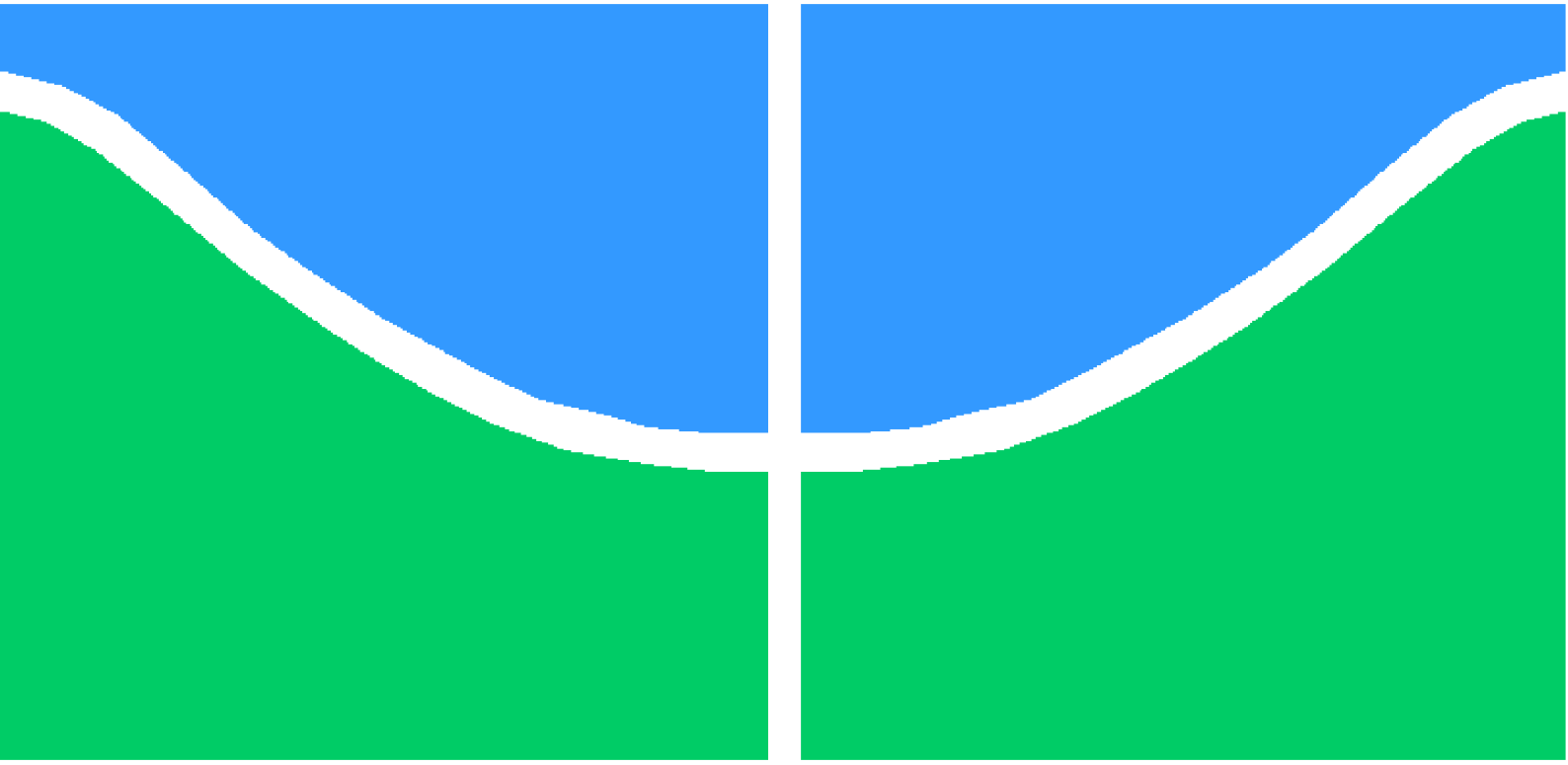


Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Engenharia de Software

# Detecção de Comentários Tóxicos em Chats e Redes Sociais com *Deep Learning*

Autor: Pedro Henrique Vieira de Lima

Orientador: Prof. Dr. Fabricio Ataides Braz

Brasília, DF

2023

Pedro Henrique Vieira de Lima

# Detecção de Comentários Tóxicos em Chats e Redes Sociais com *Deep Learning*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Fabricio Ataides Braz

Brasília, DF

2023

Pedro Henrique Vieira de Lima

# Detecção de Comentários Tóxicos em Chats e Redes Sociais com *Deep Learning*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 14/07/2023

---

**Prof. Dr. Fabricio Ataides Braz**
Orientador

---

**Dr. Nilton Correia da Silva**
Convidado 1

---

**Dr. Henrique Marra Taira Menegaz**
Convidado 2

Brasília, DF
2023

# Agradecimentos

Gostaria de expressar minha profunda gratidão aos meus familiares, especialmente à minha mãe, Cristiane, por todo o suporte e orientação que me foram concedidos desde o início desta jornada e também em outras caminhadas. Sem as suas orientações e apoio incansável, eu não estaria nem perto de ter alcançado o patamar em que me encontro atualmente. Agradeço sinceramente por estar ao meu lado e por acreditar em mim ao longo dessa trajetória acadêmica.

Aos meus amigos mais próximos, gostaria de expressar minha mais sincera gratidão. Suas contribuições valiosas, conselhos inspiradores, apoio incondicional, motivação constante e alegria compartilhada foram fundamentais para o meu progresso e sucesso neste projeto.

Sou verdadeiramente abençoado por ter tido a oportunidade de conhecer pessoas tão incríveis como vocês. Nossas interações e trocas de experiências enriqueceram minha jornada acadêmica e tornaram esta etapa final do meu curso ainda mais significativa.

Além disso, quero agradecer por todos os momentos divertidos e memoráveis que compartilhamos ao longo dessa jornada. Nossos momentos de descontração e o apoio mútuo foram fundamentais para manter o equilíbrio e a motivação durante os desafios enfrentados.

Também desejo expressar minha gratidão aos meus colegas de curso pelo companheirismo ao longo de todas as fases da graduação. Juntos, enfrentamos momentos desafiadores e alcançamos muitos sucessos. Sou grato por cada um de vocês e pelo apoio mútuo que nos proporcionamos ao longo desses anos.

A todos que contribuíram de alguma forma para o meu crescimento acadêmico e pessoal, meu mais sincero agradecimento. Seus encorajamentos, palavras de sabedoria e presença constante foram inestimáveis. Sou imensamente grato por ter compartilhado essa jornada com pessoas tão especiais.

*"Uma pessoa só cresce quando é capaz de superar as dificuldades. Proteção é importante, mas há certas coisas que deve-se aprender por esforço próprio."*

*(Jiraya)*

# Resumo

O crescente aumento do uso de redes sociais e aplicativos de chat online tem trazido consigo um desafio significativo relacionado aos comentários tóxicos. No entanto, a abordagem convencional para enfrentar esse problema muitas vezes não é eficaz o suficiente, especialmente quando se trata de idiomas específicos, como o português. Nesse contexto, o objetivo deste trabalho é preencher essa lacuna por meio da aplicação de tecnologias avançadas, como redes neurais recorrentes, para a detecção de comentários tóxicos em português. Através do treinamento de um modelo utilizando um conjunto de dados contendo sequências classificadas como tóxicas ou não tóxicas, espera-se desenvolver um sistema capaz de distinguir com precisão a qual classe cada sequência pertence. A avaliação e comparação desse modelo com outras abordagens existentes serão realizadas para fornecer *insights* valiosos sobre sua eficácia na detecção de conteúdo tóxico em português, contribuindo assim para a criação de ambientes online mais seguros e saudáveis.

**Palavras-chave**: Inteligência artificial, Processamento de linguagem natural, Redes neurais recorrentes, *Long Short Term Memory*.

# Abstract

The increasing use of social media and online chat applications has led to a significant phenomenon in recent years. However, this growth has also brought about problems, especially concerning toxic comments. To address this issue, the utilization of advanced technologies such as neural networks and natural language processing techniques has become increasingly important. While there already exists some related content, there is indeed a deficit when it comes to specific languages. Therefore, the objective of this work is to create a model, using recurrent neural networks, capable of distinguishing between toxic and non-toxic sequences from a dataset containing such sequences in Portuguese. The model will be evaluated and compared to other models to assess its performance.

**Key-words**: Artificial Intelligence, Natural Language Processing, Recurrent Neural Networks, Long Short Term Memory.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| UnB | Universidade de Brasília |
| IA | Inteligência artificial |
| RNN | *Recurrent neural network* |
| NLP | *Natural Language Processing* |
| LSTM | *Long Short Term Memory* |
| BiLSTM | *Bidirectional Long Short Term Memory* |

# Sumário

# 1 Introdução

A comunicação pela internet tornou-se algo essencial no cotidiano da grande maioria das pessoas. Com a pandemia de COVID-19, ferramentas que possibilitam comunicação de maneira virtual tornaram-se ainda mais relevantes, pois diversas atividades como conversar com amigos e familiares, questões de trabalho e estudo, foram forçadas a serem executadas de forma não presencial. Dentre essas ferramentas, redes sociais e chats online ganharam ainda mais força durante esse período.

Infelizmente, com o aumento do uso das redes sociais e outras plataformas de texto na internet, casos de comentários com toxicidade de linguagem e discursos de ódio se tornaram cada vez mais frequentes e aparentes. Muitas plataformas acabam por não possuir um sistema para prevenir esse tipo de conteúdo e, as plataformas que os possuem, normalmente demoram um tempo considerável para agir sobre tais ações.

Considerando esses casos, a proposta deste trabalho é utilizar técnicas de *machine learning* e *deep learning* para desenvolver um modelo de classificação de conteúdo em mensagens virtuais que tenha a capacidade de identificar quando um determinado conteúdo seja classificado como tóxico ou não.

As sessões posteriores irão se aprofundar tanto em um melhor detalhamento do problema quanto na solução, nos passos para alcançá-la e obstáculos encontrados.

## 1.1 Motivação

A seguir serão apresentados o contexto em que esse trabalho se encontra e o problema que ele explora.

### 1.1.1 Contexto

Tornou-se cada vez mais evidente que o aumento da utilização das redes além de grandes benefícios, trouxe também, uma nova gama de problemas. À medida que o número de usuários aumenta, o fluxo de conteúdo que é gerado também aumenta e caso as plataformas não consigam se adaptar de maneira rápida e eficiente a estas mudanças,

diversos problemas podem acontecer, principalmente problemas relacionados à moderação destes conteúdos. Atualmente, as plataformas empregam diversas técnicas com o objetivo de evitar problemas como a disseminação de postagens que contenham conteúdo relacionado a discursos de ódio ou com um certo grau de toxicidade, como ofensas e conteúdos obscenos.

Existem diversas maneiras de fazer a moderação desses conteúdos. Elas podem ser feitas tanto de maneira manual quanto de forma automatizada por meio de técnicas que visam detectar e controlar determinados tipos de informação (SINGH, 2019). Entre essas técnicas (manual e automatizadas) podemos citar algumas como:

**Filtros de palavras:** Filtros de palavras são um tipo de ferramenta de moderação automatizada e são, geralmente, usados para remover/esconder linguagem inapropriada, palavrões, palavras específicas ou outras palavras ofensivas que podem ser, de alguma forma, prejudiciais ao ambiente ou desrespeitosas para os outros usuários da plataforma. Filtros de palavras também podem ser configurados para remover ou até mesmo substituir automaticamente palavras que estão na sua *blacklist* por outras palavras ou símbolos pré definidos, como asteriscos ou outros sinais para substituição.

- **Ex:** A seguinte frase: "Você é a pessoa mais idiota que eu já vi.", ao passar pelo filtro de palavras, seria transformado em "Você é a pessoa mais ****** que eu já vi".

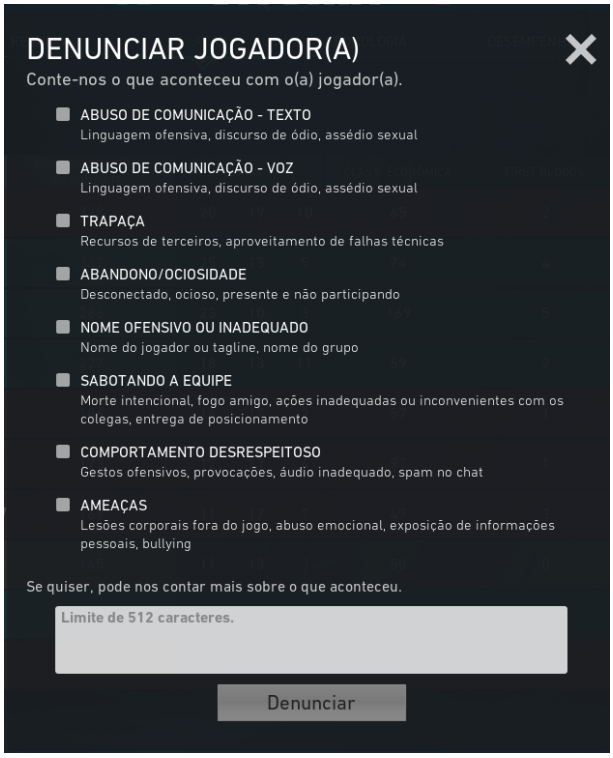
**Regras de condutas:** São regras definidas pela plataforma que, normalmente, são aceitas no momento em que o usuário realiza o seu cadastro na mesma. Essas regras dizem como é o comportamento esperado das pessoas que irão utilizar a plataforma(SINGH, 2019). Um exemplo de regra de conduta pode ser encontrado no site da empresa Epic Games, onde é dito para que se respeite as outras pessoas ao conversar, jogar ou criar (EPIC, 2022). Também é definido que interações com outras pessoas de forma predatória, ameaçadora, intimidadora, obscena, degradante, depreciativa, abusiva ou de maneira a invadir a privacidade de usuários é contra suas regras (EPIC, 2022).

