UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Mineração de texto aplicada à detecção de cyberbullying

Tácio Souza Bomfim

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Mineração de texto aplicada à detecção de cyberbullying

Tácio Souza Bomfim

USP - São Carlos
2022

Tácio Souza Bomfim

Mineração de texto aplicada à detecção de Cyberbullying

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Alneu de Andrade Lopes

USP - São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B719m

Bomfim, Tacio
Mineração de texto aplicada à detecção de cyberbullying / Tacio Bomfim; orientador Alneu Lopes. -- São Carlos, 2022.
49 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. Mineração de texto. 2. Cyberbullying. I. Lopes, Alneu, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

AGRADECIMENTOS

Aos professores do curso de MBA da USP, em especial ao meu orientador, Dr. Alneu de Andrade Lopes, que muito me ensinou contribuindo para o meu crescimento científico.

RESUMO

BOMFIM, S. T. **Mineração de texto aplicada à detecção de cyberbullying**. 2022. 49 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

O *Cyberbullying* é considerado como um grave problema de saúde pública pelo centro de controle e prevenção de doenças e têm crescido ao passar dos anos, principalmente entre as crianças e adolescentes, auxiliado pelo aumento e difusão das redes sociais. Os tipos de ofensa do cyberbullying podem ter cunhos racista, homofóbicos, sexistas, ou estar ligado a depreciação do indivíduo, através de discursos de ódio, e é considerado como crime, estando associadas àqueles crimes previstos no decreto-lei no 2.848, de 7 de dezembro de 1940, do código penal brasileiro, como calúnia, difamação, injúria, ameaça e constrangimento ilegal, podendo ainda a pena ser agravada em caso de ser realizado pela Internet. Como forma de auxiliar a mitigar o cyberbullying em meios digitais, tais como Twitter, Facebook e Instagram, o objetivo deste trabalho foi treinar algoritmos de mineração de texto existentes, com capacidade de identificar mensagens de ofensa relacionadas ao cyberbullying, realizando comparação entre eles. Dessa maneira, foram utilizados dois algoritmos para a tarefa de classificação de textos, o BERT e o *One Class* SVM. O treinamento ocorreu utilizando duas bases de dados com *tweets* em língua portuguesa. Como resultado, o modelo BERT com a utilização do BERTimbau, que é o modelo pré treinado para língua portuguesa, obteve um F1-Score total de 80% para a primeira base, com uma precisão e revocação também de 80%. O *One Class* SVM, obteve um F1-Score global de 61%, com uma precisão de 67%. Com a segunda base de textos, foi encontrado um resultado de F1-Score de 67% para o BERT e 48% para o outro modelo. Outra conclusão que ficou evidente na aplicação deste trabalho foi a dificuldade de os modelos identificarem textos com sentido figurado, sarcasmo ou ironia. Dessa forma, como trabalhos futuros, é identificado a necessidade de treinar outros algoritmos que possam ter desempenho melhores na classificação, principalmente aqueles que trarão uma boa compreensão semântica para classificar exemplos complexos. Uma outra tarefa é melhorar a base de dados de treinamento, com aplicação de modelos de pré-processamento mais robustos e cuidadosos, a fim de verificar se o desempenho do modelo aumenta nessa situação.

Palavras-chave: *Cyberbullying*; BERT;SVM .

ABSTRACT

BOMFIM, S. T. **Text mining applied to cyberbullying detection**. 2022. 49 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

Cyberbullying, considered a serious public health problem by the Center for Disease Control and Prevention, has grown over the years, especially among children and adolescents, aided by the increase and diffusion of social networks. The types of cyberbullying offense can be racist, homophobic, sexist, or be linked to the depreciation of the individual, through hate speech, and is considered a crime, being associated with those crimes provided for in Decree-Law No. of December 1940, of the Brazilian penal code, such as slander, defamation, injury, threat and illegal constraint, and the penalty may also be aggravated if carried out over the Internet. As a way to help mitigate cyberbullying in digital media, such as Twitter, Facebook and Instagram, the objective of this work was to train existing text mining algorithms, with the ability to identify offending messages related to cyberbullying, making comparisons between them. In this way, two algorithms were used for the text classification task, BERT and One Class SVM. The training took place using two databases with tweets in Portuguese. As a result, the BERT model using BERTimbau, which is the pre-trained model for the Portuguese language, obtained a total F1-Score of 80% for the first base, with accuracy and recall also of 80%. The One Class SVM, achieved an overall F1-Score of 61%, with an accuracy of 67%. With the second text base, an F1-Score result of 67% was found for BERT and 48% for the other model. Another conclusion that was evident in the application of this work was the difficulty of the models to identify texts with figurative meaning, sarcasm, or irony. Thus, as future work, the need to train other algorithms that can perform better in classification is identified, especially those that will bring a good semantic understanding to classify complex examples. Another task is to improve the training database, with the application of more robust and careful pre-processing models to verify if the model's performance increases in this situation.

Keywords: Cyberbullying; BERT; SVM.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CDC	–	*Center of Disease Control and Prevention*
ASTM	–	*American Society for Testing and Materials*
SVM	–	*Support Vector Machines*
NLP	–	Processamento de Linguagem Natural
BoW	–	*Bag of Words*
TF-IDF	–	*Term frequency-inverse document frequency*
BERT	–	*Bidirectional Encoder Representations from Transformers*
SVM	–	*Support Vector Machines*

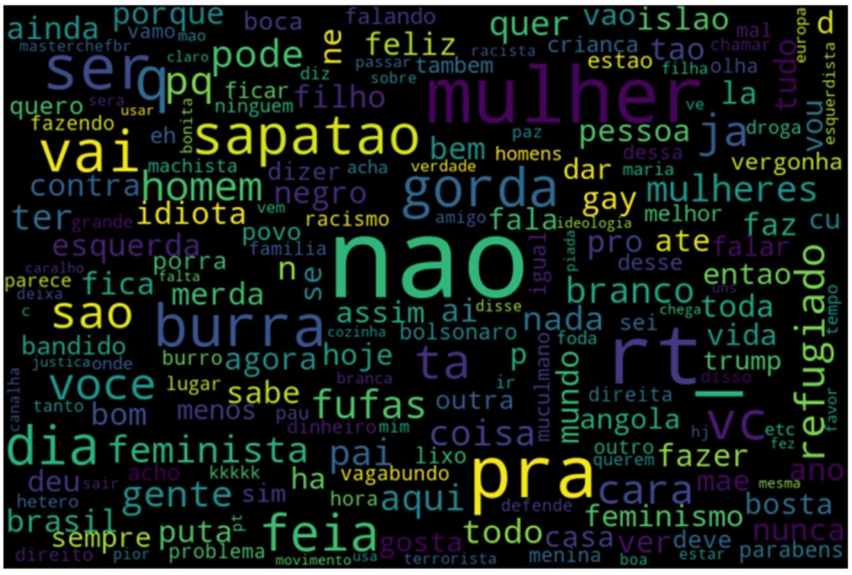
SUMÁRIO

1 INTRODUÇÃO

A evolução da Internet possibilitou a interação virtual entre indivíduos de diversas partes do mundo através das redes sociais. Em contrapartida, pessoas têm utilizado esses canais digitais para proferir mensagens ofensivas e/ou agressivas com intenção de causar danos ou desconforto a outros, o que é chamado de *cyberbullying* [1]. O *bullying* virtual é reconhecido como um sério problema de saúde pública pelo centro de controle e prevenção de doenças (CDC – *Center of Disease Control and Prevention*) que afeta crianças e adolescentes, principalmente por ser um fator de risco para o suicídio, depressão e baixo desempenho acadêmico [2,3].

Os tipos de ofensa do cyberbullying podem ter cunhos racista, homofóbicos, sexistas, ou estar ligado a depreciação do indivíduo, através de discursos de ódio, conforme pode ser verificado na Figura 1, que demonstra palavras que aparecem com mais frequência nas bases de textos utilizadas neste artigo. Apesar de muito comum, esse tipo de prática é crime, estando associadas aqueles crimes previstos no decreto-lei no 2.848, de 7 de dezembro de 1940, do código penal brasileiro, como calúnia, difamação, injúria, ameaça, constrangimento ilegal, entre outros. Cabe salientar ainda que, segundo o artigo 141, se o crime for cometido ou divulgado em quaisquer modalidades das redes sociais da rede mundial de computadores, aplica-se em triplo a pena.

Figura 1 – Palavras mais frequentes na base de discurso de ódio



Fonte: Autoria própria.

Não é incomum notícias frequentes de suicídio devido ao discurso de ódio virtual. Exemplo disso, foi o caso do jovem de 16 anos, Lucas Santos, que tirou a própria vida, em agosto de 2021, após receber uma enxurrada de comentários homofóbicos em seu perfil [4].

Apesar do exposto, o número de casos de cyberbullying está crescendo em todo o mundo [5]. A pesquisa "*The Annual Bullying Survey* 2020", que é a maior referência de comportamentos de bullying do Reino Unido, informou que o número de casos de bullying aumentou 25% em 2020, quando comparado com o ano anterior [6]. Além disso, pelo menos 30% dos entrevistados relataram sofrer *bullying* com a periodicidade semanal e 41% pelo menos uma vez ao mês, sendo que 27% de todos os casos foram através de redes sociais [6]. Ainda segundo o estudo anterior, 37% dos entrevistados acreditam que o motivo de sofrerem bullying tem relação com homofobia ou identificação de gênero, 6% devido a questões raciais e 5% referente a intolerância religiosa. Esse tipo de ofensa provocou 11% a tentativa de suicídio, 33% com pensamentos suicidas, 44% sentimentos de ansiedade e 36% problemas com depressão [6].

