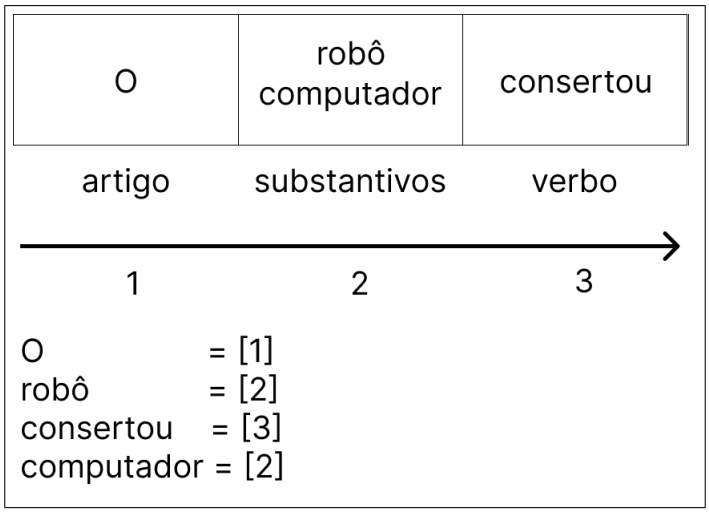
consertou = [1,0,1,0]

computador = [1,0,0,1]

Após o experimento, pode-se concluir que o objetivo de encodificar foi concluído, porém, com os vetores fornecidos não é possível estabelecer relações entre as palavras, sendo assim, para a análise de sentimentos o One-hot encoding não é a abordagem mais eficiente.

Agora, será utilizado o Word Embedding, e para isso, será feita uma abordagem diferente, serão escolhidas algumas dimensões para classificação das palavras, são elas: classes gramaticais, se é relacionado a TI (Tecnologia da Informação) e, por último, a autonomia.

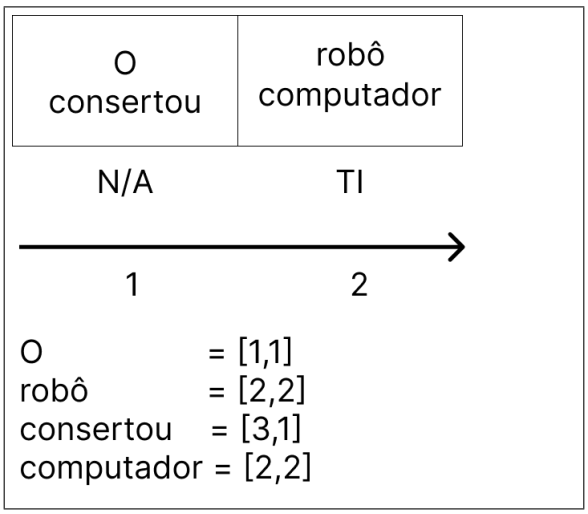
Figura 5 – Exemplo de disposição de palavras vetorizadas por word embedding por classe gramatical



Fonte: A autora, 2024

Na figura 5 acima as palavras são dispostas de acordo com sua categorização usando a dimensão de classe gramatical, "o"é o único artigo da palavra e está na primeira posição do vetor criado, então recebe [1], já "robô"e "computador"são substantivos, e recebem [2], e, por fim, consertou é um verbo que está na terceira posição, então recebe [3]. Com isso, é finalizada a primeira classificação.

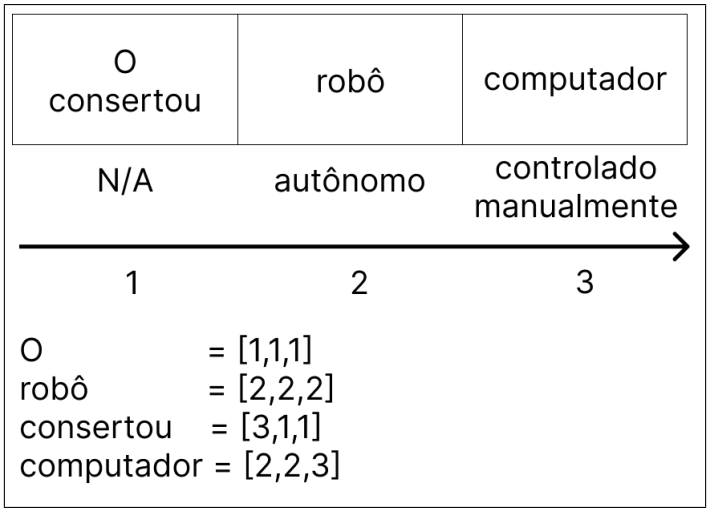
Figura 6 – Exemplo de disposição de palavras vetorizadas por word embedding por relacionamento com TI



Fonte: A autora, 2024

Na imagem acima 6 é demonstrada a classificação utilizando como critério a dimensão TI, que separa as palavras que têm relação direta a tecnologia da informação e as que não possuem essa classificação ficam como N/A. Os arrays agora ficam da seguinte forma "o"= [1,1], "robô"= [2,2], consertou = [3,1] e computador = [2,2].

Figura 7 – Exemplo de disposição de palavras vetorizadas por word embedding por autonomia



Fonte: A autora, 2024

7 Por último, será feita a classificação de acordo com o critério autonomia. O robô, por ter consertado o computador, pode ser considerado autônomo; já o computador não. Já as palavras 'o' e 'consertou' não se encaixam em nenhuma categorização e classificaram como N/A. Os arrays agora ficam da seguinte forma "o"= [1,1,1], "robô"= [2,2,2], consertou = [3,1,1] e computador = [2,2,3]. Com isso, pode-se notar a seguinte característica:

- Apesar de se tratar de uma frase curta, existe relação entre algumas palavras, por exemplo, o vetor de "robô"= [2,2,2] e o vetor de computador = [2,2,3] estão bem próximos, com diferença apenas na última posição.

Na tabela 1 são resumidas as diferenças desses diferentes tipos de codificação para uso em modelos de PLN:

Característica	**One-hot Encoding**	**Word Embedding**
Tipo de representação	Vetores binários	Vetores de números **reais**
Dimensionalidade	Alta, igual ao tamanho do vocabulário	Reduzida e fixa
Relação semântica entre palavras	Não preservada	Preservada
Similaridade entre palavras	Não representada	Representada por proximidade vetorial
Eficiência computacional	Baixa para vocabulários grandes	Mais eficiente
Aplicação em modelos de PLN	Modelos simples e introdutórios	Modelos modernos e profundos

Tabela 1 – Comparação entre One-hot Encoding e Word Embedding

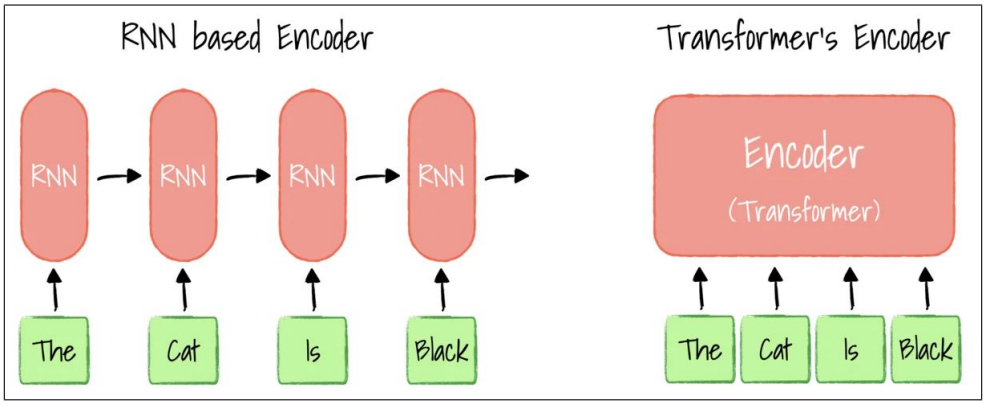
2.2 Avanços em Modelos de PLN para Análise de Sentimentos

Com o avanço das técnicas de Processamento de Linguagem Natural, diversos modelos de redes neurais passaram a ser propostos para lidar de forma mais eficiente com tarefas complexas, como a análise de sentimentos. Ao longo do tempo, essas abordagens evoluíram de modelos sequenciais, que processam o texto palavra por palavra, para arquiteturas mais robustas, capazes de capturar melhor o contexto e as relações semânticas. Nesta seção, são apresentados alguns dos principais avanços nos modelos de PLN, com foco na comparação entre redes neurais recorrentes e modelos baseados em Transformers.

2.2.1 RNN x Transformers

As redes neurais são compostas por diferentes arquiteturas, e para este trabalho é importante compreender a diferença entre dois deles: os Transformers e as Redes Neurais Recorrentes (RNN). As RNNs, ou *Recurrent Neural Networks* processam dados de forma sequencial, em que cada etapa depende da anterior. Esse tipo de arquitetura possui um mecanismo de memória, no qual informações de entradas anteriores são propagadas ao longo da sequência e influenciam a análise da entrada atual, o que pode gerar dificuldades quando se lida com sequências de dados longas. Em contraste, os Transformers adotam uma abordagem diferente, processando todas as etapas simultaneamente em paralelo, utilizando um mecanismo de atenção, que será detalhado posteriormente.