**Moderação manual:** Moderação manual é uma técnica simples que emprega a utilização de pessoas, normalmente contratadas e treinadas, para revisar os conteúdos feitos pelos usuários (SINGH, 2019). Grandes serviços de redes sociais, por exemplo, con-

tratam times focados em moderação de conteúdo de usuário. Esses times, normalmente tem como foco assegurar que as regras definidas pela plataforma sejam fielmente respeitadas (CULLEN; KAIRAM, 2022). Para tal, esses times contam com guias sobre como agir diante de determinados casos (CULLEN; KAIRAM, 2022) (SINGH, 2019). Algumas plataformas como o Reddit e Twitch, deixam com que os próprios usuários (escolhidos pelo dono do grupo do Reddit ou pelo *streamer*) tenham esse poder de moderação (dentro do seu domínio)(CULLEN; KAIRAM, 2022).

**Moderação híbrida:** A moderação híbrida de conteúdo consiste, basicamente, na junção dos métodos de moderação manual com os métodos automatizados (SINGH, 2019). Aliando-se os dois métodos é possível ter um aumento considerável na velocidade dos times de moderação manual, tendo em vista que os métodos automatizados podem levantar *flags* (sinalizar) e, de certa forma, priorizar os conteúdos mais prováveis de desrespeitar as regras e leis da plataforma (SINGH, 2019). Com as *flags* levantadas, os times de moderação manual podem tomar as devidas providências. Normalmente alia-se o sistema moderação automatizado e o sistema de moderação manual com algum outro recurso de *crowdsourcing* como na Figura 1. Ou seja, é possível receber feedback dos usuários via denúncias de conteúdos. Esses conteúdos denunciados podem passar pelos sistemas automatizados sem serem detectados mas, com usuários denunciando conteúdos tóxicos, a chance desses conteúdos não detectados ficarem na plataforma é menor.

Figura 1 – Pop Up de denúncias do jogo Valorant (Riot Games)




### 1.1.2 Problema

Apesar dos métodos apresentados acima e das punições declaradas, casos envolvendo linguagem tóxica, ofensas, abusos e também discursos de ódio aparecem com grande frequência em plataformas de chat de texto. Infelizmente, como citado anteriormente, o volume de usuários em redes sociais aumentou bastante devido a pandemia, ocasionando em um aumento direto no número de conteúdos de texto produzidos. Se o volume de dados for muito grande os métodos apresentados ficam obsoletos (uns mais, outros menos) e acabam se tornando ineficientes, demorando quantidade de tempos consideravelmente longas para aplicar a as medidas cabíveis, o que permite que o usuário que está infringindo as regras continue praticando esses atos por mais tempo.

Em 29 de agosto de 2022 a Riot Games (empresa de jogos eletrônicos) fez uma postagem em sua plataforma explicando de maneira mais detalhada como faz para ter controle sobre esses casos. Nessa postagem a empresa fornece dados a respeito do número de denúncias que ela recebe em seus jogos. Foram, em média, 240 milhões de denúncias por mês em todos os seus jogos no ano de 2021 (RIOT, 2022). Segundo a própria Riot, para acompanhar o ritmo que a empresa recebe essas denúncias, seria necessário que todos seus funcionários analisassem 6 denúncias por minuto durante 365 dias (RIOT, 2022). Esse

cenário de grande número de denúncias não se manifesta apenas em jogos, mas também em redes sociais como Twitter, Reddit e afins. Com o cenário acima apresentado, é visível a existência da necessidade de mecanismos que detectem de maneira mais precisa e rápida esses conteúdos de teor tóxico.

## 1.2 Objetivos

A seguir serão apresentados o objetivo principal deste trabalho e os objetivos específicos que serão alcançados para atingir o objetivo principal.

### 1.2.1 Objetivo Geral

Este trabalho possui como objetivo principal a implementação e uso de redes neurais LSTM e Bi-LSTM para identificação de frases consideradas tóxicas.

Para assegurar que o objetivo geral seja cumprido, alguns objetivos específicos foram definidos.

### 1.2.2 Objetivos específicos

- Definir e coletar *datasets* para fazer o treinamento do modelo.
- Realizar os tratamentos necessários *datasets* nos para treinamento do modelo.
- Fazer análise dos dados coletados.
- Criar uma prova de conceito do projeto.
- Construir e treinar modelos necessários.
- Análise de métricas para validação e comparação dos modelos.

Para este trabalho, a definição de frase tóxica é: Frases que possuem insultos, conteúdos obscenos ou que atacam determinado grupo de pessoas. Além dos modelos citados acima, serão usados também BERT e Naive Bayes para fins de comparação.

# 2 Referencial Teórico

## 2.1 Inteligência artificial

A área de inteligências artificiais (IAs) é uma ampla disciplina que faz parte da área da ciência da computação e engenharia que investiga os limites do que os computadores digitais podem fazer além da computação básica e do armazenamento de dados (JIANG, 2021). Esta área tem como foco a capacidade dos computadores de executar tarefas que normalmente requerem um grau elevado de inteligência humana, como por exemplo, jogar jogos complexos, transcrever e compreender a fala e operar carros de forma autônoma.

O termo inteligência artificial foi criado por John McCarthy em 1956 (JIANG, 2021) na faculdade de Dartmouth. Atualmente o termo inteligência artificial tem sido usado de maneira genérica para definir computadores que conseguem fazer mímicas de funções cognitivas que estão associadas à mente humana como aprender, perceber, raciocinar, etc... (JIANG, 2021).

Inicialmente, durantes os primeiros passos do campo de inteligências artificiais, eram utilizadas as chamadas inteligências artificiais simbólicas. Essas inteligências artificiais dependiam diretamente que conhecimento humano fosse repassado para ela por meio de regras e por esse motivo também são conhecidas como inteligências artificiais baseadas em regras. Todo o conhecimento humano que deveria ser repassado para a inteligência artificial era denominado base de conhecimento (JIANG, 2021). Podemos falar, a grosso modo, que essas inteligências simbólicas eram um grande conjunto de *if-else* (JIANG, 2021) que, posteriormente, ganharam representações mais refinadas como grafos. Infelizmente essa abordagem não era realmente viável em tarefas do mundo real. O autor Hui Jiang cita alguns problemas como:

- Grande dificuldade de abstrair o mundo real em regras bem formuladas.
- A quantidade de regras necessárias poderiam ser absurdamente grandes devido a complexidade do mundo real.

- A manutenibilidade de um sistema desses seria horrível devido grande ao número de suas regras.
- Incapacidade de tomar decisões perante informações incompletas ou incertas.

O primeiro problema relaciona-se à definição de regras para situações que não possuem regras explícitas. Por exemplo, apesar da facilidade dos seres humanos em detectar imediatamente um gato, é realmente complicado definir quais regras são responsáveis por definir este mesmo gato (JIANG, 2021).

O segundo problema citado aborda sobre a complexidade de algumas tarefas. Tarefas extremamente complexas e com diversos cenários e casos diferentes requerem uma quantidade imensa de regras para que todas suas variações sejam mapeadas.

O terceiro problema citado disserta sobre o quão complicado seria manter esse tipo de sistema. Mesmo que em um determinado momento todas as variações de uma atividade complexa fossem mapeadas, caso fosse necessário algum tipo de modificação em alguma regra da base de conhecimento, uma verificação deveria ser feita em todas as outras regras para evitar que essa modificação crie uma contradição. Além disso, modificar uma regra da base pode iniciar um efeito dominó sobre outras regras, que também precisariam ser reavaliadas.

Por fim, o quarto problema citado pelo autor diz a respeito da incapacidade desses modelos simbólicos de tratar casos em que não possuem a informação completa para tomar uma decisão por, provavelmente, ser ainda mais complexo definir regras para esses tratamentos.

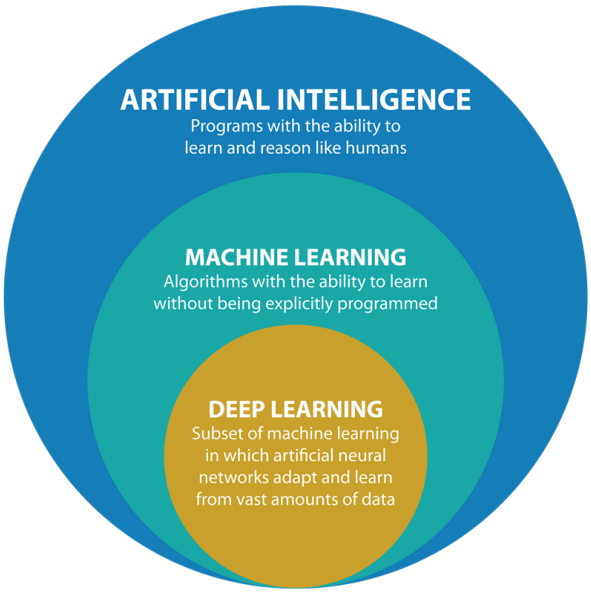
Por volta de 1980, um subcampo da área de inteligência artificial começou a se destacar. Esse campo tinha como foco principal o aprendizado automático por meio de dados para encontrar maneiras de explorar esses dados e, através disso, criar modelos matemáticos. Esse campo recebeu o nome de *machine learning* e foi nomeado em 1959 por Arthur Samuel, pesquisador da IBM.

### 2.1.1 *Machine learning*

Como mostrado na Figura 2, *Machine learning* é um subcampo da área de inteligências artificiais que tem como foco o desenvolvimento de técnicas que permitem que

os computadores aprendam e façam previsões ou tomem decisões baseadas em dados, sem serem explicitamente programados para realizar essas tarefas. Em resumo, *machine learning* é a ciência de programar computadores através de dados. (GéRON, 2019).

Figura 2 – Áreas de inteligência artificial




Fonte: UFSM [1]

*Machine learning* pode ser utilizada para uma vasta gama de tarefas, mas, tem o seu grande brilho em problemas que possuem uma complexidade grande demais para serem resolvidos de forma tradicional ou em problemas que ainda não possuem algoritmos conhecidos para a resolução do mesmo (GéRON, 2019).

Sistemas de *machine learning* podem ser divididos em alguns tipos. A classificação desses sistemas seguem determinados critérios. Uma dessas classificações leva em consideração se o sistema precisa de supervisão humana ou não em seu aprendizado (ex. aprendizado supervisionado, semi-supervisionado, não supervisionado e aprendizado por reforço) (GéRON, 2019). Também é possível classificá-los de acordo com a maneira que aprendem (*online learning* ou *batch learning*) e a maneira como trabalham (*instance based* ou *model-based learning*). Tais classificações não excluem classificações de outros critérios (GéRON, 2019).

Um dos exemplos clássicos de aprendizado supervisionado são tarefas de classificação (GéRON, 2019). Nesse tipo de tarefa, fornecemos ao modelo não somente os dados

[1] Introdução à Machine Learning. Disponível em: <https://www.ufsm.br/pet/sistemas-de-informacao/2021/05/11/introducao-a-machine-learning>. Acesso em: 17 jan. 2023.

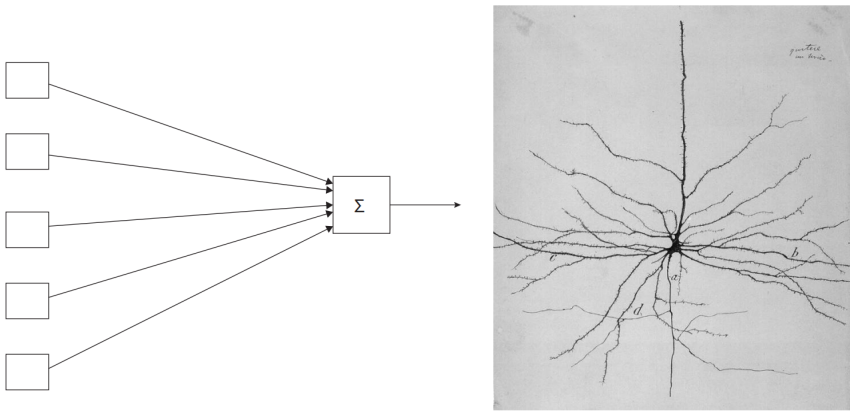
que serão utilizados para seu treinamento mas também a resposta que esperamos para esses dados. Podemos usar de exemplo análise de riscos de clientes de bancos, onde diversos dados sobre o cliente são repassados para o modelo e é esperado que o modelo nos devolva uma entre duas prováveis classes (cliente de alto risco e cliente de baixo risco) (JIANG, 2021). Como o objetivo deste trabalho é a criação e análise de algoritmos que detectam toxicidade em comentários, nos encaixamos exatamente neste tópico de aprendizado supervisionado.

### 2.1.2 *Deep learning*

Como visto na imagem 2 *deep learning* é uma área que se encontra dentro do campo de estudo de *machine learning* (CHOLLET, 2018). A ideia principal por trás da área de *deep learning* é a construção de algoritmos de *machine learning* a partir de séries de camadas interconectadas por nós ou neurônios artificiais que podem ser empilhadas, o que resulta em sua profundidade.

Redes neurais artificiais foram, em grande parte, inspiradas em redes neurais biológicas de animais e seres humanos (JIANG, 2021). Acredita-se que a inteligência dos animais esteja diretamente relacionada à vasta quantidade de neurônios presentes em seus cérebros. Essa rede neural é composta por um neurônio que está conectado a centenas de outros neurônios, que por sua vez também estão interligados a várias outras centenas de neurônios (JIANG, 2021). Cada neurônio por si só é consideravelmente simples recebendo e enviando impulsos mas, uma rede com vários neurônios pode performar funções extremamente complexas dependendo apenas das forças dessas ligações (pesos). Uma das primeiras implementações de um neurônios surgiu por volta de 1957 (JIANG, 2021) e se parecia bastante com um neurônio biológico. Esse implementação possui diversas entradas, que representam os dendritos, e apenas uma saída (axônio). Essa implementação recebeu o nome de perceptron. A Figura 3 demonstra a semelhança entre um neurônio e um perceptron.