Em relação as mídias sociais em que mais ocorrem esse tipo de crime, a pesquisa *Annual Bullying Survey* 2017 [7], indicou que 42% dos casos ocorrem no Instagram, 37% no Facebook, 31% no Snapchat, 10% no Youtube e 9% no Twitter.

Associado a todo o exposto anteriormente sobre o *cyberbullying* e suas consequências para as vítimas, a quantidade de informações nas redes sociais são enormes e crescem rapidamente. Um exemplo disso é que, em 2016, a cada minuto, 350 mil tweets são escritos, ou seja, 500 milhões de mensagem diárias [8]. Devido a essa grande quantidade e velocidade de informações, é inviável a filtragem dos conteúdos por humanos de forma manual. Para isso, é necessário que a tecnologia consiga auxiliar na mineração de texto, através dos algoritmos de inteligência artificial. Portanto, uma vez que o *Cyberbullying* ocorre em canais virtuais, é imprescindível que a tecnologia computacional existente crie métodos de mitigação para esse problema.

A mineração de texto é uma área de pesquisa tecnologia cujo objetivo é buscar padrões e tendências em textos escritos em linguagem natural [9]. Para realizar esse trabalho são utilizadas técnicas de aprendizado de máquinas para agrupamento e classificação textual. Existem algoritmos que vem sendo muito utilizados para o processo de classificação como algoritmos genéticos, *Naive-bayes, Support Vector Machines* (SVM), Redes Neurais profundas (como LSTM, GRU e RCNN), modelos baseados em árvores de decisão, modelos *boosting*, entre outros [10].

Este projeto tem o objetivo de treinar algoritmos de mineração de texto existentes, com

capacidade de identificar mensagens de ofensa relacionadas ao cyberbullying, realizando comparação entre eles, e, com isso, contribuir para mitigação desse tipo de ofensa. Neste trabalho espera-se que os modelos consigam generalizar bem, devido à grande variedade de palavras de cunho ofensivo que pode ter. Para isso, foram utilizadas duas bases de dados que contém diversas citações de ódio, na língua portuguesa, para auxiliar no processo de aprendizado. Um ponto importante a ser destacado foi a dificuldade de encontrar coleções de discurso de ódio rotulados em língua portuguesa. Diante dos desafios e problemas atualmente enfrentados para mineração desse tipo de conteúdo, foram levantadas duas questões de pesquisa: “Entre os algoritmos comumente aplicados para essa atividade, qual obtém o melhor resultado?” e “Devido à grande variação de termos ofensivos, os modelos são capazes de generalizar bem e aprender novos cenários?”.

Os objetivos específicos desse projeto foram comparar alguns tipos de algoritmos de aprendizado de máquina para verificar qual deles apresenta o melhor desempenho para o problema apresentado e realizar testes com palavras em diferentes contextos para verificar a eficiência desse modelo para novos casos de discurso de ódio.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais fundamentos para o desenvolvimento deste trabalho.

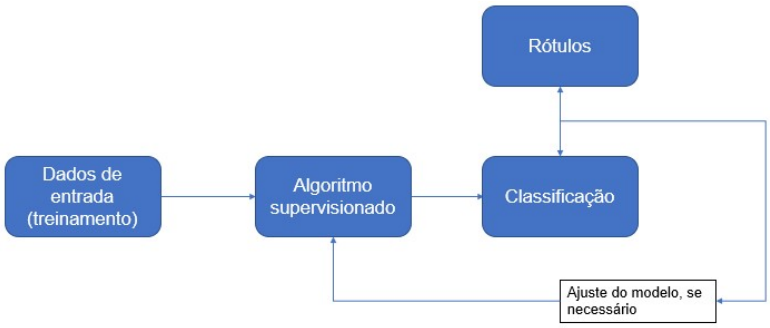
2.1 Tipos de aprendizado de máquina

Os algoritmos de aprendizado de máquina podem ser classificados de acordo com a necessidade ou não de treinamento com supervisão humana. Dessa forma, existem 4 (quatro) categorias que um modelo de *machine learning* pode ser enquadrado: supervisionado, não supervisionado, semissupervisionado e por reforço.

2.1.1 Aprendizado supervisionado

O aprendizado supervisionado, como o próprio nome sugere, são modelos que mapeiam funções matemáticas para prever ou classificar dados de entrada (*inputs*), utilizando dados rotulados, para treinar e melhorar seu desempenho, ilustrado na Figura 2 [11]. Dessa forma, os modelos supervisionados têm a resposta para cada conjunto de dados de entrada, fazendo com o algoritmo consiga comparar a inferência obtida com o resultado esperado, e a partir disso, ajustar a função em caso de erro.

Figura 2 – Esquema do treinamento supervisionado



Fonte: Autoria própria.

Uma grande dificuldade do modelo supervisionado é a necessidade da intervenção humana para realizar as rotulações dos dados. Apesar disso, esses tipos de modelo são amplamente utilizados na tarefa de classificação automática de textos. Algumas aplicações de algoritmos supervisionados, envolvendo textos, é a classificação de e-mails, detecções de spams e resposta automática [12], bem como análise de sentimentos [13]. Alguns dos modelos conhecidos são as máquinas de vetores de suporte (SVM), árvores de decisões e Naïve Bayes [14].

2.1.2 Aprendizado Não Supervisionado

Diferente dos modelos supervisionados, esse tipo de aprendizado não utiliza dados rotulados. Esse tipo de algoritmo tem um benefício de não necessitar de intervenção humana para classificar os dados, isso é muito útil para casos em que existam uma quantidade muito grande de informação, como é o caso dos textos disponíveis em *twitters* e outras mídias sociais.

Esse tipo de aprendizado de máquina é utilizado basicamente para agrupar dados, também conhecido com clusterização, e realizar associações. A clusterização é uma técnica que agrupa dados não rotulados através da semelhança ou diferença entre eles [15]. As regras de associação são utilizadas para localizar relacionamentos ou similaridades entre variáveis em um determinado conjunto de dados. Exemplo desses algoritmos são o *K-means* e o Apriori.

2.1.3 Aprendizado Semissupervisionado

O aprendizado semissupervisionado são modelos que utilizam dados rotulados e não rotulados durante o treinamento, ou seja, é uma mescla entre os aprendizados supervisionado e não supervisionado. Essa técnica é interessante quando o custo para rotular informações é alto, tanto pela complexidade da informação quanto pelo volume, se tornando inviável. Dessa forma, esse tipo de algoritmo pode utilizar uma grande quantidade de dados não rotulados e um pouco de dados rotulados [16].

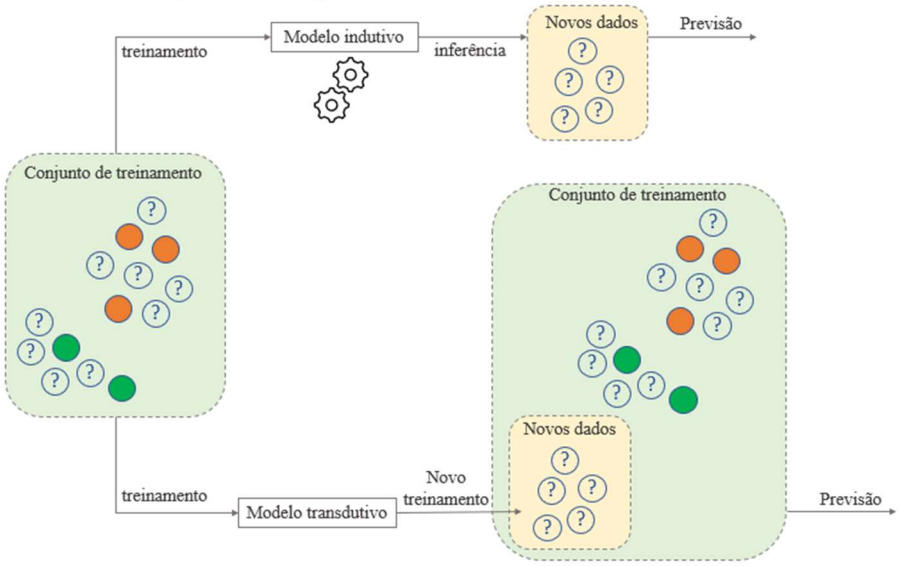
O modelo semissupervisionado pode ter uma abordagem indutiva ou transdutiva. A abordagem indutiva é basicamente o modelo supervisionado clássico, porém, é utilizado os dados (rotulados e não rotulados) do conjunto de treinamento para treinar o modelo, e, a partir disso, inferir uma função que consiga generalizar o comportamento esperado. Posteriormente, o modelo consegue predizer o rótulo de dados que ele não viu anteriormente no treinamento (conjunto de teste, que seriam os dados não rotulados). Esse tipo de modelo tem um custo computacional baixo.

A inferência transdutiva também utiliza ambos os dados, tanto o conjunto de treinamento rotulado quanto os não rotulados, para realizar o treinamento, porém, diferentemente do tipo indutivo, ele não gera um modelo capaz de prever dados não vistos anteriormente. Dessa forma, um ponto negativo dos modelos transdutivos, é que eles não geram uma função geral de predição, e, portanto, se um novo ponto de dados for adicionado ao conjunto de teste, o modelo terá que começar todo o processo de treinamento novamente.

Exemplificando as diferenças entre o método indutivo e transdutivo, a Figura 3 ilustra um conjunto de treinamento com dados rotulados (laranja e verde) e não rotulados (sem preenchimento). No caso do modelo semissupervisionado indutivo, seria utilizado os dados de treinamento para gerar um modelo e os novos dados a serem classificados seriam inferidos sem a necessidade de um novo treinamento. No caso do modelo transdutivo, seria realizado um novo treinamento com a inclusão dos novos dados no conjunto.

O modelo transdutivo ganha vantagem ao utilizar todos os dados do conjunto para realizar o treinamento, inclusive aqueles não vistos, e com isso pode realizar clusters mais precisos, utilizando dados de vizinhança, podendo ser mais assertivo que o modelo indutivo muitas vezes.

