ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

DOUGLAS DE OLIVEIRA TRAJANO

DETECÇÃO DE LINGUAGEM TÓXICA APLICADA A TEXTOS EM PORTUGUÊS

Porto Alegre
2023

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DETECÇÃO DE LINGUAGEM TÓXICA APLICADA A TEXTOS EM PORTUGUÊS

DOUGLAS DE OLIVEIRA TRAJANO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rafael Heitor Bordini

Porto Alegre
2023

Ficha Catalográfica

T768d Trajano, Douglas de Oliveira

Detecção de linguagem tóxica aplicada a textos em Português / Douglas de Oliveira Trajano. – 2023.

94 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Heitor Bordini.

1. Processamento de linguagem natural. 2. Detecção de linguagem tóxica. 3. Classificação de texto. 4. Reconhecimento de entidades. 5. Aprendizado profundo. I. Bordini, Rafael Heitor. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

DOUGLAS DE OLIVEIRA TRAJANO

DETECÇÃO DE LINGUAGEM TÓXICA APLICADA A TEXTOS EM PORTUGUÊS

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 27 de fevereiro de 2023.

BANCA EXAMINADORA:

Profª. Drª. Soraia Raupp Musse (PPGCC/PUCRS)

Profª. Drª. Viviane Pereira Moreira (PPGC/UFRGS)

Prof. Dr. Rafael Heitor Bordini (PPGCC/PUCRS - Orientador)

DEDICATÓRIA

Dedico este trabalho à minha mãe, Rosa Amélia Rosa de Oliveira, que mesmo sem ter a oportunidade de realizar o ensino superior, não mediu esforços para me apoiar em meus estudos e objetivos. Sua determinação e coragem diante das adversidades da vida são admiráveis e me ensinaram a nunca desistir dos meus sonhos. Dedico também à minha namorada, Bruna da Silva Crespo, que foi compreensiva nos momentos em que me ausentei e que me apoiou com suas palavras e carinho durante essa jornada. Amo vocês, e espero corresponder à tudo que fizeram por mim.

“All we have to decide is what to do with the time that is given us.”
(Gandalf – J.R.R. Tolkien)

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Rafael Heitor Bordini, e à Profª. Drª. Renata Vieira, que apoiaram a contribuíram para o sucesso deste projeto. Vocês me desafiaram e confiaram em mim quando os objetivos eram audaciosos, mas também me acolheram e guiaram nos momentos de incertezas e dificuldades. Vocês foram essenciais para o meu crescimento. Muito obrigado!

Ao Daniel de Los Reyes e André Giordani pela parceria durante o mestrado.

Ao Daniel Güths e Cristofer Weber pelos conselhos e direcionamos que me fizeram chegar até aqui.

À PUCRS e à Uol EdTech, por financiarem e assim permitirem a realização desta pesquisa e pela oportunidade de me tornar mestre.

DETECÇÃO DE LINGUAGEM TÓXICA APLICADA A TEXTOS EM PORTUGUÊS

RESUMO

As redes sociais têm revolucionado a forma como a sociedade se comunica, graças à sua natureza descentralizada que permite a interação entre os usuários. No entanto, as mensagens que circulam nas redes sociais podem conter expressões de opinião, mensagens de apoio e, mas também discurso de ódio. O discurso de ódio é um problema crescente na esfera digital, geralmente causado pela polarização de opiniões ou pela falsa sensação de impunidade. Os *haters*, usuários que disseminam o discurso de ódio, podem ser encontrados em uma variedade de tópicos, incluindo debates políticos, entretenimento, jogos online e ambientes corporativos. A área de Processamento de Linguagem Natural (PLN) pode contribuir com ferramentas para assegurar uma comunicação saudável e garantir os direitos dos usuários no mundo digital, agindo de forma rápida, padronizada e automatizada, evitando a necessidade de moderação manual deste tipo de conteúdo.

Neste estudo, utilizamos técnicas avançadas de aprendizado de máquina e aprendizado profundo para desenvolver um sistema de detecção de linguagem tóxica em mensagens em Português. O conjunto de dados utilizado para o treinamento dos modelos é composto por 6.354 (com possibilidade de extensão para 13.538) comentários anotados manualmente por especialistas. Este conjunto de dados, disponibilizado como parte do trabalho, possui anotações para 5 tarefas de PLN, utilizando um esquema de anotação hierárquico com diferentes níveis de granularidade. Os resultados dos experimentos demonstram a utilidade desse conjunto de dados para o desenvolvimento de sistemas de PLN voltados para a detecção de linguagem tóxica em textos em Português.

Palavras-Chave: Processamento de Linguagem Natural, Extração de Informações, Classificação de Texto, Reconhecimento de Entidades, Detecção de Discurso de Ódio, Linguagem Tóxica, Comentário Ofensivo, Segurança Online, Comentário Tóxico, Toxicidade, Racismo, Homofobia, Xenofobia.

TOXIC LANGUAGE DETECTION APPLIED TO PORTUGUESE TEXTS

ABSTRACT

The advent of social media has transformed the way in which individuals and communities interact and communicate. However, the messages on social media may contain expressions of opinion, and support messages, but they can also hate speech. The proliferation of hate speech in the digital sphere has become an increasingly pressing issue, with polarized opinions and a sense of anonymity and impunity among users often serving as contributing factors. The *haters*, users who spread hate speech, can be found in a variety of topics, including political discussions, entertainment, gaming, and corporate environments. The Natural Language Processing (NLP) area can contribute with tools to ensure healthy communication and protect users' rights online. NLP applications are efficient, standardized, and automated, eliminating the need for manual moderation of such content.

In this study, we used advanced machine learning and deep learning techniques to develop a toxic language detection system in Portuguese messages. The dataset used for training the models consists of 6,354 (with the possibility of extending to 13,538) comments manually annotated by experts. This dataset, made available as part of the work, has annotations for 5 NLP tasks, using a hierarchical annotation scheme with different levels of granularity. The results of the experiments demonstrate the usefulness of this dataset for the development of NLP systems aimed at detecting toxic language in texts in Portuguese.

Keywords: Natural Language Processing, Information Extraction, Text Classification, Named-Entity Recognition, Hate Speech Detection, Toxic Language, Offensive Comment, Toxic Comment, Toxicity, Racism, Homophobia, Xenophobia.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

AM – Aprendizado de Máquina

API – *Application Programming Interface*

AP – Aprendizado Profundo

BERT – *Bidirectional Encoder Representations from Transformers*

BILSTM – *Bidirectional Long Short-Term-Memory*

CBT – Classificação Binária de Texto

CMCT – Classificação Multiclasse de Texto

CMRT – Classificação Multirrótulo de Texto

CNN – *Convolutional Neural Network*

CT – Classificação de Texto

EN-US – Inglês dos Estados Unidos

FECAP – Fundação Escola de Comércio Álvares Penteado

FFN – *Feed-Forward Network*

FN – Falso Negativo

FP – Falso Positivo

GPT – *Generative Pre-trained Transformer*

HLPHSD – *Hierarchically-Labeled Portuguese Hate Speech Dataset*

IA – Inteligência Artificial

LSTM – *Long Short-Term Memory*

NCCVG – *Hate Speech Detection Dataset using Brazilian Imageboards*

NER – *Named Entity Tags*

OLID-BR – *Offensive Language Identification Dataset for Brazilian Portuguese*

OLID – *Offensive Language Identification Dataset*

PLN – Processamento de Linguagem Natural

PT-BR – Português do Brasil

RE – Reconhecimento de Entidades

RNA – Redes Neurais Artificiais

RNR – Redes Neurais Recorrentes

SVM – *Support Vector Machines*

TM – Tradução de Máquina

TOLD-BR – *Toxic Language Dataset for Brazilian Portuguese*

VN – Verdadeiro Negativo

VP – Verdadeiro Positivo

SUMÁRIO

1. INTRODUÇÃO

As redes sociais transformaram o estilo de vida de nossa sociedade em vários aspectos. Um dos grandes benefícios das redes sociais e blogs é proporcionar uma forma fácil dos usuários se relacionarem. Elas também possibilitam que pessoas fisicamente distantes e/ou de grupos sociais diferentes se relacionem no mundo digital. As redes sociais permitem que os usuários compartilhem ideias ou interajam com outros usuários sem barreiras que existem no mundo físico (Siddiqui et al., 2016). Todos os dias, milhares de mensagens de texto são publicadas em canais de comunicação online. Esses comentários podem expressar ideias, opiniões, mensagens de apoio, mas também podem conter discurso de ódio. O discurso de ódio é definido como qualquer comunicação que deprecie uma pessoa ou um grupo com base em suas características como cor, etnia, gênero, orientação sexual, nacionalidade, raça, religião, etc. (Levy et al., 2000). Devido à grande quantidade de dados gerados todos os dias nas plataformas de redes sociais, a moderação manual é inviável, levando a um aumento na demanda por ferramentas automatizadas que realizem a moderação de conteúdo impróprio (Alonso et al., 2020).

O discurso de ódio é um crime que tem crescido nos últimos anos, principalmente devido às interações online entre as pessoas. Diversos fatores contribuem para isso, como a sensação de anonimato que estes ambientes podem possuir, mas também pela maior disposição dos usuários em expressarem suas opiniões, o que inflama discursos de ódio e favorece a polarização dos usuários (Fortuna e Nunes, 2018). Em Whillock e Slayden (1995), os autores destacam o uso de discurso de ódio para alcançar objetivos políticos e sociais, o que vemos como combustível para a polarização política presente nos dias de hoje. Além disto, a exposição à linguagem tóxica pode afetar a saúde mental dos usuários e o acesso democrático à internet (Alonso et al., 2020). O tema ganhou bastante popularidade nos últimos anos devido a grande cobertura da imprensa sobre o problema e aos efeitos políticos e sociais que causa (Fortuna e Nunes, 2018). Em Fortuna e Nunes (2018), são citados alguns motivos para focar no desenvolvimento de soluções que detectem linguagem tóxica, como por exemplo, iniciativas de governos para reduzirem o discurso de ódio, a falta de soluções automatizadas que consigam atuar de forma rápida, a falta de dados sobre discurso de ódio, a necessidade de empresas e plataformas removerem este tipo de conteúdo de seus ambientes. Segundo levantamento realizado pelo Instituto Ipsos em 2018[1], o Brasil ficou em segundo lugar no ranking em que pais ou responsáveis relataram que os filhos já foram vítimas de violência online, ficando na frente de Estados Unidos e África do Sul, por exemplo. Nessa mesma pesquisa, 76% dos entrevistados consideram que as medidas anti-bullying existentes são insuficientes. Muitas dessas ferramentas

[1] https://www.ipsos.com/en/global-views-cyberbullying

atuam de forma manual e após o ato cometido, requisitando um processo burocrático na polícia ou no judiciário.

Além dos motivos citados em Fortuna e Nunes (2018), como maior atenção governamental com políticas na área e investimento de empresas de tecnologia para garantir a segurança e a liberdade de expressão dos usuários em suas plataformas, o autor do trabalho acredita ser de nossa responsabilidade, como cientistas da computação e programadores que desenvolvem o mundo digital, fornecer um ambiente seguro para que diferentes opiniões sejam ouvidas garantindo o respeito aos envolvidos e a liberdade de expressão de todos. As redes sociais são constantemente cobradas para impedir a publicação de mensagens que contenham linguagem tóxica[2]. Muitos avanços já foram feitos nessa área, as empresas procuram atualizar suas políticas conforme as mudanças da sociedade [3][4] e desenvolver sistemas inteligentes que sejam capazes de detectar automaticamente se um determinado comentário é ofensivo ou não.

A área de Processamento de Linguagem Natural (PLN) pode contribuir com ferramentas que permitem detectar os comentários tóxicos e/ou extrair informações relevantes sobre o(s) tipo(s) de linguagem tóxica existente(s) nos textos, aprimorando assim os sistemas de moderação de conteúdo impróprio (Alrehili, 2019). Os *haters*, termo usado para se referir aos usuários que disseminam discurso de ódio, também podem sofrer restrições ou terem suas contas excluídas com base em determinados tipos de linguagem tóxica. Mais recentemente, algoritmos de aprendizado profundo (AP) vêm demonstrando excelentes resultados nas mais variadas áreas da inteligência artificial (IA), incluindo PLN, sendo uma das mais importantes arquiteturas usadas neste contexto o Transformers (Alonso et al., 2020). A literatura abordada no Capítulo 3 mostra que modelos gerados por esta arquitetura demonstram grande capacidade de extrair padrões e generalizar problemas complexos, principalmente pelo seu mecanismo de atenção que permite entender o contexto das palavras em uma determinada frase. A maioria dos trabalhos relacionados que utilizam essa arquitetura focam principalmente na língua inglesa. Os poucos trabalhos que abordam a língua portuguesa utilizam algoritmos de aprendizado de máquina (AM) como Floresta Aleatória (*Random Forest*) combinados com técnicas de transformação de textos em vetores como por exemplo TF-IDF. Ainda que modelos baseados na arquitetura transformers sejam bem-sucedido nas tarefas de Classificação de Texto (CT), por serem algoritmos supervisionados, seu grau de sucesso está diretamente relacionado ao conjunto de dados utilizado no treinamento. Uma análise preliminar realizada pelo autor deste trabalho aponta que os mecanismos existentes para detecção de linguagem tóxica priorizam a língua inglesa e as ferramentas que suportam a língua

[2] https://noticias.uol.com.br/ultimas-noticias/rfi/2021/09/24/tiktok-e-acusado-de-permitir-cyberbullying-e-fake-news-na-franca.htm

[3] https://g1.globo.com/economia/tecnologia/noticia/2020/12/02/twitter-amplia-diretrizes-para-combater-discurso-de-odio-com-base-em-raca-etnia-ou-nacionalidade.ghtml

[4] https://olhardigital.com.br/2021/09/30/videos/chega-de-assedio-twitch-adota-regras-mais-rigidas-para-evitar-cyberbullying/

portuguesa são mais deficitárias. Outro ponto identificado na análise da literatura e suportado pelos trabalhos relacionados aponta que a Classificação Binária de Texto (CBT), em alguns casos, não fornece informação suficiente para a tomada de decisão em um sistema de moderação de conteúdo impróprio, que eventualmente precisam destacar o tipo de linguagem tóxica presente no texto como racismo, sexismo, homofobia, xenofobia, entre outros. Também pode ser necessário destacar a(s) parte(s) do texto que o identificam como um comentário tóxico ou identificar possíveis alvos dos comentários tóxicos. Nesse sentido, o PLN oferece diversas tarefas para atender essas necessidades.

Devido à falta de conjunto de dados em Português de boa qualidade, as empresas acabam adicionando uma nova tarefa de Tradução de Máquina (TM) para converter o texto em português para inglês conforme documentação da Perspective API[5]. Essa técnica eventualmente pode distorcer o significado de algumas palavras, prejudicando a efetividade do sistema. Encontramos alguns conjuntos de dados com comentários tóxicos em português, como o OFFCOMBR[6] e o HLPHSD[7], porém, os conjunto de dados encontrados oferecerem anotações apenas para a Classificação Binária de Texto (CBT) que determina se o comentário é tóxico ou a Classificação Multirrótulo de Texto (CMRT) para alguns tipos de linguagem tóxica. E se, em vez de simplesmente identificarmos se um comentário é tóxico ou não, formos além e explorarmos diferentes dimensões do mesmo comentário tóxico? Por exemplo, podemos identificar os tipos de linguagem tóxica presentes no comentário e examinar para quem o comentário é direcionado e qual parte do texto é considerada tóxica. Essas questões nortearam a pesquisa relatada nesta dissertação, que foi organizada da seguinte maneira: o Capítulo 2 descreve os recursos utilizados e conceitos fundamentais para o desenvolvimento do trabalho; o Capítulo 3 apresenta os trabalhos relacionados que motivaram e suportaram essa pesquisa; o Capítulo 4 aborda a construção do conjunto de dados OLID-BR; o Capítulo 5 detalha o software e os modelos treinados com base no OLID-BR; o Capítulo 6 descreve os experimentos realizados para avaliação da aplicabilidade do conjunto de dados nas tarefas propostas nessa pesquisa; o Capítulo 7 apresenta as conclusões da pesquisa e possibilidades de trabalhos futuros.

As principais contribuições deste trabalho são:

- Conjunto de dados com textos em Português com 5 tarefas de PLN anotadas;
- Classificador multirrótulo de linguagem tóxica;
- Classificador binário de comentários tóxicos direcionados;
- Classificador multiclasse para o tipo de comentário tóxico direcionado;
- Detector da(s) parte(s) do texto que o tornam tóxico.

[5] https://developers.perspectiveapi.com/s/about-the-api-model-cards
[6] https://github.com/rogersdepelle/OffComBR
[7] https://github.com/paulafortuna/Portuguese-Hate-Speech-Dataset

- Disponibilização de artefatos e código fonte para reprodutibilidade e aplicação prática da pesquisa, na forma de Software Livre.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo iremos apresentar os conceitos fundamentais para a compreensão deste trabalho. Na Seção 2.2 será apresentada a definição da tarefa de Classificação de Texto e suas respectivas subtarefas. A Seção 2.3 apresenta a definição da tarefa de Reconhecimento de Entidades Nomeadas. A Seção 2.4 apresenta algoritmos de inteligência artificial e suas características. Na Seção 2.5 será apresentada os processos e conceitos no processamento de dados. A Seção 2.6 apresenta as possíveis métricas para avaliar a abordagem proposta.

2.1 Linguagem tóxica e discurso de ódio

Decidir se um comentário possui discurso de ódio não é uma tarefa trivial, mesmo para humanos. Devido às diferentes experiências e relações pessoais de cada pessoa, as próprias nuances da linguagem, é possível perceber uma baixa concordância entre anotadores no processo de construção dos conjuntos de dados (Fortuna e Nunes, 2018). Por isso, definir adequadamente os termos pode facilitar o processo de anotação proposto em 4.2.

Em Poletto et al. (2021) uma revisão sistemática é feita e os termos vistos na literatura são esclarecidos. Discurso de ódio é considerado uma instância da linguagem abusiva/tóxica. Agressividade é definida como a intenção de ser agressivo, incitar atos violentos contra um determinado alvo. Ofensividade é qualquer forma de linguagem inaceitável (profana), rude, vulgar, que utiliza palavras ofensivas para insultar. As relações entre os conceitos são apresentadas na Figura 2.1.

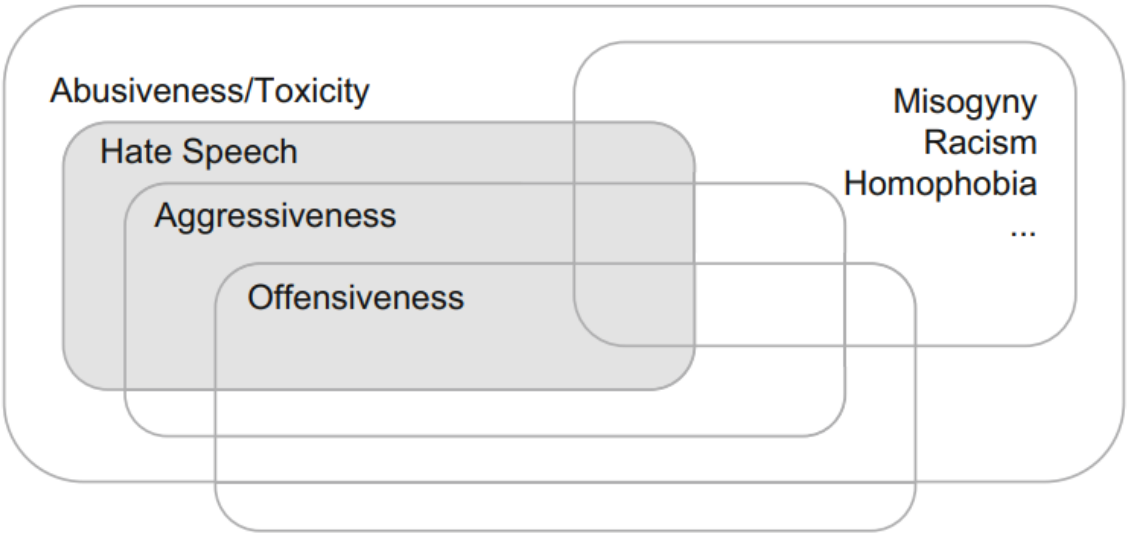

Figura 2.1: Conceitos relacionados com discurso de ódio. Fonte: Poletto et al. (2021)

Discurso de ódio (*Hate Speech*) é a linguagem que ataca ou procura diminuir, que incita violência ou ódio contra grupos, com base em suas características específicas como aparência física, raça, etnia, religião, descendência ou origem, orientação sexual, identidade de gênero, entre outras. Pode ocorrer de forma sutil ou com o uso do humor (Fortuna e Nunes, 2018).

Cyberbulling é a prática de atacar, diminuir, insultar, entre outros, uma pessoa em específico (Zampieri et al., 2019).

Linguagem tóxica ou **ofensiva** é definida por Gaydhani et al. (2018) como um texto que contém insultos, termos depreciativos.

2.2 Classificação de Texto

Na história antiga da IA, as técnicas de classificação de texto foram usadas principalmente para sistemas de recuperação de informação. Entretanto, com os avanços tecnológicos dos últimos anos, a Classificação de Texto tem ajudado na categorização de textos em muitos domínios como medicina, ciências sociais, saúde, direito, engenharia, entre outros. Também tem sido utilizada na análise do comportamento humano em corpus de texto contendo mensagens eletrônicas, publicações em redes sociais, etc. A Classificação de Texto pode ser aplicada em diferentes níveis de escopo: documento, parágrafo, frase, sub-sentença (Kowsari et al., 2019).

A Classificação de Texto pode se dividir em subtarefas de acordo com o problema a ser resolvido. Cada uma das subtarefas pode exigir um formato diferente do conjunto de dados (Er et al., 2016). A Subseção 2.2.1 apresenta a Classificação Binária de Texto. A Subseção 2.2.2 apresenta a Classificação Multiclasse de Texto. A Subseção 2.2.3 apresenta a Classificação Multirrótulo de Texto.

2.2.1 Classificação Binária de Texto

A Classificação Binária de Texto (CBT) tem como objetivo classificar um determinado texto em apenas uma de duas classes (Er et al., 2016). A CBT se assemelha a uma decisão binária em que apenas dois caminhos possíveis são permitidos. Pode ser utilizada para identificar um comentário como tóxico ou não tóxico conforme apresentado na Tabela 2.1.

Classe	**Valor**	**Texto**
Offensive	1	USER Você é um lixo.
Not offensive	0	USER Crime é invadir a casa dos outros.

Tabela 2.1: Exemplo de Classificação Binária de Texto. Fonte: Elaborada pelo autor.

2.2.2 Classificação Multiclasse de Texto

A Classificação Multiclasse de Texto (CMCT) tem como objetivo classificar um determinado texto em uma das classes possíveis em um conjunto de classe que possua mais do que duas classes (Er et al., 2016). A CMCT pode auxiliar em problemas mais complexos em que existem mais do que duas possíveis classificações. Pode ser utilizada para identificar o tipo do alvo em um comentário tóxico direcionado conforme exemplificado na Tabela 2.2.

Classe	**Valor**	**Texto**
Individual	0	USER USER Além de ladrão é mentiroso; afinal, tu nunca leu nenhum livro porque não gosta de ler, não é isso???
Group	1	Esses políticos tudo farinha do mesmo saco !!
Other	2	NÃO COMPREM DESSA LOJA FASCISTA!

Tabela 2.2: Exemplo de Classificação Multiclasse de Texto. Fonte: Elaborada pelo autor.

2.2.3 Classificação Multirrótulo de Texto

A Classificação Multirrótulo de Texto permite associar rótulos a um determinado texto. A quantidade de rótulos não é fixa e pode variar dinamicamente (Er et al., 2016). A CMRT pode ser empregada na detecção dos diferentes tipos de linguagem tóxica presente em um determinado texto. A Tabela 2.3 apresenta alguns exemplos. No campo Rótulos é

apresentado os possíveis rótulos para cada comentário no campo Texto, o campo Valores mostra os rótulos aplicáveis em cada comentário no campo Text.