Figura 8 – Encoder RNN e Transformer



Fonte: (JING, 2020)

Na figura 8 é mostrada a diferenca entre o modelo RNN e transformers À

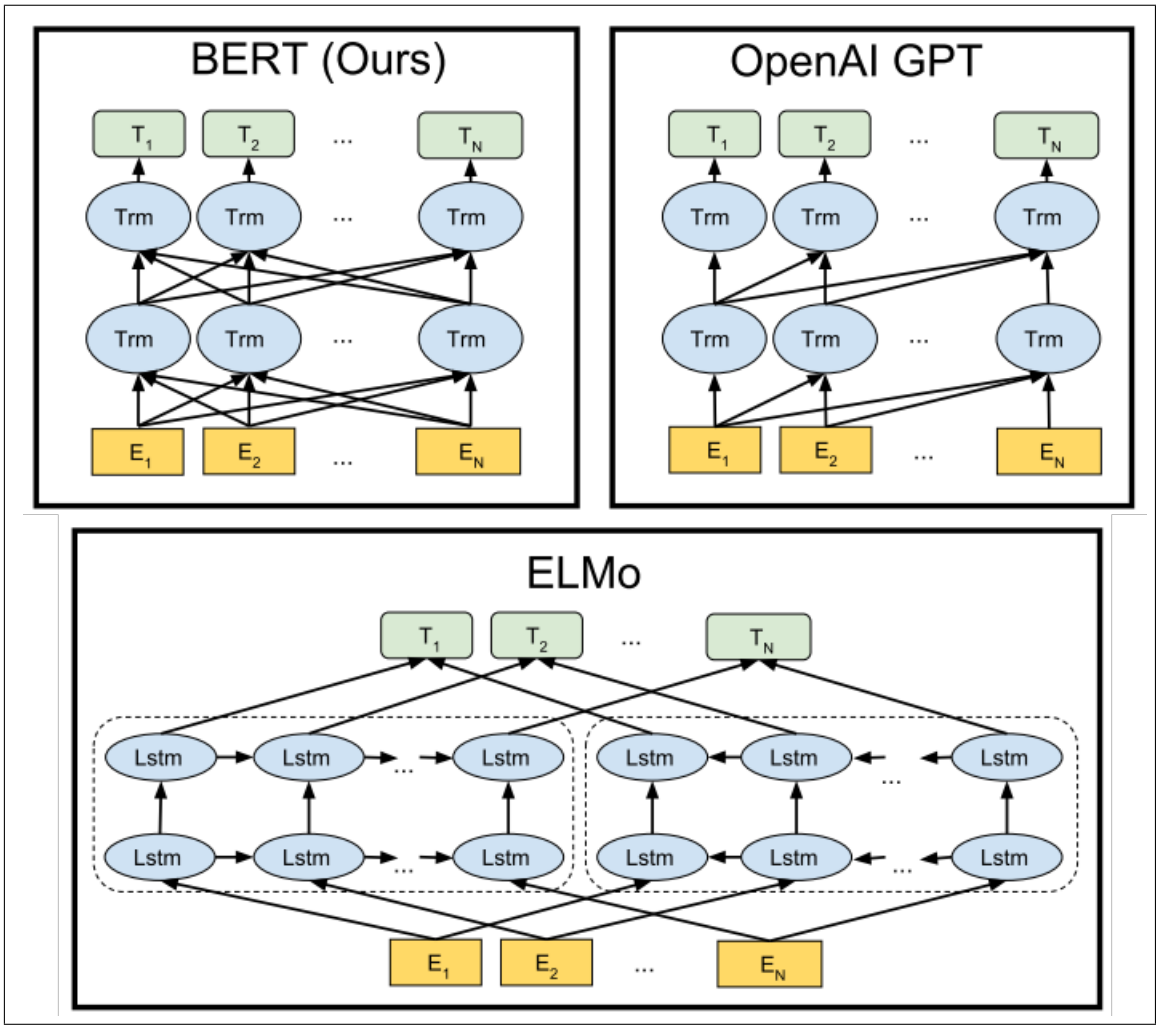
esquerda pode-se notar que o texto "O gato é preto" é processado palavra por palavra e em ordem. Em uma RNN, as palavras são processadas uma a uma, sempre utilizando a mesma camada recorrente, que considera a entrada atual e o estado oculto anterior.

À direita, o transformer processa todas as palavras da frase "o gato é preto" ao mesmo tempo. Ao contrário do RNN, o Transformer não precisa processar expressões lineares. Todas as palavras são processadas no codificador de uma vez, utilizando os mecanismos de atenção mencionados anteriormente.

As arquiteturas apresentadas constituem a base para o desenvolvimento de modelos de linguagem mais avançados, que se apoiam nos conceitos de processamento sequencial e atenção para compreender o contexto textual. Entre esses modelos, destaca-se o BERT, que utiliza a arquitetura Transformer para capturar relações semânticas complexas e será abordado no Capítulo 2.2.3.

2.2.2 Modelos bidirecionais

Figura 9 – Diferença de arquitetura em modelos pré-treinados



Fonte: (DEVLIN et al., 2019)

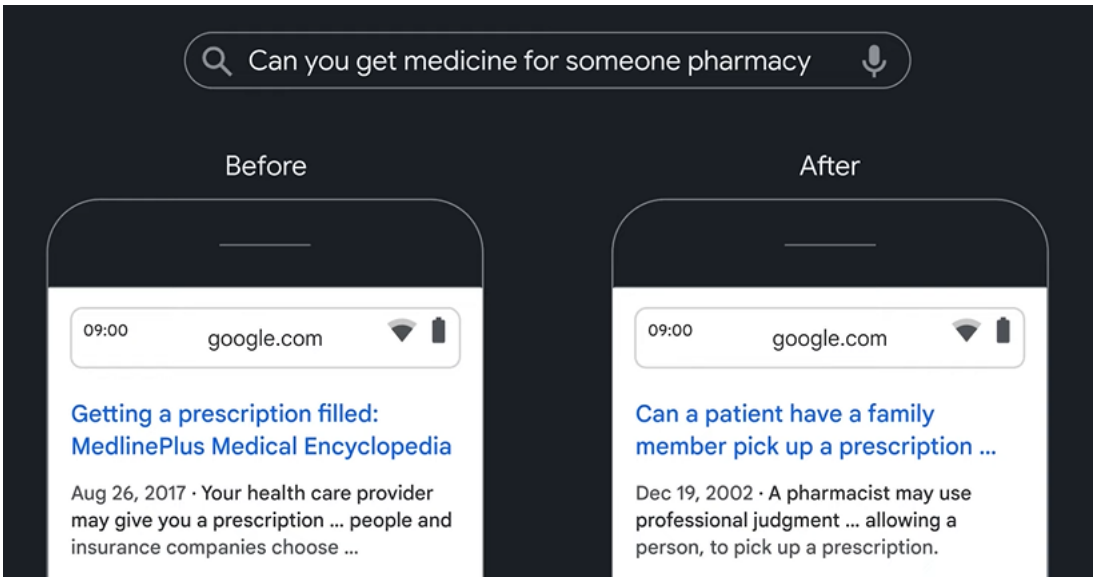
O BERT, que é o modelo usado nesse trabalho utiliza transformers juntamente com mecanismos de atenção, ou seja, não depende da sequência, atribuindo mais atenção a entradas mais importantes da sequência. Conforme a imagem acima, os modelos unidirecionais como o OpenAI GPT processam as entradas da esquerda para a direita de maneira sequencial, considerando apenas as entradas anteriores. Já o ELMo é pseudo bidirecional (Embeddings from Language Models) utiliza LSTM (Long Short-Term Memory), que é uma RNN que foi projetada para lidar com uma das fraquezas desse tipo de modelo: sequências longas. Essa arquitetura faz isso por meio do processamento por ambas as direções (esquerda-direita e direita-esquerda),

semelhante ao BERT, porém, essa arquitetura não utiliza transformers e mecanismos de atenção.

2.2.3 BERT

O BERT (Bidirectional Encoder Representations from Transformers) é um modelo que utiliza os mecanismos de atenção, o que trouxe uma melhora significativa para motores de busca no geral. Uma das características do BERT é que esse modelo é pré-treinado para que tenha um melhor desenvolvimento. O BERT foi pré-treinado com duas bases de dados que juntas, totalizam mais de 3.000.000 de palavras: A BooksCorpus e a *English Wikipedia* (DEVLIN et al., 2019).Além disso, o BERT adota a abordagem de fine-tuning, que consiste em partir de um modelo previamente treinado em grandes bases de dados para tarefas generalistas e refinar seus parâmetros por meio de um novo treinamento direcionado a uma tarefa específica, utilizando dados do problema em questão.

Figura 10 – Evolução da pesquisa no Google com BERT



Fonte: (REDDY, 2023)

Na figura 10 acima são exibidos dois resultados da mesma pesquisa no Google, sendo que o resultado da esquerda não usa o BERT e o da direita sim. A frase 'Can you get medicine for someone pharmacy' em tradução livre significa 'É possível pegar remédios para alguém na farmácia'. O primeiro resultado traz um link com uma matéria que explica como obter remédios na farmácia (por meio de uma receita

médica), ou seja, não foi feita uma boa associação entre a primeira parte da frase e a segunda. Já o segundo resultado traz um link que tem como matéria 'O parente de um paciente pode buscar a prescrição médica...', ou seja, por meio do mecanismo de atenção, o BERT conseguiu associar que o que o usuário queria era saber se podia buscar remédios para outra pessoa.

Pode-se entender então que o BERT tem habilidade de capturar contextos, ou seja, a arquitetura transformer tem a capacidade de entender o significado semântico de uma sentença.

2.2.4 CLS

De acordo com (DEVLIN et al., 2019), o token [CLS] é inserido no início de toda sequência de entrada do BERT e funciona como um token de agregação. Ao longo do processamento, ele passa a representar uma síntese das informações semânticas da sequência completa, sendo utilizado em tarefas de classificação como base para a decisão final do modelo.