Figura 3 – Perceptron e neurônio lado a lado




Fonte: (CHARNIAK, 2018)

Explicando de maneira simples, a implementação do perceptron inicia-se com um vetor de pesos que representam as forças das ligações. Em seguida, acontece uma iteração sobre todo dado separado para treinamento. Quando algum erro é encontrado, o vetor de peso será atualizado seguindo uma determinada regra. Ao final, espera-se que os pesos estejam adaptados para que o modelo não erre mais.

Conforme o passar do tempo diversos tipos de redes neurias surgem. Algumas das mais comuns são:

- **Redes Neurais Feedforward:** São as redes neurais mais simples e comuns. Nesse tipo de rede as informações fluem em uma direção, da camada de entrada para a camada de saída, sem voltar para trás. Essas redes são usadas para tarefas de classificação e regressão (GOODFELLOW; BENGIO; COURVILLE, 2016).

- **Redes Neurais Convolucionais:** Redes neurais convolucionais são projetadas para trabalhar com imagens e são comumente usadas em tarefas de visão computacional, como reconhecimento de imagens e detecção de objetos. Elas usam camadas de filtros para extrair características importantes das imagens (GOODFELLOW; BENGIO; COURVILLE, 2016).

- **Redes Neurais Recorrentes:** Essas redes possuem conexões de *feedback*, ou seja, as informações podem fluir de volta para a camada anterior, permitindo que a rede "memorize" informações anteriores. Normalmente essas redes são utilizadas

para tarefas de processamento de séries temporais, como previsão de tendências (GOODFELLOW; BENGIO; COURVILLE, 2016).

Esses são apenas alguns exemplos dos tipos de redes neurais disponíveis. Novos tipos de redes neurais continuam sendo desenvolvidos.

#### 2.1.2.1 *Long Short Term Memory*

O principal problema que podemos encontrar em redes neurais recorrentes simples é que apesar de que tenham a capacidade de "lembrar" informações já vistas, elas também esquecem esses dados de maneira relativamente rápida caso os dados sejam longos demais (CHARNIAK, 2018). Para que esse problema fosse superado, foi desenvolvida a arquitetura de rede neural conhecida como *Long Short-Term Memory* (LSTM). LSTMs são um tipo de rede neural recorrente mais poderosa e avançada que as redes recorrentes mais simples (CHARNIAK, 2018) e foram desenvolvidas para solucionar os problemas de perda de informação a longo prazo que ocorrem nas RNNs simples. A estrutura das LSTMs é mais complexa e possui componentes que ajudam na memorização a longo prazo como, por exemplo, células de memória e estados ocultos, que permitem ao modelo capturar e manter as dependências a longo prazo presentes nos dados de entrada. Por possuir essa característica, LSTMs são muito úteis em atividades relacionadas com processamento de sequência como, por exemplo, previsão de séries temporais e processamento de linguagem natural.

## 2.2 NLP

Processamento de linguagem natural é um área de pesquisa que consiste na criação de algoritmos que podem receber ou até mesmo produzir linguagem natural desestruturada (GOLDBERG, 2017) ou seja, a área de processamento de linguagem natural tem por objetivo fazer com que máquinas obtenham a capacidade de entender nossa linguagem, seja ela em sua forma escrita ou falada (GANEGEDARA; LOPATENKO, 2022).

Enquanto os seres humanos tem a capacidade de entender, produzir e perceber expressões de maneira extremamente fácil, para computadores é uma tarefa realmente complicada (GOLDBERG, 2017) pois a linguagem humana possui um grande nível de ambiguidade (banco monetário vs banco de praça), existem inúmeras maneiras distintas

de criar expressões que possuem o mesmo sentido e está em constante mudança e evolução (GOLDBERG, 2017). É válido relembrar que cada linguagem possui a sua própria gramática, sintaxe e também o seu vocabulário o que acaba dificultando ainda mais os desafios da área de processamento de linguagem natural.

### 2.2.1 Tarefas

Apesar desses desafios, a área de NLP traz grandes recompensas para aqueles que conseguem superá-los. Processamento de linguagem natural possui uma grande variedade de aplicações no mundo real atual. Seguem algumas dessas aplicações:

- ***Question answering:*** Sendo, provavelmente, a tarefa que possui maior valor comercial, é normalmente encontrada em *chatbots* e assistentes inteligentes, como por exemplo a Alexa da Amazon ou a Siri da Apple (GANEGEDARA; LOPATENKO, 2022). Atualmente, *question answering* também tem sido bastante utilizado para dar suportes para consumidores de diversos produtos online ou sanar possíveis dúvidas dos mesmos. Essas tarefas de resposta a perguntas dependem fortemente de outra tarefa também performada na área de processamento de linguagem natural que é a desambiguação de palavras.

- ***Named Entity Recognition (NER):*** Esta atividade busca fazer a identificação e classificação de entidades nominais em um texto, como nomes de pessoas, lugares, organizações e outras entidades mencionadas no texto. O objetivo final dela é extrair informações estruturadas do texto não estruturado. Podemos citar como exemplos de aplicações, extração de contatos de um e-mail e recuperação de informações em bancos de dados. (GANEGEDARA; LOPATENKO, 2022).

- ***Part of speech (PoS) tagging:*** Um pouco semelhante à atividade de NER, *part of speech tagging* tem como objetivo identificar e classificar a função gramatical de cada palavra em um corpo de texto, como substantivos, verbos, adjetivos, etc (GANEGEDARA; LOPATENKO, 2022). A atividade busca fornecer uma análise estrutural do texto, que pode ser usada para diversas aplicações, como a análise de sentimentos, a geração de resumos e a tradução automática. *Part of speech tagging* também é uma etapa imprescindível na análise sintática de uma sentença, que por sua vez, é a base para a compreensão do significado das palavras e frases.

- ***Sentence classification:*** Tem por objetivo classificar uma sentença em uma determinada categoria ou etiqueta (GANEGEDARA; LOPATENKO, 2022). É uma maneira de análise de texto, que pode ser usada para análise de sentimentos, detecção de spam, classificação de notícias, etc...

- ***Text generation:*** Como descrito pela próprio nome da atividade, esta tem como objetivo a criação de novos textos a partir de textos aprendidos durante o treinamento do modelo (se usado com redes neurais) (GANEGEDARA; LOPATENKO, 2022).

- ***Machine translation:*** Possui o objetivo de realizar a transformação de frases de uma linguagem específica para alguma outra linguagem selecionada (GANEGEDARA; LOPATENKO, 2022). Essa tarefa parece simples inicialmente, porém as diferentes estruturas sintáticas entre as linguagens acabam por dificultar que ela seja concluída.

Existem também atividades relacionadas ao pré-processamento de textos como:

- **Tokenização:** Tokenização consiste na atividade de, dado um corpo de texto, fazer a separação deste corpo em pedaços atômicos (palavras ou caracteres). A primeira vista pode parecer uma tarefa simples porém, em linguagens como o japonês por exemplo as palavras não são delimitadas por espaços ou pontuação (GANEGEDARA; LOPATENKO, 2022).

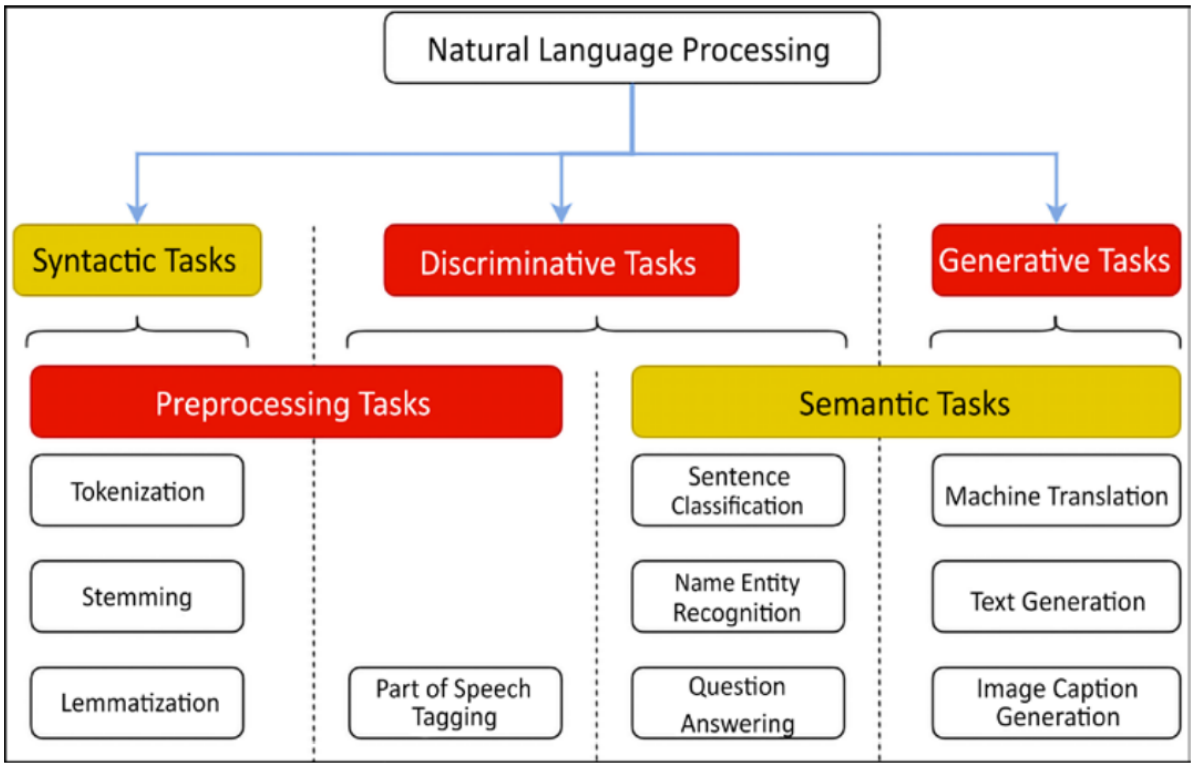
- ***Stemming:*** *Stemming* é a atividade que, a partir das palavras de um corpo de texto, faz a redução dessas palavras a sua forma básica, também conhecida como "radical" ou "raiz" da palavra. Isso é feito removendo sufixos, prefixos e outras variações da palavra, deixando apenas a parte mais importante e significativa (INDURKHYA; DAMERAU, 2010).

- ***Lemmatization:*** A Lematização é um processo semelhante ao *stemming*, porém é mais preciso e sofisticado. Enquanto o *stemming* simplesmente remove sufixos e prefixos da palavra, a *lemmatization* utiliza uma análise morfológica mais aprofundada para determinar a forma canônica (ou "lemma") de uma palavra. Isso envolve o uso de dicionários de linguagem natural e regras gramaticais para encontrar a forma

base da palavra, incluindo sua categoria gramatical (verbo, substantivo, adjetivo, etc.)(INDURKHYA; DAMERAU, 2010).

Como pode ser percebido, cada atividade pode se encaixar em uma categoria de atividade de processamento de linguagem natural, como mostrado na Figura 4:.

Figura 4 – Categorias de atividades de NLP




Fonte: Tushan, (2022)

### 2.2.2 Redes neurais e NLP

Com o avanço das redes neurais, tornou-se possível a utilização dessas tecnologias em problemas de processamento de linguagem natural. Para alcançar esses resultados, é necessário adicionar uma camada específica às redes neurais, que tem a responsabilidade de mapear os vetores de símbolos contínuos para um espaço matematicamente operável e relativamente menor. Além disso, também existem diversos tipos de redes neurais que podem ser utilizadas para obter bons resultados em tarefas de processamento de linguagem natural (NLP). Dentre elas, podemos citar três tipos mais comuns: Redes neurais *feed-forward*, redes neurais recorrentes e a recente arquitetura de *transformers*. (GOLDBERG, 2017).

As redes *feedforward* são muito eficientes na identificação de padrões locais nos dados, identificando características independentes do seu posicionamento ao longo do documento completo (GOLDBERG, 2017). Por esse motivo, elas possuem ótimos resultados em algumas tarefas de processamento de linguagem natural. Já as redes neurais recorrentes são muito boas para performar tarefas que possuem dados sequenciais (GOLDBERG, 2017) e também possuem a capacidade de "memorizar" informações, o que ajuda bastante no entendimento de contexto. Por esses motivos, são os modelos de redes neurais bem interessantes para tarefas de processamento de linguagem natural.