Rótulos	Valores	Texto
[*Health*, *Insult*, *Obscene*, *Physical aspects*]	[0, 1, 1, 0]	USER USER VTNC SEU GOLPISTA AGIOTA SAFADO. URL
[*Health*, *Insult*, *Obscene*, *Physical aspects*]	[0, 0, 0, 1]	USER Deveriam ter escolhido uma modelo menos gordinha, ela parece que não está passando fome !!
[*Health*, *Insult*, *Obscene*, *Physical aspects*]	[0, 1, 0, 0]	USER Você é um lixo.
[*Health*, *Insult*, *Obscene*, *Physical aspects*]	[1, 1, 0, 0]	USER Velhinho tô ligado que nem vc acredita no que diz.

Tabela 2.3: Exemplo de Classificação Multirrótulo de Texto. Fonte: Elaborada pelo autor.

2.3 Reconhecimento de Entidades Nomeadas

A tarefa de Reconhecimento de Entidades Nomeadas (REN) é definida como a tarefa de identificar e classificar entidades nomeadas em um determinado texto, como por exemplo, pessoas, organizações, locais, datas, etc. (Grishman e Sundheim, 1996). A tarefa de REN pode utilizar regras linguísticas a partir do conhecimento de domínio e gramatical, porém, além de exigir um alto grau de especialização em ambos os campos para alcançar um bom desempenho (Nadeau e Sekine, 2007)., também restringe a aplicação da tarefa em contextos com diferentes formas de expressão devido a sua capacidade de generalização limitada. Outra forma de realizar a tarefa de REN por meio de técnicas de aprendizado de máquina (Nadeau e Sekine, 2007), que são capazes de extrair padrões de um conjunto de dados anotados. As anotações associadas a cada entidade nomeada são denominadas de *Named Entity Tags (NER tags)*.

Shelar et al. (2020) apresentam uma revisão entre diferentes ferramentas de PLN para a tarefa de REN, como por exemplo, SpaCy, Apache OpenNLP e TensorFlow e uma comparação entre elas. Essas ferramentas disponibilizam modelos pré-treinados que incluem as entidades nomeadas mais comuns, como por exemplo, pessoas, organizações, locais, datas, etc., e também permitem a criação de modelos personalizados para a tarefa de REN de acordo com o domínio de interesse. Na comparação entre as ferramentas, os autores destacam que a SpaCy obteve os melhores resultados (avaliados por meio da métrica F1) para a tarefa de REN, além de ser a ferramenta mais robusta e com recursos adicionais que facilitam a melhoria do desempenho da tarefa de REN.

No contexto desta pesquisa, procuramos encontrar apenas uma entidade no texto chamada de *toxic_spans* em que as palavras que justificam o texto ser tóxicos são identificadas para facilitar o processo de moderação de conteúdo impróprio. Um modelo de reconhecimento de entidades nomeadas deve ser capaz de extrair as *toxic_spans* destacados em negrito conforme os exemplos a seguir:

- USER **Vaitomanocú vagabundo**
- USER A fazenda, pq esse programa ainda existe? **Ódioooo**
- USER Parabéns por expor tão claramente sua **insignificância**. Jamais será eleito

2.4 Inteligência Artificial

Russell e Norvig (2010) descrevem a Inteligência Artificial (IA) como uma área de estudo que procura desenvolver agentes inteligentes que podem pensar, simular ou agir em atividades que atualmente são praticadas por humanos como tomada de decisão, solução de problemas, etc. Nas subseções a seguir iremos abordar os conceitos de Processamento de Linguagem Natural, Aprendizado de Máquina e Aprendizado Profundo.

2.4.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é o campo da inteligência artificial que se dedica a compreender e reproduzir a linguagem humana. Suas aplicações incluem a tradução automática, análise e classificação de texto, reconhecimento de fala e modelagem de diálogo. A abordagem do PLN é multifacetada, podendo utilizar técnicas baseadas em regras linguísticas, representações vetoriais de palavras combinadas com algoritmos de aprendizado de máquina, além de técnicas mais recentes como o uso de algoritmos de aprendizado profundo especializados em tarefas de PLN, como por exemplo, o Transformer, que tem se mostrado mais eficiente devido ao seu poder de representação de contexto (Otter et al., 2021).

2.4.2 Aprendizado de Máquina

Uma das áreas mais conhecidas da IA é a área de aprendizado de máquina (*Machine Learning)*, muitas pesquisas e serviços para sistemas corporativos foram desenvolvidos usando algoritmos de aprendizado de máquina para as mais variadas tarefas.

Alguns trabalhos relacionados explorados nessa pesquisa utilizaram algoritmos de AM como *Random Forest* e *Support Vector Machines* (SVM) em conjunto com modelos semânticos como (*word embeddings)*. A ideia principal por trás do AM é aprender a reconhecer padrões complexos automaticamente e tomar decisões baseadas em dados (Han et al., 2011). O AM pode se dividir em quatro diferentes tipos de abordagens conforme visto em Han et al. (2011).

- **Aprendizado supervisionado (*Supervised learning*)**: Técnica que aprende um modelo capaz de reconhecer padrões através de dados previamente rotulados;
- **Aprendizado não supervisionado (*Unupervised learning*)**: Técnica que analisa dados não rotulados para agrupar amostras dos dados que compartilham os mesmos padrões;
- **Aprendizado semi-supervisionado (*Semi-supervised learning*)**: Técnica em que utiliza dados rotulados para aprender um modelo e dados não rotulados para refinar os limites entre as classes;
- **Aprendizado semi-supervisionado (*Active learning*)**: Técnica que interage diretamente com os usuários para rotular os dados e adquirir o conhecimento esperado.

Zampieri et al. (2019) e de Pelle e Moreira (2017) utilizaram o algoritmo SVM para treinar modelos que são capazes de reconhecer comentários tóxicos. Ambos os trabalhos também testaram seus conjuntos de dados com modelos de aprendizado profundo como *Bidirectional Long Short-Term-Memory* (BiLSTM), *Convolutional Neural Network* (CNN), e *Word2Vec*. As técnicas de aprendizado profundo apresentaram melhores resultados em comparação com os modelos de aprendizado de máquina.

2.4.3 Aprendizado Profundo

Os métodos de Aprendizado Profundo (*Deep Learning*) têm demonstrado excelentes resultados em tarefas como classificação de imagens, processamento de áudio e de linguagem natural, etc. (Kowsari et al., 2019; Pouyanfar et al., 2018). Os trabalhos que abordam o uso de AP em PLN utiliza modelos de linguagem e a composição de vetores de palavras para resolver diferentes tarefas (Mikolov et al., 2013; Collobert et al., 2011). Um dos desafios ao desenvolver soluções que utilizam AP é a grande quantidade de dados necessária em comparação com modelos tradicionais de AM, o que pode inviabilizar o uso dessa técnica em problemas com conjuntos de dados com pouca quantidade de dados (Kowsari et al., 2019), O uso de AP alcançou o estado da arte em vários domínios,

incluindo tarefas de PLN. Uma das facilidades de métodos de AP em comparação com métodos de AM é a capacidade de realizar a engenharia de recursos (*feature engineering*) de forma automatizada (Pouyanfar et al., 2018).

Redes Neurais Artificiais

Russell e Norvig (2010) compara Redes Neurais Artificiais (RNA) a estrutura do cérebro humano. As RNAs são compostas de objetos comumente chamados de neurônios que são conectados uns aos outros, semelhante ao funcionamento do nosso cérebro. A Figura 2.2 mostra um modelo matemático simples de um neurônio. Os neurônios são organizado em camadas, cada camada recebe a conexão da camada anterior e fornece conexões para as próximas camadas em uma parte escondida da rede. A entrada da rede pode ser obtida através de vetores de palavras gerados por modelos de linguagens. A saída da rede é personalizada de acordo com o resultado esperado para cada tarefa (Kowsari et al., 2019). Por exemplo, em tarefas de CBT a saída da rede é apenas dois valores, para tarefas de CMCT ou CMRT a saída da rede é igual ao número de classes ou rótulos que foram definidos.

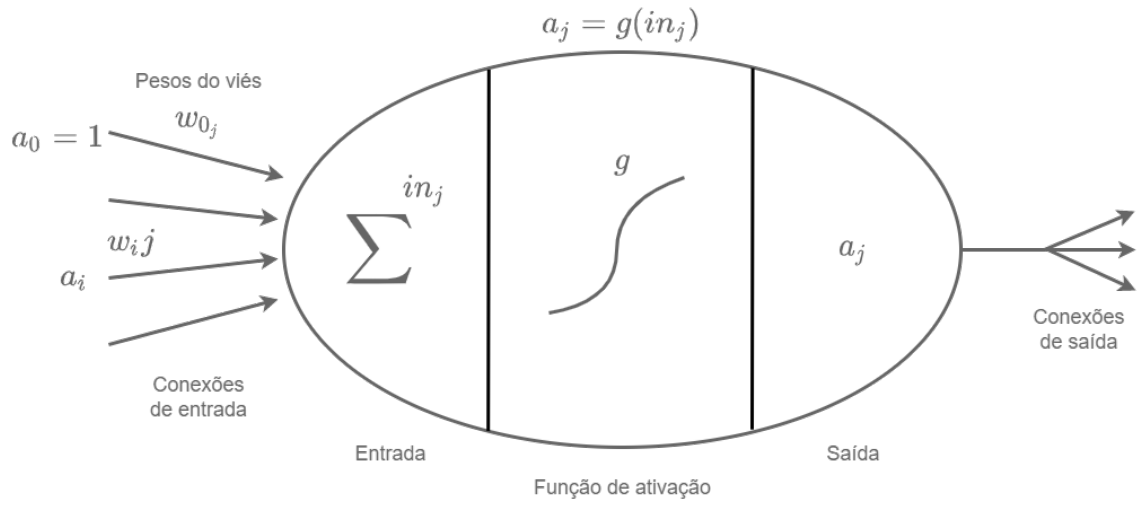

Figura 2.2: Modelo matemático simples de um neurônio. Fonte: Elaborado pelo autor, adaptada de Russel Russell e Norvig (2010).

As redes neurais artificiais são uma coleção de neurônios, argumenta Russell e Norvig (2010). O primeiro passo é implementar o modelo matemático do neurônio e então, conectá-los para formar uma rede. Ainda em Russell e Norvig (2010), descreve algumas topologias de redes como a *Feed-Forward Network* (FFN) e a *Recurrent Neural Network* (RNN), A FFN é uma estrutura mais simples, pois apenas conecta em uma direção, cada neurônio recebe a entrada dos neurônios anteriores e fornece saída para os neurônio posteriores. A RNN, por sua vez, é uma estrutura mais complexa, ela alimenta suas saídas

de volta com as próprias entradas, a resposta da rede a uma determinada entrada pode variar a depender do seu estado inicial, que pode depender de entradas anteriores, em 2.4.3 abordaremos com mais detalhes essa arquitetura.

Redes Neurais Recorrentes

As Redes Neurais Recorrentes (*Recurrent Neural Networks*) foram propostas inicialmente em Rumelhart et al. (1986), diversas variações surgiram desde então, RNR é uma das principais arquiteturas para lidar com dados sequenciais, como textos. A RNR atribui pesos aos pontos de dados de uma determinada amostra, sendo assim, uma ferramenta poderosa para tarefas de CT, pois permite que a semântica do texto seja utilizada como contexto adicional ao problema a ser resolvido (Kowsari et al., 2019). A Figura 2.3 ilustra o funcionamento de uma RNR.

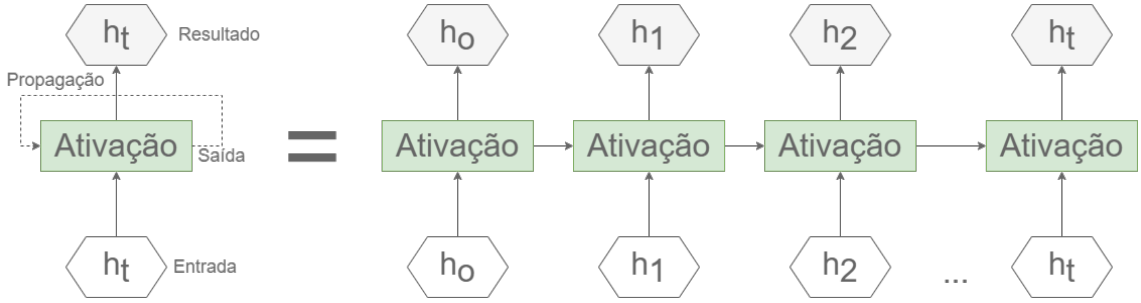

Figura 2.3: Representação de uma Rede Neural Recorrente. Fonte: Elaborado pelo autor, adaptada de Goodfellow et al. (2016).

As RNRs simples sofrem do *Vanishing Gradient Problem* em que os gradientes que são propagados pelas camadas da rede acabam desaparecendo ou explodindo por sucessivas transformações na informação da entrada (Pouyanfar et al., 2018). Esse problema acaba limitando o tamanho das redes e por isso, várias outras arquiteturas surgiram com o objetivo de mitigar este problema, uma dessas arquiteturas é a LSTM que veremos em 2.4.3.

Long Short-Term Memory

As *Long Short-Term Memory (LSTM)* foram introduzidas em Hochreiter e Schmidhuber (1997) como uma evolução das RNRs capaz de solucionar o *Vanishing Gradient Problem*, as células LSTM conseguem preservar o erro na retro-propagação fornecendo blocos de memória em suas conexões recorrentes. Cada bloco de memória inclui uma estrutura que armazena os estados temporais da rede. Adicionalmente, inclui unidades de controle de fluxo para filtrar o que deve ser armazenado no bloco de memória (Pouyanfar et al., 2018).

Goodfellow et al. (2016) definem uma LSTM como células que são conectadas recorrentemente entre si, substituindo as unidades ocultas (*hidden units*) usuais das RNRs simples. A informação é acumulada pelas células e o controle de memória é feito pelos portões (*gates*). O *Input Gate* adiciona informações úteis ao estado da célula baseado no resultado binário de uma função sigmoide usando as entradas $h_t - 1$ (saída da célula anterior) e x_t (entrada no passo de tempo atual). A informação, caso seja armazenada, passa por uma função *tanh* que gera um vetor que contém todos os valores possíveis das entradas, o vetor e os valores regulados são multiplicados para obter as informações úteis. *Forget Gate* é responsável por esquecer as informações que não são mais úteis, utilizando as mesmas entradas do *Input Gate*, elas são multiplicadas por matrizes de peso, seguidas pela adição dos pesos dos vieses. Então, o resultado é passado por uma função de ativação que fornece uma saída binária que decide se a informação é esquecida ou retida para uso futuro. *Output Gate* tem como responsabilidade extrair informações úteis do passo de tempo atual para ser apresentadas como uma saída. Um vetor é gerado através da função de ativação *tanh*, na sequência, a informação é regulada usando uma função sigmoide que filtra valores a serem armazenados usando as entradas $h_t - 1$ e x_t. Os valores do vetor e os valores regulados são multiplicados e enviados como uma saída e entrada para a próxima célula LSTM. A Figura 2.4 ilustra uma representação de uma célula LSTM a cada passo de tempo *t*.

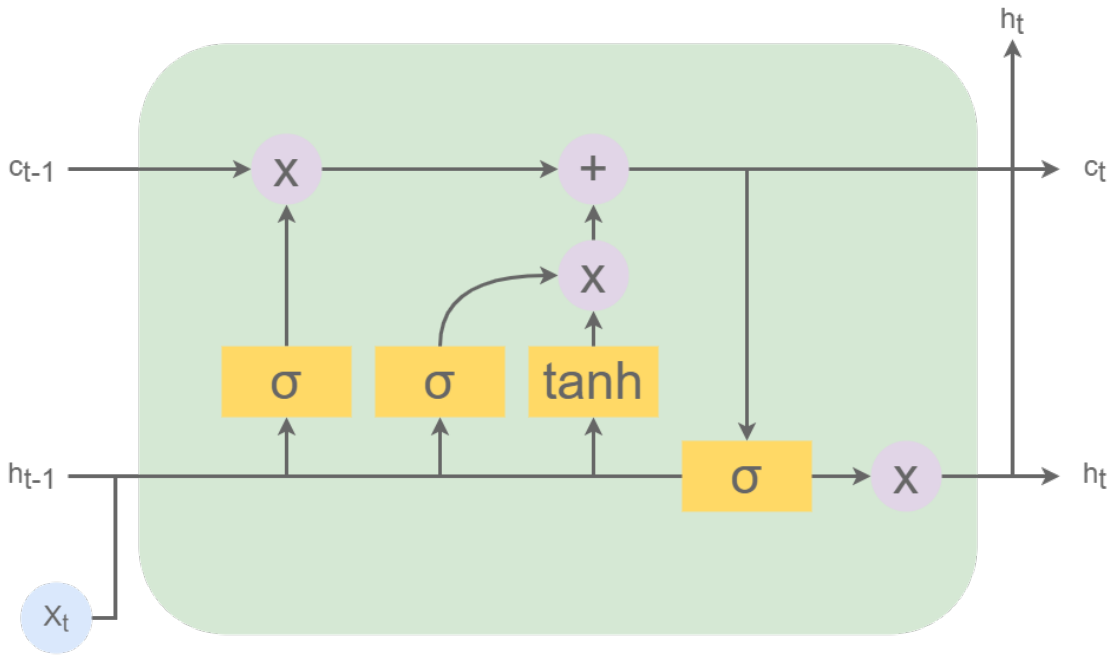

Figura 2.4: Representação de uma célula de uma *Long Short-Term Memory (LSTM)*. Fonte: Elaborado pelo autor, adaptada de Goodfellow et al. (2016).

Goodfellow et al. (2016) citam alguns casos de uso em que a LSTM obteve bons resultados como reconhecimento de escrita, reconhecimento de fala, tradução de máquina, entre outros. Kowsari et al. (2019) relatam que LSTM é um mecanismo mais ade-

quado para a tarefa de classificação de texto em comparação com RNR pela sua capacidade de preservar memória por um longo período.

Transformers

Em 2017, uma nova arquitetura que utiliza um mecanismo de atenção foi apresentado em Vaswani et al. (2017), O Transformers é composto de dois componentes chamados de codificador (*encoder*) e decodificador (*decoder*). O codificador é, na verdade, um conjunto de codificadores, o mesmo acontece com o decodificador, que na verdade, é um conjunto de decodificadores. Os codificadores são idênticos em sua estrutura, que é composta de uma *Feed-Forward Network* e um *Self-Attention*. O decodificador possui as mesmas camadas com adição de uma camada de atenção, chamada *Encoder-Decoder Attention* que permite ao decodificador focar nas partes relevantes de uma determinada sentença.

O primeiro passo é converter as palavras de uma determinada sentença em vetor utilizando representações vetoriais de palavras (*word embeddings*) (Vaswani et al., 2017), para esse estágio, normalmente se utiliza representações vetoriais pré-treinadas (Kowsari et al., 2019). O mecanismo de atenção *Self-Attention* então cria, através da multiplicação da representação vetorial das palavras nos passos de tempo, três vetores chamados de *Query*, *Key* e *Value*. Então uma pontuação de atenção é obtida para cada palavra através do produto escalar dos vetores *Query* e *Key*, então a pontuação é dividida pela raiz quadrada do vetor *Key* para que tenhamos gradientes mais estáveis. Uma função softmax é utilizada para normalizar a pontuação. O resultado da função softmax representa o quanto uma determinada palavra no texto é relevante para a palavra processada no passo de tempo atual. Na próxima etapa, cada valor do *Value* é multiplicado pela pontuação softmax para que seja possível manter os valores das palavras que serão focadas e reduzir os valores das palavras irrelevantes multiplicando por números minúsculos. Por fim, a soma ponderada do vetor *Value* representa a saída da camada de *Self-Attention* no passo de tempo atual. Este processo gera matrizes para *Query*, *Key* e *Value* pela multiplicação de todas as entradas *X*, as matrizes então permitem calcular mais facilmente as saídas da camada *Self-Attention* (Vaswani et al., 2017).

O mecanismo de atenção utilizado em Vaswani et al. (2017) é chamado de *Multi-Head Attention*, pois ele utiliza múltiplas representações de subespaços, tendo não um conjunto de matrizes *Query*, *Key* e *Value*, mas múltiplos conjuntos de matrizes, a atenção então é calculada para cada uma das *n heads* do mecanismo e posteriormente multiplicada com uma matriz de pesos gerada no treinamento do modelo para que apenas uma matriz seja passada para a camada FFN. Para representar a ordem das palavras em uma determinada sentença, é adicionado um vetor para cada *Input Embedding* chamado de *Positional Encoding* determinando então a posição das palavras ou a distância entre as

palavras da sentença. Dentro de cada subcamada (*Self-Attention*, FFN) temos uma conexão residual seguida por uma etapa de normalização. O decodificador funciona quase que da mesma forma, ele utiliza os vetores *K* e *V* gerados pelo codificador para focar nas posições apropriadas da sentença e então fornecer a saída. Para finalizar a rede, a saída do decodificador, um vetor de números reais, é então submetida a uma *Linear Layer* e *Softmax Layer* que aplica uma transformação linear e captura o índice do vetor com o maior valor respectivamente, assim, fornecendo o índice associado da palavra na representação vetorial de palavras utilizada no treinamento. A Figura 2.5 ilustra a arquitetura Transformers.

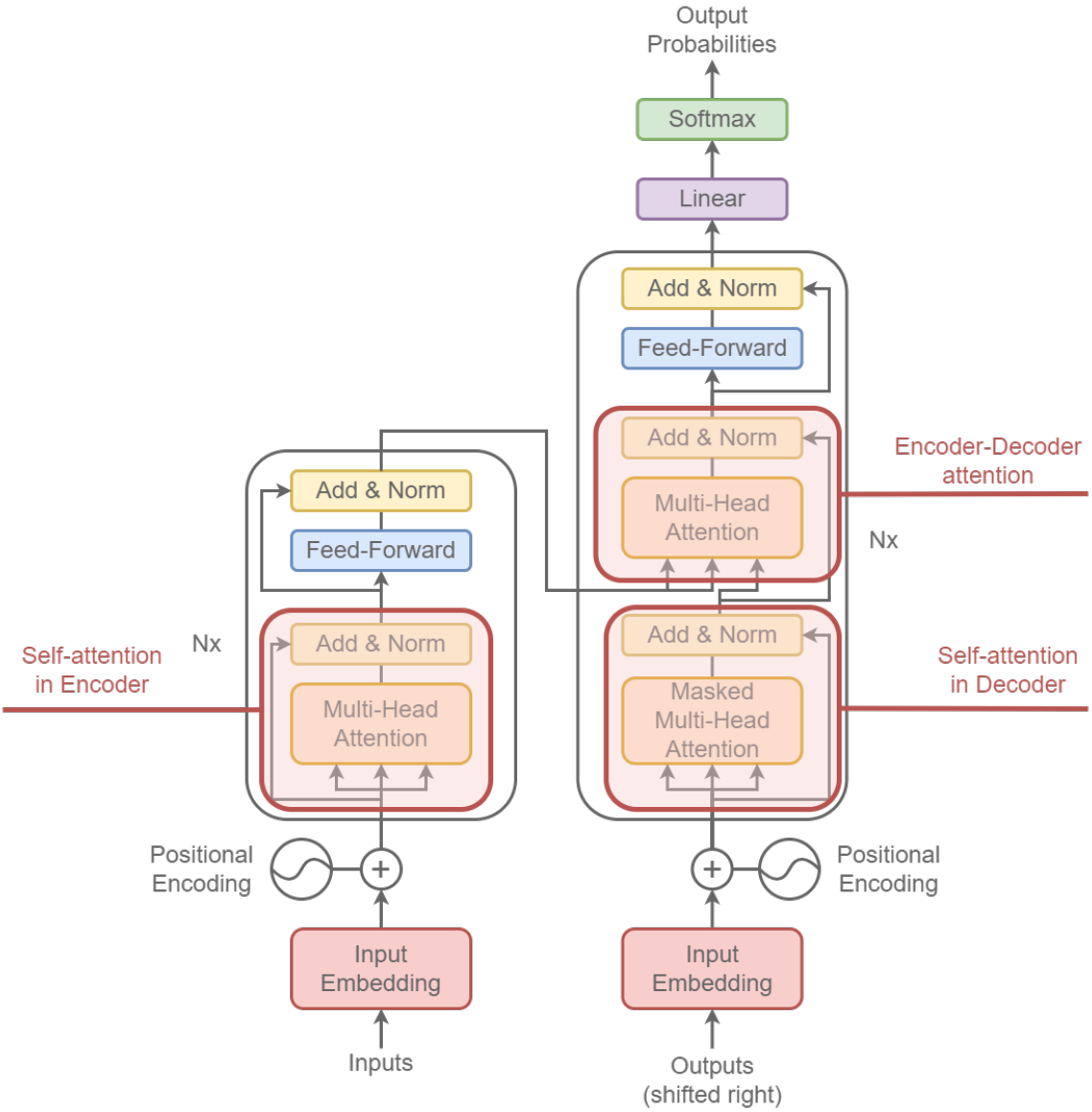

Figura 2.5: Representação da arquitetura Transformers. Fonte: Elaborado pelo autor, adaptada de Vaswani et al. (2017).