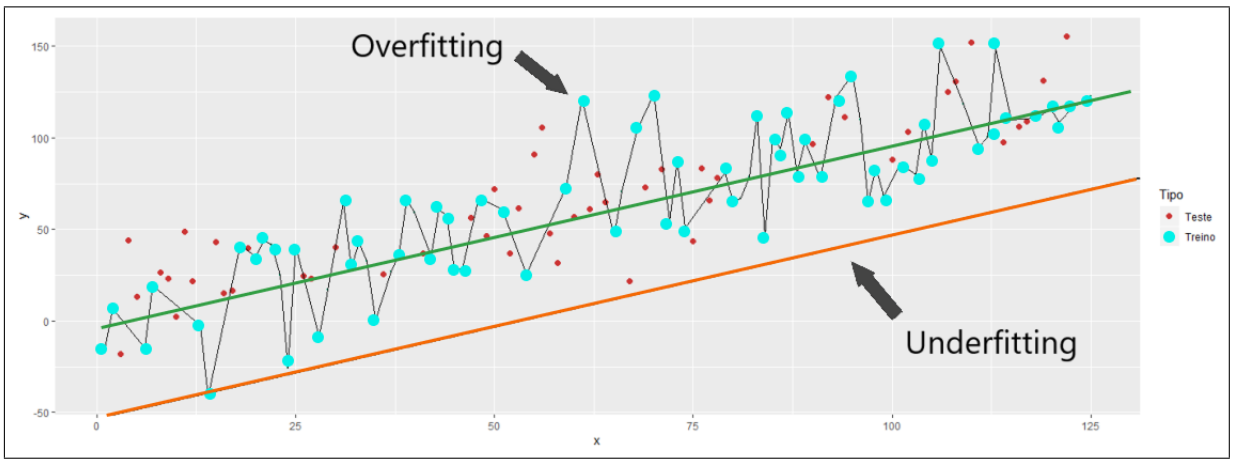
2.3 Desafios da PLN

Apesar dos avanços em PLN, os modelos ainda enfrentam limitações importantes, especialmente relacionadas ao ajuste aos dados e à interpretação semântica. Nesta seção, são discutidos desafios como underfitting, overfitting e questões semânticas da linguagem.

2.3.1 Underfitting e Overfitting

O Processamento de Linguagem Natural tem evoluído ao longo dos anos; no entanto, alguns desafios ainda persistem. Modelos que não são adequadamente ajustados aos dados podem apresentar dificuldades em desempenhar suas tarefas de forma satisfatória. Por outro lado, modelos excessivamente ajustados podem sofrer de overfitting, fenômeno que ocorre quando o modelo alcança alta acurácia nos dados de treinamento por memorizar padrões específicos. Nesse caso, ao ser avaliado com dados de teste, o desempenho tende a ser inferior, uma vez que esses dados diferem dos padrões aprendidos durante o treinamento.

Figura 11 – Exemplo de underfitting e overfitting



Fonte:Adaptado de (Didática Tech, 2024)

Na figura 11, é possível visualizar a diferença entre underfitting e overfitting. Os dados de treino estão representados pelos pontos azuis, e os de teste, pelos pontos vermelhos. A reta verde indica o cenário ideal de ajuste do modelo, de acordo com a regressão linear.

A linha laranja representa um caso de underfitting, em que o modelo é muito simples e não consegue capturar os padrões nem nos dados de treino nem nos de teste. Já a linha preta representa o overfitting, onde o modelo se ajusta perfeitamente aos dados de treino, mas tem baixo desempenho nos dados de teste, indicando que ele memorizou os exemplos em vez de aprender a generalizar.

Além das dificuldades relacionadas ao ajuste dos modelos, como underfitting e overfitting, a própria interpretação da semântica de um texto continua sendo um desafio, mesmo com o uso de modelos de PLN mais avançados.

2.3.2 Semântica

Entender a linguagem pode ser desafiador até para humanos. A semântica implica em muitas situações, como, por exemplo, a variedade de sentidos de uma palavra. Por exemplo, na frase "A manga está suja", pode-se interpretar que a manga (fruta) está suja ou que a manga de uma camiseta está suja, com base no contexto que se identifica. Então, na mesma frase, tem-se a questão do contexto e da ambiguidade da palavra. Outra dificuldade dentro da semântica é a ironia e o sarcasmo, pois, com eles, a expressão pode ter seu sentido invertido.

Na figura 12, tem-se a reação de três personagens ao atraso de um colega. O primeiro diz: "Olha só quem chegou! Não está nem um pouco atrasado!"O segundo diz: "Chegou cedo para a reunião de amanhã!"E o terceiro diz: "Ih! Lá vem o prêmio Nobel da pontualidade."Nos três casos os comentários não refletem o real: que o colega chegou atrasado. Ensinar esse comportamento a modelos é uma tarefa complexa. De acordo com (WICK-PEDRO; VALE, 2020) a detecção de ironias depende de diversos fatores, incluindo a compreensão do contexto em que a sentença está sendo escrita, o conhecimento prévio do tema em discussão, o tom usado pelo autor e o reconhecimento de pistas linguísticas específicas, como o uso de figuras de linguagem, ou contrastes claros, e a presença de elementos como aspas, emoticons ou exageros. Além disso, a identificação eficaz da ironia requer muitas vezes uma análise pragmática e cognitiva da informação, uma vez que a ironia pode alterar o significado literal das palavras, criando um desafio significativo para os sistemas de processamento de linguagem natural.

Figura 12 – Ironia, sarcasmo e deboche

Fonte: (Didática Tech, 2024)

A compreensão dos fundamentos das redes neurais, bem como de seus desafios e técnicas, é fundamental para o desenvolvimento de soluções em Processamento de Linguagem Natural. Nesse contexto, torna-se necessário apresentar como o problema abordado neste trabalho foi solucionado. Assim, esta seção descreve os materiais e métodos utilizados, detalhando a metodologia usada nesse estudo.

3 Materiais e métodos

Este capítulo apresenta as ferramentas utilizadas, os dados selecionados e as etapas seguidas para o desenvolvimento da solução. São descritos o modelo de linguagem adotado, os processos de coleta e preparação dos dados, além dos métodos aplicados para treinamento e avaliação. O objetivo é garantir clareza e permitir a reprodução do trabalho por outros interessados.

3.1 BERT

O BERT, conforme explicado anteriormente é um modelo generalista. A escolha desse modelo foi feita devido a alguns fatores, entre eles:

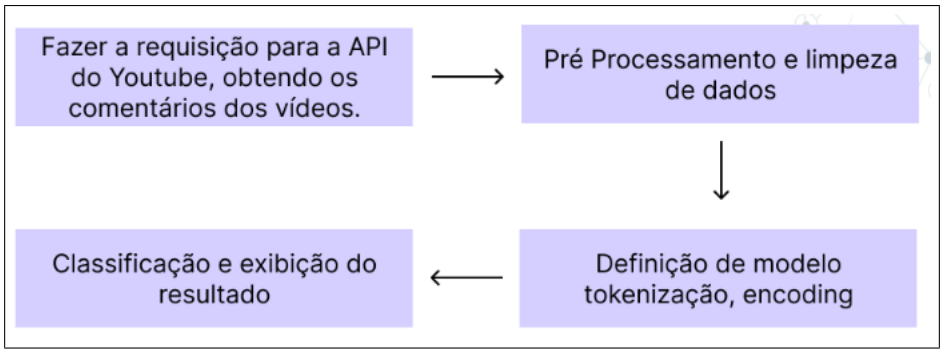
- O fato do modelo ser bidirecional, o que ajuda no melhor entendimento do contexto das frases.
- O modelo apesar de pré-treinado pode ser aprimorado para tarefas em contextos específicos.
- A eficácia do BERT foi comprovada no estudo de Devlin et al. (2019), no qual o modelo foi apresentado e avaliado em diferentes tarefas de Processamento de Linguagem Natural. O artigo "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"apresentou o BERT e mostrou que ele alcançou ótimos resultados em várias tarefas de PLN, como a classificação de sentimentos, respostas a perguntas, entre outras.

Porém, o BERT originalmente é treinado com bases em inglês e é recomendado para atividades que envolvam esse idioma, o que a princípio não atenderia aos objetivos desse trabalho. Para solucionar esse problema, foi utilizado o BERTimbau, que é um modelo baseado no BERT pré-treinado com a língua portuguesa.

3.2 Google API Client

Para buscar os comentários do Youtube foi necessário utilizar a Google API Client, que é a API do Google que concede acesso a diversas ferramentas da empresa, entre elas, o YouTube, que foi utilizado nesse trabalho para buscar os comentários dos vídeos.

Figura 13 – Fluxograma das atividades desenvolvidas



Fonte: A autora, 2024

3.3 Dataset: IMDB PT-BR

Uma das bases utilizadas foi a IMDB PT-BR, um dataset que contém análises de filmes enviadas do site IMDB, classificadas como positivas ou negativas. As avaliações são originalmente provenientes de uma base em inglês e foram traduzidas e disponibilizadas em português por pesquisadores para uso acadêmico. A base original foi proposta por (MAAS et al., 2011), enquanto a versão em português utilizada neste trabalho foi obtida por meio do repositório disponibilizado por (FRED, 2023) na plataforma Kaggle.

3.4 Dataset: Comentários tóxicos PT-BR

Outra base utilizada para aprimorar o modelo foi a "Comentários tóxicos PT-BR", organizada por (NETO, 2021). Essa base é um agrupamento de outras bases disponíveis pela internet, com foco em comentários negativos. Para que os textos não atrapalhassem a análise, o autor fez a normalização, como mostram as colunas da base:

- text: O comentário sem pré-processamento;
- text_norm: Comentário normalizado;
- toxic: O indicador de que o comentário é tóxico ou não, com 1 sendo tóxico e 0 sendo não tóxico.