Figura 3 – Esquema do treinamento supervisionado



Fonte: Autoria própria.

2.1.4 Aprendizado por reforço

O aprendizado por reforço, ou *reinforcement learning*, tem sido muito eficiente para problemas de jogos e robótica, devido ao modelo ser treinado para tomar sequências de decisões. De forma prática, os algoritmos que executam esse tipo de aprendizado, utilizam a técnica de tentativa e erro, ou seja, são executados comandos e a medida em que o modelo acerta, é lhe dado uma recompensa, ao passo que quando erra, é aplicado uma penalidade. O objetivo do algoritmo está em maximizar os ganhos de recompensa [17].

Um exemplo clássico para esse tipo de aprendizado é o jogo de labirinto, em que existe vários caminhos, porém apenas um deles leva a saída. Dessa forma o algoritmo realiza um

mapeamento de ações possíveis para que seja encontrado o resultado que irá maximizar a recompensa.

2.2 Representação textual e algoritmos para mineração de texto

Para utilização de algoritmos computacionais, como modelos estatísticos e aprendizado de máquinas, no processo de mineração de texto, é necessário transformar as palavras em caracteres numéricos, sendo mais comum a utilização de vetores para essa finalidade. Dois métodos muito utilizados para representar dados textuais para a tarefa de classificação são as *bags of words* e *word embeddings*.

2.2.1 *Bag of Words*

O *Bag of Words (BoW)*, que pode ser traduzido livremente para o português como "saco de palavras", é uma forma de extrair recursos do texto para serem utilizadas em algoritmos de aprendizado de máquina, por exemplo. Basicamente essa técnica utiliza a ocorrência das palavras presentes nos documentos, não dando relevância para ordem em que ela aparece. Uma forma de criar termos mais complexos e com mais informação semântica, é a utilização de N-grams, que são associação entre duas ou mais palavras, a depender do fator "N" escolhido. Um exemplo de um 2-gram, também chamado de bigram, seria "por favor" e "estou indo". É possível notar que o BoW que contém apenas palavras simples é um 1-gram.

Uma forma utilizada no contexto de BoW é uma indicação binária, ou seja, o algoritmo computa 1 para termos que estão contidos em um determinado documento e 0 para aqueles que não estão. Para exemplificar essa situação, na Tabela 1 estão descritos três *tweets* coletados da base de dados utilizadas neste trabalho.

Tabela 1 – *Tweets* da base de dados

Documento	**Texto**
Doc.1	Bom dia macaco branco
Doc. 2	Balofo um dia vai morrer pela boca
Doc. 3	Cala boca idiota

Fonte: Autoria própria.

A Tabela 2 demostra como é a aplicação do BoW com valores binários, utilizando 0 ou 1 para as palavras contidas em cada documento. É importante salientar que cada *twitter* é considerado um documento, sendo composto por termos, que são as palavras utilizadas no conjunto do *Bag of Words*.

Tabela 2 – Matriz com valores binários

	Bom	Dia	Macaco	Branco	Balofo	Um	Vai	Morrer	Pela	Boca	Cala	idiota
Doc.1	1	1	1	1	0	0	0	0	0	0	0	0
Doc. 2	0	1	0	0	1	1	1	1	1	1	0	0
Doc. 3	0	0	0	0	0	0	0	0	0	1	1	1

Fonte: Autoria própria.

A técnica TF-IDF, cuja sigla significa "*term frequency – inverse document frequency*", é utilizada no contexto de BoW para criar pesos numéricos para as palavras que estão presentes em cada documento. Dessa forma, o TF-IDF dá um peso de ponderação menor para as palavras que mais se repetem no conjunto de documentos e um peso maior é atribuído a termos que se repetem poucas vezes. Essa lógica é interessante, porque artigos, por exemplo, são termos encontrados com frequência, mas que pouco importa para o entendimento geral do documento. Essas palavras, que são conhecidas como *stopwords*, são facilmente eliminadas com essa técnica.

Para realizar o cálculo desse método é utilizado o produto de dois termos, sendo o primeiro a frequência em que uma determinada palavra aparece em um documento, representado com **tf**, e o segundo que é um fator de ponderação para as palavras que aparecem no conjunto total de documentos [18], conforme Equação 1:

$$w\ (t,d) = tf_{t,d}\ x\ idf_t \tag{1}$$

A primeira parcela do TF-IDF é calculada conforme a Equação 2, no qual é considerado o logaritmo da quantidade de vezes que um determinado termo é repetido em documento. Nota-se que se o termo não estiver presente no documento, essa parcela será o log (1) = 0.

$$tf_{t,d} = log_{10}(\ count(t,d) + 1) \tag{2}$$

O termo **idf**, é a parte referente ao inverso da frequência. O cálculo consiste no logaritmo da razão entre o número total de documentos existentes na base e a quantidade de documentos que existe o termo ao menos uma vez.

$$idf_t = log_{10}(\frac{N}{df_t}) \quad (3)$$

A Tabela 3 demonstra os resultados ao aplicar o cálculo do TF-IDF para o caso abordado. É possível observar que as palavras que são mais repetidas têm um fator menor do que as palavras menos repedidas. Dessa forma, é possível verificar a relevância de alguns termos para o conteúdo abordado, como, por exemplo: Balofo, Macaco e idiota.

Tabela 3 – Matriz TF-IDF

	Bom	Dia	Macaco	Branco	Balofo	Um	Vai	Morrer	Pela	Boca	Cala	idiota
Doc.1	0,17	0,07	0,17	0,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Doc. 2	0,00	0,07	0,00	0,00	0,17	0,17	0,17	0,17	0,17	0,07	0,00	0,00
Doc. 3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,07	0,17	0,17

Fonte: Autoria própria.

2.2.2 *Word Embeddings*

Uma limitação da utilização da representação textual utilizando a técnica *Bag of Words* é a dificuldade de relacionar palavras similares ou sinônimas. Por exemplo, se considerarmos as expressões “Elizabeth conversou com a imprensa” e “A Rainha falou com a mídia”, a representação do BoW seria conforme Tabela 4. Ao calcular a similaridade do cosseno, percebe-se que a esse valor seria zero, ou seja, não haveria nenhuma correlação entre os dois documentos.

Entretanto é possível notar que existe de fato similaridade entre as frases, porque “Elizabeth” e “Rainha”, “Conservou” e “falou” e “imprensa” e “mídia” são palavras que dentro de um mesmo contexto, tem significados semânticos muito parecidos.

Dessa forma, as *word embeddings* tem o objetivo de representar palavras semanticamente parecidas em forma de vetores de *n* dimensões, chamados de *word vectors*, de forma que os termos que aparecem com frequência no mesmo contexto, vão receber vetores similares.

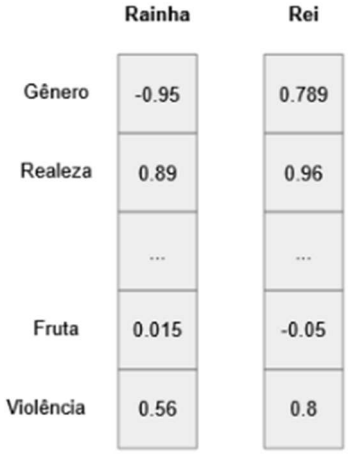
Tabela 4 – Representação *bag of Words*

	Elizabeth	conversou	imprensa	Rainha	falou	mídia
Doc.1	1	1	1	0	0	0
Doc. 2	0	0	0	1	1	1

Fonte: Autoria própria.

Podemos imaginar que os vetores são formados por alguns atributos, que pode variar numa escala de -1 a 1, representando assim situações antagônicas [19]. Para ficar mais claro, podemos considerar a Figura 4, em que o gênero mais perto de -1 significa feminino e mais perto de 1 o masculino, assim como a realeza, em que -1 seria a plebe e o 1 seria a realeza.

Figura 4 – Modelo de representação de *word embeddings*



Fonte: [19].

Com isso, é possível calcular através de alguma métrica, como a distância euclidiana ou similaridade de cossenos, quais são as palavras que tem maior correlação. A similaridade dos cossenos é um produto interno dos vetores, cujo resultado pode variar de -1 a 1, sendo zero quando o ângulo for de 90º.

Existem várias técnicas para extração de *word embeddings* em um córpus textual, como a Word2Vec que é uma das mais conhecidas, utilizando redes neurais para essa finalidade. Existem modelos pré-treinados que podem ser utilizados, uma vez que é necessário de um córpus razoavelmente grande e um bom processamento computacional para treinar as embeddings, como a google que pré treinou uma rede com 100 bilhões de palavras, resultando em 3 milhões de embeddings, com dimensão 300, todas na língua inglesa [20]; O facebook que gerou *embeddings* para 2 milhões de palavras em múltiplas línguas, também com dimensão 300 e o NILC, que é o laboratório do ICMC da USP que tem 1,3 bilhão de tokens em língua portuguesa [21].