Um dos modelos mais conhecidos baseado na arquitetura Transformers é o *Bidirectional Encoder Representations from Transformers* (BERT) proposto em Devlin et al.

(2019). Este modelo é atualmente usado na pesquisa do buscador do Google[1]. Os resultados obtidos com esse modelo são surpreendentes, conseguindo atingir o estado da arte em 11 tarefas de PLN, incluindo o *GLUE score* com 80,5% (aumento de 7,7%), *MultiNLI accuracy* com 86,7% (aumento de 4,6%), entre outros (Devlin et al., 2019).

Em Radford et al. (2019) os autores utilizaram a arquitetura Transformers para treinar grandes modelos que possam ser utilizados em diferentes tarefas de PLN sem necessitar de treinamentos especializados por tarefas. Os modelos fazem parte da família *Generative Pre-trained Transformer* (GPT). Na versão mais atual, apresentado em Brown et al. (2020), os pesquisadores da OpenAI e Johns Hopkins University treinaram um modelo com 175 bilhões de parâmetros, 10 vezes maior do que seu antecessor. O conjunto de dados utilizado foi o *Common Crawl Dataset*, fornecido pela Common Crawl[2], uma organização que rastreia a internet e fornece seus dados publicamente, após a filtragem dos dados, 570GB de dados foram utilizados no treinamento dos modelos.

2.5 Mineração de dados

Han et al. (2011) apresentam a mineração de dados como um método que tem como objetivo manipular os dados afim de alcançar um uso desejado, o processamento de dados é composto de diferentes tarefas, como por exemplo, a limpeza dos dados que tem como objetivo preencher valores faltantes, identificar e amenizar valores discrepantes (*outliers*) e resolver demais inconsistências, também há a transformação dos dados em um formato que atenda aos objetivos propostos. Em textos, a mineração de dados pode ser utilizada para a extração de informações, geração de metadados ou *features* que auxiliem a compreensão do texto (Han et al., 2011).

2.5.1 Conjunto de dados

Han et al. (2011) define um conjunto de dados como um coleção de dados relevantes extraídos de uma ou mais fonte(s) de dados, como banco de dados, de forma a atender um uso específico. A entrada de uma pipeline de um problema de CT consiste de um conjunto de dados que tenha textos brutos, geralmente uma sequência de textos como $X = X_1, X_2, , ..., X_N$, onde X_i se refere a uma instância ou amostra de dado (um texto ou sentença). Cada instância de dado é anotada de acordo com a(s) tarefa(s) que o conjunto de dados procura atender (Kowsari et al., 2019).

[1] https://blog.google/products/search/search-language-understanding-bert/
[2] https://commoncrawl.org/

2.5.2 Análise exploratória de dados

Todo processamento de dados efetivo necessita de uma análise exploratória de dados, argumenta Han et al. (2011). A análise exploratória de dados é um processo preliminar que tem como objetivo compreender como um determinado conjunto de dados é estruturado, como os dados são distribuídos, suas respectivas granularidades e dimensões. Para isso, visualizações de dados, sumarização e outras técnicas são empregadas, conforme descrito em Dasu e Johnson (2003). A visualização de dados permite representar os dados de forma agregada e com elementos gráficos que facilitam a compreensão humana, como por exemplo, gráficos de barras, histogramas, gráficos de dispersão, entre outros (Han et al., 2011).

2.5.3 Anotação dos dados

O processo de anotação dos dados é utilizado para associar classes ou rótulos às instâncias dos dados que não são rotulados a fim de utilizar o conjunto de dados em algoritmos de aprendizado supervisionado e semi-supervisionado (Han et al., 2011). A anotação de dados é um processo que pode ser realizado de forma manual ou automatizada. Quando realizada a anotação manual dos dados, especialistas no domínio podem ser necessários para garantir a consistência das anotações (Poletto et al., 2021). Quatro técnicas de anotação diferentes foram empregadas nos trabalhos da área seguindo Poletto et al. (2021). A primeira técnica utiliza especialistas (anotadores ou os próprios pesquisadores que possuam conhecimento na área) para realizar as anotações. Outra técnica comumente utilizada é a anotação por não especialistas na área, normalmente estudantes. Alguns trabalhos utilizaram empresas que prestam serviços de anotação como FigureEight/Appen[3], Amazon Mechanical Turk[4], por último, a classificação automática também foi uma das técnicas utilizadas nos trabalhos estudados.

2.5.4 Transformação dos dados

Segundo Han et al. (2011), a transformação de dados permite manipular ou consolidar os dados de forma que o conjunto de dados atenda aos objetivos de uso. Kowsari et al. (2019) apresenta diversas técnicas que normalmente são aplicadas em tarefas de PLN como *Tokenization* que transforma uma sentença em uma lista de unidades menores

[3] https://appen.com/
[4] https://www.mturk.com/

chamadas de *tokens*, que podem ser palavras, caracteres ou subpalavra. *Punctuation Removal* que remove pontuações como vírgulas, pontos, exclamações, etc., para normalizar textos que possuem diferentes formas de escrita, mas que possuem o mesmo significado. *Stop Words Removal* em que palavras comuns como artigos, preposições, conjunções, etc., são removidas do texto por possuírem pouca informação relevante para a análise de PLN, *Spelling Correction* que corrige erros de ortografia frequentemente encontrados em conjuntos de dados de mídias sociais. *Stemming* que reduz palavras flexionadas à sua base, por exemplo, “estudando” para “estudar”. *Lemmatization* que remove a parte flexionada da palavra, por exemplo, “estudando” para “estud” (Jurafsky e Martin, 2009).

2.6 Métricas de avaliação de modelos

Entender como um modelo está performando é parte fundamental no desenvolvimento de métodos de classificação de texto (Kowsari et al., 2019). A literatura oferece uma série de métricas de avaliação que podem ser empregadas afim de quantificar os erros e acertos de um modelo aplicado a um determinado problema. A abordagem proposta nesta pesquisa utiliza dois tipos diferentes de algoritmos de inteligência artificial conforme as tarefas definidas na Seção 2.2 e 2.3. Segundo He et al. (2020), modelos de Reconhecimento de Entidades Nomeadas compartilham as mesmas métricas dos modelos de Classificação de Texto. Por isso, vamos abordar as principais métricas de avaliação que poderão ser utilizadas pelos modelos proposto nesse trabalho.

A avaliação da performance dos modelos de Classificação de Texto e de Reconhecimento de Entidades Nomeadas é baseada na contagem de dados de saída correta e incorreta fornecidas pelos modelos treinados. A comparação do valor real e do valor predito pelo modelo é atribuído em uma das quatro possíveis classes a seguir:

- **Verdadeiro Positivo (VP)**: O modelo predisse a classe positiva e o valor real é a classe positiva;
- **Verdadeiro Negativo (VN)**: O modelo predisse a classe negativa e o valor real é a classe negativa;
- **Falso Positivo (FP)**: O modelo predisse a classe positiva e o valor real é a classe negativa;
- **Falso Negativo (FN)**: O modelo predisse a classe negativa e o valor real é a classe positiva.

Os dados então atribuídos às classes acima são agregados em uma tabela chamada Matriz de Confusão (Han et al., 2011). A Tabela 2.4 ilustra uma matriz de confusão

exemplificando uma tarefa de Classificação Binária de Texto para detectar comentários tóxicos.

		Valor Real	
		Positivo	Negativo
Valor predito	Positivo	VP	FP
	Negativo	FN	VN

Tabela 2.4: Exemplo de Matriz de Confusão. Fonte: Elaborada pelo autor.

Essa matriz permite quantificar precisamente os erros e acertos de um determinado modelo. É com base nessa matriz que podemos calcular métricas para avaliar a performance dos modelos. As métricas comumente utilizadas para avaliação de modelos de classificação são a acurácia, precisão, abrangência e *F-Measure* que serão detalhadas em 2.6.1, 2.6.2, 2.6.3 e 2.6.4 respectivamente.

2.6.1 Acurácia

A Acurácia (*Accuracy*) é conhecida na literatura como uma métrica geral para avaliar um modelo, essa métrica é fortemente afetada pelo problema de classes desbalanceadas em que uma das classes é muito rara (Han et al., 2011). O uso da acurácia em conjunto de dados que possuem classes desbalanceadas pode levar a uma avaliação incorreta do modelo, por exemplo, como visto em 4.4 os comentários tóxicos representam 90% dos comentários no conjunto de dados OLID-BR, sendo assim, um classificador que sempre infere que os comentários são tóxicos, obterá uma acurácia de 90%, porém não terá a capacidade de diferenciar comentários tóxicos de comentários não tóxicos.

$$Accuracy = \frac{(VP + VN)}{(VP + VN + FP + FN)} \tag{2.1}$$

2.6.2 Precisão

A Precisão (*Precision*) pode ser interpretada como uma métrica de exatidão, ou seja, qual a porcentagem de instâncias rotuladas como positivas são realmente positivas? Essa métrica pode ser utilizada em problemas onde as classes estão desbalanceadas, pois é possível obter essa métrica individualmente para cada classe do classificador (Han et al., 2011).

$$Precision = \frac{VP}{VP + FP} \tag{2.2}$$

2.6.3 Abrangência

A Abrangência (*Recall* ou *Sensitivity*) pode ser interpretada como uma métrica de integridade, ou seja, qual a porcentagem de instâncias positivas são de fato positivas? Essa métrica calcula a capacidade de um classificador encontrar todas as instâncias positivas (Han et al., 2011).

$$Recall = \frac{VP}{VP + FN} \tag{2.3}$$

2.6.4 F-Measure

A F-measure (F_1 *score* ou *F-score*) é a média harmônica entre a Precisão e a Abrangência. Ela permite que a Precisão e a Abrangência sejam utilizadas em conjunto (Han et al., 2011).

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.4}$$

2.7 Confiabilidade entre anotadores

A confiabilidade entre anotadores (*Inter-Rater Reliability*) é um experimento onde dois ou mais indivíduos, referidos como anotadores, realizam independentemente a atribuição de classes ou categorias predefinidas ao mesmo conjunto de objetos de uma pesquisa científica. A extensão em que essas duas categorizações coincidem representa o que é chamado de confiabilidade entre anotadores. Se a confiabilidade entre anotadores for alta, ambos os anotadores podem ser usados de forma intercambiável sem que vieses de anotadores específicos sejam introduzidos nas anotações. A intercambiabilidade dos anotadores é o que justifica a importância da confiabilidade entre anotadores. Se a intercambiabilidade for garantida, então as categorias nas quais os objetos são classificados podem ser usadas com confiança sem se preocupar qual anotador as categorizou.Gwet (2014)

Selecionar coeficientes para avaliar a confiança entre anotadores não é uma tarefa trivial, a literatura usualmente utiliza o coeficiente Kappa de Cohen (k), que varia de -1 a 1, onde -1 representa um acordo aleatório e 1 representa um acordo perfeito e que é calculado como $\frac{p_a - p_e}{1 - p_e}$, onde p_a é o percentual de concordância (*percent agreement*) e p_e é o percentual de concordância corrigida ao acaso (*percent change-corrected agreement*) (Gwet, 2014).

Entretanto, existem diversos problemas que podem levar a uma interpretação incorreta do experimento de confiança entre anotadores ao utilizar o k, tais problemas são referidos na literatura como paradoxos de Kappa, como por exemplo, quando o percentual de concordância pela chance é alto, o processo de correção pode converter um valor relativamente alto para um valor relativamente baixo de k (Feinstein e Cicchetti, 1990). Por isso, Eugenio e Glass (2004) sugere que usar múltiplos coeficientes com diferentes formas de calcular p_e pode ser mais revelador do que usar um único coeficiente. Gwet (2014) oferece um guia que ajuda a escolher os coeficientes apropriados de acordo com as características do experimento de confiabilidade entre anotadores e também aborda em detalhes os paradoxos existentes nos coeficientes baseados em k. Desta forma, utilizamos esse guia para selecionar os coeficientes considerando diferentes formas de calcular p_e e assim, oferecer uma visão mais abrangente dos resultados no experimento. Nas subseções a seguir, iremos abordar os coeficientes utilizados neste trabalho.

2.7.1 Percentual de Concordância

O Percentual de Concordância (*Percent Agreement*) é a forma mais básica e intuitiva de avaliar a concordância entre anotadores, não é corrigido pela chance de acordo ao acaso. O percentual de concordância é calculado dividindo o total de objetos em que os anotadores concordaram pelo total de objetos anotados, conforme ilustra a equação 2.5 (Gwet, 2014).

$$p_a = \frac{n_{\text{concordando}}}{n_{\text{total}}} \tag{2.5}$$

Percentual de concordância. Fonte: Gwet (2014)

Alguns coeficientes utilizam variações do percentual de concordância de acordo com algumas necessidades, como por exemplo, para tratar anotações ausentes, alguns coeficientes calculam o percentual de concordância apenas para os objetos em que temos todas as anotações dos anotadores do experimento (Gwet, 2014).

2.7.2 AC_1 de Gwet

O coeficiente AC_1 foi sugerido em Gwet (2008) como uma alternativa mais resistente aos paradoxos enfrentados por coeficientes baseados em k. Ele se baseia na mesma equação de percentual de concordância p_a utilizada por k, porém com uma fórmula para calcular o percentual de concordância pela chance p_e que leva em consideração a probabilidade de concordar em objetos mais difíceis, multiplicando então, pela classificação

aleatória (Gwet, 2014). O coeficiente AC_1 de Gwet é formalmente definido da seguinte forma:

$$\widehat{\kappa}_{\mathrm{G}} = \frac{p_a - p_e}{1 - p_e}, \text{onde } p_e = \frac{1}{q(q-1)} \sum_{k=1}^{q} \hat{\pi}_k \left(1 - \hat{\pi}_k\right) \tag{2.6}$$

Coeficiente AC_1 de Gwet. Fonte: Gwet (2014)

onde $\hat{\pi}_k$ é dado pela seguinte equação: $\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \frac{r_{ik}}{r_i}$

2.7.3 Alfa de Krippendorff

O Alfa de Krippendorff (α_{K}) é um coeficiente versátil que se diferencia de outros coeficientes por poder ser aplicado independentemente do número de observadores, da quantidade de objetos, da ausência de dados ou do tipo de categoria observado, como categorias nominais, ordinais, intervalares ou proporcionais (Hayes e Krippendorff, 2007).

O α_{K} é calculado da seguinte forma:

$$\alpha_{\mathrm{K}} = \frac{p'_a - p_e}{1 - p_e}, \text{ onde } \begin{cases} p_e = \widehat{\pi}_1^2 + \left(1 - \widehat{\pi}_1\right)^2, \\ p'_a = \left(1 - \varepsilon_n\right) p_a + \varepsilon_n \end{cases} \tag{2.7}$$

Alfa de Krippendorff. Fonte: Gwet (2014)

onde p_a é a fórmula apresentada em 2.5, porém aplica as anotações par-a-par.

2.7.4 Escala de referência

A escala de referência auxilia na interpretação e comunicação dos resultados de um experimento de confiabilidade entre anotadores (Gwet, 2014). A Tabela 2.5 apresenta o modelo proposto por Landis e Koch (1977) e que é adotado neste trabalho.

Pontuação	**Nível de acordo**
< 0,0	Ruim
0,01 até 0,20	Leve
0,21 até 0,40	Regular
0,41 até 0,60	Moderado
0,61 até 0,80	Substancial
0,81 até 1,00	Quase perfeito

Tabela 2.5: Escala de referência para coeficientes de confiabilidade entre anotadores. Fonte: Elaborada pelo autor, adaptada de Landis e Koch (1977).

3. TRABALHOS RELACIONADOS

Neste capítulo, iremos apresentar os trabalhos relacionados no domínio de linguagem tóxica que foram fundamentais para sustentar os objetivos desta pesquisa. Iremos abordar as diferenças nas abordagens, os desafios e as limitações apontadas pelos pesquisadores nesta área.

Zampieri et al. (2019) procura identificar linguagem tóxica e alvos de discurso de ódio em comentários de redes sociais, para isso, os autores desenvolveram um conjunto de dados chamado OLID com 14.100 comentários anotados na língua Inglês dos Estados Unidos (EN-US). Para realizar a anotação do conjunto de dados, cada comentário foi anotado por dois anotadores diferentes, em caso de discordância, um terceiro anotador foi utilizado e então a técnica de voto majoritário foi empregada. Algumas competições acadêmicas como a OffensEval[1], que foi realizada em 2019 e 2020, utilizaram o OLID como conjunto de dados referência para diversos trabalhos. O OLID possui esquema de anotação multinível. O primeiro nível (*Offensive Language Detection*) é uma identificação binária entre textos ofensivos e textos não ofensivos. No segundo nível (*Categorization of Offensive Language*), o texto também é classificado de forma binária entre textos direcionados a um alvo ou não direcionados. O terceiro nível (*Offensive Language Target Identification*) só será informado caso o texto seja direcionado a um alvo, então ele será classificado entre indivíduo, grupo ou outros. Esse esquema de anotação é apresentado na Figura 3.1.

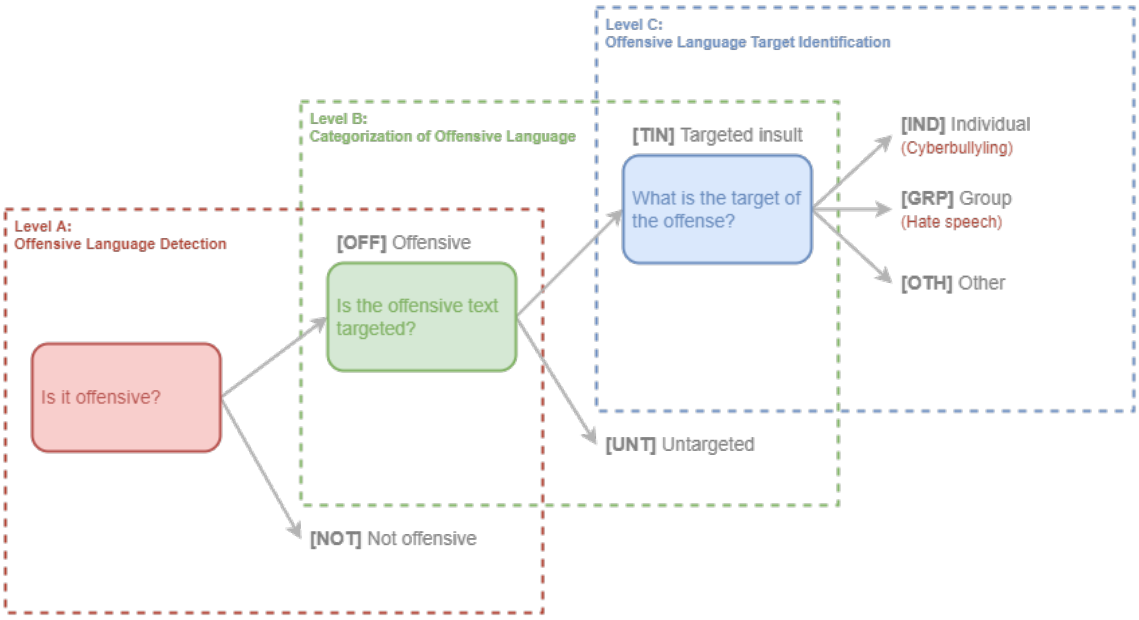

Figura 3.1: Esquema de anotação do OLID. Fonte: Elaborada pelo autor, adaptada de Zampieri et al. (2019).

[1] https://sites.google.com/site/offensevalsharedtask/home

A partir desse conjunto de dados, outros conjuntos de dados foram gerados para os seguintes idiomas: árabe, dinamarquês, grego e turco. Em Zampieri et al. (2020), o OLID com comentários em EN-US foi incrementado com mais de 9 milhões de novos comentários, devido aos trabalhos realizados no ano anterior. Ainda em Zampieri et al. (2019), três modelos foram gerados a partir dos seguintes algoritmos *Support Vector Machines* (SVM), *Bidirectional Long Short-Term-Memory* (BiLSTM) e *Convolutional Neural Network* (CNN). Os resultados obtidos foram 0,76, 0,75 e 0,80 respectivamente na medida F, demonstrando que técnicas de aprendizado profundo podem alcançar melhores resultados em comparação com técnicas de aprendizado de máquina.

de Pelle e Moreira (2017) o autor relata a dificuldade em identificar comentários ofensivos devido à capacidade dos *haters* de modificarem as palavras ofensivas inserindo asteriscos, espaços entre as letras ou trocando caracteres por outro similar. Técnicas que utilizam listas de palavras para capturar estes comentários falham na medida que os usuários descobrem possíveis variações das palavras que são permitidas pelos sistemas. O trabalho então apresenta como contribuição um conjunto de dados anotado, chamado OFFCOMBR[2], com 1.250 textos em Português do Brasil (PT-BR) para que a comunidade acadêmica possa treinar modelos de identificação de comentários tóxicos. O esquema de anotação proposto não é compatível com os objetivos desta pesquisa, entretanto, os comentários poderão ser reclassificados e agregados ao conjunto de dados proposto nessa pesquisa. O projeto também treinou modelos de classificação para reconhecer comentários tóxicos, *Support Vector Machines* (SVM) e *Naive Bayes* obtiveram 0,80 e 0,75 respectivamente na medida F.

Em Poletto et al. (2021) uma revisão sistemática sobre discurso de ódio foi realizada com o objetivo de identificar recursos e *benchmarks* disponíveis na comunidade de ciência da computação. Esse trabalho seguiu os métodos para revisão sistemática propostos em Kitchenham (2004). O trabalho menciona o repositório https://hatespeechdata.com/ que tem como objetivo reunir todos os conjuntos de dados sobre discurso de ódio e fenômenos relacionados. Essa revisão sistemática evidencia a disparidade entre recursos disponíveis em EN-US em relação aos outros idiomas, são 24 conjuntos de dados com texto em EN-US contra 2 conjuntos de dados com textos em PT-BR que foram citados pelos autores. A maior parte dos trabalhos utiliza busca por palavra-chaves como técnica para encontrar os comentários tóxicos em conteúdos públicos como publicações em blogs e redes sociais, o próprio trabalho alerta sobre os vieses presente nessa técnica e apresenta diferentes estratégias usadas pelos autores para mitigar o problema. O trabalho apresenta dois conjuntos de dados com textos em PT-BR: NCCVG[3] e o OFFCOMBR[2] ambos conjuntos de dados não atendem aos requisitos do projeto, por não terem o esquema de anotação proposto, mas os comentários poderão ser utilizados como fonte de dados con-

[2]https://github.com/rogersdepelle/OffComBR
[3]https://github.com/LaCAfe/Dataset-Hatespeech

forme explicado na Seção 4.1. As fontes de dados exploradas nos trabalhos são bastante variadas, mas o Twitter[4] é de longe a fonte de dados mais utilizada pelos pesquisadores, segundos os autores de Poletto et al. (2021), isso acontece devido ao tamanho dos textos serem limitados e uma política amigável para divulgação dos dados coletados. Outra importante contribuição desse trabalho é o entendimento dos conceitos relacionados que foram definidos com base na literatura revisada, termos como agressividade/toxicidade, discurso de ódio, ofensividade, entre outros são explicados com exemplos extraídos dos conjuntos de dados analisados.