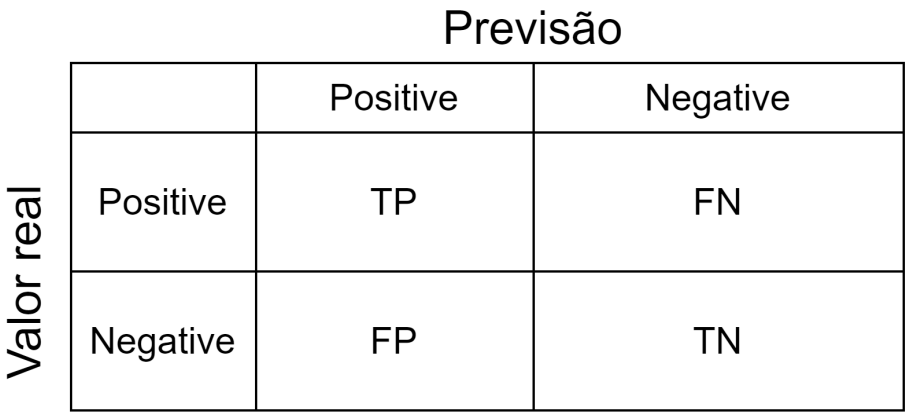
A arquitetura mais recente, conhecida como *transformers*, tem sido amplamente adotada em diversas aplicações na área de Processamento de Linguagem Natural (PLN) e tem alcançado resultados de ponta em várias delas. Uma das principais vantagens dessa arquitetura reside em seus mecanismos de atenção. Esses mecanismos operam ao calcular a atenção entre pares de elementos da sequência de entrada, permitindo que a rede focalize elementos específicos da sequência durante o seu processamento (GANEGEDARA; LOPATENKO, 2022). Essa capacidade possibilita que a rede lide com contextos mais abrangentes e estabeleça dependências de longo alcance, o que, por sua vez, aumenta significativamente a eficácia em tarefas de PLN.

### 2.2.3 Métricas para modelos de classificação de texto

As métricas de avaliação de modelos de classificação são utilizadas para mensurar a performance de um modelo que realiza essa tarefa específica. Essas métricas desempenham um papel crucial ao identificar os pontos fortes e fracos do modelo, além de fornecer *insights* sobre seu desempenho em relação aos conjuntos de dados utilizados tanto no treinamento quanto no teste. Além disso, as métricas de avaliação possibilitam a comparação entre diferentes modelos de classificação, auxiliando na seleção daquele que melhor se adequa ao problema em questão. Ao analisar essas métricas, é possível tomar decisões embasadas e fundamentadas na busca pela eficiência e precisão do modelo.

Uma das principais ferramentas para fazer essa avaliação é a matriz de confusão. Essa ferramenta apresenta uma relação entre as classes reais dos dados e as classes apontadas pelo modelo (HOSSIN; M.N, 2015). A Figura 5 representa a matriz de confusão de um modelo de classificação binária.

Figura 5 – Exemplo de matriz de confusão




A matriz de confusão apresentada (Figura 5) pode ser interpretada da seguinte maneira:

- ***True positive* (TP):** Representa quantas vezes a classe era positiva e o modelo a previu como positiva (quantos positivos o modelo efetivamente acertou).
- ***True Negative* (TN):** Representa quantas vezes a classe era negativa e o modelo a previu como negativa (quantos negativos o modelo efetivamente acertou).
- ***False Positive* (FP):** Representa quantas vezes a classe era negativa e o modelo a previu como positiva (quantos falsos positivos o modelo cometeu).
- ***False Negative* (FN):** Representa quantas vezes a classe era positiva e o modelo a previu como negativa (quantos falsos negativos o modelo cometeu).

Ao observarmos a quantidade de vezes em que o modelo classificou corretamente as instâncias positivas e negativas, podemos obter uma boa noção do desempenho do modelo em relação a um conjunto de dados específico. Da mesma forma, ao analisarmos os casos de falsos positivos e falsos negativos, podemos identificar possíveis deficiências do modelo.

A partir dessas estatísticas iniciais que a matriz de confusão nos proporciona, podemos derivar mais algumas métricas que são extremamente importantes para a análise de performance do modelo de classificação. Seguem as métricas e o que elas significam:

- **Acurácia:** É a proporção da quantidade de acertos que o modelo teve (TP + TN) sobre todas as previsões feitas (certas e erradas). Essa medida mostra o quão bom

o modelo é em acertar as classes corretas.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

- **Precisão:** É a proporção entre a quantidade de previsões corretas (TP ou TN) e o número total de previsões da mesma classe (TP + FP ou TN + FN). A precisão indica o quão preciso é o modelo em identificar os casos corretos dessas classes.

$$\frac{TP}{TP + FP} \quad \text{ou} \quad \frac{TN}{TN + FN} \tag{2.2}$$

- ***Recall:*** É a proporção entre as previsões positivas corretas (TP) (ou negativas corretas) e o número total de casos positivos ou negativos. A medida de *recall* mostra o quão bom o modelo é em identificar os positivos de cada classe.

$$\frac{TP}{TP + FN} \quad \text{ou} \quad \frac{TN}{TN + FP} \tag{2.3}$$

- ***F1-Score:*** É a média harmônica entre precisão e *recall*. O *f1-Score* é útil quando tanto precisão quanto *recall* são importantes para a tarefa em execução. Quanto mais alto for o *f1-Score*, melhor é o equilíbrio entre precisão e *recall*.

$$\frac{2 \times precisão \times recall}{precisão + recall} \tag{2.4}$$

A partir dessas métricas podemos avaliar a performance do nosso modelo e incrementa-lo quando e onde for necessário.

# 3 Materiais e Métodos

## 3.1 Considerações Iniciais

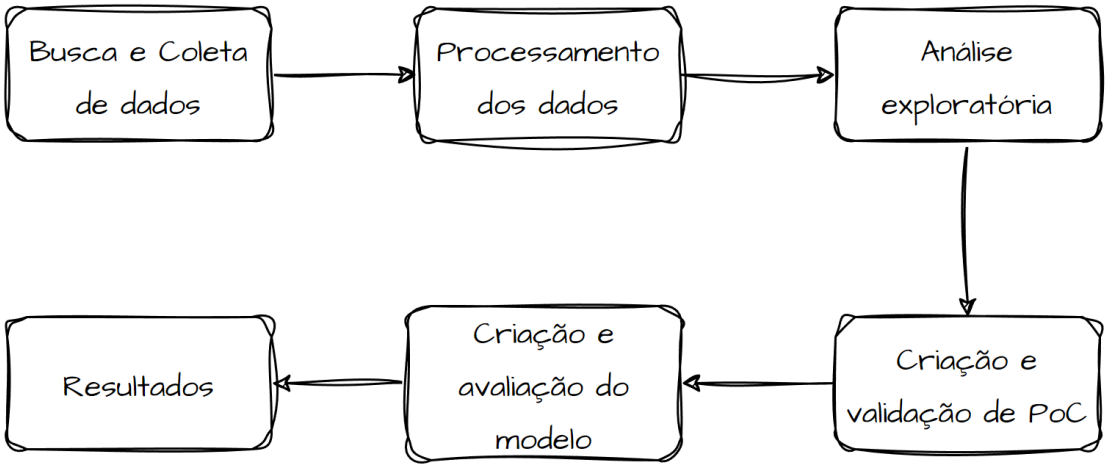
Neste capítulo apresenta-se o plano metodológico adotado para alcançar o objetivo deste trabalho, isto é, quais passos serão seguidos para que os objetivos específicos sejam completamente cumpridos e, por consequência, o objetivo principal.

## 3.2 Plano Metodológico

O plano metodológico adotado para esse trabalho possui seis fases apresentadas a seguir e na Figura 6.

- Procura e coleta dos dados necessários para realização do projeto;
- Processamento e análise desses dados;
- Criação de prova de conceito para análise de viabilidade do projeto;
- Implementação modelos propostos;
- Utilização de métricas para análise dos modelos;
- Disponibilização de resultados e discussões.

Figura 6 – Etapas do Projeto



### 3.2.1 Busca e coleta de dados

A busca e a coleta de dados representam os pontos de partida essenciais em diversos projetos de inteligência artificial e análise de dados. O *dataset*, por sua vez, constitui-se como uma compilação de informações utilizadas para o treinamento e teste de modelos de aprendizado de máquina. Essa coleção de dados abrange uma ampla gama de tipos, como imagens, textos, números e outras formas de informação. Tanto a qualidade quanto a quantidade dos dados disponíveis têm um impacto significativo na precisão e no desempenho dos modelos aos quais esse *dataset* é aplicado.

A busca e a coleta de dados envolvem a identificação de fontes confiáveis e relevantes, seguida pela organização e consolidação desses dados de forma consistente. Esse processo pode incluir a exploração de bases de dados públicas, a extração de dados da internet por meio de técnicas como *web scraping*, a coleta de informações por meio de questionários e outras fontes pertinentes. É crucial avaliar a qualidade dos dados coletados, eliminando duplicatas e inconsistências, além de verificar a precisão e a validade dessas informações. Além disso, é fundamental documentar todas as etapas do processo de coleta de dados, incluindo as fontes utilizadas, os procedimentos adotados e quaisquer dificuldades encontradas. Essa documentação torna mais fácil a revisão e a validação dos dados, promovendo a transparência e a confiabilidade do estudo.

### 3.2.2 Processamento de dados

Após a etapa de busca e aquisição dos dados necessários, procede-se ao processamento e análise desses dados. Nessa fase, os dados coletados anteriormente passam por um conjunto de atividades que visam sua limpeza, organização e preparação para o treinamento dos modelos. Essas tarefas envolvem a identificação e remoção de dados duplicados, a tomada de decisões a respeito de dados ausentes, a normalização dos dados e a transformação desses em um formato adequado para seu uso posterior.

### 3.2.3 Análise de dados

Essa fase também compreende a realização de uma análise exploratória dos dados, visando obter uma compreensão mais profunda das características dos dados e identifi-

car possíveis problemas ou tendências relevantes. A análise exploratória dos dados pode abranger os seguintes aspectos:

- **Visualização de dados:** Consiste em utilizar gráficos para visualizar a distribuição das palavras e classes de sentenças nos dados coletados.
- **Análise de frequência de palavras:** Consiste em calcular a frequência de palavras para identificar as palavras mais comuns e, a partir disso, fazer comparações e observar padrões e tendências entre as classes de textos. Isso pode dar uma ideia tanto das palavras mais comuns quanto das que estão presentes nas duas classes de texto.

O objetivo final dessa etapa é ter uma compreensão sólida dos dados e garantir que eles sejam adequados para o que o objetivo proposto seja alcançado.

### 3.2.4 Criação e validação de PoC

PoC é a sigla para "Prova de Conceito" (em inglês, *Proof of Concept*). É um experimento ou demonstração utilizado para validar a viabilidade de uma ideia ou projeto. Ela geralmente é usado para testar se uma tecnologia ou abordagem específica é capaz de atender aos objetivos e requisitos de um projeto antes que o investimento de tempo e recursos em uma implementação completa seja feito.

A prova de conceito será feita com base em um modelo probabilístico. Um modelo probabilístico é, em resumo, a aplicação de princípios da estatística em análise de dados. (CHOLLET, 2018). Esses modelos são as formas mais primitivas de algoritmos de *machine learning* mas, apesar disso, ainda são amplamente utilizados. (CHOLLET, 2018).

Sendo assim, o modelo escolhido para realização da prova de conceito foi o Naive Bayes. Esse modelo é um tipo de classificador de aprendizado de máquina baseado no teorema de Bayes, que assume que as características dos dados de entrada são independentes entre si (CHOLLET, 2018).

### 3.2.5 Implementação do modelo proposto e avaliação

A fase de implementação do modelo proposto é o estágio em que o modelo selecionado é efetivamente codificado e treinado utilizando os dados coletados nas etapas

anteriores. Nessa fase, o modelo é construído e testado com base nas métricas estabelecidas para avaliar seu desempenho em uma tarefa de classificação de dados em português. Os modelos escolhidos para este trabalho são as redes LSTM (*Long Short Term Memory*) e Bi-LSTM (*Bidirectional Long Short Term Memory*), conhecidas por sua capacidade de "memórização" de informações de longo prazo. As métricas adotadas para avaliação das mesmas são:

- *Precision*
- *Recall*
- *F1-Score*
- *Accuracy*

### 3.2.6 Disponibilização de resultados

A fase de disponibilização de resultados é a etapa final do trabalho onde os resultados obtidos ao longo do projeto são apresentados e discutidos. É nesta fase que o trabalho é finalizado e o resultado é compartilhado com o público interessado. Códigos e demais dados também são disponibilizados para os mesmos.

## 3.3 Materiais

### 3.3.1 *Datasets*

Diversos conjuntos de dados foram coletados para este estudo, com ênfase na língua portuguesa. Infelizmente, a disponibilidade de conjuntos de dados específicos ou extensos sobre o tema é bem limitada. Portanto, foi necessário utilizar múltiplos conjuntos de dados para alcançar resultados aceitáveis.

O primeiro *dataset* (e o principal) adotado foi o ***Toxic Language Dataset for Brazilian Portuguese*** (ToLD-Br). O ToLD-Br é um extenso conjunto de dados composto por tweets em português brasileiro que contêm conteúdo tóxico. Esses dados foram coletados por meio de uma abordagem de *crowdsourcing*, envolvendo a contribuição de 42 anotadores selecionados a partir de um grupo inicial de 129 voluntários. Os anotadores

foram selecionados com o objetivo de criar um grupo plural em termos de demografia (etnia, orientação sexual, idade, gênero). Cada *tweet* foi rotulado por três anotadores em 6 categorias possíveis: Fobia LGBTQ+, xenofobia, conteúdo obsceno, insulto, misoginia e racismo. (LEITE et al., 2020). Além desse, outros *datasets* complementares foram coletados:

- **Olid-br:** É um *dataset* para detecção de toxicidade em português contendo 6.354 comentários anotados manualmente usando um esquema hierárquico com múltiplos níveis de granularidade (TRAJANO; BORDINI; VIEIRA, 2022)
- **Offcombr:** Um *dataset*, por (PELLE; MOREIRA, 2017), com aproximadamente 1.033 comentários anotados nas seguintes categorias: racismo, sexismo, homofobia, xenofobia, intolerância religiosa e ofensa.
- ***Portuguese Hate Speech Dataset*:** Um *dataset* por (FORTUNA et al., 2019) que possui aproximadamente 5.670 comentários classificados entre conteúdo com discuros de ódio e sem discuros de ódio.