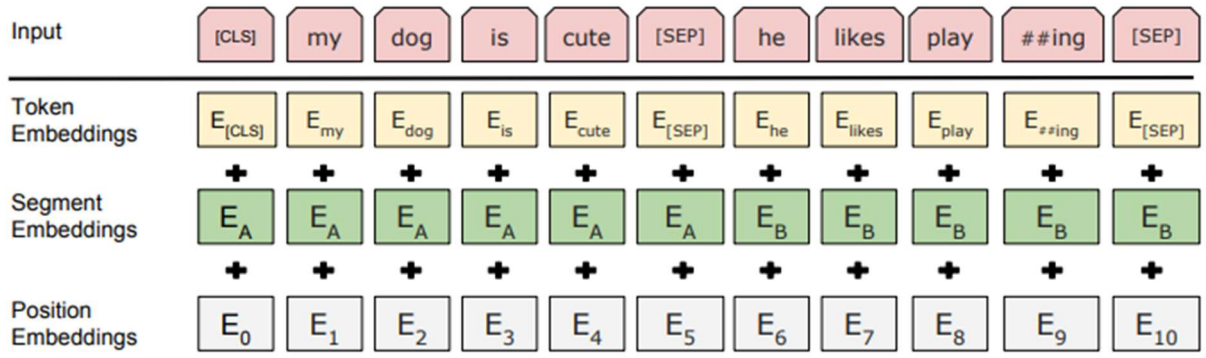
2.2.3 BERT

O BERT (*Bidirectional Encoder Representation from Transformers*) é um algoritmo de processamento de linguagem natural criado pelo Google no ano 2018, utilizando redes neurais profundas, que utilizou um corpus textual com mais de 33 milhões de itens. Esse modelo utiliza os componentes *transformers,* que são arquiteturas utilizadas para tratar dados de entradas sequenciais, ou seja, ele é um modelo que leva em consideração o contexto. Além disso, o BERT possui inicialmente 30 mil tokens mapeados em um vocabulário pré-definido através da técnica *WordPiece embeddings* [22].

Conforme Figura 5, esse modelo tem a capacidade de receber como dados de entrada (*inputs*) pares de sentenças. Existe no modelo dois tokens especiais: O CLS, que é um token inicial do BERT, é responsável por representar o embedding de toda a entrada e SEP que é responsável por separar os textos da sequência de entrada.

O BERT gera três conjuntos de embeddings, sendo o primeiro de cada um dos tokens, incluindo os tokens especiais, o segundo por segmento e o terceiro por cada posição, ou seja, é uma estratégia para lidar com a ordem das palavras.

Figura 5 – Representação do funcionamento do BERT



Fonte: [22]

O algoritmo realiza o treinamento em duas etapas: Pré-treinamento e *Fine-tuning*. Para pré-treinamento, o BERT utiliza um grande corpus textual não rotulado, para realizar duas tarefas. A primeira é a previsão de palavras mascaradas (*Masked Language Model*), onde são extraídas aproximadamente 15% das palavras e substituídos por um token especial chamado "*mask*", ou seja, o algoritmo corrompe o texto original para que a rede consiga prever as palavras mascaradas, explorando o contexto das palavras não mascaradas, conforme ilustrado na Figura 6. A outra tarefa é a previsão da próxima sentença, em que o objetivo é que o algoritmo preveja se uma determinada sequência textual A é subsequente ou não a uma sequência B.

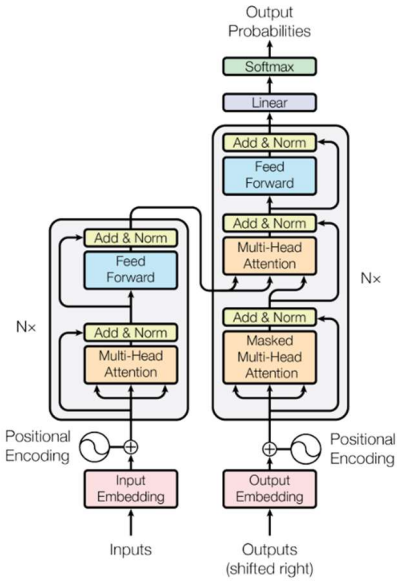
Figura 6 – Representação do funcionamento de máscara do BERT

Fonte: Própria

Por ser um trabalho muito custoso computacionalmente, geralmente é utilizando um modelo previamente treinado, fazendo apenas o processo de *fine-tuning*. A etapa de *fine-tuning* é um ajuste fino das embeddings geradas no pré-treinamento, ou seja, é um processo de baixo custo computacional quando comparado com a etapa inicial e tem o objetivo de moldar as embeddings para uma tarefa específica. Nesse trabalho, o fine-tuning é o ajuste dos pesos iniciais para classificação de discursos de ódio, através de um corpus textual específico para esse problema.

A arquitetura dos *transformers* é formada pelos módulos de *encoder* e *decoder,* que são os blocos na esquerda e direita da Figura 7, respectivamente. A função dos codificadores é processar a entrada, gerando informações sobre quais partes são relevantes entre si. Já os decodificadores fazem o caminho oposto, ou seja, utiliza as informações passadas nos *encoders* e gera uma sequência na saída. Um ponto importante e diferencial na arquitetura *transformers* é o mecanismo de atenção, que traz o contexto do processamento de cada palavra, ou seja, é um método que percebe a influência de diferentes partes dos dados da entrada para gerar o resultado final [23].

Figura 7 – Arquitetura *Transformers*



Fonte: [23]

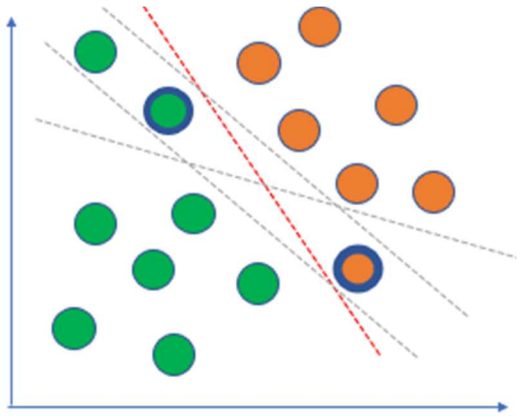
2.2.4 SVM

O SVM (*Support Vector Machine*), ou ainda máquina de vetores de suporte, é um algoritmo supervisionado de aprendizado de máquinas, muito utilizado para tarefas de classificação, apesar de também serem utilizados para atividades de regressão. Um ponto muito interessante do SVM é que ele cria limites de decisão não lineares ao projetar os dados com dimensões mais altas no espaço usando sua função não linear. Este hiperplano é utilizado no processo de separação dos dados de uma classe para outra [25], ou seja, de forma resumida, o SVM utiliza o conceito de hiperplanos para separar da maneira mais eficiente possível um conjunto de dados em um espaço *n*-dimensional (onde *n* é a quantidade coordenadas dos vetores).

Se considerarmos os elementos em um plano de 2 dimensões (x e y), o hiperplano que irá classificar ou separar o conjunto dos elementos pode ser representado por uma função de primeiro grau, com f(x) = ax + b, conforme Figura x.

No exemplo abaixo é possível observar que dos vários hiperplanos propostos, aquele que está em vermelho foi o que teve o melhor desempenho, ou seja, cometeu a menor quantidade de erros de classificação com uma boa margem de distância entre os elementos de fronteiras. Um conceito importante, que dá o nome para esse modelo, são os vetores de suporte (ou *support vectors*, em inglês), que são os elementos utilizados como fronteira entre os dois conjuntos, como pode ser visto na Figura 8, os elementos que estão destacados com borda azul.

Figura 8 – Separação de amostras por um hiperplano bidimensional



Fonte: própria

O modelo utilizado nesse trabalho foi o *one class Support Vector Machine,* é um modelo não supervisionado de *machine learning,* baseado no conceito tradicional de SVM, conforme explicado anteriormente. Essa técnica, como o próprio nome sugere, tem como alvo uma única

classe, ou seja, para o problema exposto, seria a classe de discurso de ódio. Cabe salientar que é apesar do problema ter duas classes (frases com ou sem discurso de ódio), precisamos identificar apenas aquelas que contém o discurso de ódio e nesse modelo a classe não alvo é chamada de 'outlier' [23].

Uma grande vantagem do método SVM de uma classe é que por utilizar dados não rotulados, esse algoritmo concede mais liberdade para ser aplicados em casos que tenham grandes volumes de elementos sem classificação, além disso, este modelo não tem o problema comum em que base de dados seja desbalanceada [26].

2.3 Trabalhos relacionados

Foram encontrados alguns trabalhos similares ao tema proposto neste artigo, que é aplicação de modelos de *machine learning* para classificação de textos de discurso de ódio, em língua portuguesa.

Em [26], os autores propuseram a verificação do uso de alguns algoritmos clássicos de aprendizado de máquina para a tarefa de detecção de discurso de ódio em tweets escritos em português, testando quatro modelos diferentes (SVM, MLP (*Multilayer Perceptron*), *Logistic Regression* e *Naıve Bayes)* com diferentes configurações. Nesse trabalho foi utilizado a base de textos [27], que é um dos acervos que foram utilizados no presente documento. Dos resultados obtidos pelo autor, foi verificado que o SVM obteve um F1-Score de validação de 0,6946 em situação com pré-processamento dos dados e 0,7130 para a base de teste, conforme Tabela 5 abaixo. O presente artigo apresenta uma contribuição utilizando mais de uma base para resultados, além de utilizar diferentes algoritmos destes utilizados na Tabela 5.

Tabela 5 – F1-Score dos modelos apresentados no artigo [24]

Modelo	Validação	Teste
NB	0,4647	0,4560
LR	0,6889	0,6929
SVM	0,6946	0,7130
MLP	0,6621	0,6911

Fonte: [26]

O trabalho "*Toxic language detection in social media for brazilian portuguese: new dataset and multilingual analysis*" [28], propôs um novo conjunto de dados em larga escala para o português brasileiro com tweets anotados como tóxicos ou não tóxicos ou em diferentes

tipos de toxicidade. Além disso, este arquivo constatou que o Modelos BERT foi capaz de atingir 76% de pontuação macro-F1 usando dados monolíngues no caso binário.

Em [29], no artigo nomeado "*Automatic Hate Speech Detection on Social Media: A brief survey",* os autores propuseram uma pesquisa que demonstra o estado da arte para utilização da técnica de processamento de linguagem natural (NLP) utilizada na detecção automática do discurso de ódio em redes sociais online. Foi verificado que houve aplicações utilizando o algoritmo SVM que obteve um indicador de F1-Score de 87%, enquanto outros envolvendo LSTM ficaram em torno de 85%.