3.5 Dataset: Tweets Ekman pt-br

A outra base utilizada para aprimorar o treinamento é a de Tweets Ekman pt-br, essa base é interessante para o treinamento pois traz comentários do Twitter, que assim como o Youtube, é uma rede social, ou seja, os tweets têm abreviações, gírias e expressões típicas da linguagem da internet. Esse dataset foi escolhido na expectativa de tornar o modelo mais preciso. É interessante notar que essa base possui as seguintes colunas:

- Texto: tweet
- Sentimento: Emoção

A coluna emoção possui 5 possíveis valores: felicidade, raiva, nojo, medo e tristeza. De acordo com (SOUZA, 2023) essas são as 5 emoções de Ekman. Para utilizar essa base, foi necessário converter as emoções em valores de 0 ou 1, sendo 0 como negativo e 1 positivo. A atribuição foi feita da seguinte forma:

- Felicidade:1
- Raiva:0
- Nojo:0
- Medo:0
- Tristeza:0

3.6 Fluxograma geral

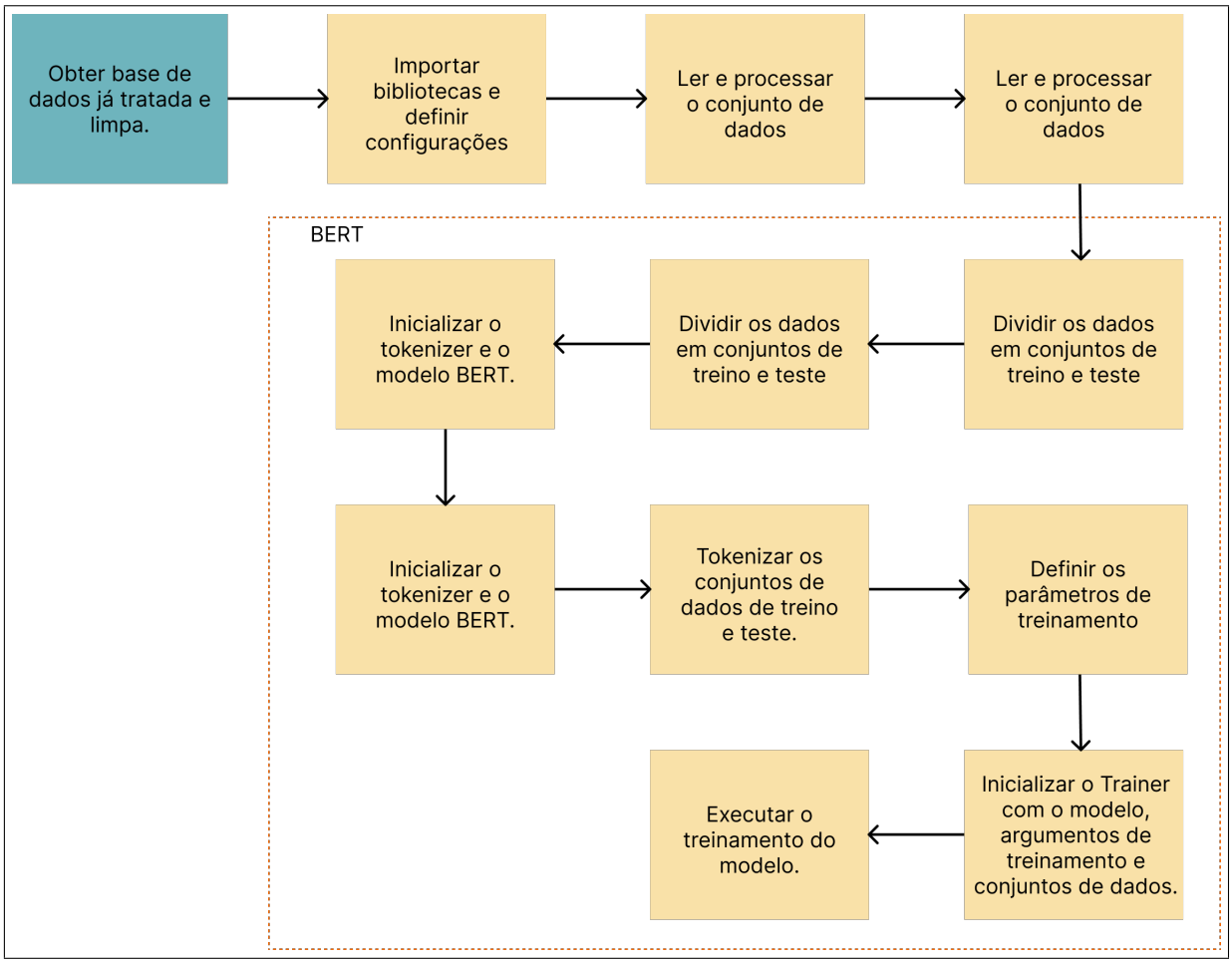
A figura 14 abaixo apresenta um fluxograma que resume o processo completo de preparação, tokenização e treinamento do modelo BERT, utilizado neste trabalho. O processo se inicia com a obtenção de uma base de dados previamente tratada e limpa. Em seguida, são importadas as bibliotecas necessárias e definidas as configurações iniciais do ambiente de execução.

Após essa etapa, os dados são lidos, processados e divididos em conjuntos de treino e teste. Com os dados prontos, o modelo BERT e seu tokenizador são inicializados. A tokenização transforma os textos em sequências de números compreensíveis para o modelo.

Posteriormente, são definidos os parâmetros de treinamento, como número de épocas, taxa de aprendizado e batch size. Com essas configurações, inicializa-se o objeto Trainer, que reúne o modelo, os dados e os argumentos de treinamento. Por fim, o treinamento é executado, permitindo que o modelo aprenda a classificar os textos com base nos dados fornecidos.

Esse fluxo representa a sequência prática adotada para personalizar o modelo BERT com os dados deste projeto, e serve como guia visual para compreensão das etapas envolvidas no processo.

Figura 14 – Fluxograma de treinamento do projeto



Fonte: A autora, 2024

3.7 Treinamento e parâmetros utilizados

Após a preparação das bases de dados, foi realizado o treinamento do modelo utilizando a biblioteca Transformers, da plataforma Hugging Face. O modelo base utilizado foi o BERTimbau, uma versão do BERT treinada com textos em português, o que torna sua aplicação mais adequada para a análise de sentimentos em comentários escritos nesse idioma.

A configuração do treinamento foi feita por meio do objeto *TrainingArguments*, que permite definir os principais parâmetros utilizados durante o ajuste do modelo. Nele foram definidos aspectos como número de épocas, tamanho dos lotes, estratégia de avaliação, estratégia de salvamento, métrica principal, *warmup_ratio*, *weight_decay* e registro das métricas no TensorBoard. A Figura 15 apresenta o trecho de código com os principais parâmetros aplicados.

Figura 15 – Parâmetros de treinamento definidos no objeto *TrainingArguments*

```
training_args = TrainingArguments(
    output_dir="/content/drive/My Drive/BERT/results_toxicos_30ep",
    num_train_epochs=30,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    eval_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True,
    metric_for_best_model="f1",
    greater_is_better=True,
    warmup_ratio=0.1,
    weight_decay=0.01,
    logging_dir="/content/drive/My Drive/BERT/logs_toxicos_30ep",
    logging_steps=100,
    save_total_limit=1,
    report_to=["tensorboard"]
)
```

Fonte: A autora, 2024.

Além do trecho de código, os principais hiperparâmetros foram organizados na Tabela 2. Essa tabela tem como objetivo facilitar a leitura e tornar mais claro o papel de cada configuração aplicada no treinamento. Os parâmetros apresentados controlam aspectos importantes do processo, como a quantidade de épocas, o tamanho dos lotes de treinamento e avaliação, a forma de salvamento dos checkpoints, a métrica utilizada para escolha do melhor modelo e os ajustes usados para melhorar a estabilidade do

treinamento.

Tabela 2 – Hiperparâmetros utilizados no treinamento do modelo

Parâmetro	**Valor utilizado**	**Descrição**
Modelo base	BERTimbau	Modelo pré-treinado em português utilizado como base para o fine-tuning.
Número de épocas	30	Quantidade máxima de vezes em que o modelo percorreu a base de treinamento.
Batch size de treinamento	8	Quantidade de amostras processadas por vez durante o treinamento.
Batch size de avaliação	8	Quantidade de amostras processadas por vez durante a avaliação.
Estratégia de avaliação	A cada época	O modelo foi avaliado ao final de cada época de treinamento.
Estratégia de salvamento	A cada época	Um checkpoint do modelo foi salvo ao final de cada época.
Melhor modelo ao final	Sim	Ao final do treinamento, foi carregado o checkpoint com melhor desempenho.
Métrica principal	F1-score	Métrica utilizada para escolher o melhor modelo durante o treinamento.
Critério de escolha	Maior valor	Como o F1-score mede desempenho, valores maiores indicam melhores resultados.
Warmup ratio	0,1	Percentual inicial do treinamento usado para aumentar gradualmente a taxa de aprendizado.
Weight decay	0,01	Técnica de regularização utilizada para reduzir o risco de overfitting.
Logging steps	100	Frequência de registro das métricas durante o treinamento.
Limite de checkpoints salvos	1	Apenas o melhor checkpoint foi mantido salvo.
Ferramenta de acompanhamento	TensorBoard	Ferramenta utilizada para visualizar as métricas e curvas de treinamento.

Fonte: A autora, 2026.

A métrica principal utilizada foi o F1-score, pois ela combina precisão e recall em uma única medida. Essa escolha foi importante porque, em problemas de classificação de sentimentos, a acurácia isolada pode não representar bem o desempenho do modelo, principalmente quando há diferença na quantidade de exemplos entre as classes. Dessa forma, o F1-score permite avaliar de maneira mais equilibrada se o modelo consegue identificar corretamente comentários positivos e negativos.

Também foram utilizados parâmetros voltados à estabilidade e à generalização do treinamento. O *warmup_ratio* ajuda a tornar o início do treinamento mais estável, aumentando gradualmente a taxa de aprendizado. Já o *weight_decay* funciona como uma forma de regularização, reduzindo o risco de o modelo se ajustar excessivamente aos dados de treino.