No contexto deste trabalho, será adotada uma abordagem de conversão das múltiplas classificações dos datasets em classificações binárias relacionadas ao nível de toxicidade das respectivas frases. Portanto, frases que pertençam a classes como xenofobia, misoginia, insultos, entre outras, serão anotadas em uma nova coluna que representará, de forma binária (0 ou 1), se a frase é considerada tóxica ou não. Essa conversão permitirá simplificar a análise e focar na classificação geral da toxicidade das frases.

## 3.4 Ferramentas

### 3.4.1 Colab

Este trabalho utilizará o *Colaboratory*, conhecido como Colab, como ferramenta principal para análise de dados e desenvolvimento dos modelos propostos. O Colab é uma plataforma gratuita, que permite acesso às ferramentas de aprendizado de máquina e processamento de dados do Google. Com esta ferramenta, será possível realizar análises detalhadas dos dados e desenvolver modelos precisos e eficientes (GOOGLE, 2019). A Tabela 1 apresenta as especificações do ambiente de desenvolvimento do Colab.

| **Hardware** | **Especificações** |
|---|---|
| CPU | 2 núcleos, 2.00GHz |
| Memória | 13GB DDR3 |
| GPU (quando disponível) | NVIDIA Tesla T4, 16GB GDDR6 |
| Armazenamento | 100GB (dinâmico, depende do uso) |

Tabela 1 – Especificações de *hardware* gratuitas para Google Colab

### 3.4.2 Bibliotecas

Com o intuito de alcançar os objetivos mencionados anteriormente, serão empregadas várias bibliotecas que disponibilizam ferramentas de aprendizado de máquina e análise de dados. As principais bibliotecas utilizadas serão as seguintes:

- **TensorFlow:** TensorFlow é uma biblioteca de código aberto para aprendizado de máquina desenvolvida pelo Google. Ele permite que os usuários construam e treinem modelos de aprendizado de máquina de forma eficiente e escalável. Ele também oferece uma ampla variedade de ferramentas para visualização e depuração de modelos (ABADI et al., 2015).

- **Keras:** Keras é uma biblioteca de redes neurais de alto nível para o TensorFlow. Ele permite que os usuários construam modelos de redes neurais de forma fácil e rápida, sem se preocupar com os detalhes de baixo nível. Ele também oferece uma ampla variedade de camadas pré-treinadas e modelos para uso imediato (CHOLLET et al., 2015).

- **Pandas:** Pandas é uma biblioteca de análise de dados para Python. Ele permite que os usuários manipulem e analisem dados de forma eficiente e intuitiva, usando estruturas de dados como *DataFrames* e *Series*. Ele também oferece uma ampla variedade de funções de manipulação de dados, como agrupamento, junção e filtragem (TEAM, 2020).

- **Matplotlib:** Matplotlib é uma biblioteca de visualização de dados para Python. Ele permite que os usuários criem gráficos e *plots* de alta qualidade de forma fácil e rápida. Ele também oferece uma ampla variedade de estilos e personalizações para gráficos, bem como suporte a vários formatos de saída (HUNTER, 2007).

- **Scikit-learn:** Scikit-learn é uma biblioteca de aprendizado de máquina para Python. Ele oferece uma ampla variedade de algoritmos de aprendizado supervisionado e não

supervisionado, incluindo regressão, classificação, agrupamento e redução de dimensionalidade (PEDREGOSA et al., 2011).

- **Simpletransformers:** O Simple Transformers (RAJAPAKSE, 2021) é uma biblioteca de aprendizado de máquina de código aberto desenvolvida por Thilina Rajapakse. Ela é projetada para simplificar a tarefa de treinar, avaliar e fazer previsões com modelos de linguagem baseados em transformers.

## 3.5 Considerações Finais

Neste capítulo, foi apresentado o plano metodológico adotado para se atingir os objetivos desta pesquisa e também as ferramentas que servirão de suporte para tal. No próximo capítulo apresenta-se os resultados iniciais derivados desse plano.

# 4 Resultados

## 4.1 Considerações Iniciais

Nesta seção, serão apresentados os resultados preliminares de acordo com o plano metodológico e suas etapas descritas na seção 3. Esta seção seguirá fielmente a ordem do plano, abordando os processos, apresentando os resultados obtidos e realizando discussões sobre os mesmos.

## 4.2 Coleta de dados

Os dados utilizados para este trabalho possuem como origem trabalhos e artigos semelhantes encontrados pela internet. O primeiro trabalho encontrado sobre o tema foi o ***Toxic Language Detection in Social Media for Brazilian Portuguese: New Dataset and Multilingual Analysis*** por (LEITE et al., 2020). Esse artigo diz sobre a criação e análise de um novo *dataset*, majoritariamente em portugues, e a análise dele após finalizada a coleta e anotação das classes. A partir deste artigo foi possível encontrar trabalhos correlatos a este tema, o que facilitou a busca por outras fontes de dados semelhantes. Com isso, os *datasets* complementares citados na seção 3 também foram coletados, baixados para um armazenamento local e depois enviados para a ferramenta de armazenamento em nuvem da Google (Google drive) para posterior utilização via Colab.

## 4.3 Pré-processamento e análise dos dados

O pré-processamento de dados consiste em limpar os dados coletados anteriormente para facilitação da análise e utilização posterior durante a fase de implementação dos modelos. Esse pré-processamento é aplicado a todos os *datasets* coletados. Antes de iniciarmos os pré-processamentos, devemos considerar que as frases têm como fonte a rede social Twitter e por isso, marcações características da rede social devem ser limpas

também. Durante o pré processamento, novas colunas serão adicionadas ao *dataset* final. Essas colunas significam *checkpoints* de mudanças significativas nos textos.

### 4.3.1 Pré-processamento dos textos

Inicialmente é necessário passar todos os textos para letras minúsculas para que, logo após isto, partes indesejadas do texto sejam removidas (marcações de usuários, *links*, *hashtags*, *emojis*, palavras características da plataforma, etc...) como na Tabela 2. Em seguida também são removidos pontuações e números.

| **Antes** | **Depois** |
|---|---|
| RT USER USER pensei a mesma coisa HASHTAG | pensei a mesma coisa |

Tabela 2 – Exemplo após a limpeza simples inicial

A seguir removemos *stop words*. A remoção de *stop words* é uma técnica amplamente utilizada em tarefas de processamento de linguagem natural. Alguns exemplos de *stop words* são “é”, “de”, “em”, “o”, “a”, etc... Essas palavras aparecem frequentemente em todos os tipos de texto e não possuem um significado específico na frase ou agregam muito pouco significado a ela, mas podem distorcer ou prejudicar a análise de dados (ruído), aumentando a quantidade de dados sem informação relevante (INDURKHYA; DAMERAU, 2010). A lista de *stop words* foi adquirida a partir das bibliotecas **nltk** e **spacy**.

| **Antes** | **Depois** |
|---|---|
| pensei a mesma coisa | pensei mesma |

Tabela 3 – Exemplo de remoção de *stop words*

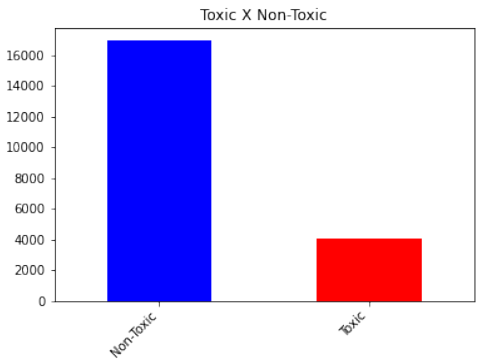
### 4.3.2 Balanceamento do *dataset*

Após limparmos os *datasets*, podemos passar para a etapa de análise exploratória dos dados. Nesta etapa buscamos entender melhor as características dos dados coletados e tratados. Durante uma primeira análise, é possível observar que todos os *datasets* (analisados individualmente) estão desbalanceados (possuem um grande número de dados pertencentes a classe X e poucos dados pertencentes a classe Y, por exemplo). A Tabela 4 e a Figura 7 mostram a quantidade de linhas por *dataset* e a distribuição de frases tóxicas ou não tóxicas.

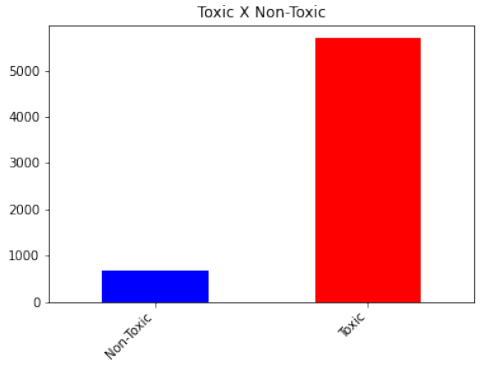
| **Dataset** | **Linhas** | **Qtd. Tóxicos** | **Qtd. Não Tóxicos** |
|---|---|---|---|
| told-br | 21000 | 4063 | 16937 |
| olid-br | 6354 | 5691 | 663 |
| phds | 5670 | 1788 | 3882 |
| offcom-br | 1033 | 202 | 831 |
| **Total** | 34057 | 11744 | 22313 |

Tabela 4 – Linhas por *dataset*

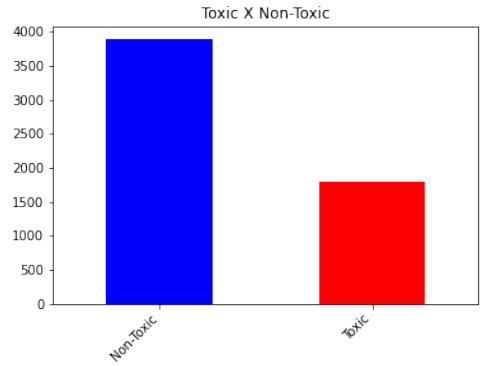
Figura 7 – Balanceamento dos dados



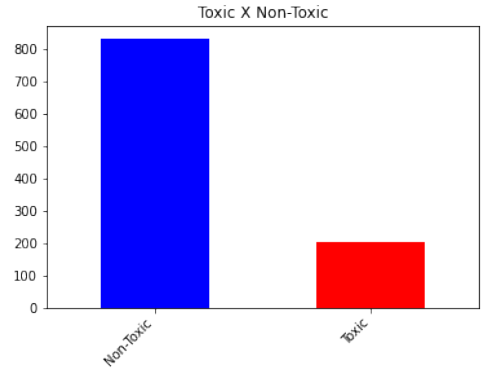

Told-br



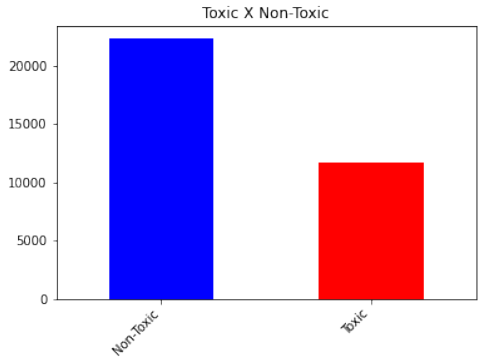

Olid-br




PHDS




Offcom-br

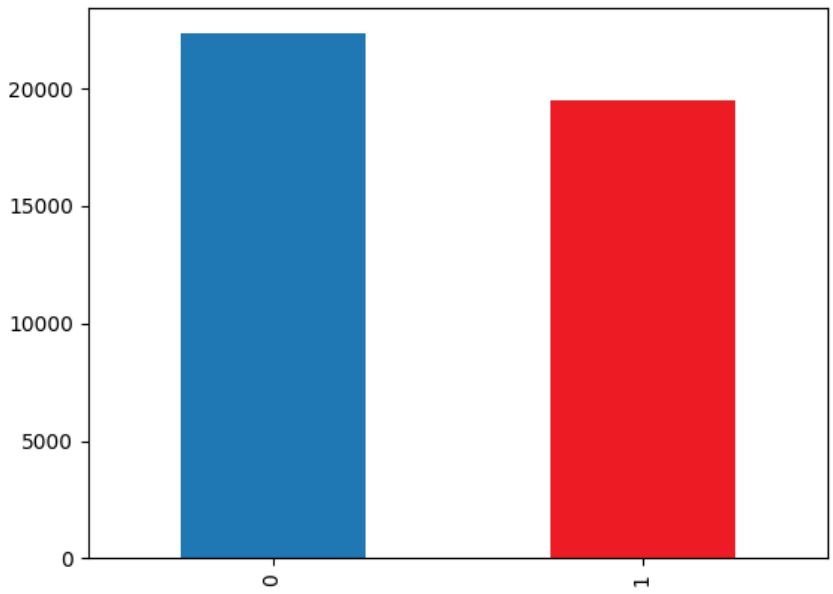



*Datasets* Juntos

Existem algumas técnicas que podemos utilizar para tratarmos desse desbalanceamento. Podemos, por exemplo, fazer o *undersampling* dos dados, isto é, reduzir o número de dados da classe que possui mais, igualando a quantidade de dados do *dataset*.

A abordagem utilizada para lidar com o desbalanceamento do *dataset* foi o *oversampling*. Essa escolha se deu pelo fato de o volume de dados já ser reduzido, e a aplicação do *undersampling* reduziria ainda mais esse volume. Com o objetivo de evitar ou mitigar possíveis problemas de *overfitting*, as frases duplicadas passaram por modificações que não alteraram seu contexto, como substituições de palavras por sinônimos. Após esse pré-processamento dos dados, um *dataset* unificado foi criado (Figura 8).