De maneira não exaustiva, com a pesquisa de artigos similares realizada em acervos como IEEE-xplore (https://ieeexplore.ieee.org) , ACM library (https://dl.acm.org/) e Research Gate (https://www.researchgate.net/), utilizando palavras chaves como "*Hate speech*", "discurso de ódio", "Aprendizado de máquinas", "*machine learning*", "algoritmos", "*Text Mining*", "*Natural Language Processing*", entre outras, é possível verificar que apesar de ser um tema bastante discutido no meio acadêmico, especificamente para a língua portuguesa, ainda existe pouca contribuição e base de dados rotuladas disponíveis.

3 METODOLOGIA

Este capítulo tem o objetivo de trazer o resultado da aplicação prática do problema de classificação de *cyberbullying* no modelo BERT e no modelo *One-Class* SVM.

3.1 Corpus

Para compor o corpus que servirá para fine-tuning e teste do modelo, foram utilizadas duas bases disponíveis em português, através de mensagens do Twitter. A base intitulada "*Portuguese Hate Speech Twitter Dataset*" [27], é um conjunto de dados que contém 5.668 mensagens no Twitter, de 1.156 usuários distintos e classificadas manualmente. A outra base utilizada contém 1.253 instâncias [30], sendo a classe atribuída a cada comentário foi a escolhida por pelo menos dois dos juízes. Ao total a base utilizada nesse processo contém 6.921 mensagens de discurso de ódio, todas classificadas de forma binária, ou seja, são considerados apenas duas classes: *hate* ou *unhate*, quando é ou não um comentário de *cyberbullying*, respectivamente.

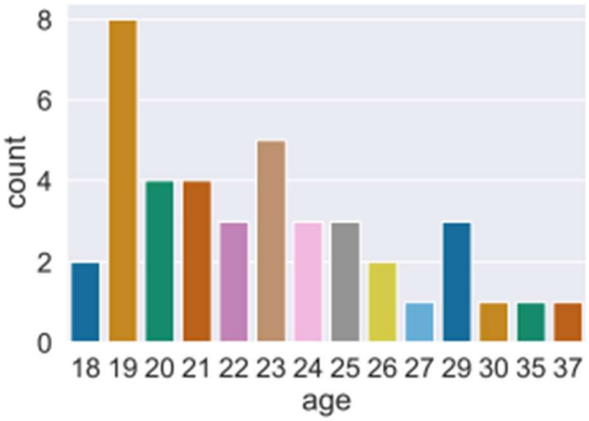
Um outro *dataset* utilizado, como forma de comparação entre os resultados dos dois modelos, foi o [31], que contém 21 mil tweets anotados entre várias categorias de discurso de ódio. O processo de anotação dessa grande base de dados foi realizado com 42 pessoas voluntárias através de uma consulta pública na universidade federal de São Carlos.

É importante salientar, conforme Figura 9, que a distribuição dos avaliadores que anotaram a base de dados foi dividida entre gênero, orientação sexual, etnia e idade. Essa distribuição é muito importante para evitar uma rotulação enviesada ou errônea de algum discurso, uma vez que discursos envolvendo homofobia, sexismo e racismo são mais bem reconhecidos por aqueles que são diretamente alvo. Além disso, como muitos discursos de ódio nas redes sociais são feitos por adolescentes, é importante também essa separação dos avaliadores por faixa etária, de forma a garantir uma rotulação consistente.

Figura 9 – Informação dos classificadores da base de textos [31]

	Categories	# annotators
Sex	Male	18
	Female	24
Sexual orientation	Heterosexual	22
	Bisexual	12
	Homosexual	5
	Pansexual	3
Ethnicity	White	25
	Brown	9
	Black	5
	Asian	2
	Non-Declared	1

Table 1: Annotators demographic information.



Fonte: [29].

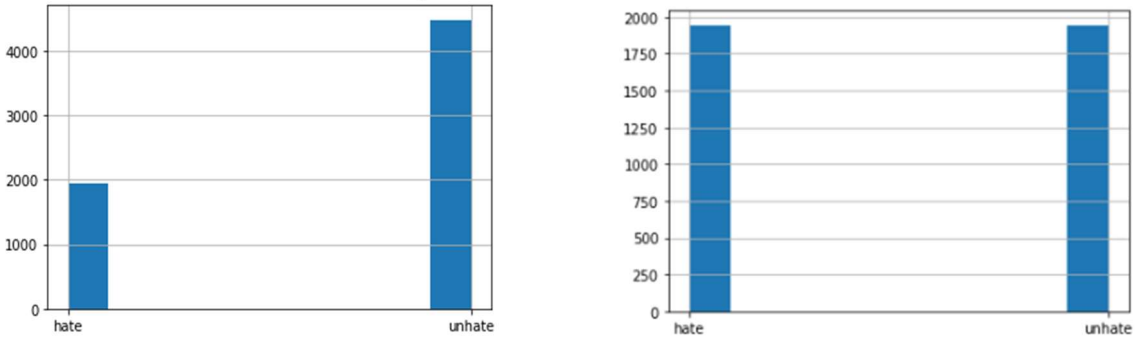
Todos os conjuntos de textos utilizados neste trabalho contêm exemplos de mensagens envolvendo sexismo, que é o discurso de ódio contra mulher; discurso de ódio baseado no corpo; xenofobia; homofobia; racismo; intolerância religiosa e outras como insultos relativos a condições de saúde, como pessoas com deficiência [27, 30, 31].

3.2 BERT

3.2.1 Base de dados 1

A base de texto foi dividida randomicamente com aproximadamente 93% das instâncias para treino e 7% para teste. Um ponto importante na base de dados de treino é que ela estava altamente desequilibrada, ou seja, possuía muito mais exemplos da classe "*unhate*" (não discurso de ódio), do que da classe "*hate*", Figura 10a. Dessa forma, foi utilizada a técnica de *undersampling* para balancear a base, resultando em um total de 1.945 *tweets* para cada classe, Figura 10b.

Figura 10 – Base textual de treino 1



a) b)

Fonte: própria.

Conforme a Tabela 6, após a técnica para o balanceamento entre as classes, 3.890 exemplos foram utilizados para treino e validação (cerca de 89% do novo conjunto balanceado), os outros 11% foram usados como teste, sendo divididos em 53% da classe *unhate* e 47% da classe *hate.*

Tabela 6 – Separação da base de texto 1

Classe	Treino/Validação	Teste
Unhate	1.945	262
Hate	1.945	238

Fonte: Própria

Após o treinamento do modelo utilizando o BERTimbau, que é uma versão pré-treinada do algoritmo BERT para língua portuguesa, foi realizado o *fine-tuning* utilizando a base de dados de treino apresentada anteriormente. Foi realizado o processamento de apenas duas épocas, devido ao modelo não obter evolução com mais épocas, com um learning rate de 0.00002 e o max-len de 52, Figura 11.

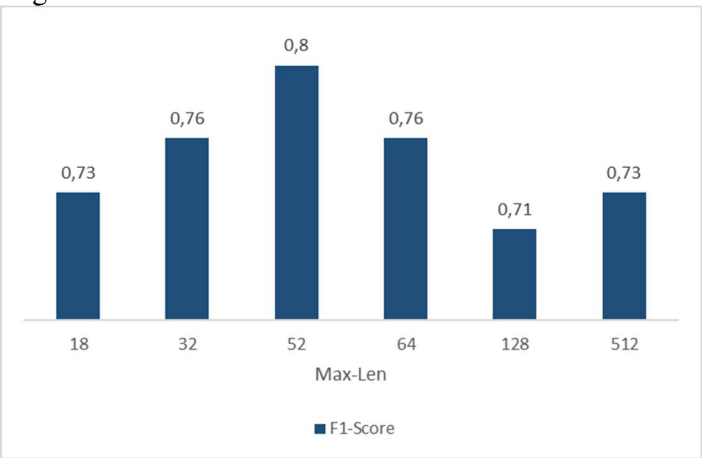
Figura 11 – *Fine-Tuning* do modelo

```
Epoch 1/2
98/98 [==============================] - 56s 317ms/step - loss: 0.6612 - accuracy: 0.5922 - val_loss: 0.6068 - val_accuracy: 0.6697
Epoch 2/2
98/98 [==============================] - 25s 257ms/step - loss: 0.5308 - accuracy: 0.7429 - val_loss: 0.5851 - val_accuracy: 0.7108
<keras.callbacks.History at 0x7fda264b4410>
```

Fonte: própria.

O ajuste do parâmetro max-len é importante para o melhor desempenho do modelo, podendo receber valores até 512, porém para textos curtos, como é o caso da base em questão, é interessante trabalhar com números menores, conforme demonstrado na Figura 12.

Figura 12 – Treinamento com diversos valores de max-len



Fonte: própria.