3.8 Código

Durante o desenvolvimento, foram realizados dois treinamentos diferentes, utilizando combinações de bases de dados com textos em português.

O primeiro treinamento utilizou comentários do Twitter combinados com a base IMDB PT-BR, composta por avaliações de filmes traduzidas para o português. Já o segundo treinamento agrupou os comentários do Twitter com a base de comentários tóxicos em português, que reúne mensagens com linguagem mais próxima da realidade encontrada em redes sociais que foram classificados como tóxicos ou não.

Apesar de ambos seguirem o mesmo fluxo de preparação, o segundo modelo apresentou melhores resultados nos testes, com maior capacidade de identificar corretamente os sentimentos expressos nos comentários do YouTube.

3.8.1 Fluxo 1: Treinamento do Modelo

3.8.1.1 Limpeza dos Dados

Algorithm 1: Limpeza e normalização dos dados

Resultado: Comentários limpos e prontos para o treinamento do modelo

Carregar os comentários da base original
Remover linhas vazias ou com dados ausentes
foreach *comentário* **do**
    Remover emojis
    Remover links, menções e hashtags
    Remover pontuação e símbolos
    Converter o texto para letras minúsculas
    Normalizar espaços em branco
end
Salvar os comentários limpos em um novo arquivo CSV

3.8.1.2 Treinamento com BERTimbau (Twitter + Comentários Tóxicos)

Algorithm 2: Treinamento do modelo com BERTimbau

Resultado: Modelo treinado com base nas bases Twitter e Comentários Tóxicos

Carregar os comentários já limpos
Atribuir rótulos de sentimento (positivo ou negativo)
Dividir os dados em treino e teste na proporção 80/20
Carregar o modelo `BERTimbau` e seu tokenizador
foreach *comentário* **do**
    Tokenizar o texto
    Gerar tensores de entrada
end
Definir parâmetros de treinamento (número de épocas, batch size, etc)
Executar o processo de treino com o `Trainer`
Salvar o modelo treinado para uso posterior

3.8.2 Fluxo 2: Aplicação do Modelo em Comentários do YouTube

3.8.2.1 Coleta de Comentários com YouTube API

Algorithm 3: Coleta de comentários do YouTube

Resultado: Lista de comentários extraídos de vídeos de um canal do YouTube e salvos em CSV

```
Montar conexão com o Google Drive
Importar bibliotecas (Google API, pandas, transformers, etc)
Carregar chave da API com dotenv
Inicializar serviço da YouTube API
Definir ID do canal
Listar todos os vídeos do canal
foreach vídeo na lista de vídeos do
    Obter ID do vídeo
    Buscar comentários com a YouTube API
    Adicionar comentários à lista final
end
Salvar os comentários em um arquivo CSV
```

3.8.2.2 Classificação com Modelo Treinado

Algorithm 4: Classificação de sentimentos dos comentários do YouTube

Resultado: Comentários do YouTube classificados como positivos ou negativos

```
Carregar os comentários coletados e limpos
Carregar o modelo treinado
Carregar o tokenizador correspondente
foreach comentário do
    Tokenizar o texto
    Realizar a predição com o modelo
    Armazenar o resultado (positivo ou negativo)
end
Salvar os resultados com rótulos em um novo arquivo CSV
```

Com os dados preparados, as ferramentas definidas e o modelo treinado, a próxima etapa foi avaliar sua aplicação prática. No capítulo seguinte, são apresentados os testes realizados, os resultados obtidos e a análise do desempenho da solução proposta.

4 Resultados e discussões

Neste capítulo são apresentados os resultados obtidos com o modelo treinado, incluindo a análise dos gráficos de treinamento, as métricas de desempenho, a matriz de confusão e exemplos reais classificados a partir de comentários do YouTube. A intenção é mostrar não apenas se o modelo funcionou, mas também onde ele teve mais dificuldade.

4.1 Processo de desenvolvimento

O desenvolvimento foi dividido em etapas. Primeiro, foram preparadas as bases de dados utilizadas no treinamento. Depois, foram realizados testes com diferentes combinações de bases e, por fim, o modelo treinado foi aplicado em comentários reais extraídos do YouTube.

Foram testadas duas combinações principais de bases: Twitter com IMDB PT-BR e Twitter com Comentários Tóxicos PT-BR. A primeira combinação apresentou limitações porque a base IMDB é composta por avaliações de filmes, com textos mais longos e mais formais. Esse tipo de texto é diferente dos comentários encontrados em redes sociais, que costumam ser mais curtos, informais e diretos.

A segunda combinação, utilizando Twitter e Comentários Tóxicos PT-BR, apresentou melhor resultado para a proposta deste trabalho. Isso ocorreu porque essas bases possuem uma linguagem mais próxima da usada em comentários online, com expressões mais informais, críticas diretas e maior presença de termos negativos. Por esse motivo, essa versão foi escolhida como a melhor para aplicação nos comentários do YouTube.

Além do treinamento, também foi necessário ajustar a coleta dos comentários por meio da API do YouTube. Após a coleta, os comentários passaram pelo processo de limpeza e foram classificados com o modelo treinado.

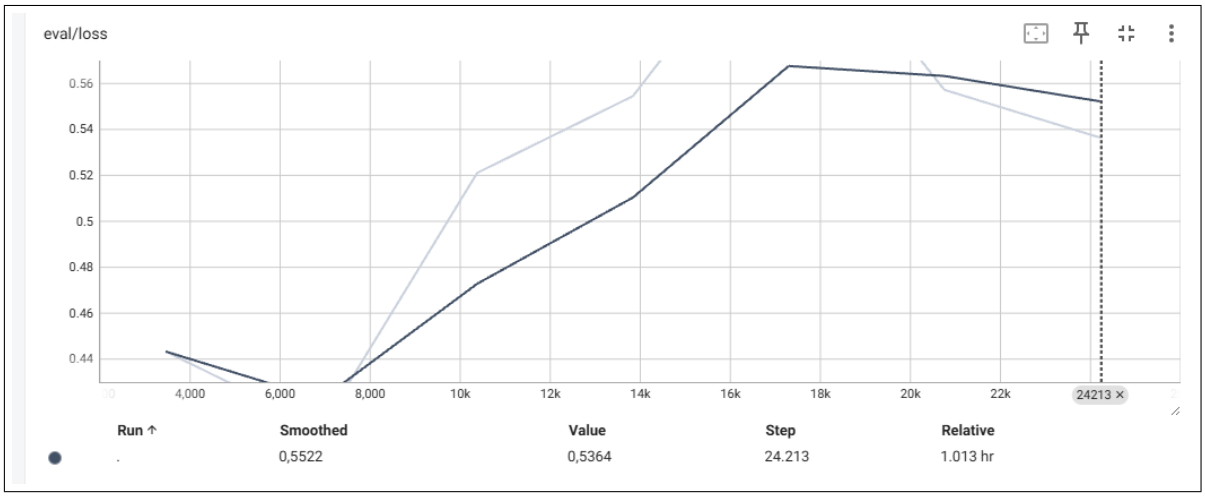
4.2 Desempenho do modelo durante o treinamento

Durante os testes com a base combinada de Twitter e Comentários Tóxicos PT-BR, o modelo apresentou bons resultados nos primeiros checkpoints. No entanto, depois de algumas épocas, os gráficos começaram a indicar sinais de overfitting. Isso

significa que o modelo continuava aprendendo os dados de treino, mas começava a perder desempenho nos dados de validação.

A Figura 16 mostra a curva de *eval_loss*, que representa a perda nos dados de validação. Quanto menor esse valor, melhor é o comportamento do modelo em dados que ele não usou diretamente para treinar. No gráfico, é possível observar que a perda de validação começa a subir depois dos primeiros checkpoints, indicando que o modelo estava começando a se ajustar demais aos dados de treino.

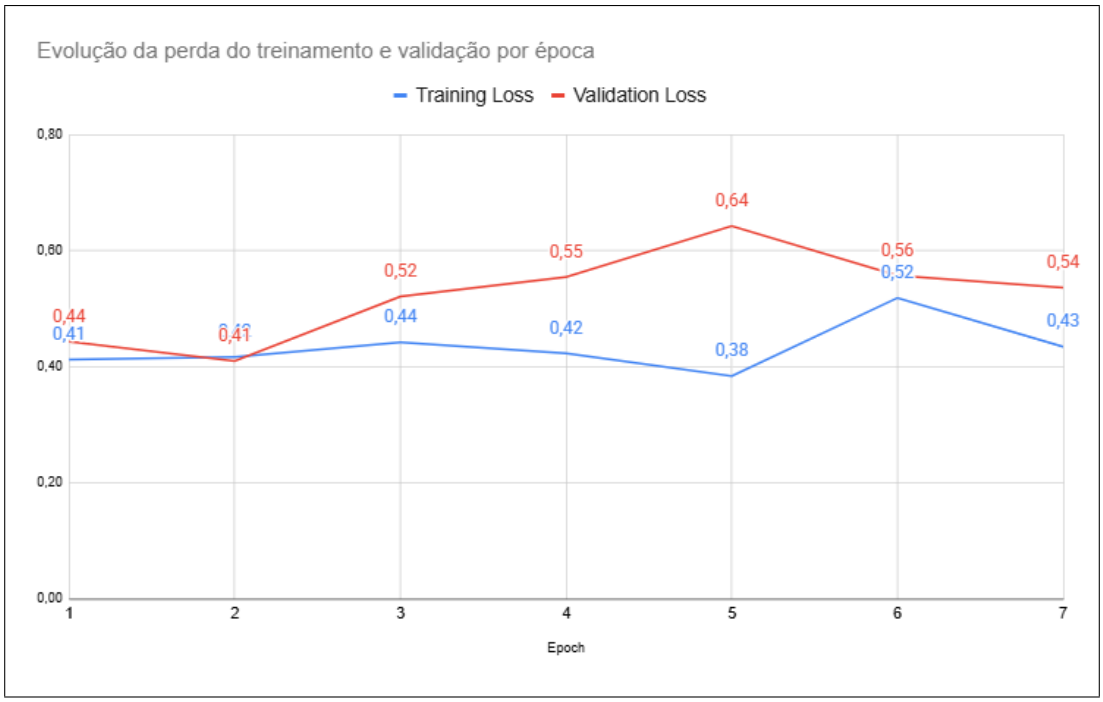
Figura 16 – Curva de perda de validação (*eval_loss*) no treinamento com base de comentários tóxicos



Fonte: A autora, 2026.