Em Fortuna et al. (2019), os autores desenvolveram um conjunto de dados com 5.668 *tweets* em PT-BR, coletados entre janeiro e março de 2017. Para coletar os dados, os autores definiram alguns perfis de usuários e palavras-chave relacionadas a discurso de ódio. O conjunto de dados conta com duas tarefas anotadas, uma classificação binária e uma classificação multirrótulo semelhante a uma das tarefas que propomos nessa pesquisa, os rótulos foram protocolados a medida em que eram detectados pelos anotadores. Ao final, 9 rótulos foram definidos:

- *sexism* ’sexismo’: Discurso de ódio baseado em gênero.
- *body* ’corpo’: Discurso de ódio baseado no corpo, como obesidade, altura, etc.
- *origin* ’origem’: Discurso de ódio baseado no local de origem.
- *homophobia* ’homofobia’: Discurso de ódio baseado na orientação sexual.
- *racism* ’racismo’: Discurso de ódio baseado na etnia.
- *ideology* ’ideologia’: Discurso de ódio baseado na ideologia de indivíduos ou grupos.
- *religion* ’religião’: Discurso de ódio baseado na religião.
- *health* ’saúde’: Discurso de ódio baseado em condições de saúde.
- *other-lifestyle* ’outros estilo de vida’: Discurso de ódio baseado em hábitos, como vegetarismo, veganismo, etc.

Em Pavlopoulos et al. (2021) é apresentada uma tarefa de extração das parte(s) tóxica(s) de um texto tóxico, segundo os autores, os sistemas de detecção de toxicidades atuais classificam textos em tóxicos ou não tóxicos, porém, em alguns casos, mais informações precisam ser fornecidas para auxiliar a moderação de conteúdo tóxico. O conjunto de dados *Toxic Spans Dataset* fornece comentários tóxicos e as respectivas posições dos caracteres que justificam a toxicidade de um texto tóxico.

O conjunto de dados proposto em Fortuna et al. (2019) é similar ao proposto nesta pesquisa, porém com algumas significativas diferenças. O conjunto de dados usado

[4] https://twitter.com/

neste trabalho possui em duas tarefas, a Classificação Binária de Texto (CBT) e a Classificação Multirrótulo de Texto (CMRT) e foi totalmente coletado do Twitter. O trabalho proposto nesta pesquisa procura atender 5 possíveis tarefas de PLN e coletar dados de diferentes fontes de dados para que seja possível capturar diferentes comportamentos dos usuários. Os comentários tóxicos disponíveis em HLPHSD[5] poderão ser utilizados como fonte de dados para o conjunto de dados proposto nessa pesquisa.

Leite et al. (2020) também relata o maior foco em pesquisas na área para a língua inglesa em comparação com a língua portuguesa e por isso, construiu um conjunto de dados com 21.000 *tweets* (termo usado para se referir as publicações na plataforma Twitter) com anotação binária (tóxico e não tóxico) e anotação multirrótulo (*LGBTQ+phobia*, *Insult*, *Xenophobia*, *Misogyny*, *Obscene*, *Racism*). A coleta dos dados no Twitter utilizou duas abordagens, a primeira é através do uso de palavra-chaves específicas em cada rótulo, a segunda é o monitoramento de perfis que são normalmente afetados por comentários tóxicos. A anotação foi realizada manualmente por 42 voluntários diferentes, sendo que cada *tweet* foi anotado por três anotadores. O trabalho fornece uma comparação entre as anotações e é possível ver uma grande divergência entre os anotadores. Considerando apenas os resultados em que os anotadores concordaram, o conjunto de dados possui 19.510 comentários não tóxicos e 1.490 comentários tóxicos. Já na anotação multirrótulo, temos *LGBTQ+phobia* com 74 tweets, *Insult* com 517 tweets, *Xenophobia* com 15, *Misogyny* com 29 tweets, *Obscene* com 612 tweets e *Racism* com 6 tweets. Podemos perceber que existem poucos casos para a maioria dos rótulos, o que torna difícil identificar e generalizar os padrões em poucas amostras de dados. O trabalho também destaca a importância de se usar conjuntos de dados específicos por linguagem para alcançar resultados melhores do que soluções multi-línguas.

Neste trabalho, busca-se construir um sistema que utiliza inteligência artificial (IA) para responder múltiplas perguntas de um mesmo comentário tóxico, como os conjuntos de dados em PT-BR abordados neste capítulo não possuem todas as anotações propostas nesta pesquisa, foi decidido que uma das contribuições do trabalho é a construção de um conjunto de dados em PT-BR com múltiplas tarefas relacionadas com a detecção de linguagem tóxica. Como ponto de partida, foi utilizada a hierarquia de conceitos elaborada em Poletto et al. (2021), o esquema de anotação proposto em Zampieri et al. (2019) e Fortuna et al. (2019) com algumas adaptações, como por exemplo, os tipos de linguagem tóxica que foram definidos levando em consideração todos os trabalhos relacionados abordados nesta análise da literatura. Adicionalmente, incluímos a tarefa proposta em Pavlopoulos et al. (2021) que procura mapear a(s) parte(s) tóxica(s) do texto.

[5] https://github.com/paulafortuna/Portuguese-Hate-Speech-Dataset

4. CONJUNTO DE DADOS (OLID-BR)

O conjunto de dados *Offensive Language Identification for Brazilian Portuguese* (OLID-BR) é um dos principais entregáveis desta pesquisa, visto a falta de conjunto de dados com múltiplas tarefas de PLN na área de detecção de linguagem tóxica explorada no Capítulo 3. No total, 6.354 comentários anotados usando um esquema de anotação com 5 tarefas de PLN foi gerado através de anotações de três anotadores contratados pelo autor deste trabalho. Adicionalmente, outros 7.184 comentários foram disponibilizados com apenas uma ou duas anotações e que podem estender o conjunto de dados. O conjunto de dados OLID-BR foi disponibilizado publicamente nas plataformas Hugging Face (https://huggingface.co/datasets/dougtrajano/olid-br) e Kaggle (https://www.kaggle.com/datasets/dougtrajano/olidbr) com o intuito de aumentar os trabalhos acadêmicos com textos em Português nesta área.

4.1 Coleta dos dados

A coleta dos dados ocorreu em plataformas de redes sociais e conjuntos de dados relacionados. Ao final, os comentários coletados passaram por técnicas de processamento de texto para filtragem e anonimização dos dados. A Figura 4.1 ilustra o processo de desenvolvimento do OLID.BR, onde na primeira parte, os dados foram coletados das diferentes fontes de dados, logo em seguida, foi realizada a filtragem apenas dos comentários tóxicos utilizando o grau de toxicidade fornecido pela Perspective API como referência e também de comentários que estão em Português, a anonimização dos dados ocorreu antes de inserir os dados no Amazon S3 (serviço de armazenamento de objetos da Amazon Web Services). Com os dados brutos já preparados, em cada iteração do processo de anotação foi feita a amostragem dos dados estratificada pela origem dos dados, garantindo que a mesma proporção de cada fonte de dados fosse respeitada em cada iteração, os dados foram adicionados ao Label Studio (Tkachenko et al., 2020), uma ferramenta de anotação de textos, para que fossem anotadas pelos anotadores. Os dados já anotados foram processados para gerar o conjunto de dados conforme detalhado nas subseções a seguir. Na subseção 4.1.1 abordaremos as fonte dos dados utilizadas, em 4.1.2 detalhamos o processo empregado para seleção dos dados, por fim 4.1.3 explicaremos os processos de anonimização que foram realizados.

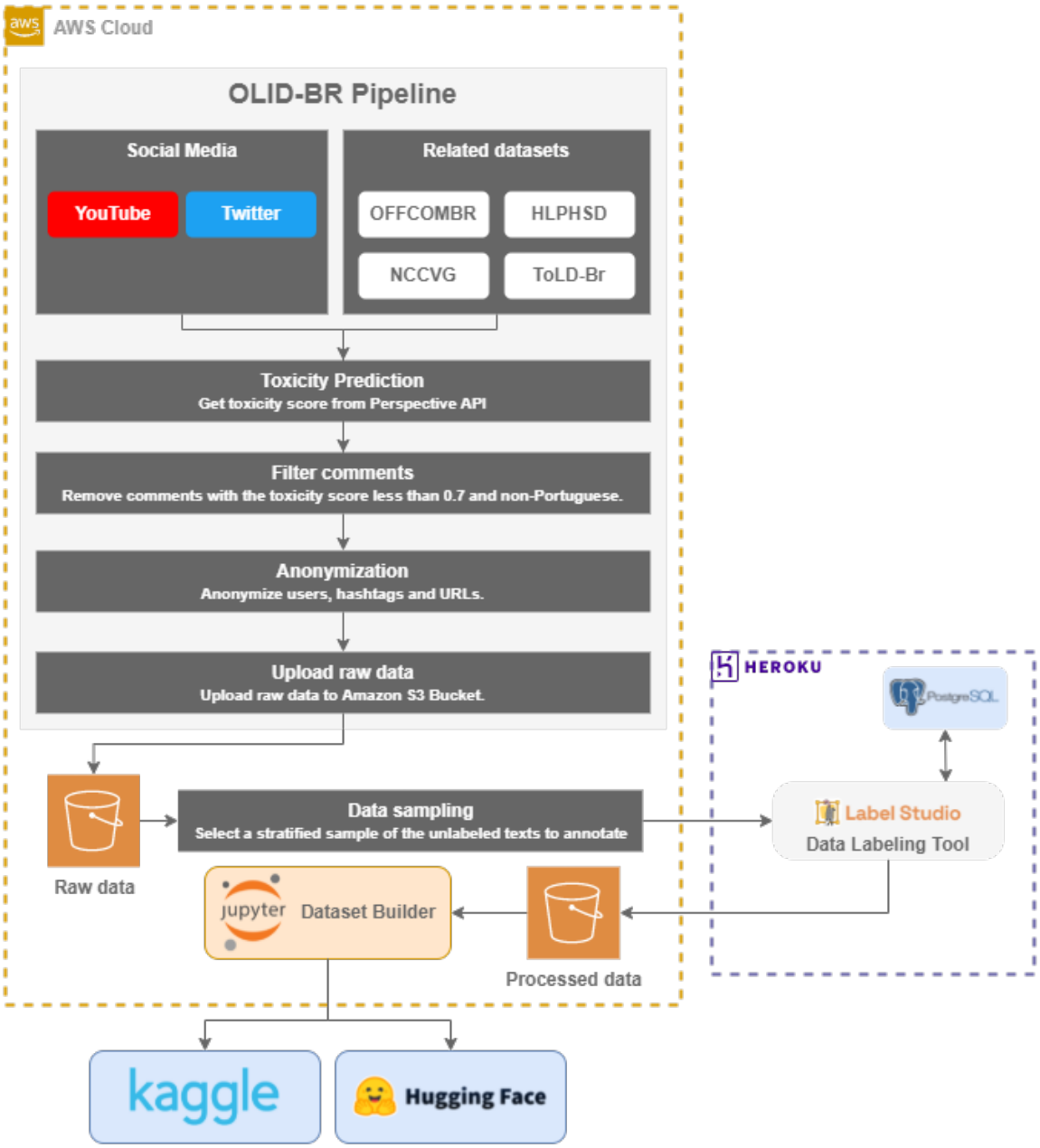

Figura 4.1: Processo de desenvolvimento do OLID-BR. Fonte: Elaborada pelo autor.

4.1.1 Fonte dos dados

As fontes de dados foram definidas com o objetivo de fornecer diversidade na população dos dados e reduzir possíveis vieses. Como abordado no Capítulo 1, as redes sociais proporcionam o ambiente ideal para a coleta deste material devido a grande interação social, por isso, realizamos um cadastramento nos serviços do YouTube e do Twitter para coletar os dados para fins de pesquisa científica.

Como utilizamos diferentes fontes de dados, a coleta dos dados foi adaptada de acordo com o contexto de cada fonte de dados. No YouTube, todos os comentários de vídeos previamente selecionados pelo autor da pesquisa utilizando como critério a alta

possibilidade de sofrerem ou incitarem discurso de ódio foram coletados. No Twitter, duas abordagens foram utilizadas para coleta dos dados. Na primeira, perfis com alta possibilidade de sofrerem ou incitarem discurso de ódio foram previamente selecionados pelo autor da pesquisa e então, seus *tweets* (postagens) e as respectivas respostas foram coletados. Após o piloto do processo de coleta e anotação dos dados, utilizamos as partes dos textos classificadas como tóxicas pelos anotadores como palavras-chaves para procura de novos *tweets* e respostas com o objetivo de maximizar a quantidade de exemplos coletados. Além dos comentários coletados das redes sociais, os comentários em Português dos conjuntos de dados OFFCOMBR[1], NCCVG[2], HLPHSD[3] e ToLD-Br[4] foram agregados sem suas anotações originais.

No total, **249.162** comentários foram coletados e passarão pelas etapas de seleção e anonimização dos dados abordados nas subseções a seguir.

4.1.2 Seleção dos dados

Como o objetivo desta pesquisa é estudar os comentários tóxicos em profundidade, detectando possíveis tipos de linguagem tóxica, alvos dos ataques e parte dos textos que são consideradas tóxicas, precisamos selecionar os dados que atendem estes requisitos. Para isso, os comentários foram filtrados com o auxilio da Perspective API[5] (desenvolvida pela Google Jigsaw) que fornece um grau de toxicidade para cada comentário que é enviado para a *Application Programming Interface* (API). No piloto do processo de anotação, selecionamos os comentários com grau de toxicidade superior a 0,5, porém nossos anotadores reclassificaram vários comentários previamente classificados como tóxicos para não tóxicos. Após cruzar a quantidade de comentários reclassificados para não tóxico com o grau de toxicidade, identificamos que utilizar um grau de 0,7 seria mais adequado.

Adicionalmente ao filtro de toxicidade, comentários em outros idiomas, duplicados, ou que não são legíveis foram descartados. **153.559** comentários foram selecionados e considerados elegíveis para o processo de anotação.

[1] https://github.com/rogersdepelle/OffComBR
[2] https://github.com/LaCAfe/Dataset-Hatespeech
[3] https://github.com/paulafortuna/Portuguese-Hate-Speech-Dataset
[4] https://github.com/JAugusto97/ToLD-Br
[5] https://www.perspectiveapi.com/

4.1.3 Anonimização dos dados

Anonimização dos dados é o processo de desidentificação de dados preservando o formato e tipo do dado. O dado anonimizado pode receber uma sequência especial de caracteres ou um valor aleatório para a informação que se deseja anonimizar. Por exemplo, a substituição do nome “Don Quixote” por “Ron Edwards” em uma sentença é um exemplo de anonimização de dado, a substituição de “Don Quixote” por “XXXXX” é um exemplo de mascaramento de dados (Raghunathan, 2013). Entretanto, ambos os termos mascaramento de dados e anonimização de dados são intercambiáveis, segundo Raghunathan (2013).

Aplicamos este processo aos comentários tóxicos com o objetivo de remover informações que possam identificar os usuários envolvidos, sejam eles agressores (*haters*) ou vítimas. Alguns padrões foram estabelecidos usando expressões regulares para remover os nomes dos usuários nas redes sociais, links e *hashtags* que possam estar contidos nos comentários analisados. Também utilizamos análise morfológica através do modelo em Português disponibilizado na biblioteca em Spacy para identificar possíveis nomes pessoais e então anonimizá-los. Todos os comentários do OLID-BR passaram por este processo.

4.2 Anotação dos dados

Nesta seção, abordamos o processo de anotação dos dados utilizado na construção do conjunto de dados desta pesquisa. Com base no conhecimento apresentado na Subseção 2.5.3, combinamos técnicas de anotação automatizada e manual para atender o esquema de anotação proposto. A anotação automática foi empregada na classificação binária que identifica os comentários tóxicos (tarefa 1), para as demais tarefas do esquema de anotação foi utilizado anotadores qualificados (especialistas) que foram treinados com recursos fornecidos pelo autor da pesquisa. A subseção 4.2.1 apresenta as diretrizes de anotação e os requisitos necessários para os anotadores qualificados. A subseção 4.2.2 apresenta o esquema de anotação hierárquico do OLID-BR. A subseção 4.2.3 apresenta a ferramenta de anotação utilizada neste projeto. A subseção 4.2.4 apresenta o processo para avaliação de concordância entre os anotadores.

4.2.1 Diretrizes de anotação e anotadores qualificados

A maioria dos trabalhos relacionados analisados no Capítulo 3 apontaram dificuldades em convergir os anotadores no mesmo entendimento sobre os conceitos, isso se deve a vários fatores, conforme apontam os autores, como diferentes experiências pessoais e relações entre grupos sociais, falta de definição clara dos conceitos, entre outros. Para mitigar este problema, estabelecemos diretrizes de anotação que foram seguidas pelos anotadores durante o processo de anotação. As diretrizes de anotação foram disponibilizadas em https://dougtrajano.github.io/olid-br/annotation/guidelines.html. Também estabelecemos requisitos técnicos para que um anotador seja qualificado para realizar as anotações. Os anotadores precisaram atender os seguintes requisitos para serem elegíveis:

- Inglês básico, já que este é o idioma usado pela ferramenta de anotação (Label Studio);
- Português nativo, para entender os textos que serão anotados;
- Bom entendimento de linguagem tóxica e como detectá-la utilizando os conceitos apresentados nesse trabalho.

Para atender ao requisito de entendimento de linguagem tóxica, os anotadores realizaram o curso Comunicação Não Violenta[6], fornecido gratuitamente pela Fundação Escola de Comércio Álvares Penteado (FECAP) e com carga horária de 4 horas. Adicionalmente, um treinamento com o autor da pesquisa foi realizado onde explicamos as diretrizes de anotação e apresentamos a ferramenta de anotação utilizada neste projeto.

4.2.2 Esquema de anotação

O esquema de anotação do OLID-BR é baseado no esquema de anotação do *Offensive Language Identification Dataset* (OLID) proposto em Zampieri et al. (2019) e que foi utilizado em competições acadêmicas como o OffensEval[7], com isso, acreditamos que futuras competições acadêmicas possam incluir o idioma Português através do uso do OLID-BR. Duas tarefas adicionais também foram introduzidas, uma tarefa de classificação multirrótulo adaptada de Fortuna et al. (2019) fornece a possibilidade de identificar os tipos de linguagem tóxica encontrados nos comentários tóxicos, a segunda tarefa, adaptada de Pavlopoulos et al. (2021), possibilita extrair a(s) parte(s) do texto que tornam-o

[6] https://www.fecap.br/curta-duracao/comunicacao-nao-violenta-1/
[7] https://sites.google.com/site/offensevalsharedtask/home

tóxico, fornecendo um nível maior de profundidade que auxilia a moderação humana, permitindo realçar as possíveis partes tóxicas do texto, caso existem. Ao todo, o OLID-BR possui anotações para 5 tarefas de processamento de linguagem natural. A Figura 4.2 ilustra o esquema de anotação do OLID-BR.

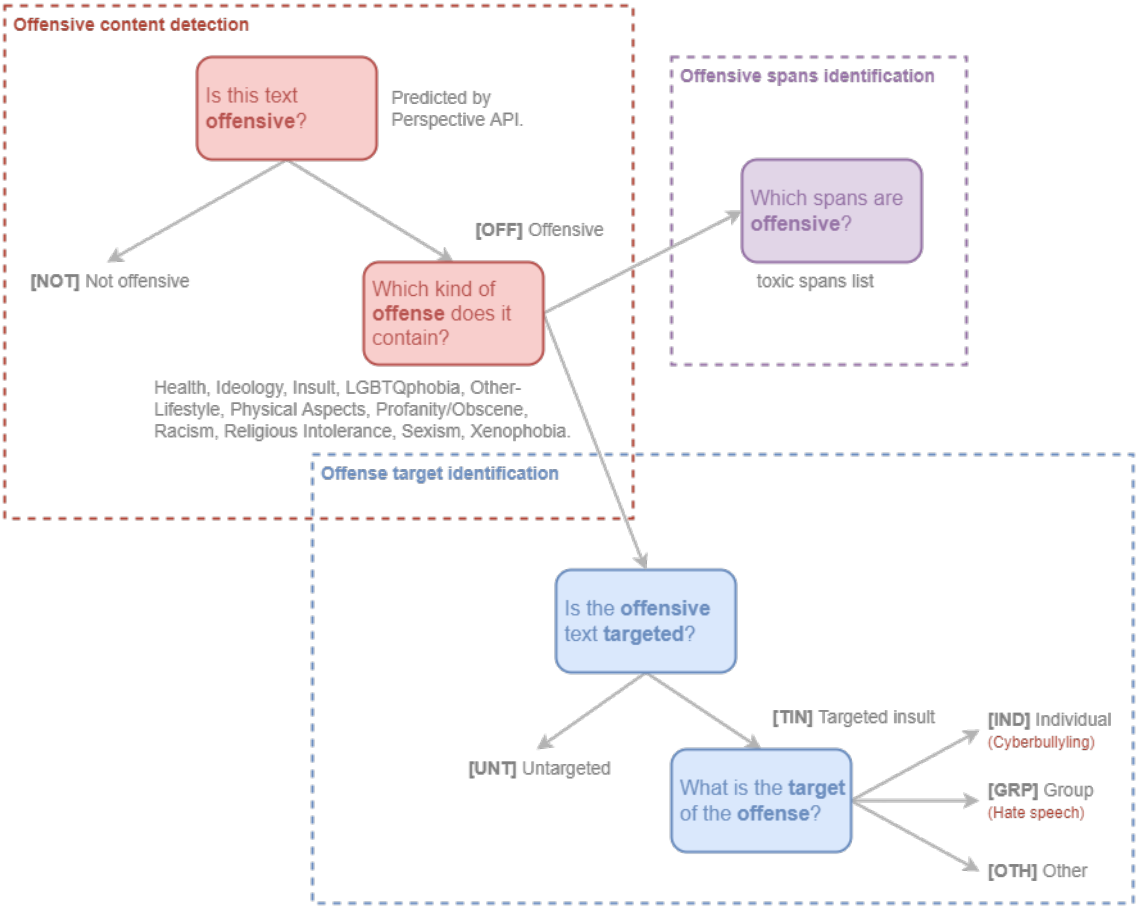

Figura 4.2: Esquema de anotação do OLID-BR. Fonte: Elaborada pelo autor.

Tarefa 1 - Detecção de comentários tóxicos

A detecção de comentários tóxicos é uma tarefa de CBT que obteve sua anotação através do grau de toxicidade fornecido pela Perspective API[8] conforme explicado em 4.1.2 e que foi revisada pelos anotadores qualificados. Alguns comentários previamente identificamos como tóxicos foram reclassificados como não tóxicos de acordo com a estratégia de atribuição de anotação do campo **is_offensive** visto em 4.2.5. A Tabela 4.1 apresenta os possíveis valores do campo **is_offensive**.

[8]https://www.perspectiveapi.com/

Classe	Valor	Descrição
Offensive	OFF	O comentário é ofensivo ou tóxico.
Not offensive	NOT	O comentário não é ofensivo ou tóxico.

Tabela 4.1: Classes de comentários no OLID-BR. Fonte: Elaborada pelo autor.

Tarefa 2 - Classificação do tipo de linguagem tóxica

A classificação do tipo de linguagem tóxica é uma tarefa de CMRT para identificar todos os rótulos aplicáveis em um determinado comentário tóxico. Os rótulos foram definidos com base em análise preliminar feita pelo autor da pesquisa, no glossário de termos da Universidade de Washington[9], na documentação da Perspective API[10] e nos trabalhos relacionados Fortuna et al. (2019) e Poletto et al. (2021). O anotador pode selecionar opcionalmente os rótulos de linguagem tóxica que se aplicam ao texto. Abaixo, apresentamos os rótulos de linguagem tóxica, seus respectivos campos no conjunto de dados, descrições e exemplos.

Health (health): Discurso de ódio com base em condições de saúde, como deficiência física, discriminação por idade, etc.

- USER Igualzinho a você, usa uma massa por fora pra esconder que por dentro é frango.
- USER velho broxa
- Parece um ET, você é um monstro, tem que morrer

LGBTQphobia (lgbtqphobia): Discurso de ódio com base na identidade de gênero ou orientação sexual.

- USER Meu deus gay hauahuahaua
- USER O problema não é sexualidade e sim competência. É diferente ferrar só um rabo ou ferrar de milhares.
- USER Você é uma vergonha para os gays pq tu é um gay ladrão de dinheiro da Pandemia.

Ideology (ideology): Discurso de ódio com base nas ideologias, como feminista, esquerda política, etc.