Para avaliar o desempenho do modelo, na parte de teste, foi utilizado três indicadores bem conhecidos: precisão, revocação e f1-score. A precisão é matematicamente definida pela razão entre a soma de todos os objetos corretamente detectados (TP) e o total de exemplos detectados como positivos pela rede (TP e FP), conforme a fórmula 4 [32]. Ela pode ser interpretada como o percentual de predições realmente corretas do total de detecções que a rede apontou, ou seja, quanto mais próxima de 100%, significa que as detecções do modelo são corretas para as duas classes (*hate* e *unhate*).

$$Precisão = \frac{TP}{TP + FP} \tag{4}$$

A revocação indica o percentual de acertos da rede em relação a todas as anotações realizadas, ou seja, quanto mais próxima de 100%, demonstra que a rede está acertando tudo que foi anotado como *ground-truth* e não tendo falhas de detecção. Matematicamente definida pela razão entre a soma dos verdadeiros positivos e a soma de todos os objetos anotados (TP + FN), conforme equação 5.

$$Revocação = \frac{TP}{TP + FN} \tag{5}$$

O F1-Score é a relação da média harmônica entre o precision e o recall, conforme equação 6. Isso significa que quanto maior for o recall e o precision, maior também será o F1-Score. O maior problema dessa métrica é o fato da dificuldade de interpretação.

$$F1 - Score = 2 * \frac{precisão\ x\ revocação}{(precisão + revocação)} \tag{6}$$

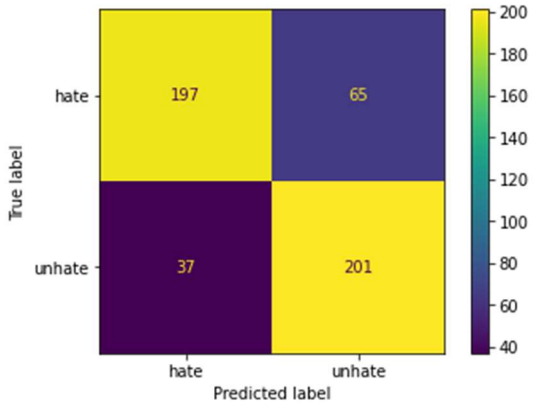
Cabe salientar que conceitualmente para esse problema, a revocação é muito importante, porque o objetivo é cometer o mínimo de erros possíveis na classificação de discurso de ódio, ou seja, evitar que mensagens de cyberbullying não sejam identificadas. Por esse motivo, neste trabalho a revocação terá um peso maior que a precisão, quando do critério de escolha como um bom desempenho.

Conforme Figura 13, os resultados obtidos no teste com o modelo BERT na versão em português, foi uma precisão e revocação média de 80%. Quando individualmente analisadas, a classe "*hate*" que é o maior objetivo do trabalho, obteve uma precisão de 84%, porém uma revocação de 75%. Na matriz de confusão, percebe-se que apesar de existir 65 falsos negativos,

o modelo acertou 197 classificação de sentenças da classe alvo, o que é um bom indicativo para o desempenho da rede.

Figura 13 – Resultados obtidos no modelo BERT para base 1

	precision	recall	f1-score
hate	0.84	0.75	0.79
unhate	0.76	0.84	0.80
accuracy			0.80
macro avg	0.80	0.80	0.80
weighted avg	0.80	0.80	0.80



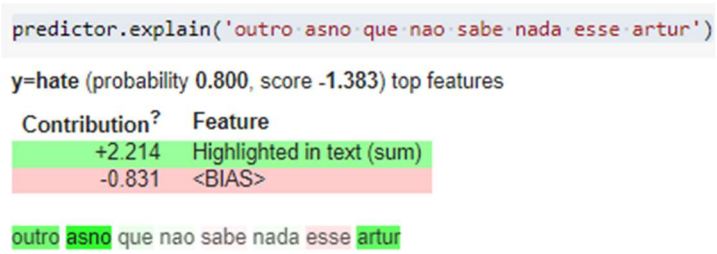
Fonte: própria.

Foi utilizado o algoritmo "Eli5", que é um pacote em python para fazer a explicabilidade da classificação do modelo, com o objetivo de achar um padrão para os erros de falso negativo cometidos na classe hate. É importante salientar que na tarefa de PNL, identificar textos com sentido Figurado, sarcasmo ou ironia, ainda é uma atividade muito complexa para os modelos. Essa problemática fica claro na Figura 14a, em que a frase "vai se tratar doente" foi classificada pelo modelo como *unhate,* com uma probabilidade de aproximadamente 66%. Acontece, porém, que essa frase pode ser confundida até mesmo por humanos, a depender do contexto, uma vez que é possível pensar na frase como um conselho para uma pessoa que está doente.

Em contrapartida, na Figura 14b, percebe-se que o texto com uma narrativa ofensiva foi mais fácil para o modelo classificar corretamente. Na explicabilidade da frase "Outro asno que não sabe nada, esse Artur", é possível verificar que o modelo identificou a palavra asno como uma ofensa, e classificou a sentença com uma probabilidade de 80%.

Figura 14 – Exemplo da explicabilidade do modelo BERT

a)



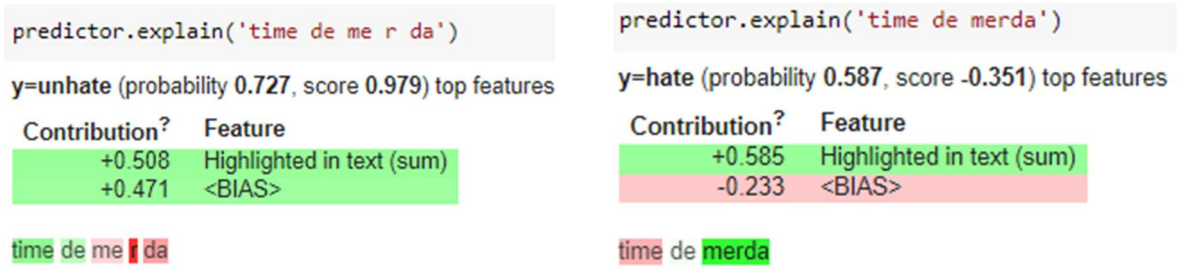
b)

Fonte: própria.

Outros tipos de sentenças que foram classificadas de forma errada pelo modelo, em que o *ground-truth* é da classe *hate* e a predição foi da classe *unhate*, tiveram um grau de complexidade maior devido a situações específicas, como por exemplo:

- "**p i r a n h a**" – Essa sentença teve todas as letras da palavra separadas por espaços.
- "**time de me r da**" – Similar ao exemplo anterior, essa frase também teve a palavra separada, o que faz com que o modelo tenha dificuldade de prever. Essa afirmação fica claro, se observarmos a Figura 15, em que o modelo classifica errado quando a palavra "merda" está separada e corretamente quando está junta.

Figura 15 – Classificação de frases complexas pelo modelo BERT



Fonte: própria.

- "**Colar velcro perdeu o meu respeito**" – Essa frase é um discurso de ódio envolvendo homofobia, porém com um sentindo Figurado. Seria necessário um bom entendimento de contexto para que essa expressão pejorativa fosse corretamente identificada.
- "**So digo outra coisa fdss**" – Uma outra sentença complicada de ser corretamente classificada pelo modelo devido ao termo 'fdss' ser uma abreviação não trivial para um insulto.
- "**vc é jumento**", "**aprende a escrever cavalo**" – Esses são exemplos em que os substantivos referentes a animais são utilizados de forma Figurada e depende do contexto para uma compreensão correta.

3.2.2 Base de dados 2

Conforme a Tabela 7, a segunda base de textos foi dividida com 18.900 exemplos foram utilizados para teste e validação (cerca de 90%) do conjunto inteiro, os outros 10% foram usados como teste, sendo divididos em 77% da classe *unhate* e 23% da classe *hate.* Apesar dos conjuntos estarem desbalanceados, a princípio, propositalmente para esse teste não foi utilizada

a técnica de balanceamento, como foi feito no item anterior. Esse propósito foi para analisar uma base de textos mais próximos do que é encontrada em ambientes reais.

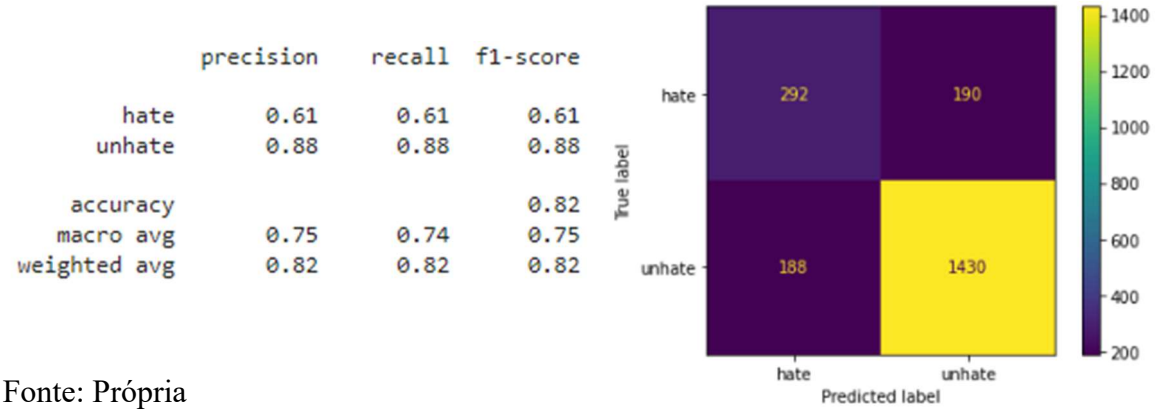
Tabela 7 – Separação da base de texto 2

Classe	Treino/Validação	Teste
Unhate	14.566	1.618
Hate	4.334	482

Fonte: Própria

Aplicando o mesmo modelo na segunda base de dados, conforme Figura 16, verifica-se que o BERT obteve uma precisão média de 75%, uma revocação de 74% e um F1-score de 75%. Quando analisado separadamente por classes, é possível notar que o grupo "*unhate*" obteve melhores indicadores do que a classe *hate*. 190 exemplos em que o rótulo original era ofensivo foi classificado como não ofensivos, e 188 casos aconteceu o contrário, ou seja, o *ground-truth* era "*unhate*" e foi erroneamente classificado como "*hate*".