Para entender melhor esse comportamento, também foi analisada a comparação entre a perda de treinamento e a perda de validação por época, apresentada na Figura 17. Nesse gráfico, a linha de treinamento mostra como o modelo se comportou nos dados usados para aprender, enquanto a linha de validação mostra como ele se comportou em dados separados para avaliação.

Figura 17 – Evolução das perdas de treinamento e validação por época



Fonte: A autora, 2026.

A partir da Figura 17, nota-se que a época 2 apresentou um dos melhores equilíbrios entre treinamento e validação. Nessa etapa, a perda de validação ficou em aproximadamente 0,41, com a perda de treinamento também próxima desse valor. Esse comportamento indica que o modelo ainda estava aprendendo sem se ajustar excessivamente aos dados de treino.

Já na época 5, a perda de validação aumentou para aproximadamente 0,64, enquanto a perda de treinamento continuou menor. Esse afastamento entre as duas curvas é um sinal de overfitting, pois mostra que o modelo estava indo bem nos dados que já conhecia, mas piorando em dados novos.

A Figura 18 apresenta a curva de *train_loss* registrada pelo TensorBoard ao longo dos steps de treinamento. Esse gráfico ajuda a observar o comportamento interno do treinamento. Apesar das oscilações, é possível perceber uma queda da perda de treinamento no início, o que mostra que o modelo conseguiu aprender padrões presentes na base.

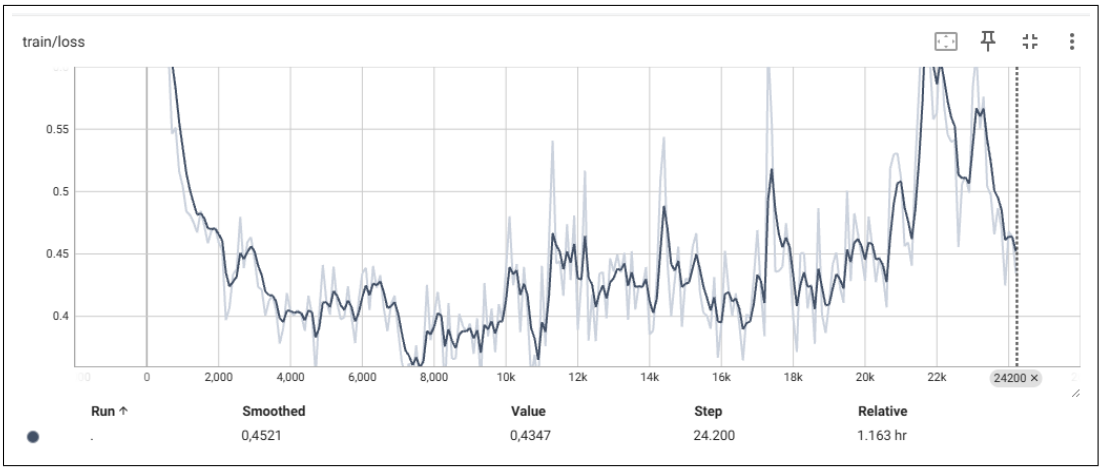


Figura 18 – Curva de perda de treinamento (*train_loss*) no modelo treinado com base de comentários tóxicos

Fonte: A autora, 2026.

Também foi analisada a evolução do F1-score ao longo das épocas, conforme apresentado na Figura 19. Essa métrica foi importante porque considera, ao mesmo tempo, precisão e recall, permitindo uma avaliação mais equilibrada do modelo.

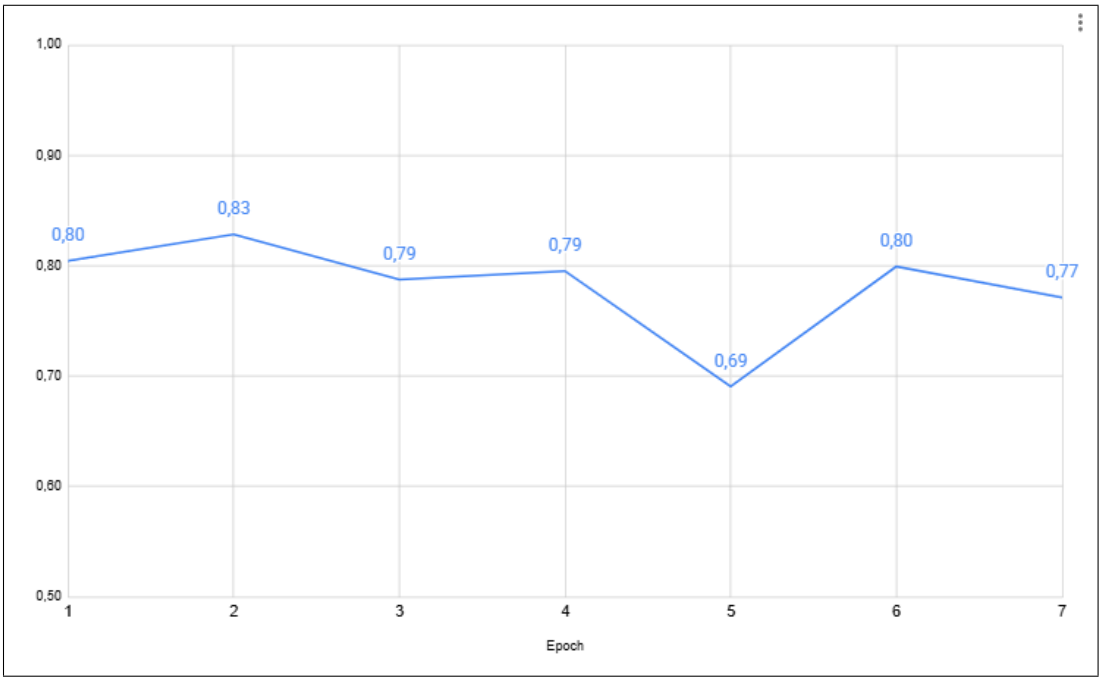


Figura 19 – Evolução do F1-score do modelo ao longo das épocas

Fonte: A autora, 2026.

Na Figura 19, observa-se que o melhor desempenho ocorreu próximo da época

2, quando o F1-score ficou em torno de 0,83. Depois disso, o desempenho oscilou e teve queda mais acentuada na época 5, chegando a aproximadamente 0,69. Esse comportamento reforça a mesma leitura feita a partir das perdas: o modelo teve seu melhor momento nos primeiros checkpoints e depois começou a apresentar sinais de overfitting.

Com base nesse comportamento, foi escolhido o checkpoint 6918 como melhor ponto de parada. Essa escolha buscou equilibrar o aprendizado do modelo com sua capacidade de generalizar para dados novos, evitando manter uma versão treinada por mais tempo, mas com pior desempenho em validação.

Durante o treinamento, também foi observado que os gráficos do TensorBoard apresentaram poucos pontos de amostragem. Isso ocorreu porque cada época levou um tempo considerável para ser concluída, devido ao tamanho do modelo BERTimbau e ao volume das bases utilizadas. Mesmo assim, os gráficos foram suficientes para identificar a tendência de overfitting e apoiar a escolha do melhor checkpoint.

4.3 Métricas de desempenho

Para avaliar quantitativamente o desempenho do modelo treinado, foram calculadas as métricas de precisão, recall, F1-score e acurácia no conjunto de teste. O conjunto utilizado para avaliação correspondeu a 20% da base final, totalizando 6.917 exemplos, sendo 3.271 da classe *TOXICO* e 3.646 da classe *NAO_TOXICO*.

A precisão mostra o quanto o modelo acerta quando escolhe uma classe. Por exemplo, quando ele classifica um comentário como *TOXICO*, a precisão indica quantos desses comentários realmente eram tóxicos. Já o recall mostra quantos comentários daquela classe o modelo conseguiu encontrar. Em outras palavras, ele ajuda a perceber quantos comentários tóxicos o modelo conseguiu identificar e quantos acabaram escapando. O F1-score junta essas duas informações em uma única métrica, buscando um equilíbrio entre acertar bem as classificações e deixar passar o menor número possível de casos.

É importante destacar que não foi definido previamente um valor mínimo de F1-score como critério obrigatório de sucesso do experimento. Neste trabalho, o F1-score foi utilizado principalmente como métrica de comparação entre os checkpoints e entre as versões treinadas do modelo, auxiliando na escolha da configuração com melhor equilíbrio entre precisão e recall.

A Tabela 3 apresenta os resultados obtidos pelo modelo treinado no conjunto de teste.

Tabela 3 – Métricas de desempenho do modelo treinado no conjunto de teste

Classe	**Precisão**	**Recall**	**F1-score**	**Suporte**
TOXICO	0,8855	0,7777	0,8281	3271
NAO_TOXICO	0,8202	0,9098	0,8627	3646
Acurácia	0,8473			
Média macro	0,8529	0,8438	0,8454	6917
Média ponderada	0,8511	0,8473	0,8463	6917

Fonte: A autora, 2026.