- USER Então porque as professoras,as políticas femininas ganham um salário de miséria

[9] https://students.wustl.edu/glossary-bias-terms/
[10] https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages.

- Bolsonaro: “Como é duro ser patrão no Brasil”. 19 milhões de brasileiros passam fome e a grande preocupação do cretino é com os patrões.
- USER Esquerda querendo falar de corrupção É de matar kkkk

Insult (insult): Discurso de ódio que possui insulto, injúria, xingamento. Tem o objetivo de inflamar uma discussão ou irritar uma parte.

- USER Só não falou mais merda porque não tem como
- USER Você é um lixo!
- USER Foi você quem mandou fechar tudo, ajudou quebrar, seu mau caráter

Other-Lifestyle (other_lifestyle): Discurso de ódio com base em hábitos de vida, como vegetariano, vegano, fumante, etc.

- USER A fazenda, pq esse programa ainda existe? Ódioooo
- Povo da ração com milho he he
- USER crackudo voz de fumante

Physical aspects (physical_aspects): Discurso de ódio com base em características físicas, como gordofobia, tamanhismo, etc.

- USER Forte esse Chandon em gordola.
- USER Próximo major já sabem, galera: caçar o gordo e extermina-lo.
- USER Deveriam ter escolhido uma modelo menos gordinha , ela parece que não está passando fome !!

Profanity/Obscene (profanity_obscene): Discurso de ódio possui palavras obscenas, vulgar, pornográficas, etc.

- USER Comer cu de curioso
- USER Mas que grande fdp.
- USER Passar porte de vacina é o meu pau

Racism (racism): Discurso de ódio com base na raça ou etnia.

- Nego Ney URL

- Essas filhas de pedreiro africano e foda
- Não tenho tv colorida pra ficar olhando essa preta nao

Religious intolerance (religious_intolerance): Discurso de ódio com base na religião, culto, prática religiosa.

- USER Um governo de Evangelicos. Um governo de Corruptos. Um governo de Criminosos.
- Uma macumbeira a menos
- Que diabo de Ogum, por isso que não vai pra frente

Sexism (sexism): Discurso de ódio com base no gênero ou sexo.

- USER E tem como parar de ouvir mulher??
- Se eu fosse o Temer criava o Ministério da Cozinha e da Limpeza e colocava uma mulher, só pro pessoal chorar menos. hahahaha
- USER Uma calcinha dessa fio-dental dentro da bunda não tem que resistir uma tentação dessa?

Xenophobia (xenophobia): Discurso de ódio contra pessoas estrangeiras ou de outras culturas.

- USER O brasileiro merece o que tá passando povo ignorante e sem a menor consciência bando de fudido do caralho
- USER USER PUTAQUEPARIUUUUUU.....ESSAS PORCARIAS SÓ ACONTECE NO BRASIL.... RAÇA MALDITA ESSES BRASILEIROS....EXTINÇAO
- Nordestino é uma desgraça cambada de demônio

Tarefa 3 - Detecção de comentários ofensivos direcionados

A detecção de comentários ofensivos direcionados é uma tarefa de CBT para identificar comentários tóxicos que foram direcionados ou não. Essa tarefa foi extraída do resultado da tarefa 3 descrita em 4.2.2. Caso a anotação tenha alguma das possíveis classes disponíveis na tarefa 3, o valor do campo **is_targeted** é “TIN”, caso o anotador não tenha selecionado nenhuma das possíveis classes, o valor no campo **is_targeted** é “UNT”. A Tabela 4.2 apresenta as possíveis classes, valores e descrições presentes no campo **is_targeted**.

Classe	Valor	Descrição
Targeted Insult	TIN	O comentário ofensivo é direcionado a um indivíduo, grupo ou outro.
Untargeted	UNT	O comentário ofensivo não é direcionado.

Tabela 4.2: Classes de comentário tóxico direcionado. Fonte: Elaborada pelo autor.

Tarefa 4 - Classificação do tipo de alvo do comentário tóxico direcionado

A classificação do tipo de alvo do comentário tóxico direcionado é uma CMCT em que apenas uma classe poderá ser atribuída a um comentário tóxico que tenha sido classificado como comentário tóxico direcionado. Essa classificação é opcional e por isso poderá ter valores nulos caso o comentário não seja classificado como direcionado na tarefa 4.2.2. A Tabela 4.3 apresenta as possíveis classes, seus valores e descrições presentes no campo **targeted_type**.

Classe	Valor	Descrição
cyberbullying *Individual*	IND	O alvo da ofensa é um indivíduo, geralmente definido como *cyberbullying*.
Group	GRP	O alvo da ofensa é um grupo de pessoas ou comunidade.
Other	OTH	O alvo da ofensa não pertence a nenhuma das duas classes anteriores. Por exemplo, uma empresa, um evento, um problema.

Tabela 4.3: Classes do tipo de alvo de comentário tóxico direcionado. Fonte: Elaborada pelo autor.

Tarefa 5 - Extração da parte tóxica do texto

A extração da parte tóxica do texto é uma tarefa de RS que busca identificar a(s) parte(s) do texto que fazem um texto tóxico. O anotador poderá selecionar a(s) parte(s) do texto assinalando-as caso existam. O campo **toxic_spans** possui uma lista com a posição dos caracteres tóxicos no texto presente no campo **text**. A Tabela 4.4 mostra um exemplo da anotação proposta.

4.2.3 Ferramenta de anotação

Utilizamos o Label Studio[11], uma ferramenta disponível no formato de Software Livre, para fornecer um ambiente em que os anotadores pudessem avaliar os textos e realizar as anotações propostas em 4.2.2. Essa ferramenta permite uma ampla customização

[11] https://labelstud.io/

Spans	**Comentário**
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25]	USER **Vaitomanocú vagabundo**
[47, 48, 49, 50, 51, 52, 53]	USER A fazenda, pq esse programa ainda existe? **Ódi-oooo**
[43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57]	USER Parabéns por expor tão claramente sua **insignificância**. Jamais será eleito.

Tabela 4.4: Exemplo de extração de partes tóxicas em textos tóxicos. Fonte: Elaborada pelo autor.

da tela de anotação, o que foi crucial para permitir o uso de perguntas agregadas afim de otimizar a atividade de anotação. Assim, todas as tarefas foram anotadas na mesma tela e com o menor número de perguntas possíveis, conforme é visto na Figura 4.3.

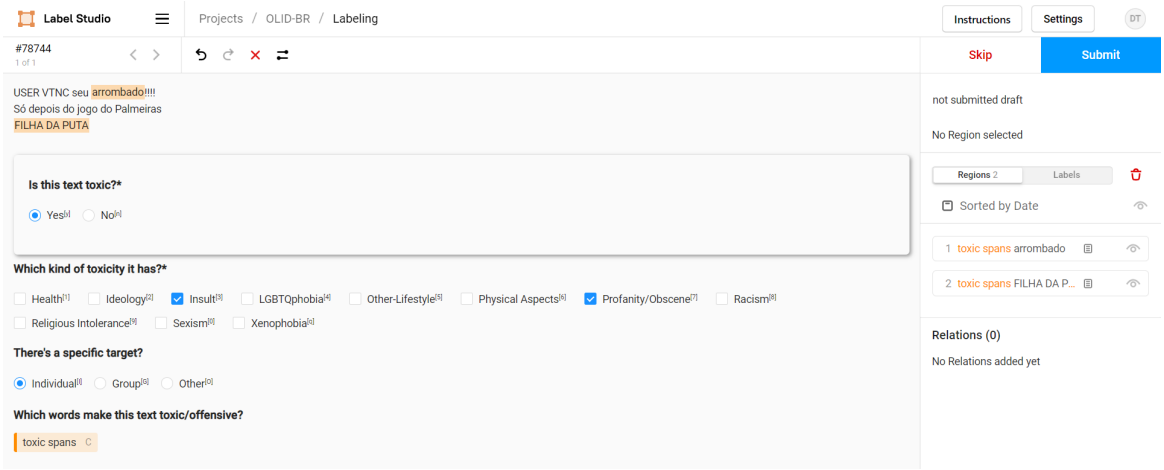

Figura 4.3: Tela da ferramenta de anotação. Fonte: Elaborada pelo autor.

A primeira pergunta (*"Is this text toxic?"*) possui como valor padrão *"Yes"*, pois todos os comentários foram previamente classificados como tóxico pela Perspective API[12], mas caso os anotadores não concordassem com essa classificação, era possível alterar para *"No"* e todas as outras perguntas eram ocultadas. Essa opção existe, pois durante o piloto do processo de anotação realizado pelo autor da pesquisa, foram identificados falso positivos na classificação automática da tarefa 1 (4.2.2). Na segunda pergunta (*"Which kind of toxicity does it have"*), os anotadores poderiam selecionar os rótulos de linguagem tóxica que se aplicam ao texto tóxico. Na terceira pergunta (*"Is there a specific target?"*), os anotadores poderiam marcar se o texto tóxico era direcionado a um indivíduo, grupo de pessoas ou outros. A quarta e última pergunta (*"Which words make this text toxic/offensive?"*) forneceu aos anotadores a possibilidade de assinalar a(s) parte(s) da sentença que ele considerava tóxica ou ofensiva.

[12] https://www.perspectiveapi.com/

Ainda na tela de anotação, havia dois botões onde o anotador poderia confirmar as anotações realizadas pressionando o botão *"Submit"* ou pular para o próximo texto utilizando o botão *"Skip"* caso ele não pudesse realizar a anotação de um determinado texto. Conforme explicado em 4.2.1, os anotadores receberam treinamento adequado para entender os conceitos e como utilizar a ferramenta corretamente. Os anotadores foram instruídos a pularem textos que não pudessem ser compreendidos sem o contexto suficiente ou que não se sentiam capazes de interpretar corretamente o significado do texto, os textos pulados por pelo menos um anotador não foram considerados no conjunto de dados descrito no Capítulo 4.

4.2.4 Concordância entre anotadores

A maioria dos trabalhos relacionados da área não apresentaram de forma detalhada os experimentos de confiabilidade entre anotadores, o que dificulta nossa capacidade de realizar comparações com os resultados dessa pesquisa. Zampieri et al. (2019) utilizou o Kappa de Fleiss em apenas 21 *tweets* classificados por cinco anotadores, e apenas para a tarefa binária (ofensivo ou não ofensivo), o valor foi de 0,83 (alta concordância). Fortuna et al. (2019) também utilizou o Kappa de Fleiss na avaliação da tarefa binária, o valor foi de 0,17 (concordância ruim) considerando todas as 5.668 mensagens rotuladas por três anotadores. Os autores sugerem que o uso de anotadores não especialistas foi o motivo da baixa concordância nessa tarefa, pois para a tarefa multirrótulo (rótulos de toxicidade), os anotadores especialistas categorizaram 500 mensagens, essas anotações foram avaliadas usando o Kappa de Cohen, que resultou em 0,72 (concordância substancial). de Pelle e Moreira (2017) calculou o Kappa de Fleiss para todas as 1.250 mensagens classificadas pelos três anotadores, o valor foi 0,71, considerado uma concordância substancial. Em nossa análise, Leite et al. (2020) demonstrou o estudo de confiabilidade entre avaliadores mais maduro, pois utilizou o alfa de Krippendorff para todos os 21.000 *tweets* com anotações de três diferentes anotadores, a média para todos os rótulos foi de 0,55 (concordância moderada), e os autores também disponibilizaram as anotações para cada instância, permitindo assim, a reprodutibilidade dos resultados. Em resumo, todos os trabalhos relacionados demonstraram que alcançar um bom acordo entre anotadores é um desafio nesta área.

Na construção do OLID-BR, cada texto foi anotado por três anotadores distintos que foram treinados e remunerados especificamente para realizar essa atividade. Ao final de cada iteração do processo, realizamos um experimento de confiabilidade entre anotadores para avaliar os resultados parciais e direcionar ações de melhoria no processo de anotação para a próxima iteração, desta forma, conseguimos realizar ajustes no processo com o objetivo de maximizar o entendimento dos anotadores do processo de anotação

adotado neste trabalho. Alguns anotadores foram substituídos durante o processo devido a indisponibilidade para continuar a atividade de anotação, ao total, 5 diferentes anotadores participaram desta atividade. Como visto em 2.7, utilizaremos o percentual de concordância, AC_1 de Gwet e o Alpha de Krippendorff para avaliar a concordância entre os anotadores.

Na **iteração 1**, ou piloto do processo de anotação, nosso objetivo era validar o processo de coleta, anotação dos dados e a ferramenta de anotação escolhida no projeto. Este anotador voluntário classificou 706 sentenças e forneceu sugestões de melhorias ao processo de anotação e diretrizes, o que foi fundamental para ajustar o processo para as iterações seguintes. Como houve algumas mudanças críticas ao processo, o autor do conjunto de dados revisou as anotações do voluntário e renomeou algumas categorias que sofreram alterações devido às mudanças sugeridas pelo anotador voluntário. Nesta iteração, não realizamos a análise de confiabilidade entre anotadores porque as mudanças no processo de anotação levaram a um conjunto diferente de anotações, o autor do conjunto de dados revisou todas as anotações e rotulou novamente algumas anotações considerando o processo de anotação atualizado. No processo de atribuição de rótulos, as anotações do autor prevaleceram sobre as anotações do voluntário.

Na **iteração 2**, com o processo de anotação revisado e validado, introduzimos três anotadores contratados que classificaram 2,996 sentenças. Os anotadores contratados foram treinados conforme descrito em 4.2.1 e receberam incentivo financeiro de acordo com a quantidade de anotações realizadas. Na análise de confiabilidade entre anotadores, identificamos que houve uma incorreta interpretação por um dos anotadores que atribuiu uma resposta na pergunta 3 para todos os textos, onde deveria ser respondida apenas se o comentário tóxico fosse direcionado, isto levou a uma alta discordância entre os anotadores, mas que não compromete significativamente a atribuição final dos rótulos, pois para os rótulos extraídos desta pergunta, utilizamos a estratégia de voto majoritário conforme visto em 4.2.5.

A Tabela 4.5 apresenta o resultado dos coeficientes para cada rótulo na iteração 2, conforme a escala de referência apresentada na Tabela 2.5, obtivemos uma concordância ruim em **is_targeted**, uma concordância leve em **is_offensive** e nos **rótulos de toxicidade** (*toxicity labels*), e uma concordância regular em **targeted_type** e **toxic_spans**.

Rótulo	**Percentual de concordância**	α	**Gwet's AC_1**
is_offensive (pergunta 1)	0,7277	0,0595	0,7750
Toxicity labels (pergunta 2)	0,1877	0,1962	N/A
is_targeted (pergunta 3)	0,1610	-0,1348	-0,1029
targeted_type (pergunta 3)	0,0641	0,2461	0,4978
toxic_spans (pergunta 4)	0,1220	0,2703	N/A

Tabela 4.5: Confiabilidade entre anotadores da iteração 2. Fonte: Elaborada pelo autor.

Na **iteração 3**, um dos anotadores contratados deixou o projeto e foi substituído por outro anotador que também foi treinado pelo autor do conjunto de dados. Após analisar os resultados do experimento de confiabilidade entre anotadores da iteração anterior, nossos anotadores classificaram mais 2,999 sentenças. A Tabela 4.6 apresenta os resultados dos coeficientes analisados na iteração 3, onde obtivemos uma concordância leve em **is_offensive** e **is_targeted**, e uma concordância moderada nos **rótulos de toxicidade** (*toxicity labels*), **targeted_type** e **toxic_spans**, resultados significativamente melhores em comparação com a iteração anterior.

Rótulo	**Percentual de concordância**	α	**Gwet's AC_1**
is_offensive (pergunta 1)	0,6509	0,1777	0,6754
Toxicity labels (pergunta 2)	0,2758	0,4653	N/A
is_targeted (pergunta 3)	0,3551	0,1072	0,6754
targeted_type (pergunta 3)	0,1975	0,4887	0,6300
toxic_spans (pergunta 4)	0,1757	0,4427	N/A

Tabela 4.6: Confiabilidade entre anotadores da iteração 3. Fonte: Elaborada pelo autor.

Na **iteração 4**, 2,013 comentários foram categorizados pelos três anotadores contratados, obtivemos uma concordância leve em **is_targeted**, uma concordância regular em **is_offensive** e uma concordância regular nos **rótulos de toxicidade** (**toxicity labels**), **targeted_type** e **toxic_spans**. Os resultados da iteração 4 foram similares aos resultados da iteração anterior conforme apresentado na Tabela 4.7.

Rótulo	**Percentual de concordância**	α	**Gwet's AC_1**
is_offensive (pergunta 1)	0,5847	0,2174	0,5716
Toxicity labels (pergunta 2)	0,2769	0,4424	N/A
is_targeted (pergunta 3)	0,4253	0,1825	0,2790
targeted_type (pergunta 3)	0,2223	0,4840	0,5756
toxic_spans (pergunta 4)	0,2249	0,4760	N/A

Tabela 4.7: Confiabilidade entre anotadores da iteração 4. Fonte: Elaborada pelo autor.

Após finalizada a atividade de anotação, os dados de todas as iterações foram consolidadas para gerar o conjunto de dados OLID-BR. Obtivemos uma concordância leve em **is_offensive** e **is_targeted**, e uma concordância moderada nos **rótulos de toxicidade** (**toxicity labels**), **targeted_type** e **toxic_spans** conforme mostra a Tabela 4.8.

Rótulo	Percentual de concordância	α	Gwet's AC_1
is_offensive (pergunta 1)	0,6641	0,1733	0,6929
Toxicity labels (pergunta 2)	0,2435	0,3648	N/A
is_targeted (pergunta 3)	0,3000	0,0355	0,0960
targeted_type (pergunta 3)	0,1505	0,4149	0,5689
toxic_spans (pergunta 4)	0,1679	0,3918	N/A

Tabela 4.8: Confiabilidade entre anotadores do OLID-BR. Fonte: Elaborada pelo autor.

4.2.5 Estratégias de atribuição de rótulos

A atribuição de rótulos é o processo em que múltiplas anotações são agregadas para gerar apenas uma anotação para uma tarefa no conjunto de dados. Podemos aplicar diferentes estratégias para agregar as anotações de acordo com a necessidade e o contexto. Se definirmos que uma instância será positiva somente se todos os anotadores concordarem com isso, podemos inserir um viés no modelo no sentido de que só pode prever o rótulo se for muito evidente. Esta opção pode ser útil para modelos que serão usados para tomar ações proibitivas sem a supervisão de um humano. Também podemos usar uma estratégia de voto majoritário, pois a escolha da maioria dos anotadores resulta no valor final do rótulo, gerando um modelo menos restritivo. Também podemos considerar uma instância como positiva se pelo menos um anotador classificou-o como positivo, essa estratégia pode ser útil para treinar modelos de suporte à moderação de conteúdo realizada por humanos (Leite et al., 2020). Neste trabalho, aplicamos diferentes estratégias na atribuição dos rótulos, de acordo com as características individuais de cada tarefa. Para os rótulos **is_offensive**, **is_targeted** e **targeted_type** consideramos o voto majoritário, onde a decisão da maioria dos anotadores resulta no valor final do rótulo, pois existe um melhor equilíbrio entre a quantidade de amostras em cada classe. Para o campo **toxic_spans** consideramos todos os caracteres selecionados pelos três anotadores. Por fim, os **rótulos de toxicidade** (*toxicity labels*) utilizaram a estratégia de ter pelo menos uma instância positiva para considerar o rótulo como positivo, visto que existe um grande desbalanceamento na quantidade de amostras positivas como é possível ver em 4.4.

4.3 Amostragem e formato dos dados

Com o conjunto de dados anotado e refinado, dividimos em três subconjuntos: dados de treino, dados de teste privado e dados de teste público. O objetivo desta divisão é disponibilizar o OLID-BR já distribuído entre treino e teste a fim de facilitar a comparação entre diferente soluções, pois estarão utilizando o mesmo conjunto de dados para treino

e para teste (avaliação). Os dados de teste privado não foram publicados, pois este subconjunto foi reservado para futuras competições acadêmicas. A divisão do conjunto de dados foi estratificada considerando todos os rótulos, exceto **toxic_spans**, para garantir que cada subconjunto de dados tenha uma distribuição semelhante de amostras de todo o conjunto de dados. A Tabela 4.9 apresenta a quantidade de amostras em cada subconjunto de dados e a proporção em relação ao total de amostras do conjunto de dados.

Subconjunto de dados	Quantidade de amostras (% do total)
Treino	4.765 (60%)
Teste privado	1.589 (20%)
Teste público	1.589 (20%)

Tabela 4.9: Quantidade de amostras em cada subconjunto de dados. Fonte: Elaborada pelo autor.

O conjunto de dados foi disponibilizado em dois formatos: *Comma-separated values (CSV)* e *JavaScript Object Notation (JSON)*. No formato CSV os rótulos foram definidos conforme as estratégias de atribuição de rótulos descritas em 4.2.5, no formato JSON, as amostras possuem o texto, metadados e todas as anotações realizadas pelos anotadores. Os metadados disponibilizados possuem informações adicionais sobre o contexto em que os comentários foram coletados como origem, horário de criação e coleta, grau de toxicidade (gerado pela Perspective API), e também dados sobre os anotadores como gênero, idade, tipo do anotador, formação técnica e escolaridade. Os metadados poderão ser úteis em futuras pesquisas para entender possíveis vieses em anotações realizadas por homens ou mulheres, por exemplo. Na opção CSV, o conjunto de dados contém três arquivos: *train.csv* (dados de treino), *test.csv* (dados de teste público) e *metadata.csv* (metadados para todos os textos nos dados de treino e teste público). Os arquivos *train.csv* e *test.csv* contêm as seguintes colunas: *id*, *text*, *is_offensive*, *is_targeted*, *targeted_type*, *toxic_spans*, *health*, *ideology*, *insult*, *lgbtqphobia*, *other_lifestyle*, *physical_aspects*, *profanity_obscene*, *racism*, *religious_intolerance*, *sexism* e *xenophobia*. O arquivo *metadata.csv* contém as seguintes colunas: *id*, *source*, *created_at*, *collected_at*, *toxicity_score*, *annotator_id*, *gender*, *age*, *education_level* e *annotator_type*.

Na opção JSON, o conjunto de dados contém dois arquivos: *train.json* (dados de treino) e *test.json* (dados de teste público). Cada arquivo contém uma lista de dicionários, onde cada dicionário é uma instância de dado usando o seguinte esquema:

```
{
  "id": "string",
  "text": "string",
  "metadata": {
    "source": "string",
    "created_at": "string",
    "collected_at": "string",
    "toxicity_score": "float"
  },
  "annotations": [
    {
      "annotator_id": "int",
      "is_offensive": "string",
      "is_targeted": "string",
      "targeted_type": "string",
      "toxic_spans": ["int"],
      "health": "bool",
      "ideology": "bool",
      "insult": "bool",
      "lgbtqphobia": "bool",
      "other_lifestyle": "bool",
      "physical_aspects": "bool",
      "racism": "bool",
      "religious_intolerance": "bool",
      "sexism": "bool",
      "xenophobia": "bool"
    },
    "..."
  ]
}
```


Adicionalmente, publicamos um arquivo chamado *additional_data.json* contendo 7.184 comentários com anotações incompletas (ou seja, menos de 3 anotadores) que não foi usado no conjunto de dados, mas que podem ser úteis para aumentar o número de amostras dos dados de treino. Estes dados requerem uma análise prévia, pois não foram validados no experimento de concordância entre os anotadores.

4.4 Análise dos dados

Como visto em 4.3, o OLID-BR é dividido entre dados de treino, dados de teste público e dados de teste privado. Nesta seção, iremos analisar os dados de treino para entender as características do conjunto de dados, como os conjuntos de dados são estratificados, é esperado que a distribuição de rótulos de cada subconjunto seja similar a distribuição de rótulos do conjunto de dados inteiro. Os dados de treino possuem 4.765 amostras com 15 rótulos disponíveis. 11 rótulos são binários (rótulos de toxicidade) e 4 rótulos são categóricos (**is_offensive**, **is_targeted**, **targeted_type** e **toxic_spans**). A maioria dos rótulos do OLID-BR são desbalanceados, como podemos ver a seguir.