Figura 16 – Resultados do modelo BERT para base 2



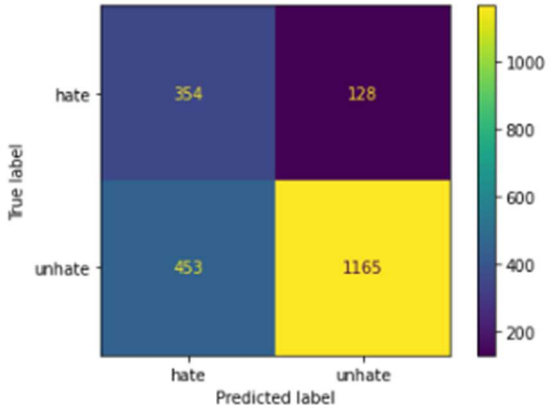
Fonte: Própria

Após aplicar a técnica de *undersampling,* e com isso balancear o acervo de dados utilizados no treinamento do modelo, cada uma das classes ficou com 4.334 dados, resultando em número total de 8.668 dados de treinamento. O modelo foi novamente aplicado para essa configuração balanceada e resultou em uma precisão média de 67%, uma revocação de 73% e um F1-score de 67%, conforme Figura 17.

Apesar do desempenho global ser um pouco inferior do que o modelo antes do balanceamento, o número de erros de classificação do grupo *hate* foi inferior, cerca de 32%, o que é muito positivo, considerando que é muito importante que frases de discursos de ódio não deixem de ser identificadas.

Figura 17 – Resultados do modelo BERT para base 2 após *undersampling*

	precision	recall	f1-score
hate	0.44	0.73	0.55
unhate	0.90	0.72	0.80
accuracy			0.72
macro avg	0.67	0.73	0.67
weighted avg	0.79	0.72	0.74



Fonte: Própria

3.3 One-Class Support Vector Machines

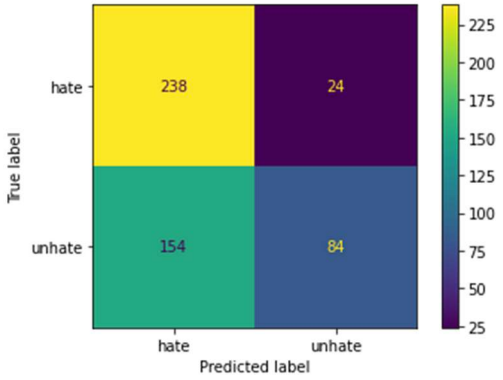
3.3.1 Base de dados 1

Uma outra abordagem para o problema foi coletar as embeddings do modelo BERT para treinar o algoritmo *One-Class Support Vector Machines* (SVM), utilizando apenas a classe de interesse (*hate*), como o próprio nome do modelo sugere. Essa técnica é interessante porque todo o treinamento é focado em uma única classe, podendo melhorar significativamente o desempenho em relação aos modelos binários ou multiclasses.

Após o treinamento do SVM utilizando os embeddings obtidos do modelo BERT, para a mesma base de dados, porém utilizando apenas a classe *hate* para treinamento, obteve-se uma revocação de 91% para a classe alvo, com uma precisão de 61% e um f1-score de 73%, conforme Figura 18. É importante salientar novamente, que a revocação é considerado o parâmetro prioritário, uma vez que o objetivo é que o modelo não deixe de classificar discursos de ódios em comentários das redes sociais.

Figura 18 – Resultados obtidos no modelo *One-Class* SVM para base 1

	precision	recall	f1-score
hate	0.61	0.91	0.73
unhate	0.78	0.35	0.49
accuracy			0.64
macro avg	0.69	0.63	0.61
weighted avg	0.69	0.64	0.61

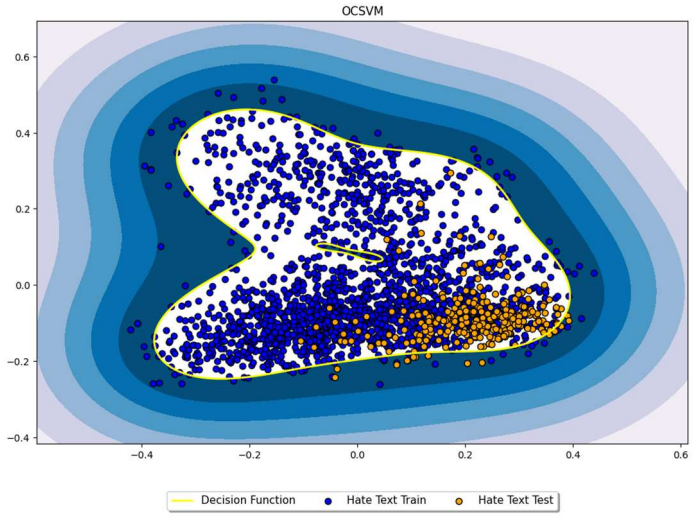


Fonte: própria.

A Figura 19 representa a delimitação da função que agrega todos os resultados, após a

redução de dimensionalidade PCA para 2 dimensões apenas. Observa-se que os dados que estão dentro do hiperplano são aqueles que o modelo classificou como classe *hate*, em azul as sentenças de treinamento e em laranja as sentenças de teste.

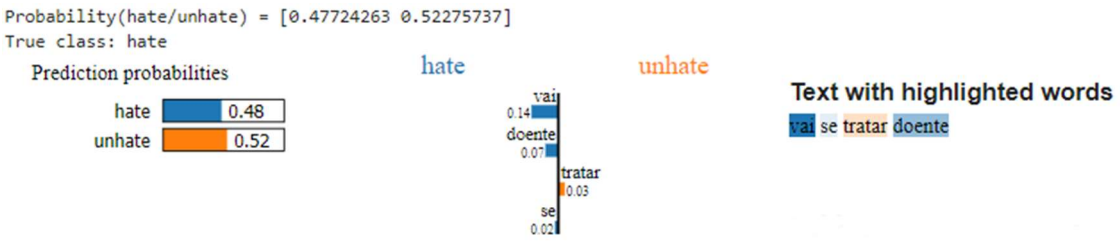
Figura 19 – Visualização do hiperplano do SVM após redução de dimensionalidade



Fonte: própria.

Os autores de [33] propuseram um método de explicabilidade de PLN para o modelo utilizando SVM. Apesar da revocação desse algoritmo ter sido superior ao BERT, é possível perceber que ele também cometeu erro de classificação em situações mais complexas, como sentido Figurado, conforme Figura 20.

Figura 20 – Exemplo da explicabilidade do modelo *One-Class* SVM



Fonte: própria.

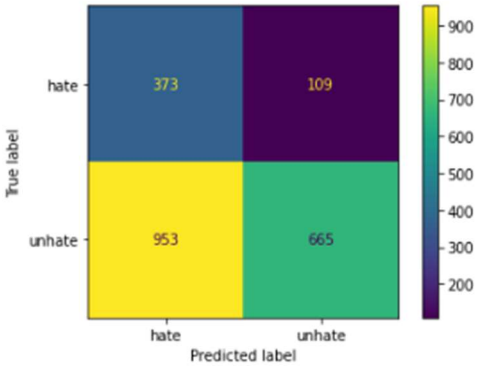
3.3.2 Base de dados 2

Aplicando o modelo *one class* SVM, com a base de treino 2, utilizando apenas os dados da classe *hate*, uma vez que esse algoritmo utiliza apenas dados de uma única classe,

é possível verificar que foi obtida uma revocação média global de 59%, com uma precisão de 57% e um f1-score de 48%, conforme Figura 21. Apesar de ter um desempenho global inferior ao modelo BERT, é importante salientar que a revocação da classe *hate* novamente foi superior nesse *dataset*.

Figura 21– Resultado do *One Class* SVM para base de dados 2

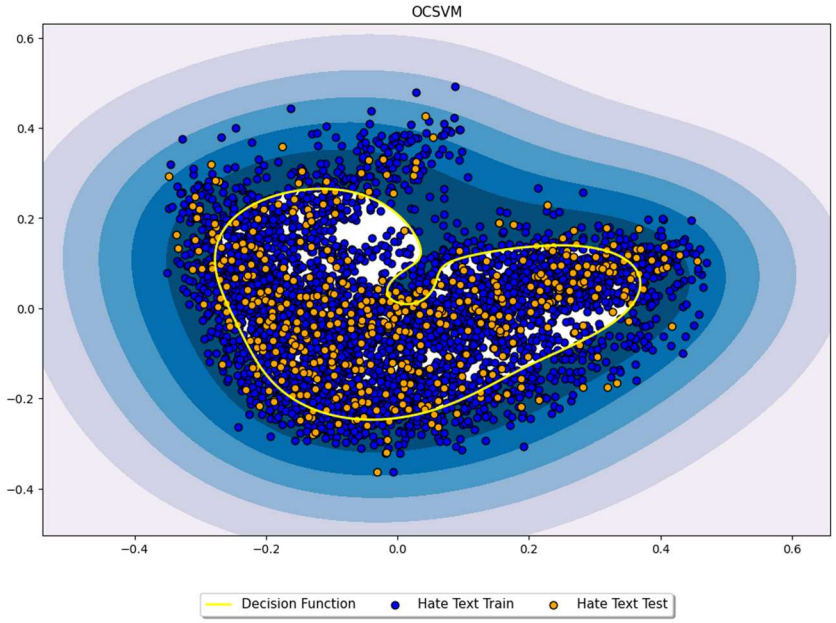
	precision	recall	f1-score
hate	0.28	0.77	0.41
unhate	0.86	0.41	0.56
accuracy			0.49
macro avg	0.57	0.59	0.48
weighted avg	0.73	0.49	0.52



Fonte: própria.

A Figura 22 representa a delimitação da função que agrega todos os resultados, após a redução de dimensionalidade PCA para 2 dimensões apenas, conforme aplicado no item anterior. Observa-se que os dados que estão dentro do hiperplano são aqueles que o modelo classificou como classe *hate*, em azul as sentenças de treinamento e em laranja as sentenças de teste.

Figura 22 – Visualização do hiperplano do SVM após redução de dimensionalidade



Fonte: própria.