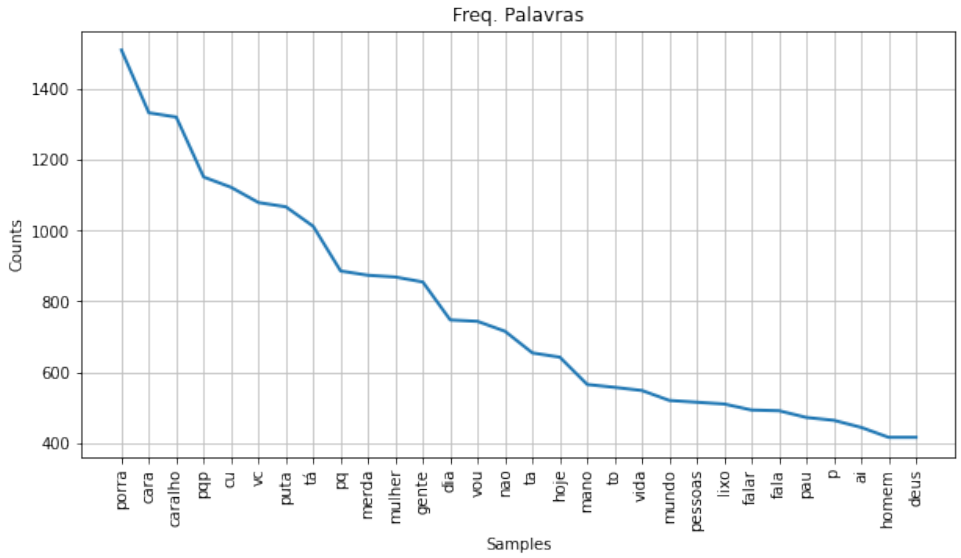
Figura 8 – *Dataset* após *oversampling*




### 4.3.3 Análise de frequência de palavras:

A seguir faremos uma análise de frequência de palavras para termos uma noção melhor sobre as sentenças do nosso *dataset*. Essa análise nos dará uma forma de explorar e visualizar como está a distribuição das palavras nos dados. A análise será feita sobre o *dataset* gerado a partir do *dataset* unificado. O gráfico da Figura 9 apresenta a frequência das palavras mais comuns.

Figura 9 – Gráfico de frequência de palavas




A partir da Figura 9 podemos observar uma grande presença de palavrões e expressões que parecem inicialmente tóxicas. Este é um dos grandes desafios a serem superados neste trabalho pois, nem todas as frases que possuem palavrão são consideradas tóxicas mas, normalmente, frases tóxicas contém palavrões. A partir deste raciocínio fica evidente a importância de um modelo que consiga levar em consideração o contexto completo da sentença no momento da sua classificação.

## 4.4 Prova de conceito

Como citado anteriormente, para termos uma noção sobre a viabilidade do trabalho, criaremos uma prova de conceito. O modelo escolhido para testar a viabilidade deste trabalho, como já citado, é o Naive Bayes. O modelo foi instanciado e treinado usando a biblioteca TensorFlow. Seguindo o plano metodológico, as métricas que nos possibilitam analisar os resultados do modelo são: Precisão, acurácia, *recall*, *F1-Score*. A Figura 10 apresenta a matriz de confusão resultante das previsões do modelo e a Figura 11 apresentam as métricas que foram ser calculadas a partir da matriz de confusão.

Figura 10 – Matriz de confusão - Naive Bayes

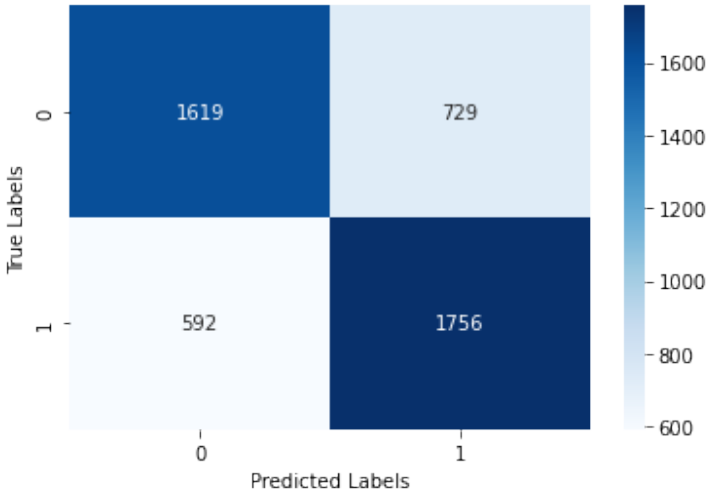



Figura 11 – Resultados de classificação - Naive Bayes

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.73 | 0.69 | 0.71 | 2348 |
| 1 | 0.71 | 0.75 | 0.73 | 2348 |
| accuracy | | | 0.72 | 4696 |
| macro avg | 0.72 | 0.72 | 0.72 | 4696 |
| weighted avg | 0.72 | 0.72 | 0.72 | 4696 |

Em uma primeira análise das métricas, podemos ver que o modelo possui uma acurácia de 72%, precisão de 73% ao prever frases não tóxicas e 72% ao prever frases tóxicas, o que não é exatamente ruim. Analisando também a métrica de *recall* e a parte superior direita da matriz de confusão, podemos observar que o modelo teve um pouco de dificuldade em classificar corretamente frases que não possuem conteúdo tóxico. Isso provavelmente se deve a falta de entendimento de contexto, visto que o modelo utilizado não possui esta capacidade. Deste momento em diante, utilizaremos as métricas apresentadas (Figura 11) como base de comparação para os modelos subsequentes.

## 4.5 LSTM

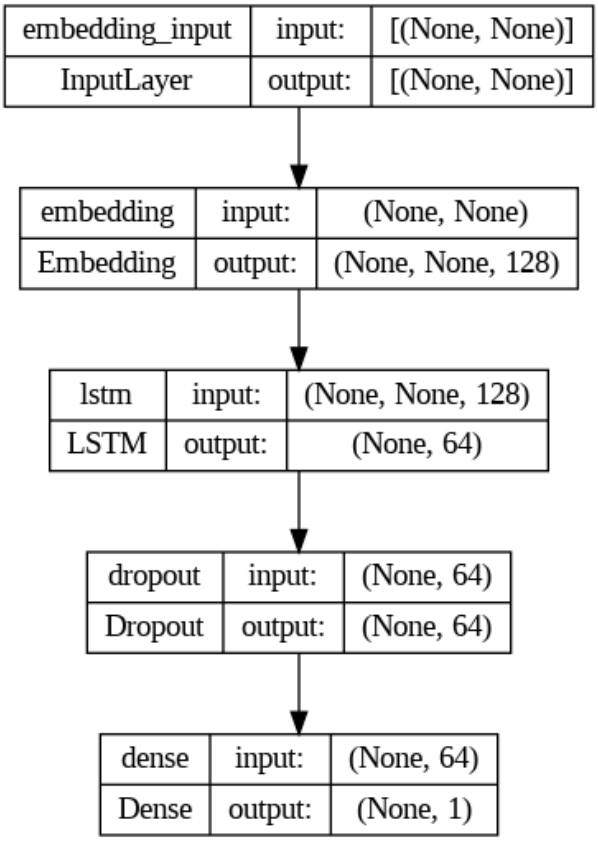
### 4.5.1 Treinamento

Inicialmente, como citado anteriormente, foi realizada a preparação dos dados, incluindo a etapa de pré-processamento textual, que envolveu a remoção de pontuações, e a remoção de stopwords. Posteriormente, o corpus de comentários foi dividido em conjuntos de treinamento e teste, garantindo a robustez da avaliação dos resultados.

Durante o treinamento do modelo LSTM, foram considerados vários parâmetros importantes para otimização, tais como taxa de aprendizagem, dimensão dos vetores de palavras, tamanho da camada LSTM e número de épocas. Para encontrar a configuração ótima que maximizasse a acurácia e minimizasse as taxas de falso positivo e falso negativo na classificação de comentários tóxicos, foi utilizada a técnica de busca em grade (*Grid Search*).

Após uma fase de experimentação abrangente, foi possível identificar a estrutura que demonstrou os resultados mais promissores, conforme ilustrado na Figura 12.

Figura 12 – Camadas LSTM




Além disso, por meio da aplicação da técnica de *Grid Search*, foi possível descobrir hiperparâmetros que desempenham um papel crucial na otimização dos resultados alcançados pelo modelo, conforme apresentado na Tabela 5.

### 4.5.2 Resultados e métricas

O modelo LSTM demonstrou uma capacidade significativa de aprender as representações dos comentários e realizar previsões consideravelmente precisas em relação à to-

| **Épocas** | 10 |
|---|---|
| ***Batch size*** | 32 |
| **Otimizador** | Adam |
| **Taxa de aprendizado** | 0.0001 |
| **Regularizadores** | L2 e *Dropout* |

Tabela 5 – Hiperparâmetros LSTM

xicidade dos mesmos. Durante a fase de treinamento, o modelo foi exposto a um conjunto de comentários e, com base nesses dados, aprendeu a identificar padrões determinados e características que indicam a presença de conteúdo tóxico.

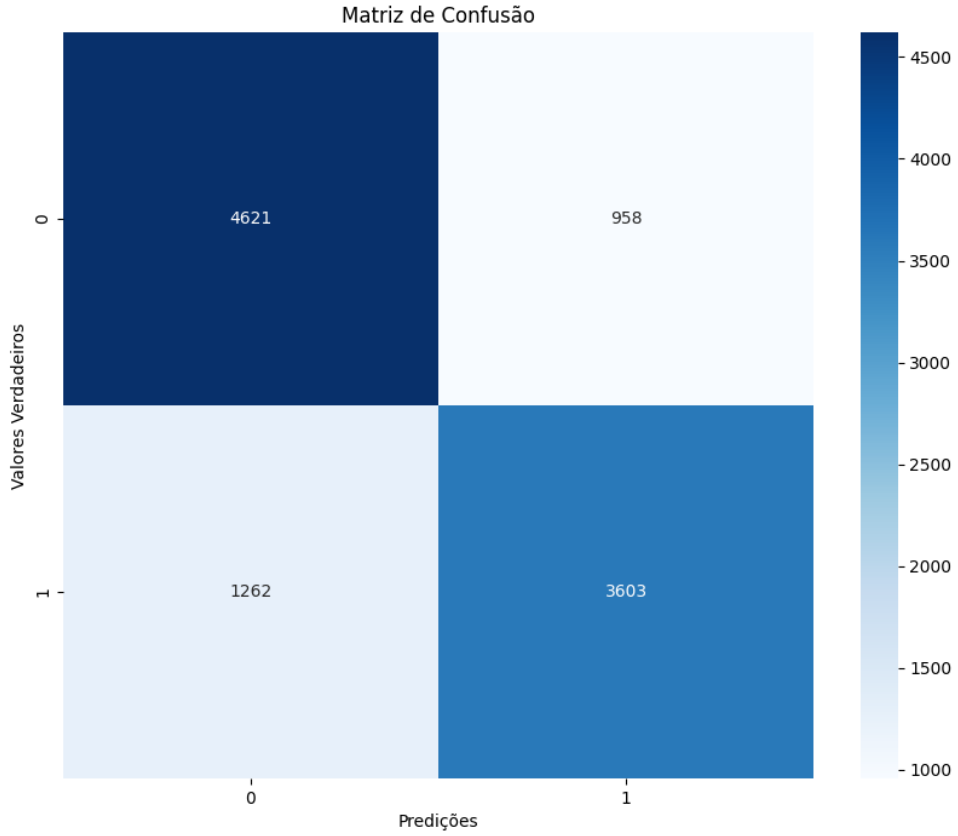
Os resultados obtidos e apresentados na figura 13 com o modelo LSTM revelaram uma boa taxa de acurácia na classificação de comentários tóxicos, fornecendo uma medida geral da capacidade do modelo em identificar adequadamente conteúdo ofensivo e prejudicial.

Figura 13 – Dados de classificação

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.83 | 0.81 | 5579 |
| 1 | 0.79 | 0.74 | 0.76 | 4865 |
| accuracy | | | 0.79 | 10444 |
| macro avg | 0.79 | 0.78 | 0.79 | 10444 |
| weighted avg | 0.79 | 0.79 | 0.79 | 10444 |

Além disso, analisaremos a matriz de confusão, disponível na figura 14, para avaliar o desempenho do modelo em termos de verdadeiros positivos e falsos positivos na detecção de comentários tóxicos.

Figura 14 – Matriz de confusão LSTM




Conforme observado no relatório de classificação para o modelo LSTM, ilustrado na Figura 13, foram avaliadas métricas-chave de desempenho. Para a classe 0, o modelo apresentou uma precisão de 0.79, *recall* de 0.83 e *F1-score* de 0.81. Para a classe 1, as métricas correspondentes foram de precisão 0.79, *recall* 0.74 e *F1-score* 0.76. Esses resultados indicam a habilidade do modelo em classificar corretamente comentários não tóxicos (classe 0) e tóxicos (classe 1). Também é possivel termos uma noção dessas métricas observando os quadrantes da matriz de confusão na Figura 14.

Essas métricas são indicadores fundamentais para avaliar a habilidade do modelo em classificar corretamente comentários não tóxicos (classe 0) e comentários tóxicos (classe 1). Conforme já explicado, a precisão reflete a proporção de verdadeiros positivos em relação a todos os exemplos classificados positivamente pelo modelo. O *recall* mede a proporção de verdadeiros positivos em relação a todos os exemplos pertencentes à classe

positiva. O *F1-score* é uma medida harmônica que combina a precisão e o *recall*, fornecendo uma medida agregada de desempenho do modelo na classificação de ambas as classes.

Os resultados obtidos evidenciam um equilíbrio entre a precisão e o *recall* para ambas as classes, destacando a capacidade do modelo LSTM em realizar previsões corretas tanto para comentários não tóxicos quanto para comentários tóxicos. Observa-se uma taxa de *recall* de 0.83 para a classe 0 (comentários não tóxicos) e de 0.74 para a classe 1 (comentários tóxicos). Esses resultados indicam que o modelo possui uma boa capacidade de identificar corretamente os comentários não tóxicos e, embora tenha um desempenho um pouco inferior na detecção dos comentários tóxicos, ainda é capaz de capturar uma parcela significativa deles. Tal capacidade é essencial para assegurar a efetividade na detecção de conteúdo prejudicial em ambientes digitais.