O campo **is_offensive** contém 4.292 amostras com “OFF” (*offensive*) e 473 amostras para “NOT” (*not offensive*). A Figura 4.4 mostra a distribuição entre comentários tóxicos e não tóxicos.

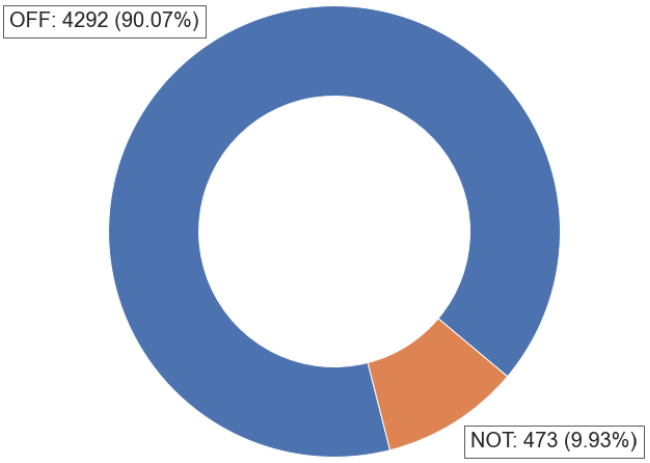

Figura 4.4: Distribuição entre comentários tóxicos e não tóxicos. Fonte: Elaborada pelo autor.

O campo **is_targeted** contém 2.982 amostras de “TIN” (*targeted insult*) e 1.783 amostras de “UNT” (*untargeted*). A Figura 4.5 mostra a distribuição entre comentários tóxicos direcionados e comentários tóxicos não direcionados.

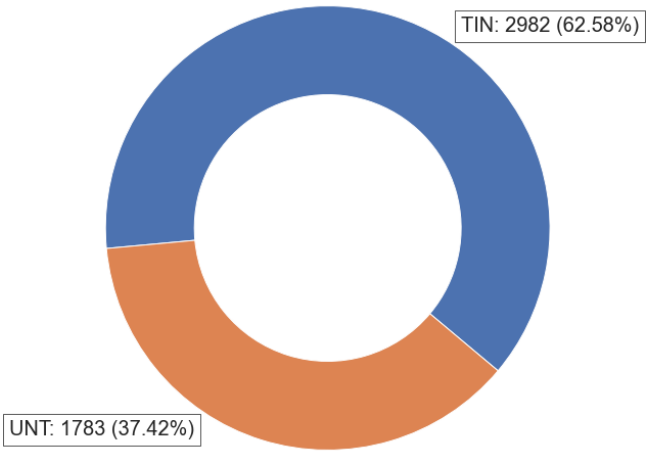

Figura 4.5: Distribuição de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

O campo **targeted_type** contém 1.753 amostras de “IND” (*individual*), 745 amostras de “GRP” (*group*) e 339 amostras de “OTH” (*other*). A Figura 4.6 mostra a distribuição entre os tipos de alvo dos comentários tóxicos direcionados.

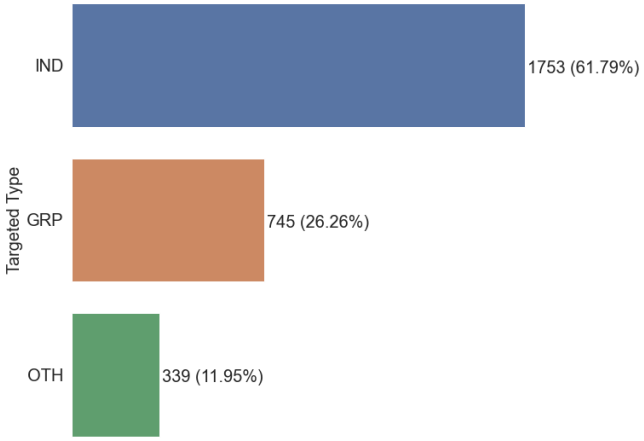

Figura 4.6: Distribuição dos tipos de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

O campo **toxic_spans** está presente em 3.930 amostras, o que representa mais de 80% do conjunto de dados. A Figura 4.7 mostra a distribuição entre os comentários que possuem **toxic_spans** e os que não possuem.

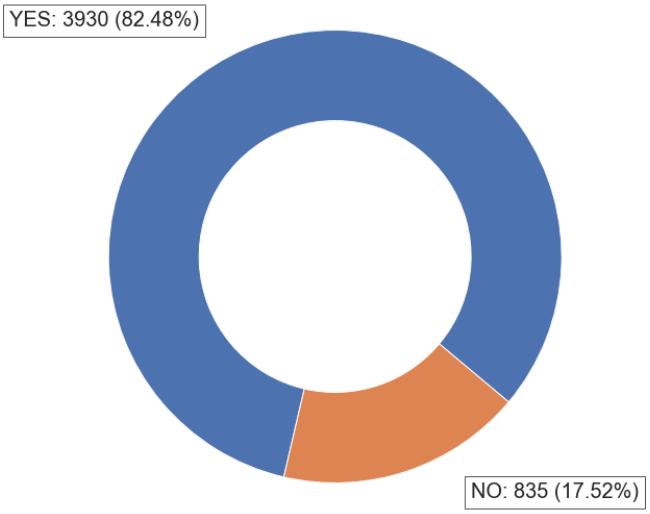

Figura 4.7: Distribuição de **toxic_spans**. Fonte: Elaborada pelo autor.

A Figura 4.8 apresenta uma nuvem de palavras que ilustra as palavras mais frequentes mapeadas pelo campo **toxic_spans**.

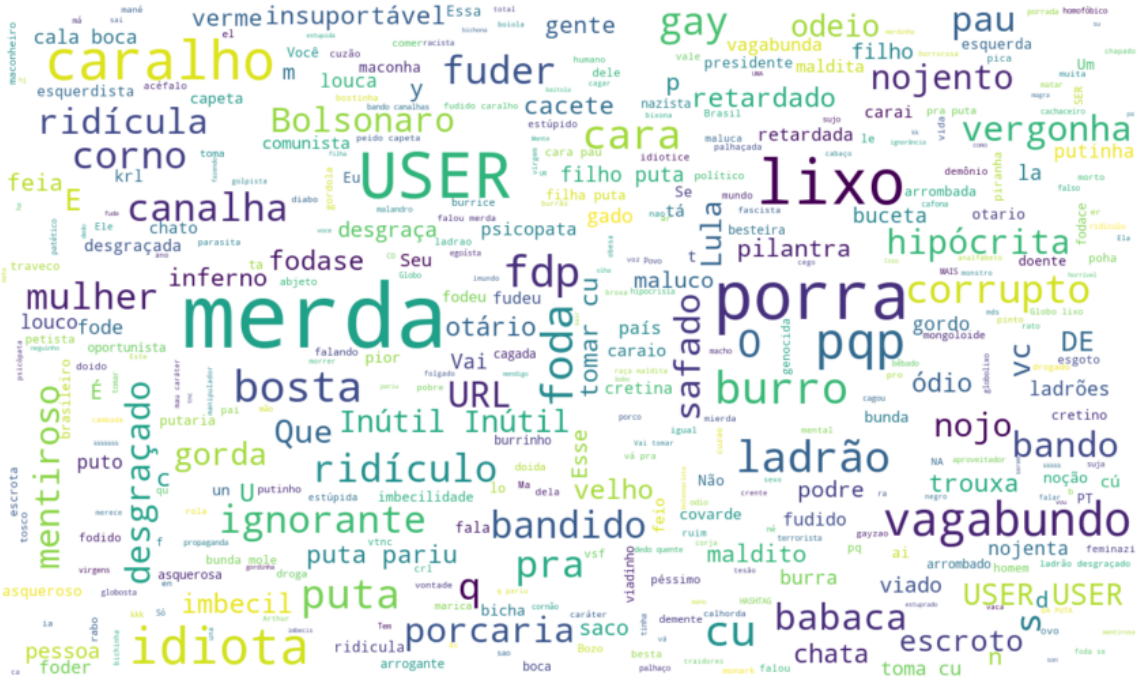

Figura 4.8: Nuvem de palavras do campo **toxic_spans**. Fonte: Elaborada pelo autor.

Os rótulos de toxicidade são os rótulos mais desbalanceados do OLID-BR. Os campos **health**, **lgbtqphobia**, **other_lifestyle**, **physical_aspects**, **racism**, **sexism**, **religious_intolerance** e **xenophobia** são os rótulos com a menor proporção de amostras positivas. Os campos **ideology**, **insult** e **profanity_obscene** são levemente desbalanceados. A Figura 4.9 apresenta a distribuição de amostras nos rótulos de toxicidade.

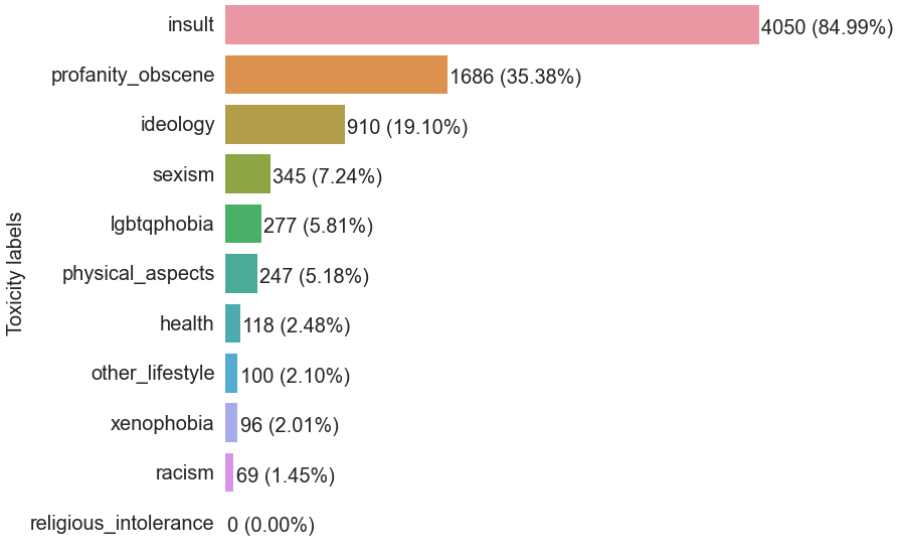

Figura 4.9: Distribuição dos rótulos de toxicidade. Fonte: Elaborada pelo autor.

A Figura 4.10 apresenta uma matriz de correlação entre os rótulos de toxicidade. Podemos observar que o campo **health** está correlacionado com o campo **physical_aspects**, sugerindo que a linguagem tóxica relacionada à aparência física e à saúde tendem a serem usadas em conjunto nos textos analisados. Também observamos que o campo **insult** está ligeiramente correlacionado com os campos **profanity_obscene**, **ideology** e **sexism**, sugerindo que insultos relacionados a ideologias ou que são sexistas apresentam palavras profanas ou vulgares com maior frequência do que outros tipos de linguagem tóxica.

	health	ideology	insult	lgbtqphobia	other_lifestyle	physical_aspects	profanity_obscene	racism	sexism	xenophobia
health	1.00	0.01	0.08	-0.03	0.03	0.33	0.01	0.00	0.01	-0.00
ideology	0.01	1.00	0.21	-0.03	0.00	-0.04	-0.09	0.00	-0.04	0.11
insult	0.08	0.21	1.00	0.05	0.04	0.09	0.14	0.04	0.10	0.05
lgbtqphobia	-0.03	-0.03	0.05	1.00	0.05	-0.01	-0.01	0.05	0.02	0.01
other_lifestyle	0.03	0.00	0.04	0.05	1.00	0.04	-0.01	0.01	-0.02	-0.02
physical_aspects	0.33	-0.04	0.09	-0.01	0.04	1.00	0.00	0.07	0.06	0.00
profanity_obscene	0.01	-0.09	0.14	-0.01	-0.01	0.00	1.00	0.00	0.08	-0.04
racism	0.00	0.00	0.04	0.05	0.01	0.07	0.00	1.00	0.03	0.05
sexism	0.01	-0.04	0.10	0.02	-0.02	0.06	0.08	0.03	1.00	0.01
xenophobia	-0.00	0.11	0.05	0.01	-0.02	0.00	-0.04	0.05	0.01	1.00

Figura 4.10: Matriz de correlação entre os rótulos de toxicidade. Fonte: Elaborada pelo autor.

5. SISTEMA PARA DETECÇÃO DE LINGUAGEM TÓXICA

Este capítulo apresenta o sistema para detecção de linguagem tóxica que atende aos objetivos de pesquisa definidos no Capítulo 1. A arquitetura apresenta os componentes do sistema e como eles se relacionam, também mostra o fluxo de processamento dos dados realizado pelo sistema. O sistema proposto recebe como entrada (*input*) um determinado texto e a primeira etapa do sistema é realizada pelo classificador de comentários tóxicos (*Toxic Comments Classifier*) descrito na Seção 5.1, caso o comentário seja classificado como tóxico, ele então é enviado paralelamente para outros três modelos, o classificador dos tipos de linguagem tóxica (*Toxicity Type Classifier*) descrito na Seção 5.2, que identificar os rótulos aplicáveis ao comentário tóxico, classificador de comentários tóxicos direcionados (*Toxicity Target Clasifier*) descrito na Seção 5.3, que identifica quais comentários tóxicos são direcionados a alguém ou algo, caso o comentário tóxico seja direcionado uma etapa adicional é realizada pelo classificador do tipo de comentário tóxico direcionado (*Toxicity Target Type Classifier*) descrito na Seção 5.4, que identifica o tipo de alvo do ataque. A última etapa executada em paralelo é realizada pelo Detector das partes tóxicas do texto (*Toxic Spans Detector*) descrito na Seção 5.5, que identifica as partes do texto consideradas tóxicas. A saída (*output*) do sistema possui todas as predições elegíveis para um determinado texto de entrada. A Figura 5.1 ilustra o funcionamento do sistema com seus respectivos componentes.

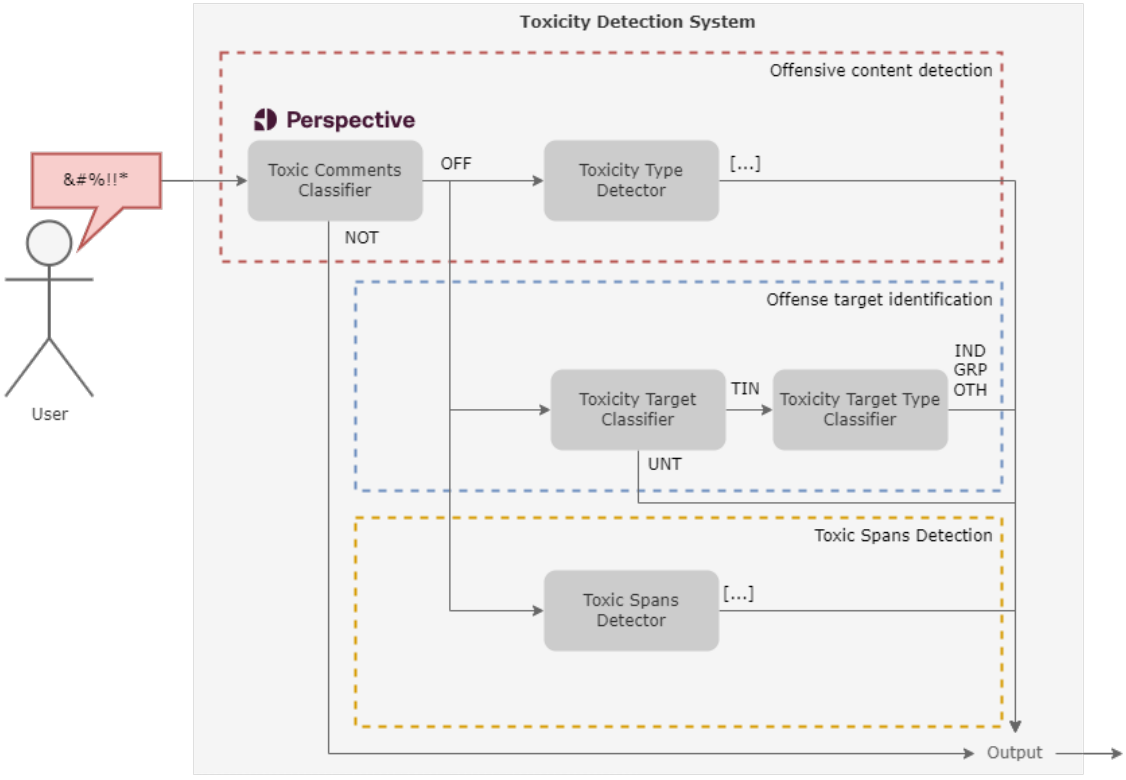

Figura 5.1: Sistema de detecção de linguagem tóxica. Fonte: Elaborada pelo autor.

Conforme visto na Seção 2.4, os algoritmos de aprendizado profundo são hoje o estado da arte nas mais variadas áreas de IA, incluindo PLN. Especificamente, a arquitetura Transformers vem apresentando grandes avanços em comparação com outras arquiteturas como RNN e LSTM, por isso, utilizaremos a arquitetura Transformers para gerar a maioria dos modelos propostos nessa abordagem, com algumas adaptações conforme a tarefa em específico. O detector das partes tóxicas do texto utiliza um modelo de reconhecimento de entidades adaptado da biblioteca de código aberto SpaCy (Honnibal e Montani, 2022). O conjunto de dados utilizado no treinamento dos modelos proposto foi desenvolvimento como parte desta pesquisa e é detalhado no Capítulo 4. Cada tarefa proposta exige uma preparação de dados específica para atender aos objetivos e é abordado nas seções a seguir.

5.1 Classificador de comentários tóxicos

O classificador de comentários tóxicos é responsável por identificar quais comentários são tóxicos e quais não são tóxicos. Para isso, o modelo fornece uma resposta binária, sendo o comentário tóxico identificado como “OFF” (*offensive*) e o comentário não tóxico identificado como “NOT” (*not-offensive*).

5.2 Classificador do tipo de linguagem tóxica

O classificador do tipo de linguagem tóxica é uma tarefa de CMRT responsável por detectar todos os rótulos de linguagem tóxica que são aplicáveis a um determinado comentário tóxico. Os rótulos que poderão ser detectados são: *health*, *ideology*, *insult*, *lgbtqphobia*, *other_lifestyle*, *physical_aspects*, *profanity_obscene*, *racism*, *religious_intolerance*, *sexism* e *xenophobia*. O classificador pode atribuir nenhum ou todos os rótulos disponíveis a um determinado comentário tóxico.

5.3 Classificador de comentários tóxicos direcionados

O classificador de comentários tóxicos direcionados é responsável por detectar quais dos comentários tóxicos são direcionados a um alvo específico. O resultado do modelo é uma decisão binária, sendo o comentário tóxico identificado como não direcionado ou direcionado.

5.4 Classificador do tipo de comentário tóxico direcionado

O classificador do tipo de comentário tóxico direcionado atua em todos os comentários tóxicos que foram identificados como direcionados. Este classificador realiza uma Classificação Multiclasse de Texto com o objetivo de atribuir uma das classes *Individual*, *Group* ou *Other*, a um determinado comentário tóxico direcionado.

5.5 Detector das partes tóxicas do texto

O Detector das partes tóxicas do texto é responsável por identificar as parte(s) do texto que foram detectadas como tóxicas. Essa informação é útil no processo de moderação de conteúdo impróprio, pois possibilita realçar as parte(s) tóxica(s) do texto para facilitar a moderação de conteúdo impróprio ou destacar aos usuários que parte(s) dos seus textos podem ser tóxicas, sugerindo a reescrita.

6. EXPERIMENTOS

Neste capítulo, relatamos os experimentos realizados para avaliar a aplicabilidade do conjunto de dados OLID-BR e treinamento dos modelos para as tarefas descritas no Capítulo 5.

Os experimentos compartilham requisitos em comum que visam garantir a validade dos resultados. Os dados de treinamento serão divididos em conjuntos de treinamento e validação. O conjunto de treinamento foi usado para treinar os modelos e o conjunto de validação foi usado para otimizar os hiperparâmetros. O conjunto de teste foi usado para avaliar o melhor modelo treinado com os hiperparâmetros otimizados. Técnicas para lidar com dados desbalanceados serão aplicadas para garantir que os modelos aprendam a classificar adequadamente as classes minoritárias. Os experimentos foram divididos em três etapas: preparação dos dados, otimização dos hiperparâmetros, treinamento e avaliação do melhor modelo. Nas seções a seguir, descrevemos os experimentos realizados para cada tarefa.

6.1 Classificador de comentários tóxicos

Como o foco do conjunto de dados construído no trabalho foi nos comentários tóxicos, foi necessário coletar comentários não tóxicos para balancear a distribuição das classes “OFF” (*offensive*) e “NOT” (*non-offensive*), para isso, coletamos comentários não ofensivos dos conjuntos de dados relacionados descritos em 4.1.1. O conjunto de treinamento, validação e teste possui 9.006, 2.252 e 3.213 instâncias, respectivamente.

O modelo BERT (*Bidirectional Encoder Representations from Transformers*) pré-treinado com textos em Português disponibilizado em Souza et al. (2020) foi utilizado através da biblioteca Transformers (Wolf et al., 2020) contendo 12 camadas e 110 milhões de parâmetros. Para fornecer a probabilidade de pertencimento de cada classe, adicionamos a função de ativação *Softmax* na saída do modelo. Além disso, adaptamos a função de perda para incluir pesos calculados de acordo com a incidência de instâncias positivas da classe **is_offensive**, a fim de mitigar o desequilíbrio da classe. Para identificar os hiperparâmetros apropriados para o modelo, realizamos uma otimização Bayesiana, utilizando a *F-Measure* ponderada (*weighted*) como métrica-alvo e considerando os seguintes intervalos de hiperparâmetros:

- *learning rate*: entre 0,00001 e 0,001;
- *weight decay*: entre 0,0 e 0,1;
- *adam beta1*: entre 0,8 e 0,999;

- *adam beta2*: entre 0,8 e 0,999;
- *adam epsilon*: entre 0,00000001 e 0,000001;
- *label smoothing factor*: entre 0,0 e 0,1;
- *optimizer*: “adamw_hf”, “adamw_torch”, “adamw_apex_fused” ou “adafactor”.

Além dos hiperparâmetros mencionados anteriormente, estabelecemos alguns outros de forma estática. O tamanho do lote (*batch size*) foi estabelecido como 8 e a quantidade de épocas (num_train_epochs) foi definida como 30. Utilizamos uma política de parada precoce (early stopping) de 2 épocas sem melhora na métrica *F-Measure* ponderada avaliada no conjunto de validação. A Tabela 6.7 apresenta o resultado da *F-Measure* ponderada e a duração de cada um dos 18 treinamentos no trabalho de ajuste de hiperparâmetros realizado.

Nome do trabalho	*F-Measure* ponderada	Duração do treinamento
pytorch-training-230202-1408-001-3a7ba5a5	0,868192315	28 minutos
pytorch-training-230202-1408-010-fa54f873	0,865946949	48 minutos
pytorch-training-230202-1408-011-51c0a2c0	0,865673721	56 minutos
pytorch-training-230202-1408-008-38cd8920	0,864526451	26 minutos
pytorch-training-230202-1408-015-edf8d39c	0,863599539	27 minutos
pytorch-training-230202-1408-018-9deed799	0,861810446	37 minutos
pytorch-training-230202-1408-012-566ef2d0	0,860839069	32 minutos
pytorch-training-230202-1408-014-ed39b6e7	0,860670984	24 minutos
pytorch-training-230202-1408-009-140c0893	0,860427082	32 minutos
pytorch-training-230202-1408-005-777c7510	0,859007239	53 minutos
pytorch-training-230202-1408-017-da98cd69	0,857631683	37 minutos
pytorch-training-230202-1408-016-b60a8aa2	0,85553056	27 minutos
pytorch-training-230202-1408-003-421033a5	0,722851872	25 minutos
pytorch-training-230202-1408-004-9bf72426	0,473337054	23 minutos
pytorch-training-230202-1408-002-5ee87efd	0,473337054	25 minutos
pytorch-training-230202-1408-007-6ead9eae	0,473337054	26 minutos
pytorch-training-230202-1408-013-6a315783	0,473337054	26 minutos
pytorch-training-230202-1408-006-fb80af01	0,473337054	21 minutos

Tabela 6.1: Resultados dos treinamentos do trabalho de ajuste de hiperparâmetros do Classificador de comentários tóxicos. Fonte: Elaborada pelo autor.