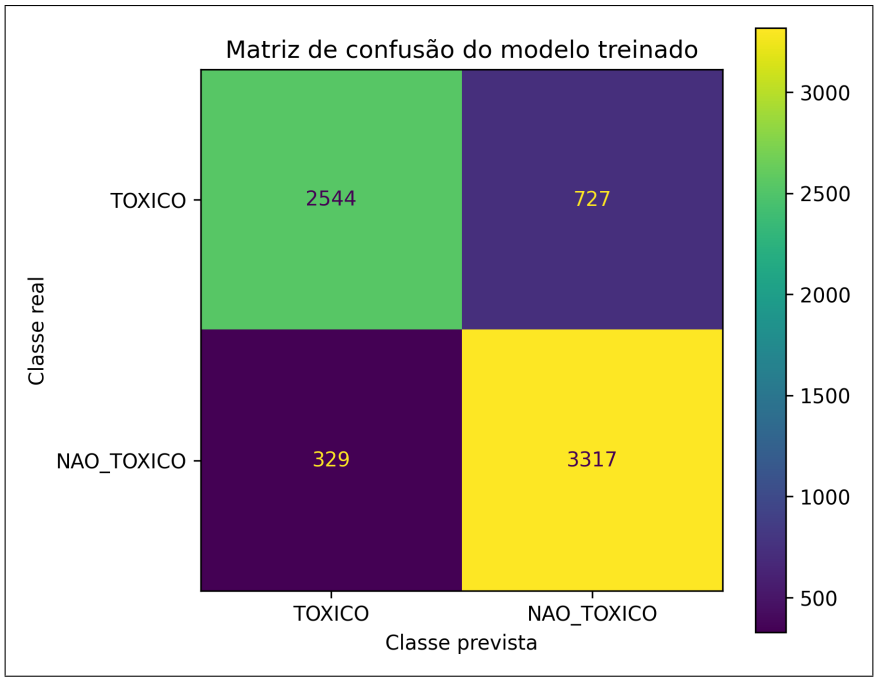
Os resultados mostram que o modelo obteve acurácia de 0,8473, classificando corretamente aproximadamente 84,73% dos exemplos do conjunto de teste. A média macro do F1-score foi de 0,8454, indicando um desempenho equilibrado entre as classes avaliadas.

Na classe *TOXICO*, o modelo apresentou precisão de 0,8855 e recall de 0,7777. Isso indica que, quando o modelo classificou um comentário como tóxico, na maior parte das vezes ele estava correto. Porém, o recall menor mostra que uma parte dos comentários tóxicos ainda foi classificada como não tóxica.

Na classe *NAO_TOXICO*, o modelo apresentou recall de 0,9098, indicando boa capacidade de identificar corretamente comentários não tóxicos. O F1-score dessa classe foi de 0,8627, valor um pouco superior ao obtido na classe tóxica.

Além das métricas, foi gerada a matriz de confusão, apresentada na Figura 20, para observar de forma mais detalhada os acertos e erros do modelo.

Figura 20 – Matriz de confusão do modelo treinado



Fonte: A autora, 2026.

Na matriz de confusão, observa-se que o modelo classificou corretamente 2.544 comentários tóxicos e 3.317 comentários não tóxicos. Por outro lado, 727 comentários tóxicos foram classificados como não tóxicos, enquanto 329 comentários não tóxicos foram classificados como tóxicos. Esse resultado mostra que o modelo teve melhor desempenho para reconhecer a classe *NAO_TOXICO*, mas ainda deixou passar parte dos comentários tóxicos.

4.4 Análise de erros

Apesar dos resultados gerais serem bons, o modelo ainda apresentou dificuldades em alguns tipos de comentário. Os principais casos problemáticos foram comentários com ironia, duplo sentido, frases muito curtas, emojis e comentários neutros.

A dificuldade com comentários neutros ocorreu porque o modelo foi treinado em uma classificação binária. Ou seja, ele precisava escolher entre *TOXICO* e *NAO_TOXICO*. Por isso, comentários sem uma polaridade clara também acabavam sendo enviados para uma dessas duas classes.

Outro ponto importante foi a remoção dos emojis durante o pré-processamento. Essa decisão ajudou a reduzir ruídos no treinamento, mas também retirou informações que poderiam indicar sentimento. Em redes sociais, emojis podem mudar o sentido de uma frase ou reforçar uma emoção, então sua remoção também representa uma limitação do modelo.

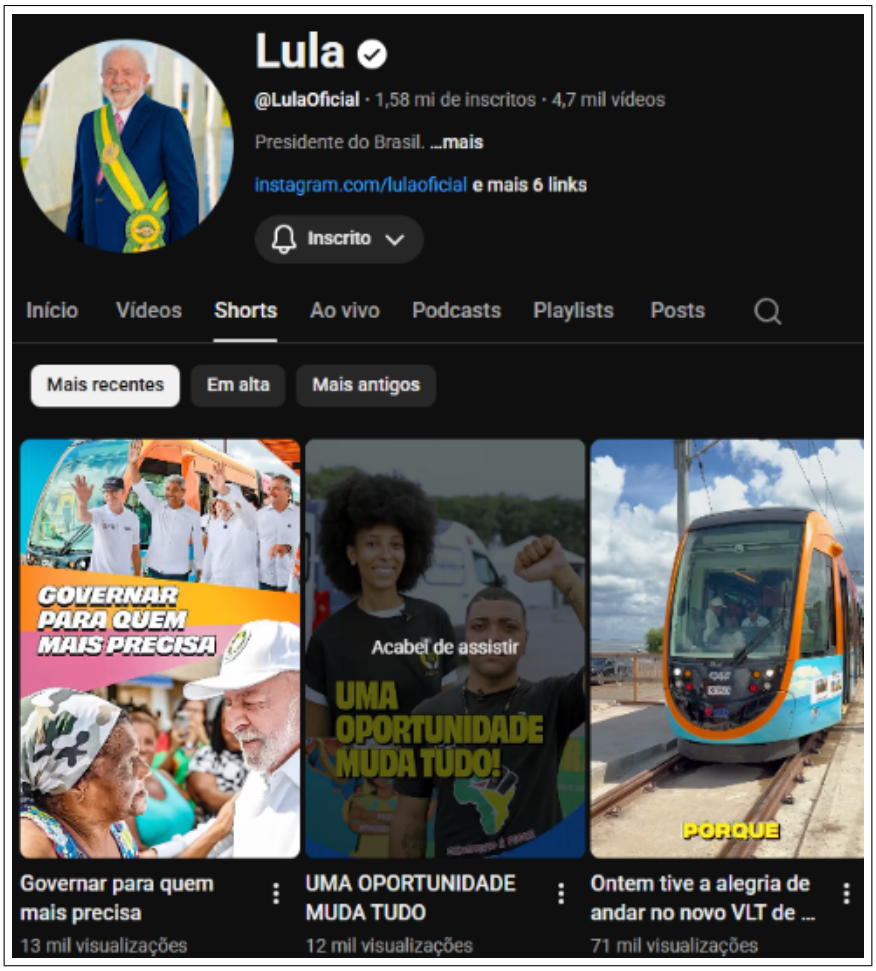
Também foram observadas dificuldades em comentários muito curtos, pois nesses casos há pouco contexto para o modelo interpretar. Frases pequenas, interjeições e expressões soltas podem depender muito do vídeo, do momento ou da conversa anterior, contexto que o modelo não recebe durante a classificação.

4.5 Análise qualitativa

Para avaliar a aplicação prática do modelo treinado, foram utilizados comentários reais extraídos de vídeos publicados no canal oficial de Luiz Inácio Lula da Silva no YouTube. A escolha desse canal foi discutida anteriormente na justificativa do trabalho, considerando que conteúdos políticos tendem a concentrar comentários com maior carga opinativa.

A Figura 21 apresenta o canal utilizado na coleta e alguns dos vídeos cujos comentários foram analisados.

Figura 21 – Canal utilizado para coleta dos comentários analisados



Fonte: YouTube, 2026.

Após a coleta, os comentários foram classificados pelo modelo treinado e comparados com a classificação gerada pelo modelo original. Essa comparação permitiu observar diferenças entre os dois modelos, principalmente na identificação de comentários com maior carga negativa.

A Figura 22 apresenta uma tabela comparativa gerada no Google Colab, contendo o título do vídeo, o autor do comentário, o texto analisado, a classificação atribuída pelo modelo treinado, a confiança dessa classificação, a classificação do modelo original e sua respectiva confiança.

Figura 22 – Comparação entre as classificações do modelo treinado e do modelo original

Título do vídeo	Autor	Comentário	Classificação (Treinado)	Confiança (Treinado)	Classificação (Original)	Confiança (Original)
Governar para quem mais precisa	@lucasvinicius_1093	A cada anos mandato fica pior	TOXICO	0.8439	NAO_TOXICO	0.5885
UMA OPORTUNIDADE MUDA TUDO	@jandirv66	A educação transforma e o com conhecimento se controi uma nação.👏👏👏👏👏👏	NAO_TOXICO	0.9835	NAO_TOXICO	0.5572
Ontem tive a alegria de andar no novo VLT de Salvador	@mlr2995	O pix é do Brasil Vamos lutar pela soberania e democracia junto com o nosso presidente Lula.	NAO_TOXICO	0.9560	NAO_TOXICO	0.5632
Segurança pública como prioridade	@oicram2843	Nunca fez nada lixo lixo !!!	TOXICO	0.902473	NAO_TOXICO	0.554211

Fonte: A autora, 2026.

Conforme observado na Figura 22, o modelo treinado conseguiu identificar comentários com teor negativo de forma mais clara do que o modelo original. Nos exemplos analisados, o modelo original classificou todos os comentários como *NAO_TOXICO*, mesmo em casos com expressões negativas, como “A cada anos mandato fica pior” e “Nunca fez nada lixo lixo !!!”. Já o modelo treinado classificou esses comentários como *TOXICO*, indicando maior sensibilidade a termos e construções negativas.