## 4.6 Bi-LSTM

### 4.6.1 Treinamento

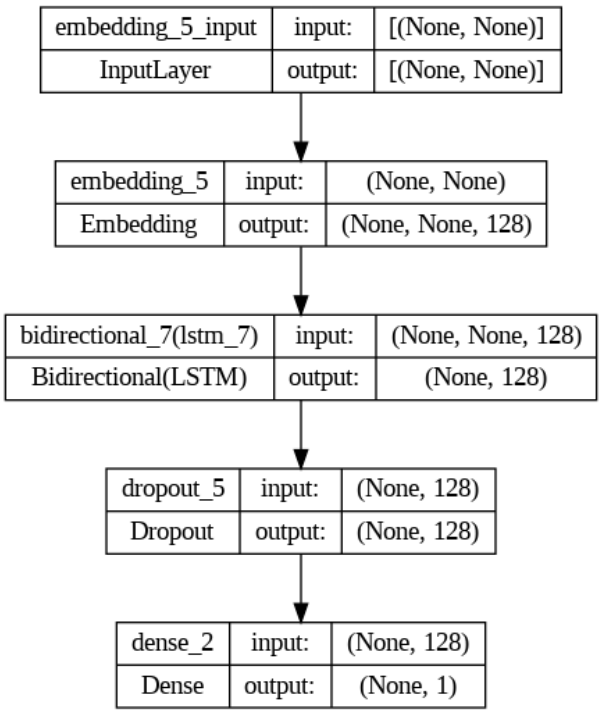
Para tentar explorar ainda mais o desempenho na detecção de comentários tóxicos, foi empregada uma abordagem baseada em Bi-LSTM (*Bidirectional Long Short-Term Memory*). Assim como no treinamento do modelo LSTM descrito anteriormente, os dados foram preparados por meio de etapas de pré-processamento textual, incluindo a remoção de pontuações.

O mesmo conjunto de comentários foi novamente dividido em conjuntos de treinamento e teste, a fim de garantir uma avaliação robusta dos resultados obtidos. Em seguida, também foram considerados diversos parâmetros durante o treinamento do modelo Bi-LSTM , como a taxa de aprendizagem, a dimensão dos vetores de palavras, o tamanho das camadas LSTM e o número de épocas.

De maneira similar à abordagem anterior com o modelo LSTM, a otimização dos hiperparâmetros do modelo Bi-LSTM também foi realizada por meio da técnica de busca em *grid* (*Grid Search*).

Assim como nos resultados apresentados anteriormente, uma extensa fase de experimentação também foi conduzida para identificar a estrutura que proporcionou os resultados mais promissores, visível na Figura 15.

Figura 15 – Camadas Bi-LSTM




A Tabela 6 apresenta os hiperparâmetros utilizados no modelo Bi-LSTM para a detecção de comentários tóxicos. Esses hiperparâmetros foram cuidadosamente ajustados visando aprimorar o desempenho do modelo.

| **Épocas** | 10 |
|---|---|
| ***Batch size*** | 16 |
| **Otimizador** | Adam |
| **Taxa de aprendizado** | 0.0001 |
| **Regularizadores** | L2 e *Dropout* |

Tabela 6 – Hiperparâmetros Bi-LSTM

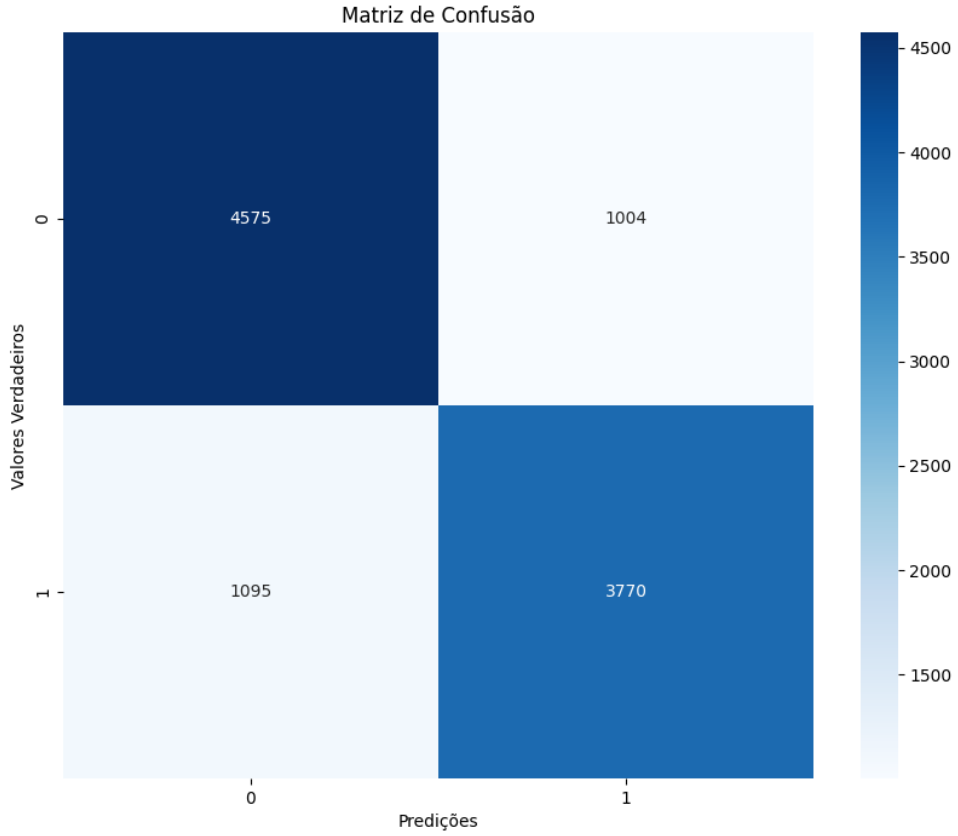
### 4.6.2 Resultados e métricas

Assim como o modelo LSTM, o modelo Bi-LSTM demonstrou uma notável capacidade de aprendizado e previsão em relação à toxicidade dos comentários. Ao ser treinado com um conjunto diversificado de comentários, o modelo Bi-LSTM adquiriu a habilidade de identificar padrões e características que indicam a presença de conteúdo tóxico.

Os resultados obtidos com o modelo Bi-LSTM revelaram uma taxa de acurácia satisfatória na classificação de comentários tóxicos, reforçando a capacidade do modelo em identificar de forma adequada conteúdo ofensivo e prejudicial. A matriz de confusão

(Figura 16) também pode ser analisada para avaliar o desempenho do modelo em termos de verdadeiros positivos e falsos positivos na detecção de comentários tóxicos.

Figura 16 – Matriz de confusão Bi-LSTM




Conforme observado na Figura 17, o modelo Bi-LSTM foi avaliado por meio de métricas-chave de avaliação. Para a classe 0, o modelo apresentou uma precisão de 0.81, *recall* de 0.82 e *F1-score* de 0.81. Para a classe 1, as métricas correspondentes foram de precisão 0.79, *recall* 0.77 e *F1-score* 0.78.

Figura 17 – Dados de classificação

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.82 | 0.81 | 5579 |
| 1 | 0.79 | 0.77 | 0.78 | 4865 |
| accuracy | | | 0.80 | 10444 |
| macro avg | 0.80 | 0.80 | 0.80 | 10444 |
| weighted avg | 0.80 | 0.80 | 0.80 | 10444 |

Os resultados obtidos também evidenciam um equilíbrio entre a precisão e o *recall* para ambas as classes, demonstrando a capacidade do modelo Bi-LSTM em realizar previsões corretas tanto para comentários não tóxicos quanto para comentários tóxicos. Observa-se uma taxa de *recall* de 0.82 para a classe 0 (comentários não tóxicos) e de 0.77 para a classe 1 (comentários tóxicos). Esses resultados indicam que o modelo, semelhantemente ao LSTM, possui uma boa capacidade de identificar corretamente os comentários não tóxicos e, embora tenha um desempenho um pouco inferior na detecção dos comentários tóxicos, foi capaz de identificá-los com um pouco mais de facilidade se comparado ao modelo LSTM.

## 4.7 BERT

No intuito de explorar uma abordagem mais avançada (e definir um teto para comparação), decidimos implementar um modelo baseado no BERT (*Bidirectional Encoder Representations from Transformers*). Especificamente, utilizamos o modelo BERTimbau Base (aka bert-base-portuguese-cased) e realizamos o processo de *fine-tuning* para adaptá-lo à tarefa de detecção de comentários tóxicos em língua portuguesa brasileira.

O modelo BERT foi treinado utilizando o mesmo conjunto de dados utilizado nos modelos Bi-LSTM e LSTM. Para facilitar o processo de treinamento e avaliação, foi utilizada a biblioteca *Simple Transformers*, que oferece uma interface simples e intuitiva para trabalhar com modelos baseados em *transformers*.

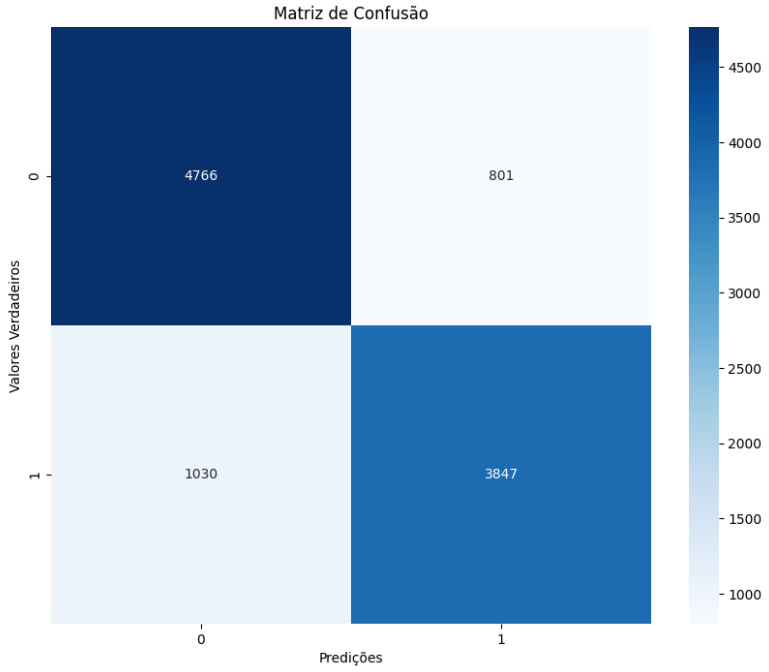
A utilização do mesmo conjunto de dados e da biblioteca *Simple Transformers* proporciona uma base sólida para a comparação entre os modelos, permitindo uma análise justa e objetiva do desempenho do BERT em relação aos modelos anteriores.

Após o treinamento e a avaliação do modelo BERT, foram obtidos os seguintes resultados, que podem ser visualizados nas Figuras 18 e 19. Essas métricas fornecem uma visão abrangente do desempenho do modelo na classificação dos comentários tóxicos, e a matriz de confusão oferece uma compreensão mais detalhada das classificações realizadas pelo modelo.

Figura 18 – Resumo de classificação BERT

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.86 | 0.84 | 5567 |
| 1 | 0.83 | 0.79 | 0.81 | 4877 |
| accuracy | | | 0.82 | 10444 |
| macro avg | 0.82 | 0.82 | 0.82 | 10444 |
| weighted avg | 0.82 | 0.82 | 0.82 | 10444 |

Figura 19 – Matriz de confusão BERT




Ao comparar os resultados obtidos com o modelo BERT em relação aos modelos LSTM e Bi-LSTM , observamos uma melhora modesta no desempenho. O modelo BERT apresentou um desempenho ligeiramente superior, conforme demonstrado pelas métricas de avaliação, como *recall*, *F1-score* e outras métricas relevantes. Vemos também que o BERT conseguiu melhorar ainda mais a capacidade de detecção de comentários tóxicos.

Essa melhoria pode ser atribuída à capacidade do modelo BERT de capturar relações contextuais complexas e utilizar informações de contexto de maneira eficiente. Através do processo de *fine-tuning*, o modelo BERT foi capaz de aprender a representação das palavras em contexto e, assim, obter uma compreensão mais refinada dos comentários tóxicos.

## 4.8 Análise

A partir dos resultados apresentados anteriormente (compilados na Tabela 7), podemos realizar análises mais detalhadas das próprias métricas e da capacidade dos modelos em compreender o contexto das frases.

| **Modelo X Métrica** | **Naive Bayes** | **LSTM** | **Bi-LSTM** | **BERT** |
|---|---|---|---|---|
| ***Precision* (não tóxicos)** | 0.73 | 0.79 | 0.81 | 0.82 |
| ***Precision* (tóxicos)** | 0.71 | 0.79 | 0.77 | 0.83 |
| ***Recall* (não tóxicos)** | 0.69 | 0.83 | 0.82 | 0.86 |
| ***Recall* (tóxicos)** | 0.75 | 0.74 | 0.77 | 0.79 |
| ***F1-score (não tóxicos)*** | 0.71 | 0.81 | 0.81 | 0.84 |
| ***F1-score (tóxicos)*** | 0.73 | 0.76 | 0.78 | 0.81 |
| ***Accouracy*** | 0.72 | 0.79 | 0.80 | 0.82 |

Tabela 7 – Métricas dos modelos

### 4.8.1 Contexto de frase

Ao analisar os modelos apresentados e suas métricas (Tabela 7) , é possível identificar aspectos importantes ao compará-los. Os modelos LSTM e Bi-LSTM demonstraram uma compreensão contextual mais avançada em relação ao Naive Bayes.