O treinamento que obteve melhor desempenho no trabalho de ajuste de hiperparâmetros utilizou os seguintes hiperparâmetros:

- *learning rate*: 3,255788747459486e-05;
- *weight decay*: 0,031031065174245122;
- *adam beta1*: 0,8445637934160373;
- *adam beta2*: 0,8338816842140165;
- *adam epsilon*: 2,527092625455385e-08;
- *label smoothing factor*: 0,07158711257743958;
- *optimizer*: “adamw_hf”.

O modelo final foi treinado utilizando os dados dos conjuntos de treinamento e validação juntos, e utilizando os hiperparâmetros acima. Após 19 épocas de treinamento, a política de parada precoce foi acionada e o modelo foi salvo. Para avaliar o modelo foi utilizado o conjunto de teste. A Tabela 6.2 apresenta os resultados gerais das métricas do modelo final para o classificador de comentários tóxicos direcionados.

Precisão (ponderada)	85,67%
Abrangência (ponderada)	85,68%
***F-Measure* (ponderada)**	85,68%

Tabela 6.2: Resultados gerais obtidos no experimento do Classificador de comentários tóxicos. Fonte: Elaborada pelo autor.

A Figura 6.1 apresenta as métricas de avaliação em cada época de treinamento.

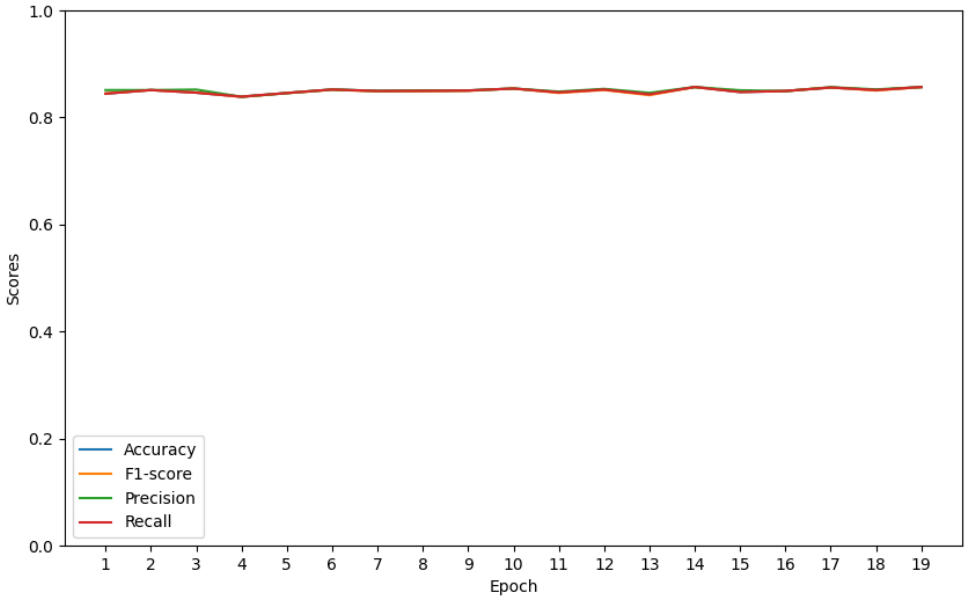

Figura 6.1: Avaliação do classificador de comentários tóxicos em cada época de treinamento. Fonte: Elaborada pelo autor.

A Tabela 6.3 apresenta os resultados das métricas para cada classe do classificador de comentários tóxicos direcionados.

Classe	Precisão	Abrangência	*F-Measure*	Exemplos
OFF (*offensive*)	84,29%	83,59%	83,94%	1.438
NOT (*non-offensive*)	86,79%	87,38%	87,09%	1.775

Tabela 6.3: Resultados obtidos por classe no experimento do classificador de comentários tóxicos. Fonte: Elaborada pelo autor.

O modelo demonstrou bons resultados para ambas as classes, com uma *F-Measure* ponderada de 85,68%, vale destacar que a tarefa de classificação de comentários tóxicos é uma tarefa mais simples de ser realizada, visto que os comentários tóxicos apresentaram características únicas o que permitem a identificação dos padrões destes comentários.

6.2 Classificador dos tipos de linguagem tóxica

Para treinar o classificador dos tipos de linguagem tóxica os dados foram filtrados para incluir apenas os comentários tóxicos (“OFF”) e que possuem pelo menos uma instância positiva dos rótulos de toxicidade (*health*, *ideology*, *insult*, *lgbtqphobia*, *other_lifestyle*, *physical_aspects*, *profanity_obscene*, *racism*, *sexism* e *xenophobia*), o rótulo *religious_intolerance* foi descartado, pois possui apenas uma instância positiva, invia-

bilizando o treinamento deste rótulo. O conjunto de treinamento, validação e teste possui 3.417, 855 e 1.438 instâncias, respectivamente.

O modelo BERT (*Bidirectional Encoder Representations from Transformers*) pré-treinado com textos em Português disponibilizado em Souza et al. (2020) foi utilizado através da biblioteca Transformers (Wolf et al., 2020) contendo 12 camadas e 110 milhões de parâmetros, adicionamos a função de ativação *Sigmoid* para fornecer a probabilidade individual para cada rótulo de toxicidade. Além disso, adaptamos a função de perda para incluir pesos (*weights*) calculados de acordo com a incidência de instâncias positivas dos rótulos de toxicidade, a fim de mitigar o desequilíbrio entre os rótulos. Para identificar os hiperparâmetros apropriados para o modelo, realizamos uma otimização Bayesiana, utilizando a *F-Measure* ponderada (*weighted*) como métrica-alvo e considerando os seguintes intervalos de hiperparâmetros:

- *learning rate*: entre 0,00001 e 0,001;
- *weight decay*: entre 0,0 e 0,1;
- *adam beta1*: entre 0,8 e 0,999;
- *adam beta2*: entre 0,8 e 0,999;
- *adam epsilon*: entre 0,00000001 e 0,000001;
- *optimizer*: “adamw_hf”, “adamw_torch”, “adamw_apex_fused” ou “adafactor”.

Além dos hiperparâmetros mencionados anteriormente, estabelecemos alguns outros de forma estática. O tamanho do lote (*batch size*) foi estabelecido como 8 e a quantidade de épocas (num_train_epochs) foi definida como 10. Utilizamos uma política de parada precoce (early stopping) de 2 épocas sem melhora na métrica *F-Measure* ponderada avaliada no conjunto de validação. A Tabela 6.4 apresenta o resultado da *F-Measure* ponderada e a duração de cada um dos 18 treinamentos no trabalho de ajuste de hiperparâmetros realizado.

Nome do trabalho	*F-Measure* ponderada	Duração do treinamento
pytorch-training-221118-2303-001-2c25be0c	0.7591	47 minutos
pytorch-training-221118-2303-006-49d87b23	0.7587	1 hora e 2 minutos
pytorch-training-221118-2303-007-1182e099	0.7258	23 minutos
pytorch-training-221118-2303-008-10c5fe8b	0.7194	25 minutos
pytorch-training-221118-2303-005-f1adc05a	0.7187	23 minutos
pytorch-training-221118-2303-004-1fdab0a9	0.7142	24 minutos
pytorch-training-221118-2303-002-c40b0d07	0.7080	23 minutos
pytorch-training-221118-2303-009-5743f9ce	0.7095	23 minutos
pytorch-training-221118-2303-012-c98d6e1c	0.6910	23 minutos
pytorch-training-221118-2303-011-cd4e57b8	0.6754	44 minutos
pytorch-training-221118-2303-003-ee3d2e41	0.6754	40 minutos
pytorch-training-221118-2303-015-f432d514	0.6718	41 minutos
pytorch-training-221118-2303-014-677d0d92	0.6718	24 minutos
pytorch-training-221118-2303-018-c70bac75	0.6653	23 minutos
pytorch-training-221118-2303-010-7c1a0ed4	0.5460	24 minutos
pytorch-training-221118-2303-017-ec8b49e7	0.5442	24 minutos
pytorch-training-221118-2303-016-f1f0f0a2	0.5442	23 minutos
pytorch-training-221118-2303-013-8440bbfd	0.5442	24 minutos

Tabela 6.4: Resultados dos treinamentos do trabalho de ajuste de hiperparâmetros do classificador dos tipos de linguagem tóxica. Fonte: Elaborada pelo autor.

O treinamento que obteve melhor desempenho no trabalho de ajuste de hiperparâmetros utilizou os seguintes hiperparâmetros:

- *learning rate*: 7,044186985160909e-05;
- *weight decay*: 0,02426675806866223;
- *adam beta1*: 0,9339215524915885;
- *adam beta2*: 0,9916979096990963;
- *adam epsilon*: 3,4435900142455904e-07;
- *optimizer*: “adamw_apex_fused”.

O modelo final foi treinado utilizando os dados dos conjuntos de treinamento e validação juntos, e utilizando os hiperparâmetros acima. Após 7 épocas de treinamento, a política de parada precoce foi acionada e o modelo foi salvo. Para avaliar o modelo foi utilizado o conjunto de teste. A Tabela 6.5 apresenta os resultados gerais das métricas do modelo final para o classificador dos tipos de linguagem tóxica.

Precisão (ponderada)	81,27%
Abrangência (ponderada)	67,10%
***F-Measure* (ponderada)**	72,50%

Tabela 6.5: Resultados gerais obtidos no experimento do Classificador dos tipos de linguagem tóxica. Fonte: Elaborada pelo autor.

A Figura 6.2 apresenta as métricas de avaliação em cada época de treinamento.

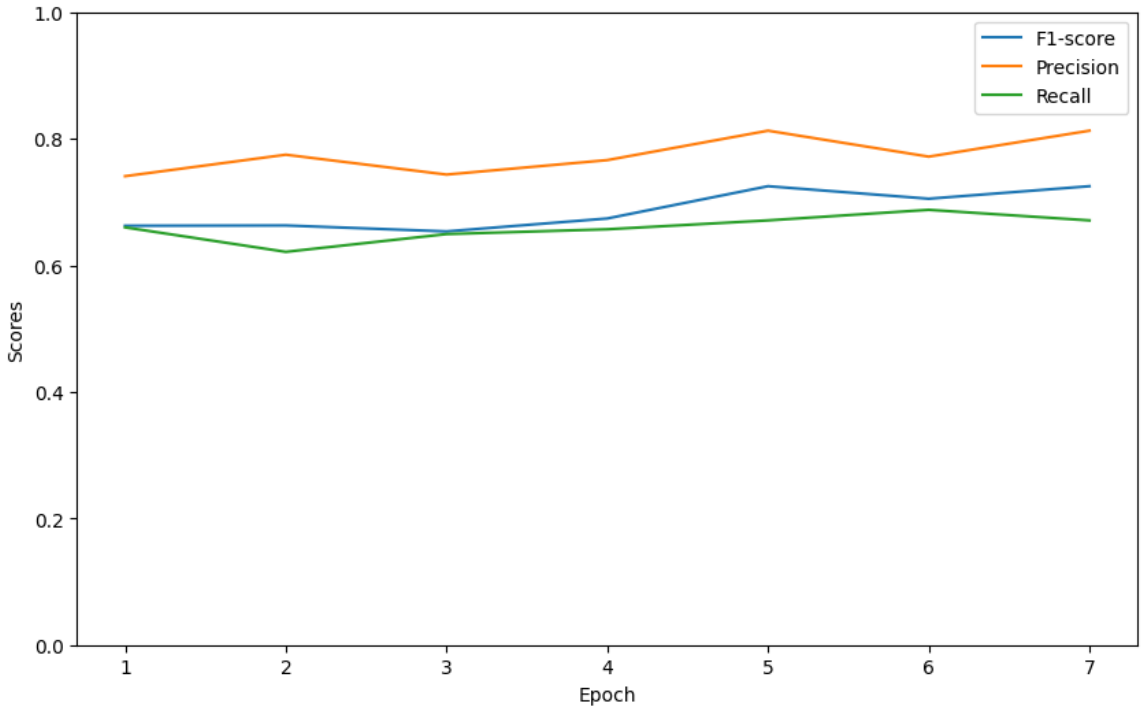

Figura 6.2: Avaliação do classificador dos tipos de linguagem tóxica em cada época de treinamento. Fonte: Elaborada pelo autor.

A Tabela 6.6 apresenta os resultados das métricas para cada rótulo do classificador dos tipos de linguagem tóxica.

Rótulo	Precisão	Abrangência	*F-Measure*	Exemplos
health	24,19%	38,46%	29,70%	39
ideology	73,90%	66,12%	69,79%	304
insult	98,60%	67,65%	80,25%	1351
lgbtqphobia	66,34%	72,83%	69,43%	92
other_lifestyle	40,00%	35,29%	35,70%	34
physical_aspects	42,86%	46,99%	44,83%	83
profanity_obscene	71,24%	76,69%	73,86%	562
racism	45,45%	43,48%	44,44%	23
sexism	34,38%	57,39%	43,00%	115
xenophobia	46,43%	40,62%	43,33%	32

Tabela 6.6: Resultados obtidos por rótulo no experimento do Classificador dos tipos de linguagem tóxica. Fonte: Elaborada pelo autor.

É possível perceber que o modelo teve dificuldades em prever rótulos com poucas amostras como *health* e *other_lifestyle*, porém, no geral, o modelo obteve um bom resultado na maioria dos rótulos. Como oportunidade futura, podemos buscar amostras dos rótulos com poucos exemplos no conjunto de dados adicionais fornecidos pelo OLID-BR, para aumentar a quantidade de exemplos e melhorar o desempenho do modelo.

6.3 Classificador de comentários tóxicos direcionados

Para treinar o classificador de comentários tóxicos direcionados os dados foram filtrados para incluir apenas os comentários tóxicos (“OFF”). O conjunto de treinamento, validação e teste possui 3.433, 859 e 1.438 instâncias, respectivamente.

O modelo BERT (*Bidirectional Encoder Representations from Transformers*) pré-treinado com textos em Português disponibilizado em Souza et al. (2020) foi utilizado através da biblioteca Transformers (Wolf et al., 2020) contendo 12 camadas e 110 milhões de parâmetros. Para fornecer a probabilidade de pertencimento de cada classe, adicionamos a função de ativação *Softmax* na saída do modelo. Além disso, adaptamos a função de perda para incluir pesos calculados de acordo com a incidência de instâncias positivas da classe **is_targeted**, a fim de mitigar o desequilíbrio da classe. Para identificar os hiperparâmetros apropriados para o modelo, realizamos uma otimização Bayesiana, utilizando a *F-Measure* ponderada (*weighted*) como métrica-alvo e considerando os seguintes intervalos de hiperparâmetros:

- *learning rate*: entre 0,00001 e 0,001;
- *weight decay*: entre 0,0 e 0,1;
- *adam beta1*: entre 0,8 e 0,999;
- *adam beta2*: entre 0,8 e 0,999;
- *adam epsilon*: entre 0,00000001 e 0,000001;
- *label smoothing factor*: entre 0,0 e 0,1;
- *optimizer*: “adamw_hf”, “adamw_torch”, “adamw_apex_fused” ou “adafactor”.

Além dos hiperparâmetros mencionados anteriormente, estabelecemos alguns outros de forma estática. O tamanho do lote (*batch size*) foi estabelecido como 8 e a quantidade de épocas (num_train_epochs) foi definida como 10. Utilizamos uma política de parada precoce (early stopping) de 2 épocas sem melhora na métrica *F-Measure* ponderada avaliada no conjunto de validação. A Tabela 6.7 apresenta o resultado da *F-Measure* ponderada e a duração de cada um dos 18 treinamentos no trabalho de ajuste de hiperparâmetros realizado.

Nome do trabalho	*F-Measure* ponderada	Duração do treinamento
pytorch-training-221118-2218-012-846ea4ca	0.6707	16 minutos
pytorch-training-221118-2218-008-246a6202	0.6705	10 minutos
pytorch-training-221118-2218-017-723e34df	0.6671	12 minutos
pytorch-training-221118-2218-016-237e4be9	0.6668	12 minutos
pytorch-training-221118-2218-007-fac3e712	0.6661	16 minutos
pytorch-training-221118-2218-005-d190f55d	0.6614	11 minutos
pytorch-training-221118-2218-015-8b41f2f4	0.6600	14 minutos
pytorch-training-221118-2218-014-858a08e5	0.6594	15 minutos
pytorch-training-221118-2218-011-e411da32	0.6592	19 minutos
pytorch-training-221118-2218-018-bfc9d8ec	0.6583	14 minutos
pytorch-training-221118-2218-003-1e86fd67	0.6581	21 minutos
pytorch-training-221118-2218-006-d659934d	0.6537	13 minutos
pytorch-training-221118-2218-002-64102474	0.6537	14 minutos
pytorch-training-221118-2218-009-8f234a8e	0.5549	11 minutos
pytorch-training-221118-2218-013-3d3a3fd3	0.5488	10 minutos
pytorch-training-221118-2218-010-ae71a96a	0.5488	11 minutos
pytorch-training-221118-2218-004-0c5f857c	0.5488	11 minutos
pytorch-training-221118-2218-001-3befd2ce	0.5488	11 minutos

Tabela 6.7: Resultados dos treinamentos do trabalho de ajuste de hiperparâmetros do classificador de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

O treinamento que obteve melhor desempenho no trabalho de ajuste de hiperparâmetros utilizou os seguintes hiperparâmetros:

- *learning rate*: 4,174021560583183e-05;
- *weight decay*: 0,05595810634526813;
- *adam beta1*: 0,9360294728287728;
- *adam beta2*: 0,9974781444436187;
- *adam epsilon*: 8,016624612627008e-07;
- *label smoothing factor*: 0,09936835309930625;
- *optimizer*: “adamw_hf”.

O modelo final foi treinado utilizando os dados dos conjuntos de treinamento e validação juntos, e utilizando os hiperparâmetros acima. Após 6 épocas de treinamento, a política de parada precoce foi acionada e o modelo foi salvo. Para avaliar o modelo foi utilizado o conjunto de teste. A Tabela 6.8 apresenta os resultados gerais das métricas do modelo final para o classificador de comentários tóxicos direcionados.

Precisão (ponderada)	67,44%
Abrangência (ponderada)	70,03%
***F-Measure* (ponderada)**	67,67%

Tabela 6.8: Resultados gerais obtidos no experimento do Classificador de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

A Figura 6.3 apresenta as métricas de avaliação em cada época de treinamento.

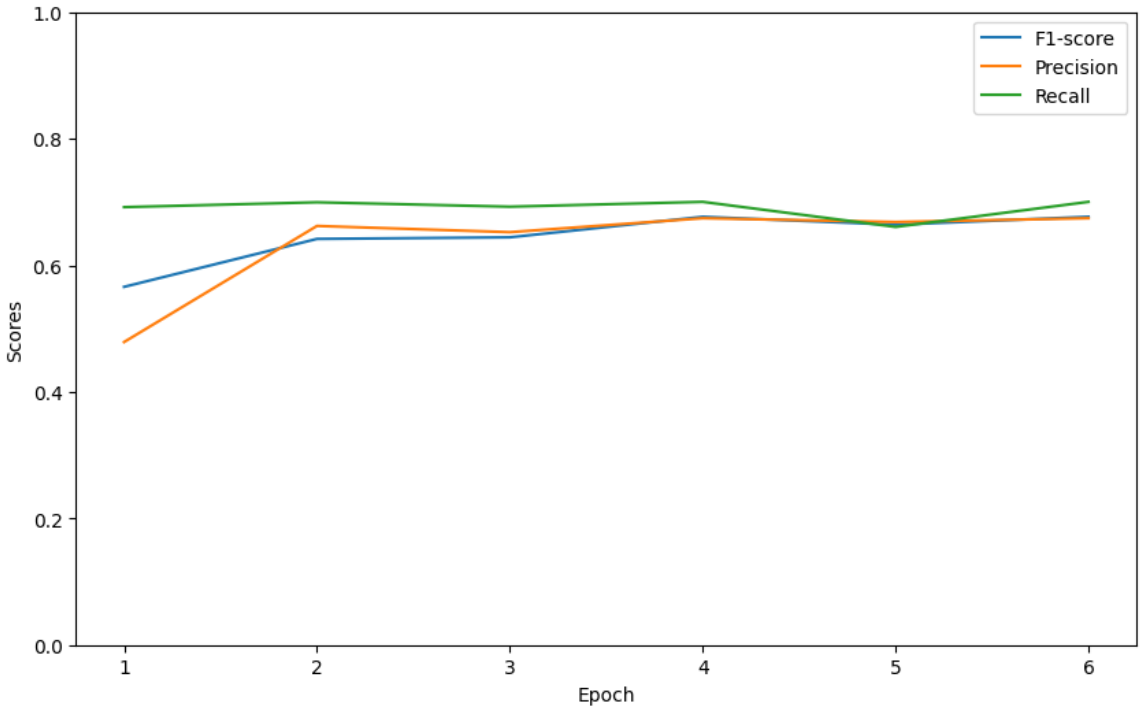

Figura 6.3: Avaliação do classificador de comentários tóxicos direcionados em cada época de treinamento. Fonte: Elaborada pelo autor.

A Tabela 6.9 apresenta os resultados das métricas para cada classe do classificador de comentários tóxicos direcionados.

Classe	Precisão	Abrangência	*F-Measure*	Exemplos
UNT (*untargeted*)	52,19%	32,28%	39,89%	443
TIN (*targeted insult*)	74,23%	86,83%	80,04%	995

Tabela 6.9: Resultados obtidos por classe no experimento do classificador de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

O modelo mostrou capacidade superior na identificação de comentários tóxicos direcionados, alcançando uma *F-Measure* ponderada de 80,04%. No entanto, o modelo apresentou dificuldade ao lidar com a classe “UNT” (*untargeted*). Ao realizarmos uma análise qualitativa dos exemplos, é possível compreender a complexidade desta tarefa, pois nem sempre há evidências claras de um possível alvo para o discurso de ódio, não podemos supor que a existência de um nome de usuário (anonimizado como “USER”) significa que o comentário tóxico é direcionado a esse indivíduo, o que torna essa tarefa ainda mais desafiadora.

6.4 Classificador do tipo de alvo de comentários tóxicos direcionados

Para treinar o classificador do tipo de alvo de comentários tóxicos direcionados os dados foram filtrados para incluir apenas os comentários tóxicos (“OFF”) e que foram categorizados como direcionados (“TIN”). O conjunto de treinamento, validação e teste possui 2.269, 568 e 946 instâncias, respectivamente.

O modelo BERT (*Bidirectional Encoder Representations from Transformers*) pré-treinado com textos em Português disponibilizado em Souza et al. (2020) foi utilizado através da biblioteca Transformers (Wolf et al., 2020) contendo 12 camadas e 110 milhões de parâmetros, adicionamos a função de ativação *Softmax* na saída do modelo para fornecer a probabilidade de pertencimento de cada classe. Além disso, adaptamos a função de perda para incluir pesos calculados de acordo com a incidência de instâncias positivas da classe **targeted_type**, a fim de mitigar o desequilíbrio da classe. Para identificar os hiperparâmetros apropriados para o modelo, realizamos uma otimização Bayesiana, utilizando a *F-Measure* ponderada (*weighted*) como métrica-alvo e considerando os seguintes intervalos de hiperparâmetros:

- *learning rate*: entre 0,00001 e 0,001;
- *weight decay*: entre 0,0 e 0,1;
- *adam beta1*: entre 0,8 e 0,999;
- *adam beta2*: entre 0,8 e 0,999;
- *adam epsilon*: entre 0,00000001 e 0,000001;
- *label smoothing factor*: entre 0,0 e 0,1;
- *optimizer*: “adamw_hf”, “adamw_torch”, “adamw_apex_fused” ou “adafactor”.

Além dos hiperparâmetros mencionados anteriormente, estabelecemos alguns outros de forma estática. O tamanho do lote (*batch size*) foi estabelecido como 8 e a quantidade de épocas (num_train_epochs) foi definida como 10. Utilizamos uma política de parada precoce (early stopping) de 2 épocas sem melhora na métrica *F-Measure* ponderada avaliada no conjunto de validação. A Tabela 6.10 apresenta o resultado da *F-Measure* ponderada e a duração de cada um dos 18 treinamentos no trabalho de ajuste de hiperparâmetros realizado.