A coluna de confiança representa o grau de segurança do modelo em relação à classificação atribuída. Esse valor é gerado a partir da probabilidade associada à classe prevista: quanto mais próximo de 1, maior é a confiança do modelo naquela classificação. Porém, esse valor não deve ser entendido como garantia absoluta de acerto, mas como um indicativo da segurança da predição.

Também é possível observar que, em alguns casos, o modelo treinado apresentou confiança superior à do modelo original. Enquanto o modelo original classificou comentários com confiança próxima de 0,55, o modelo treinado apresentou valores superiores a 0,84 e 0,90 em algumas classificações. Esse resultado indica que o treinamento contribuiu para tornar o modelo mais ajustado ao tipo de linguagem presente nos comentários analisados.

Observação: o modelo não foi treinado com a classe neutra. Por isso, comentários neutros ou informativos acabam sendo classificados em uma das classes disponíveis, conforme o padrão predominante identificado pelo modelo.

4.6 Considerações sobre os resultados

Os resultados indicam que o modelo se mostrou funcional para classificar comentários em larga escala, principalmente quando aplicado a textos com linguagem mais próxima das redes sociais. A acurácia de 0,8473 e o F1-score macro de 0,8454 mostram que o modelo teve um desempenho satisfatório no conjunto de teste.

Ao mesmo tempo, os resultados também mostram limitações importantes. A matriz de confusão indicou que parte dos comentários tóxicos foi classificada como não tóxica, o que mostra que o modelo ainda pode deixar escapar comentários negativos mais sutis, ambíguos ou dependentes de contexto.

De forma geral, o trabalho evidencia a importância da escolha das bases de dados. O modelo treinado com Comentários Tóxicos PT-BR e Tweets Ekman teve melhor aderência ao tipo de linguagem encontrada no YouTube do que a combinação com IMDB PT-BR. Isso reforça que, em tarefas de análise de sentimentos, a qualidade e a proximidade da base de treinamento com o contexto real de aplicação influenciam diretamente no resultado.

Assim, os resultados obtidos mostram que o modelo pode ser útil para apoiar a análise de grandes volumes de comentários, mas ainda precisa de ajustes para lidar melhor com comentários neutros, irônicos e mais dependentes de contexto.

5 Considerações finais

O trabalho proposto mostrou como aprimorar um modelo de linguagem para análise de sentimentos de comentários no YouTube. Por meio da aplicação do BERTimbau e do uso de bases de dados voltadas à linguagem informal, foi possível construir um modelo funcional, com bons resultados na classificação de sentimentos negativos.

Durante os testes, observou-se que o modelo apresentou overfitting após alguns checkpoints, sendo necessário interromper o treinamento no ponto ideal. Outro fator relevante foi o tempo de processamento: cada época levou um tempo considerável para ser concluída, devido ao porte do modelo e ao tamanho das bases utilizadas. Essa característica impactou diretamente a densidade dos gráficos gerados pelo TensorBoard, que apresentaram menos pontos por conta dos passos mais espaçados.

Apesar dessas limitações, a proposta demonstrou que é possível personalizar um modelo robusto como o BERTimbau para aplicações práticas em português, especialmente em plataformas como o YouTube. Do ponto de vista gerencial, a ferramenta desenvolvida pode ser aplicada para avaliar reputação digital, entender reações públicas a campanhas e orientar decisões de marketing baseadas em sentimento do consumidor.

Dessa forma, os objetivos específicos foram atingidos: foram avaliadas ferramentas disponíveis, foi implementada uma coleta automatizada de dados por meio da API do YouTube, iniciado o treinamento com classificadores, refinada a análise e validados os resultados por meio de testes com dados reais. Como trabalhos futuros, sugere-se ampliar a base de comentários extraídos de redes sociais, incluir uma terceira classe neutra e testar estratégias para reduzir o overfitting, como early stopping automático, congelamento de camadas (*layer freezing*) e ajuste parcial do modelo. Também seria interessante avaliar formas de manter ou converter emojis em descrições textuais, já que eles podem carregar informações importantes sobre sentimento em comentários informais.

Referências

AMARAL, O. E. d. The victory of jair bolsonaro according to the brazilian electoral study of 2018. *Brazilian Political Science Review*, v. 14, n. 1, p. e0004, 2020. Acesso em: 06 maio 2026. Disponível em: <https://brazilianpoliticalsciencereview.org/article/the-victory-of-jair-bolsonaro-according-to-the-brazilian-electoral-study-of-2018/>. Citado 2 vezes nas páginas 15 e 16.

CECCON, D. *Word Embedding: transformando palavras em números*. 2019. Disponível em: <https://iaexpert.academy/2019/04/12/word-embedding-transformando-palavras-em-numeros/>. Citado na página 19.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019. Citado 3 vezes nas páginas 26, 27 e 28.

DIAS, A. d. S. E-book. *Processamento de linguagem natural*. Local da Editora: Editora Saraiva, 2021. Acesso em: 12 mar. 2024. ISBN 9786589881995. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786589881995/>. Citado na página 18.

Didática Tech. *Entenda o que é Underfitting e Overfitting (Machine Learning)*. 2024. Accessed: 2024-08-06. Disponível em: <https://didatica.tech/underfitting-e-overfitting/>. Citado 2 vezes nas páginas 29 e 30.

FRED, L. *IMDb PT-BR*. 2023. Kaggle. Dataset with IMDb data translated into Brazilian Portuguese. Disponível em: <https://www.kaggle.com/datasets/luisfredgs/imdb-ptbr>. Citado na página 32.

FURTADO, M. I. V. *Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula*. Ponta Grossa, PR: Atena Editora, 2019. 11 p. ISBN 978-85-7247-326-2. Disponível em: <https://www.atenaeditora.com.br/index.php/catalogo/post/redes-neurais-artificiais-uma-abordagem-para-sala-de-aula>. Citado na página 19.

Hotjar. *4 Sentiment Analysis Examples to Help You Improve CX*. 2022. Disponível em: <https://www.hotjar.com/user-sentiment/analysis-examples/>. Citado na página 14.

IBM. *Estudo IBM: 41% das empresas no Brasil já implementaram ativamente inteligência artificial em seus negócios*. 2023. Acesso em: 04 mar. 2024. Disponível em: <www.ibm.com/blogs/ibm-comunica/estudo-ibm-41-das-empresas-no-brasil-ja-implementaram-ativamente-inteligencia-artificial-em-seus-ne
Citado na página 14.

JING, H. *Illustrated Guide to Transformer*. 2020. <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>. Accessed: 2024-08-22. Citado na página 24.

MAAS, A. L. et al. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2011. p. 142–150. Citado na página 32.

MAGALHãES, R.; VENDRAMINI, A. Os impactos da quarta revolução industrial. *GVExecutivo*, FGV EAESP, v. 17, n. 1, p. 40–43, Jan/Fev 2018. Citado na página 18.

NETO, G. *Comentários Tóxicos PT-BR*. 2021. Disponível em: <https://www.kaggle.com/datasets/gedorneto/comentrios-toxicos-ptbr>. Citado na página 32.

O GLOBO. *Nike ganha US$ 43 milhões com repercussão de campanha com Colin Kaepernick*. 2018. Acesso em: 06 maio 2026. Disponível em: <https://oglobo.globo.com/economia/nike-ganha-us-43-milhoes-com-repercussao-de-campanha-com-colin-kaepernick-23041666>. Citado na página 15.

REDDY, S. *Transformer Models BERT Model: Overview*. 2023. Acesso em: 04 mar. 2024. Disponível em: <https://www.archishman.com/post/transformer-models-bert-model-overview>. Citado na página 27.

SILVA, R. A. d. Polarização política digital: a contribuição das redes sociais na divisão sociopolítica em bolhas informativas e as consequências para a ciberdemocracia. In: *Anais do 5º Congresso Internacional de Direito e Contemporaneidade: mídias e direitos da sociedade em rede*. Santa Maria, RS: [s.n.], 2019. Acesso em: 06 maio 2026. Citado na página 16.

SOUZA, L. F. *Tweets Ekman PT-BR*. 2023. Kaggle. Dataset of tweets labeled according to Ekman's six basic emotions for sentiment analysis in Portuguese. Disponível em: <https://www.kaggle.com/datasets/luizfelipesouza/tweets-ekman-pt-br>. Citado na página 33.

SUZUKI, K. (Ed.). *Artificial Neural Networks: Methodological Advances and Biomedical Applications*. Rijeka, Croatia: InTech, 2011. ISBN 978-953-307-243-2. Disponível em: <https://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications>. Citado na página 19.

WICK-PEDRO, G.; VALE, O. A. Comentcorpus: descrição e análise de ironia em um corpus de opinião para o português do brasil. *Cadernos de Linguística*, ABRALIN, v. 1, n. 2, p. 01–15, 2020. Disponível em: <https://cadernos.abralin.org/index.php/cadernos/article/view/207>. Citado na página 30.

Apêndices

APÊNDICE A – Modelo Treinado Disponibilizado Publicamente

O modelo treinado desenvolvido neste trabalho foi disponibilizado publicamente na plataforma Hugging Face, com o objetivo de permitir a reprodutibilidade dos experimentos e a continuidade de estudos futuros relacionados à análise de sentimentos em português.

O repositório pode ser acessado no link abaixo:

<https://huggingface.co/jaina134/tcc-jaina-modelo>

APÊNDICE B – Notebook para Teste do Modelo

Além do modelo disponibilizado na plataforma Hugging Face, também foi criado um notebook no Google Colab para permitir o teste prático do modelo treinado. Esse notebook possibilita carregar o modelo publicado, inserir comentários manualmente e visualizar a classificação gerada, juntamente com a confiança da predição.

O notebook pode ser acessado pelo link abaixo:

Acessar notebook no Google Colab