A capacidade de capturar o contexto é crucial para uma detecção mais precisa da toxicidade em textos, pois permite considerar o significado geral da frase, em vez de se basear exclusivamente em palavras isoladas. Os modelos LSTM e Bi-LSTM levam em conta a sequência das palavras e as relações entre elas, o que lhes confere uma vantagem em situações em que o contexto é relevante.

Por outro lado, o Naive Bayes, embora seja um modelo simples e rápido de treinar, tende a adotar uma abordagem mais "ingênua" para a classificação de textos, considerando cada palavra de forma independente. Isso pode resultar em dificuldade na captura de nuances e contextos sutis presentes nas frases.

Nas Figuras 20 e 21, é possível notar como os modelos foram capazes de distinguir o contexto e atribuir diferentes interpretações a palavras idênticas. Essa habilidade de compreender o sentido implícito e capturar nuances semânticas é crucial para uma detecção precisa de comentários tóxicos.

Figura 20 – Exemplo sem toxidade

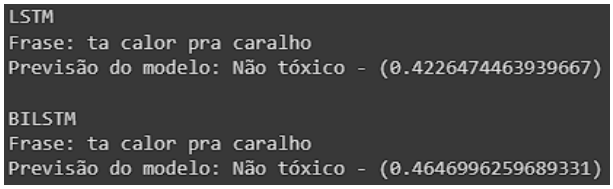
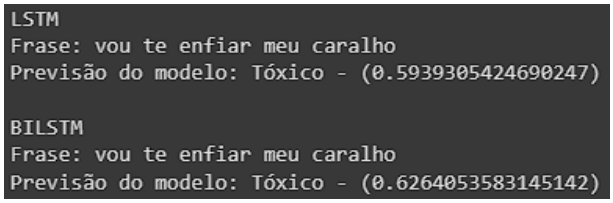



Figura 21 – Exemplo com toxidade




### 4.8.2 Métricas

Ao avaliar o desempenho dos modelos, constatamos que a LSTM apresentou um tempo de treinamento comparativamente inferior em relação aos demais modelos, tornando-a uma opção mais acessível em termos de recursos computacionais e tempo de processamento.

Quanto às métricas de avaliação (Tabela 7), tanto a LSTM quanto a Bi-LSTM superaram (em termos de métricas) o modelo de prova de conceito proposto, o Naive Bayes. Esses modelos demonstraram consistentemente sua capacidade de identificar comentários com diferentes graus de toxicidade, o que evidencia sua eficácia na tarefa de detecção de conteúdo tóxico. As métricas de precisão, *recall* e *F1-score* foram utilizadas para medir o desempenho dos modelos, e ambas as arquiteturas LSTM e Bi-LSTM apresentaram resultados superiores ao Naive Bayes nesses aspectos. Vale ressaltar que, como o objetivo é a identificação de comentários tóxicos, a métrica de *recall* deve obter uma atenção um pouco maior que as demais.

Ao analisarmos os resultados obtidos pelos modelos LSTM e Bi-LSTM observamos que não houve uma vantagem significativa do modelo Bi-LSTM em relação ao modelo LSTM na detecção de comentários tóxicos (comparando o *recall* para comentários tóxicos). Essa observação contraria a expectativa inicial de que o modelo Bi-LSTM , com sua capacidade de processar informações em ambas as direções da sequência textual, teria uma performance significativamente superior.

Uma possível (e a mais provável) explicação para a ausência de uma vantagem significativa do modelo Bi-LSTM em relação ao modelo LSTM na detecção de comentários tóxicos é o tamanho, balanceamento e a qualidade (diversidade e a representatividade dos comentários tóxicos) do *dataset* utilizado. Apesar de todos os esforços focados em melhorar os dados, eles podem não ter sido suficientemente grandes para explorar plenamente a capacidade total dos modelos.

### 4.8.3 Limitações

É necessário ressaltar que o desempenho do modelo apresenta limitações que devem ser consideradas. Uma das limitações está relacionada à natureza do *dataset* utilizado no treinamento, o qual é predominantemente composto por exemplos de insultos e conteúdo tóxico mais comumente encontrados. Como resultado, o modelo pode estar mais familiarizado e ter maior facilidade na identificação dessas categorias específicas, enquanto pode apresentar dificuldades na detecção de grupos menos representados, como discursos racistas, misóginos ou xenofóbicos. Portanto, é essencial considerar a necessidade de utilizar *datasets* mais diversificados e equilibrados, que englobem uma variedade ampla de contextos e categorias de conteúdo tóxico, para melhorar a capacidade de generalização do modelo. É importante destacar que durante a busca por *datasets* relevantes, foi constatada uma escassez de conjuntos de dados abrangentes e diversificados para a detecção de conteúdo tóxico.

No contexto da identificação de frases tóxicas, o modelo também pode apresentar limitações na detecção de expressões mascaradas ou palavras não convencionais. Essa deficiência decorre da complexidade em capturar nuances e contextos específicos que podem estar presentes em expressões utilizadas pelos usuários. Essas expressões podem ser mais sutis e requerer um entendimento mais profundo do contexto para serem identificadas corretamente. Portanto, é necessário aprimorar ainda mais a capacidade do modelo em lidar com essas formas mais complexas de conteúdo tóxico.

Esses limites apresentados pelo modelo destacam a importância contínua de pesquisas adicionais e o desenvolvimento de abordagens mais sofisticadas para a detecção de conteúdo tóxico. É fundamental explorar novas fontes de dados e aprimorar as técnicas de treinamento dos modelos para garantir uma identificação mais abrangente e precisa de

diferentes formas de conteúdo tóxico, levando em consideração sua diversidade e evolução nos ambientes online.

## 4.9 Considerações finais

Ao longo dessa seção, acompanhamos a aquisição do *dataset*, seu tratamento e sua utilização por meio de um exemplo inicial utilizando um modelo probabilístico. Em seguida, exploramos a criação, treinamento e avaliação de modelos LSTM, Bi-LSTM e BERT para a detecção de comentários tóxicos em língua portuguesa brasileira. Esses modelos foram cuidadosamente desenvolvidos e treinados utilizando o conjunto de dados processado.

Realizamos uma análise minuciosa dos resultados obtidos por cada modelo, considerando métricas como precisão, *recall* e *F1-score*, além de outras medidas de desempenho relevantes. Em seguida, comparamos esses modelos e avaliamos sua eficácia na detecção de comentários tóxicos.

Constatamos que os modelos LSTM, Bi-LSTM e BERT demonstraram capacidade promissora na classificação de comentários tóxicos. No entanto, observamos que o modelo BERT, por possuir uma arquitetura mais avançada, alcançou um desempenho ligeiramente superior em comparação aos modelos LSTM e Bi-LSTM com consideravelmente menos esforço de implementação. Também observamos as limitações que o modelo apresentou.

Essa comparação entre os modelos nos permitiu ter uma visão abrangente de suas características e potenciais. Com base nesses resultados, podemos concluir que os modelos LSTM e Bi-LSTM são, de fato, alternativas eficazes na detecção de comentários tóxicos, uma vez que apresentaram resultados semelhantes.

Essa análise comparativa nos fornece *insights* valiosos sobre a aplicação de diferentes abordagens de modelos para a detecção de conteúdo tóxico, auxiliando na escolha da abordagem mais adequada para futuras aplicações e investigações no campo da análise de sentimentos e detecção de linguagem ofensiva em ambientes digitais.

# 5 Conclusão

O presente trabalho teve como objetivo principal a implementação e utilização de redes neurais LSTM e Bi-LSTM para a identificação de frases consideradas tóxicas. Para alcançar esse objetivo, foram definidos objetivos específicos que abrangeram desde a coleta e tratamento dos dados até a análise das métricas de desempenho dos modelos.

No que diz respeito à coleta e tratamento dos dados, foram adotadas estratégias cuidadosas no intuito de garantir a qualidade e representatividade dos conjuntos de dados utilizados no treinamento dos modelos. A análise dos dados coletados permitiu uma maior compreensão das características das frases tóxicas, identificando possíveis padrões e tendências relevantes para o desenvolvimento dos modelos.

A construção dos modelos LSTM e Bi-LSTM compreendeu a implementação de arquiteturas de redes neurais recorrentes e a configuração minuciosa dos hiperparâmetros correspondentes. Utilizando técnicas de otimização e treinamento iterativo, os modelos foram ajustados para melhorar seu desempenho.

A análise dos resultados revelou que os modelos LSTM e Bi-LSTM demonstraram eficácia na detecção de frases tóxicas apesar das limitações impostas pelo *dataset* utilizado. Ambos os modelos foram capazes de classificar corretamente uma grande proporção das frases em relação à sua toxicidade. Além disso, foi possível observar uma certa vantagem de modelos *transformers* (BERT) sobre as redes alvo deste trabalho.

Devemos salientar que a detecção de conteúdo tóxico apresenta desafios significativos, devido à natureza subjetiva e contextual dessas manifestações. Portanto, ainda há espaço para aprimoramentos e pesquisas adicionais no desenvolvimento de abordagens mais sofisticadas que considerem a diversidade e a evolução dos discursos ofensivos.

Os resultados deste trabalho contribuem para o avanço do conhecimento na área de detecção de conteúdo tóxico em ambientes online. As abordagens adotadas e os *insights* obtidos podem ser aplicados em diferentes contextos, como em plataformas de mídia social e fóruns de discussão, para promover um ambiente online mais seguro e respeitoso.

Por fim, é importante destacar que a detecção de conteúdo tóxico é uma área em constante evolução, exigindo esforços contínuos para acompanhar as mudanças nos pa-

drões de comportamento e nas expressões linguísticas utilizadas pelos usuários. Portanto, este trabalho pode servir como um ponto de partida para futuras pesquisas e desenvolvimentos na área de detecção de conteúdo tóxico em ambientes virtuais.

# Referências

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>. Citado na página 34.

CHARNIAK, E. *Introduction to deep learning*. [S.l.]: Mit Press, 2018. ISBN 9780262039512. Citado 2 vezes nas páginas 21 e 22.

CHOLLET, F. *Deep Learning with Python*. [S.l.]: Manning, Cop, 2018. ISBN 9781617294433. Citado 2 vezes nas páginas 20 e 31.

CHOLLET, F. et al. *Keras*. 2015. <https://keras.io>. Citado na página 34.

CULLEN, A. L. L.; KAIRAM, S. R. Practicing moderation: Community moderation as reflective practice. *Proceedings of the ACM on Human-Computer Interaction*, v. 6, n. CSCW1, p. 1–32, Mar 2022. Citado na página 14.

EPIC. *Epic Games Comunity Rules*. Epic Games, 2022. Disponível em: <https://www.epicgames.com/site/en-US/community-rules>. Citado na página 13.

FORTUNA, P. et al. A hierarchically-labeled portuguese hate speech dataset. In: *Proceedings of the 3rd Workshop on Abusive Language Online (ALW3)*. [S.l.: s.n.], 2019. Citado na página 33.

GANEGEDARA, T.; LOPATENKO, A. *Natural Language Processing with TensorFlow*. [S.l.]: Packt Publishing Ltd, 2022. ISBN 9781838647742. Citado 4 vezes nas páginas 22, 23, 24 e 26.

GOLDBERG, Y. *Neural network methods in natural language processing*. [S.l.]: Morgan Claypool Publishers, 2017. ISBN 9781627052986. Citado 4 vezes nas páginas 22, 23, 25 e 26.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. The Mit Press, 2016. ISBN 9780262035613. Disponível em: <https://www.deeplearningbook.org/>. Citado 2 vezes nas páginas 21 e 22.

GOOGLE. *Google Colaboratory*. 2019. Disponível em: <https://colab.research.google.com/>. Citado na página 33.

GéRON, A. *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*. [S.l.]: O'Reilly Media, Inc., 2019. ISBN 9781492032649. Citado na página 19.

HOSSIN, M.; M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining  Knowledge Management Process*, v. 5, p. 01–11, 03 2015. Citado na página 26.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 34.

INDURKHYA, N.; DAMERAU, F. J. *Handbook of natural language processing*. [S.l.]: Taylor Francis, 2010. ISBN 9781420085938. Citado 3 vezes nas páginas 24, 25 e 37.

JIANG, H. *Machine learning fundamentals : a concise introduction*. [S.l.]: Cambridge University Press, 2021. ISBN 9781108837040. Citado 3 vezes nas páginas 17, 18 e 20.

LEITE, J. A. et al. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *CoRR*, abs/2010.04543, 2020. Disponível em: <https://arxiv.org/abs/2010.04543>. Citado 2 vezes nas páginas 33 e 36.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 35.

PELLE, R. P. de; MOREIRA, V. P. Offensive comments in the brazilian web: a dataset and baseline results. 2017. Citado na página 33.

RAJAPAKSE, T. *Simple Transformers*. 2021. <https://github.com/ThilinaRajapakse/simpletransformers>. Citado na página 35.

RIOT. *An Update on Player Dynamics*. 2022. Disponível em: <https://www.riotgames.com/en/news/an-update-on-player-dynamics>. Citado na página 15.

SINGH, S. *Everything in Moderation*. [s.n.], 2019. Disponível em: <https://www.newamerica.org/oti/reports/everything-moderation-analysis-how-internet-platforms-are-using-artificial-intelligence-moderate-us>. Citado 2 vezes nas páginas 13 e 14.

TEAM, T. pandas development. *pandas-dev/pandas: Pandas*. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.3509134>. Citado na página 34.

TRAJANO, D.; BORDINI, R.; VIEIRA, R. Olid-br: Offensive language identification dataset for brazilian portuguese. *OLID-BR: Offensive Language Identication Dataset for Brazilian Portuguese*, Nov 2022. Citado na página 33.

# Apêndices