Nome do trabalho	*F-Measure* ponderada	Duração do treinamento
pytorch-training-221209-0011-010-d474d283	0.7943	16 minutos
pytorch-training-221209-0011-012-713a2cab	0.7892	11 minutos
pytorch-training-221209-0011-017-26c875ea	0.7888	12 minutos
pytorch-training-221209-0011-005-d9f53221	0.7838	15 minutos
pytorch-training-221209-0011-018-bbc1a967	0.7827	8 minutos
pytorch-training-221209-0011-011-0a4772df	0.7809	11 minutos
pytorch-training-221209-0011-009-67fb584b	0.7804	12 minutos
pytorch-training-221209-0011-001-ff4aab66	0.7740	12 minutos
pytorch-training-221209-0011-006-da6d3b9e	0.7714	13 minutos
pytorch-training-221209-0011-013-14b6c947	0.7682	12 minutos
pytorch-training-221209-0011-015-61ec86e2	0.7676	13 minutos
pytorch-training-221209-0011-008-90a1fb3f	0.7668	11 minutos
pytorch-training-221209-0011-016-0c5d2f3b	0.7665	9 minutos
pytorch-training-221209-0011-003-c198c584	0.7628	11 minutos
pytorch-training-221209-0011-014-194d2e74	0.7605	11 minutos
pytorch-training-221209-0011-002-91f528d4	0.7486	11 minutos
pytorch-training-221209-0011-007-e0515402	0.4808	9 minutos
pytorch-training-221209-0011-004-0778c592	0.4808	8 minutos

Tabela 6.10: Resultados dos treinamentos do trabalho de ajuste de hiperparâmetros do classificador do tipo de alvo de comentário tóxico direcionado. Fonte: Elaborada pelo autor.

O treinamento que obteve melhor desempenho no trabalho de ajuste de hiperparâmetros utilizou os seguintes hiperparâmetros:

- *learning rate*: 3,952388499692274e-05;
- *weight decay*: 0,1;
- *adam beta1*: 0,9944095815441554;
- *adam beta2*: 0,8750000522553327;
- *adam epsilon*: 1,8526084265228802e-07;
- *label smoothing factor*: 0,047566123672759336;
- *optimizer*: “adafactor”.

O modelo final foi treinado utilizando os dados dos conjuntos de treinamento e validação juntos, e utilizando os hiperparâmetros acima. Após 5 épocas de treinamento, a política de parada precoce foi acionada e o modelo foi salvo. Para avaliar o modelo foi utilizado o conjunto de teste. A Tabela 6.11 apresenta os resultados gerais das métricas do modelo final para o classificador do tipo de alvo de comentários tóxicos direcionados.

Precisão (ponderada)	78,31%
Abrangência (ponderada)	77,48%
***F-Measure* (ponderada)**	77,83%

Tabela 6.11: Resultados gerais obtidos no experimento do classificador do tipo de alvo de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

A Figura 6.4 apresenta as métricas de avaliação em cada época de treinamento.

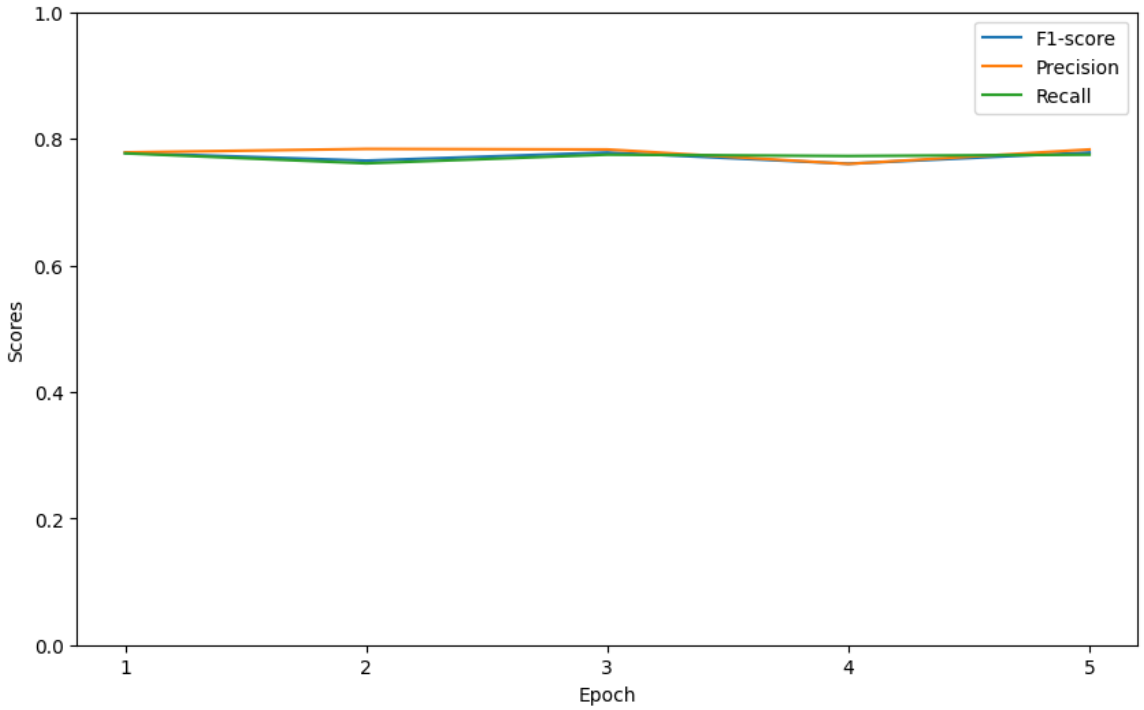

Figura 6.4: Avaliação do classificador do tipo de alvo de comentários tóxicos direcionados em cada época de treinamento. Fonte: Elaborada pelo autor.

A Tabela 6.12 apresenta os resultados das métricas para cada classe do classificador do tipo de alvo de comentários tóxicos direcionados.

Classe	Precisão	Abrangência	*F-Measure*	Exemplos
IND (*individual*)	87,86%	84,40%	86,10%	609
GRP (*group*)	65,44%	66,67%	66,05%	213
OTH (*other*)	53,47%	62,10%	57,46%	124

Tabela 6.12: Resultados obtidos por classe no experimento do classificador do tipo de alvo de comentários tóxicos direcionados. Fonte: Elaborada pelo autor.

De forma geral, o modelo apresentou resultados satisfatórios, atingindo uma *F-Measure* ponderada de 77,83%. No entanto, conforme observado na Tabela 6.9, foi notado que o modelo apresentou dificuldade em classificar corretamente os exemplos pertencentes à classe “OTH”, o que pode ser explicado pela grande variedade de possibilidades existentes dentro dessa classe, em comparação com as classes “IND” e “GRP”, que são mais específicas.

6.5 Detector das partes tóxicas do texto

Para treinar o detector das partes tóxicas do texto os dados foram filtrados para incluir apenas os comentários tóxicos (“OFF”) e que o campo **toxic_spans** não é nulo. O

conjunto de treinamento, validação e teste possui 3.433, 859 e 1.438 instâncias, respectivamente.

Em oposição aos experimentos anteriores, utilizamos um modelo pré-treinado em Português, fornecido pela biblioteca SpaCy (Honnibal e Montani, 2022), especificamente para a tarefa de Reconhecimento de Entidade (RE). Adicionamos a entidade "TOXIC" e realizamos um treinamento fino com os dados de treinamento. O modelo foi levemente adaptado para inferir *spans* ao invés de *tokens*, que é o padrão do modelo pré-treinado. Para identificar os hiperparâmetros apropriados para o modelo, realizamos uma otimização Bayesiana, utilizando a *F-Measure* ponderada como métrica-alvo e considerando os seguintes intervalos de hiperparâmetros:

- *learning rate*: entre 0,01 e 0,0001;
- *dropout*: 0,0, 0,1, 0,2, 0,3, 0,4 ou 0,5;
- *weight decay*: entre 0,0 e 0,1;
- *adam beta1*: entre 0,8 e 0,999;
- *adam beta2*: entre 0,8 e 0,999;
- *adam epsilon*: entre 0,00000001 e 0,000001;
- *optimizer*: "adam" ou "radam".

Outros hiperparâmetros foram definidos estaticamente como a quantidade de épocas (*num_train_epochs*) que foi definida como 30 com uma política de parada precoce (*early stopping*) em 5 épocas sem melhora na métrica *F-Measure* avaliada no conjunto de validação. A Tabela 6.13 apresenta o resultado da *F-Measure* ponderada e a duração de cada um dos 18 treinamentos no trabalho de ajuste de hiperparâmetros realizado.

Nome do trabalho	*F-Measure* ponderada	Duração do treinamento
pytorch-training-221219-0823-009-95f9058d	0.5666	10 minutos
pytorch-training-221219-0823-003-9224d519	0.5643	22 minutos
pytorch-training-221219-0823-017-1d5a97e9	0.5630	13 minutos
pytorch-training-221219-0823-005-e62eb39f	0.5627	20 minutos
pytorch-training-221219-0823-016-92398636	0.5544	11 minutos
pytorch-training-221219-0823-002-2a3cc18a	0.5531	15 minutos
pytorch-training-221219-0823-006-c097609d	0.5451	13 minutos
pytorch-training-221219-0823-001-d03b1c30	0.5418	21 minutos
pytorch-training-221219-0823-018-795921a0	0.5197	13 minutos
pytorch-training-221219-0823-015-ed3f4506	0.5075	15 minutos
pytorch-training-221219-0823-008-ddf7a40e	0.4915	11 minutos
pytorch-training-221219-0823-007-0e41718b	0.4861	14 minutos
pytorch-training-221219-0823-011-500b77f0	0.4832	11 minutos
pytorch-training-221219-0823-004-342f085a	0.4817	10 minutos
pytorch-training-221219-0823-012-b1d06540	0.4754	11 minutos
pytorch-training-221219-0823-010-16f0737e	0.4652	9 minutos
pytorch-training-221219-0823-013-1847c7a8	0.4569	9 minutos
pytorch-training-221219-0823-014-e808b567	0.4055	10 minutos

Tabela 6.13: Resultados dos treinamentos do trabalho de ajuste de hiperparâmetros do detector das partes tóxicas do texto. Fonte: Elaborada pelo autor.

O treinamento que obteve melhor desempenho no trabalho de ajuste de hiperparâmetros utilizou os seguintes hiperparâmetros:

- *learning rate*: 0,00038798590315954165;
- *dropout*: 0,3;
- *weight decay*: 0,1;
- *adam beta1*: 0,9978242993498763;
- *adam beta2*: 0,9988901284249041;
- *adam epsilon*: 3,12576102525027e-08;
- *optimizer*: “adam”.

O modelo final foi treinado utilizando os dados dos conjuntos de treinamento e validação juntos, e utilizando os hiperparâmetros acima. Após 10 épocas de treinamento, a política de parada precoce foi acionada e o modelo foi salvo. O modelo foi avaliado através do conjunto de teste com as métricas Precisão, Abrangência e *F-Measure* que foram adaptadas para primeiro serem calculadas em cada par de valor esperado e valor predito, e então, consolidar utilizando a média de todos os valores no conjunto de avaliação. A Tabela 6.14 apresenta os resultados das métricas do modelo final para o detector das partes tóxicas do texto.

Precisão	68,76%
Abrangência	49,18%
F-Measure	57,34%

Tabela 6.14: Resultados obtidos no experimento do detector das partes tóxicas do texto. Fonte: Elaborada pelo autor.

A Figura 6.5 apresenta as métricas de avaliação em cada época de treinamento.

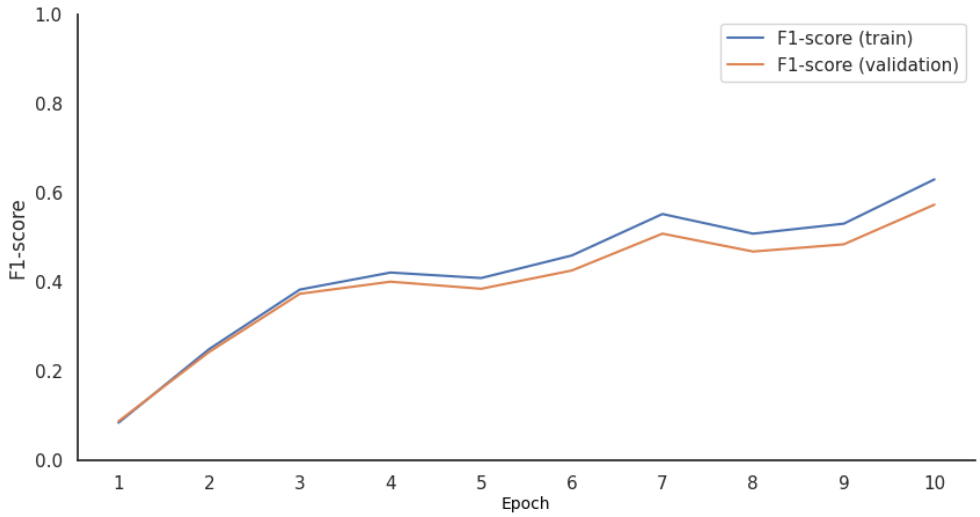

Figura 6.5: Avaliação do detector das partes tóxicas do texto em cada época de treinamento. Fonte: Elaborada pelo autor.

Com um *F-Measure* de 57,34%, o modelo apresentou resultados moderados, como visto na Tabela 6.14, a Precisão foi significativamente maior do que a Abrangência, demonstrando a dificuldade do modelo em encontrar todos os *spans* existentes no conjunto de teste.

7. CONCLUSÕES

Neste trabalho, investigamos diferentes abordagens para tarefas de Processamento de Linguagem Natural (PLN) relacionadas à detecção de linguagem tóxica, utilizando um conjunto de dados desenvolvido como parte deste trabalho e disponibilizado publicamente para estimular pesquisas e contribuições na área. Em geral, os modelos apresentaram desempenho satisfatório nas tarefas avaliadas, o que os torna úteis para ajudar na moderação de conteúdo inapropriado ou para estudos sobre o comportamento de discurso de ódio presente nas redes sociais e outros meios de interação social.

O conjunto de dados OLID-BR, desenvolvido neste estudo, contém textos em língua portuguesa com múltiplas tarefas de PLN anotadas na área de detecção de linguagem tóxica. Ele está disponível publicamente em dois formatos: CSV, com os rótulos já agregados de acordo com as estratégias de atribuição de rótulos definidas neste projeto, e JSON, contendo as três anotações para cada texto, que permite aos pesquisadores explorarem diferentes formas de definir os rótulos para as tarefas propostas no conjunto de dados. Além disso, disponibilizamos os textos com menos de três anotações, que podem ser usados para aumentar a quantidade de amostras. O código e a documentação de como utilizar o conjunto de dados OLID-BR estão disponíveis no GitHub[1] para que outros pesquisadores possam utilizá-lo em suas pesquisas. Também destacamos como contribuição deste trabalho, os modelos treinados com o OLID-BR, que apresentaram desempenho satisfatório nas tarefas avaliadas, com *F-Measure* ponderado de 75,43% para a classificação dos tipos de linguagem tóxica, 66,78% para a classificação de comentários tóxicos direcionados, 76,97% para a classificação do tipo de alvo de comentários tóxicos direcionados e 57,34% para a detecção das partes tóxicas de um texto tóxico. O código para treinamento dos modelos está disponível no GitHub[2] para que futuros trabalhos possam aperfeiçoá-los e/ou testar diferentes técnicas com o objetivo de melhorar seus resultados. O processo de desenvolvimento do OLID-BR e dos modelos treinados neste conjunto de dados foram descritos em um artigo que foi aceito para publicação em *Language Resources and Evaluation*.

A subjetividade da tarefa de anotação pode ser definida como o principal desafio deste trabalho. Percebemos que os anotadores tiveram diferentes interpretações de um mesmo texto, o que é confirmado pelos resultados moderados obtidos no experimento de confiabilidade entre anotadores (Capítulo 4). Isto pode ter sido motivado pelas diferentes formações educacionais, contextos culturais e sociais e pelas diferentes experiências de cada anotador, pois os anotadores foram selecionados a fim de minimizar possíveis vieses de anotação. Este desafio também foi citado em trabalhos anteriores na área de detecção de toxicidade, o que nos ajudou a definir estratégias para minimizar este problema, como

[1] https://github.com/DougTrajano/olid-br
[2] https://github.com/DougTrajano/ToChiquinho

desenvolver diretrizes claras para a tarefa de anotação, fornecer taxonomias e exemplos de linguagem tóxica e realizar um processo iterativo de anotação e validação, que ajudou a diminuir a discordância entre os anotadores. Os experimentos dos modelos demonstraram que os rótulos de toxicidade que possuem um número proporcionalmente menor de instâncias são mais difíceis de classificar, como esperado. Isso pode ser explicado pelo problema de dados desequilibrados, que pode ser atenuado pelo uso de técnicas mais sofisticadas, como aumento de dados. Apesar dos nossos esforços para garantir a qualidade do conjunto de dados, existem limitações que podem influenciar os resultados obtidos. A primeira delas é a quantidade de dados disponíveis, que foi limitada pela quantidade de dados anotados manualmente. Outra limitação é a origem dos dados, que foram coletados de redes sociais e conjuntos de dados públicos, o que pode não representar a diversidade global e ser generalizado para outros contextos. Os vieses potenciais nos dados incluem: vieses inerentes das redes sociais, vieses da base de usuários, as listas de palavras ofensivas/vulgares (anotados na iteração 1) usadas para filtragem de dados e vieses inerentes ou inconscientes na avaliação do conteúdo de toxicidade.

Acreditamos que o conjunto de dados OLID-BR é um recurso valioso para a comunidade acadêmica no desenvolvimento de novas pesquisas na área de detecção de linguagem tóxica. Como trabalhos futuros, podemos explorar o aprimoramento dos modelos existentes utilizando técnicas avançadas para lidar com dados desequilibrados e aumentar a quantidade de dados através do uso de dados adicionais fornecidos com o OLID-BR. Além disso, podemos utilizar técnicas de aprendizado semi-supervisionado para aumentar ainda mais a quantidade de dados de treinamento. O classificador de tipos de linguagem tóxica apresenta-se como potencialmente promissor para obter resultados significativamente melhores com o aumento de dados. Pretendemos realizar uma competição acadêmica onde os dados de teste privado, não disponibilizados publicamente, possam ser utilizados para avaliar as submissões. Além disso, os metadados dos textos também podem ser explorados para melhorar os resultados das predições ou para entender os perfis dos anotadores e características dos comentários presentes no OLID-BR.

REFERÊNCIAS BIBLIOGRÁFICAS

Alonso, P., Saini, R., and Kovács, G. (2020). Hate speech detection using transformer ensembles on the hasoc dataset. In: *Speech and Computer*, pp. 13–21. Springer, Springer International Publishing.

Alrehili, A. (2019). Automatic hate speech detection on social media: A brief survey. In: *IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–6. IEEE.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In: *Advances in Neural Information Processing Systems*, pp. 1877–1901. Curran Associates, Inc.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (Nov, 2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, vol. 12, pp. 2493–2537.

Dasu, T. and Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc.

de Pelle, R. and Moreira, V. (2017). Offensive comments in the brazilian web: a dataset and baseline results. In: *Anais do VI Brazilian Workshop on Social Network Analysis and Mining*, pp. 1–10. SBC.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics.

Er, M. J., Venkatesan, R., and Wang, N. (2016). An online universal classifier for binary, multi-class and multi-label classification. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3701–3706. IEEE.

Eugenio, B. D. and Glass, M. (Jan, 2004). The kappa statistic: A second look. *Computational linguistics*, vol. 30, pp. 95–101.

Feinstein, A. R. and Cicchetti, D. V. (Fev, 1990). High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, vol. 43, pp. 543–549.

Fortuna, P. and Nunes, S. (Jul, 2018). A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, vol. 51, pp. 1–30.

Fortuna, P., Rocha da Silva, J., Soler-Company, J., Wanner, L., and Nunes, S. (2019). A hierarchically-labeled Portuguese hate speech dataset. In: *Proceedings of the Third Workshop on Abusive Language Online*, pp. 94–104. Association for Computational Linguistics.

Gaydhani, A., Doma, V., Kendre, S., and Bhagwat, L. (Set, 2018). Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *ArXiv*, vol. abs/1809.08651, pp. 1–5.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In: *Proceedings of the 16th Conference on Computational Linguistics*, pp. 466–471. Association for Computational Linguistics.

Gwet, K. L. (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.

Gwet, K. L. (Jun, 2008). Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, vol. 61, pp. 29–48.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hayes, A. F. and Krippendorff, K. (Abr, 2007). Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, vol. 1, pp. 77–89.

He, Z., Wang, Z., Wei, W., Feng, S., Mao, X., and Jiang, S. (Nov, 2020). A survey on recent advances in sequence labeling from deep learning models. *arXiv preprint*, vol. abs/2011.06727, pp. 1–16.

Hochreiter, S. and Schmidhuber, J. (Dez, 1997). Long short-term memory. *Neural computation*, vol. 9, pp. 1735–1780.

Honnibal, M. and Montani, I. (2022). spaCy · Industrial-strength Natural Language Processing in Python. Recuperado de: https://spacy.io/. Dezembro 2022.

Jurafsky, D. and Martin, J. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall.

Kitchenham, B. (Ago, 2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, vol. 33, pp. 1–26.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (Mar, 2019). Text classification algorithms: A survey. *Information*, vol. 10, pp. 150.

Landis, J. R. and Koch, G. G. (Abr, 1977). The measurement of observer agreement for categorical data. *biometrics*, vol. 33, pp. 159–174.

Leite, J. A., Silva, D., Bontcheva, K., and Scarton, C. (2020). Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 914–924. Association for Computational Linguistics.

Levy, L., Karst, K., and Winkler, A. (2000). *Encyclopedia of the American Constitution*. Macmillan Reference USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, pp. 3111–3119. Curran Associates Inc.

Nadeau, D. and Sekine, S. (Ago, 2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, vol. 30, pp. 3–26.

Otter, D. W., Medina, J. R., and Kalita, J. K. (Fev, 2021). A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, vol. 32, pp. 604—624.

Pavlopoulos, J., Sorensen, J., Laugier, L., and Androutsopoulos, I. (2021). SemEval-2021 task 5: Toxic spans detection. In: *Proceedings of the 15th International Workshop on Semantic Evaluation*, pp. 59–69. Association for Computational Linguistics.

Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., and Patti, V. (Set, 2021). Resources and benchmark corpora for hate speech detection: a systematic review. *Lang. Resour. Evaluation*, vol. 55, pp. 477–523.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (Set, 2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, vol. 51, pp. 1–36.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (Fev, 2019). Language models are unsupervised multitask learners. *OpenAI blog*, vol. 1, pp. 9.

Raghunathan, B. (2013). *The Complete Book of Data Anonymization: From Planning to Implementation*. Auerbach Publications.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (Out, 1986). Learning representations by back-propagating errors. *nature*, vol. 323, pp. 533–536.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Shelar, H., Kaur, G., Heda, N., and Agrawal, P. (Mai, 2020). Named entity recognition approaches and their comparison for custom ner model. *Science & Technology Libraries*, vol. 39, pp. 324–337.

Siddiqui, S., Singh, T., et al. (Fev, 2016). Social media its impact with positive and negative aspects. *International journal of computer applications technology and research*, vol. 5, pp. 71–75.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In: Cerri, R. and Prati, R. C., editores, *9th Brazilian Conference on Intelligent Systems, BRACIS*, pp. 403–417. Springer-Verlag.

Tkachenko, M., Malyuk, M., Holmanyuk, A., and Liubimov, N. (2020). Label Studio: Data labeling software. Recuperado de https://github.com/heartexlabs/label-studio/. Dezembro 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In: *Advances in neural information processing systems*, pp. 5998–6008. Curran Associates, Inc.

Whillock, R. K. and Slayden, D. (1995). *Hate speech*. ERIC.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In: *Conference on Empirical Methods in Natural Language Processing*, pp. 38–45. Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). Predicting the Type and Target of Offensive Posts in Social Media. In: *Proceedings of NAACL*, pp. 1415–1420. Association for Computational Linguistics.

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., and Çöltekin, Ç. (2020). SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 1425–1447. International Committee for Computational Linguistics.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br