Observa-se que a linha de decisão não está delimitando da melhor maneira possível os exemplos de discurso de ódio de treino e teste, porém, caso fosse expandida, a revocação do

modelo hate iria melhorar muito, enquanto a classe *unhate* teria uma revocação muito pequena, ou seja, é como se o modelo estivesse classificando praticamente todos os exemplos como a classe alvo, o que não é um resultado viável para aplicação proposta de classificação eficaz.

3.4 Comparação dos resultados

Após a aplicação dos 2 (dois) algoritmos para classificação textual de mensagens de cyberbullying, é possível verificar, na Tabela 8 e 9, que o BERT, utilizando o modelo BERTimbau, que foi pré-treinado para língua portuguesa, obteve o melhor resultado para as duas bases de texto aplicadas.

Tabela 8 – Comparação entre os modelos na base de dados 1

Algoritmo	Acurácia	Precisão	Revocação	F1- Score
BERT	**80%**	**80%**	**80%**	**80%**
SVM	64%	69%	63%	61%

Fonte: própria.

Tabela 9 – Comparação entre os modelos na base de dados 2

Algoritmo	Acurácia	Precisão	Revocação	F1- Score
BERT	**72%**	**67%**	**73%**	**67%**
SVM	49%	57%	59%	48%

Fonte: própria.

Em contrapartida é importante salientar que o modelo SVM de uma classe obteve uma precisão na classe *hate* superior ao modelo BERT, para as duas bases textuais aplicadas, e esse fato é importante uma vez que o principal objetivo é identificar as mensagens de discurso de ódio, e, portanto, a detecção de verdadeiros positivos da classe "*hate*" é fundamental.

4. CONCLUSÃO

O Cyberbullying, considerado como um grave problema de saúde pública pelo centro de controle e prevenção de doenças, têm crescido ao passar dos anos, principalmente entre as crianças e adolescentes, auxiliado pelo aumento e difusão das redes sociais. Os tipos de ofensa do cyberbullying podem ter cunhos racista, homofóbicos, sexistas, ou estar ligado a depreciação do indivíduo, através de discursos de ódio, e é considerado como crime, estando associadas àqueles crimes previstos no decreto-lei no 2.848, de 7 de dezembro de 1940, do código penal brasileiro, como calúnia, difamação, injúria, ameaça e constrangimento ilegal, podendo ainda a pena ser agravada em caso de ser realizado pela Internet.

Como forma de auxiliar a mitigar o cyberbullying em meios digitais, tais como Twitter, Facebook e Instagram, o objetivo deste trabalho foi treinar algoritmos de mineração de texto existentes, com capacidade de identificar mensagens de ofensa relacionadas ao cyberbullying, realizando comparação entre eles.

Dessa maneira, foram utilizados dois algoritmos para a tarefa de classificação de textos, o BERT e o *One Class* SVM. O treinamento ocorreu utilizando duas bases de dados com *tweets* em língua portuguesa, sendo que a primeira teve um total de 6.921 mensagens de discursos de ódio, formadas pelas bases [27] e [30]. Devido ao grande desbalanceamento da base, foi aplicada a técnica de *undersampling* para equalizar os exemplos de cada classe para treinamento. A segunda base, [31], contém 21 mil tweets anotados entre várias categorias de discurso de ódio, onde o processo de anotação foi realizado por 42 pessoas voluntárias de diferentes gêneros, orientação sexual, etnia e idade.

Como resultado, o modelo BERT com a utilização do BERTimbau, que é o modelo pré treinado para língua portuguesa, obteve um F1-Score total de 80% para a primeira base, com uma precisão e revocação também de 80%. O *One Class* SVM, obteve um F1-Score global de 61%, com uma precisão de 67%. Com a segunda base de textos, foi encontrado um resultado de F1-Score de 67% para o BERT e 48% para o outro modelo.

É importante salientar que o modelo *One Class* SVM, no primeiro teste, apesar dos resultados globais serem inferiores ao BERT, obteve uma revocação de 91% para a classe alvo, sendo que a revocação é considerado o parâmetro prioritário, uma vez que o objetivo é que o modelo não deixe de classificar discursos de ódios em comentários das redes sociais.

Outra conclusão que ficou evidente na aplicação deste trabalho foi a dificuldade de os modelos identificarem textos com sentido figurado, sarcasmo ou ironia. Dessa forma, como trabalhos futuros, é identificado a necessidade de treinar outros algoritmos que possam ter

desempenho melhores na classificação, principalmente aqueles que trarão uma boa compreensão semântica para classificar exemplos complexos. Uma outra tarefa é melhorar a base de dados de treinamento, com aplicação de modelos de pré-processamento mais robustos e cuidadosos, a fim de verificar se o desempenho do modelo aumenta nessa situação.

REFERÊNCIAS

[1] Tokunaga. **Following you home from school: a critical review and synthesis of research on cyberbullying victimization**. Comput Hum Behav. 2010;26:277–87.

[2] Centers for Disease Control and Prevention (CDC). **The relationship between bullying and suicide: what we know and what it means for schools**. 2014. Disponível em: https://www.cdc.gov/violenceprevention/pdf/bullying-suicide-translation-final-a.pdf

[3] Centers for Disease Control and Prevention (CDC). **#StopBullying**. Disponível em: https://www.cdc.gov/injury/features/stop-bullying/. Acessado em 18/09/2021 às 13:54.

[4] Carta Capital. **Os alertas deixados pelo suicídio de Lucas, um adolescente vítima do ódio e da LGBTfobia no TikTok**. 2021. Disponível em: https://www.cartacapital.com.br/diversidade/os-alertas-deixados-pelo-suicidio-de-lucas-um-adolescente-vitima-do-odio-e-da-lgbtfobia-no-tiktok/. Acessado em : 01/12/2021.

[5] Comparitech. **Cyberbullying facts and statistics for 2018 – 2021**. Disponível em: https://www.comparitech.com/Internet-providers/cyberbullying-statistics/. Acessado em 18/09/2021 às 14:21.

[6] Ditch the Label. **The annual bullying survey 2020**. 2020. Disponível em: https://www.ditchthelabel.org/research-papers/the-annual-bullying-survey-2020. Acessado em: 18/09/2021 às 13:54.

[7] Ditch the Label. **The annual bullying survey 2017**. 2017. Disponível em: https://www.ditchthelabel.org/wp-content/uploads/2017/07/The-Annual-Bullying-Survey-2017-1.pdf. Acessado em: Acessado em 29/11/2021 às 20:01.

[8] **Internet live stats. Twitter usage statistics**. Disponível em: https://www.Internetlivestats.com/twitter-statistics/. Acessado em: Acessado em 18/09/2021 às 14:31.

[9] MOURA, M. F. **Proposta de utilização de mineração de textos para seleção, classificação e qualificação de documentos**. Embrapa Informática Agropecuária, 2004, ISSN 1677-9274, 2004.

[10] Kowsari, Kamran & Jafari Meimandi, Kiana & Heidarysafa, Mojtaba & Mendu, Sanjana & Barnes, Laura & Brown, Donald & Id, Laura & Barnes,. (2019). **Text Classification Algorithms: A Survey.** Information (Switzerland). 10. 10.3390/info10040150.

[11] Nasteski, Vladimir. (2017). **An overview of the supervised machine learning methods**. HORIZONS.B. 4. 51-62. 10.20544/HORIZONS.B.04.1.17.P05.

[12] B, Priya & J.M, Nandhini & Thangavel, Gnanasekaran. (2021). **An Analysis of the Applications of Natural Language Processing in Various Sectors**. 10.3233/APC210109.

[13] A. P. Jain and P. Dandannavar, "**Application of machine learning techniques to sentiment analysis**," *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016, pp. 628-632, doi: 10.1109/ICATCCT.2016.7912076.

[14] Ikonomakis, Emmanouil & Kotsiantis, Sotiris & Tampakas, V.. (2005). **Text Classification Using Machine Learning Techniques**. WSEAS transactions on computers. 4. 966-974.

[15] G, Nathiya & Punitha, S. & Punithavalli, Dr. (2010**). An Analytical Study on Behavior of Clusters Using K Means, EM and K* Means Algorithm**. International Journal of Computer Science and Information Security. 7.

[16] GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow**. O'Reilly Media, 2017.

[17] Data Science Academy. **Deep Learning Book, capítulo 6, 2021**. Disponível em: <https://www.deeplearningbook.com.br/>. Acesso em: 02 dezembro. 2021.

[18] JURAFSKY, D; MARTIN, J. **Speech and Language processing (3rd ed.)**, https://web.stanford.edu/jurafsky/spl3/, 2018.

[19] C. Fonseca. **Word Embedding: fazendo o computador entender o significado das palavras**. 2021. Disponível em: https://medium.com/turing-talks/word-embedding-fazendo-o-computador-entender-o-significado-das-palavras-92fe22745057.

[20] Word2Vec. 2018. Disponível em: https://code.google.com/archive/p/word2vec/ .

[21] Repositório de Word Embeddings do NILC. Disponível em: http://www.nilc.icmc.usp.br/embeddings.

[22] J. Devlin, et al., **BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding**. https://arxiv.org/abs/1810.04805. 2019.

[23] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Lion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia, "**Attention is all you need**" , 2017.

[24] Manevitz, Larry & Yousef, Malik. (2001). **One-Class SVMs for Document Classification**. Journal of Machine Learning Research. 2. 139-154.

[25] Zhuang, L., Dai, H. (2006). **Parameter Estimation of One-Class SVM on Imbalance Text Classification**. In: Lamontagne, L., Marchand, M. (eds) Advances in Artificial Intelligence. Canadian AI 2006. Lecture Notes in Computer Science(), vol 4013. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11766247_46

[26] Silva, Adriano & Roman, Norton. (2020). **Hate Speech Detection in Portuguese with Naïve Bayes, SVM,MLP and Logistic Regression.**

[27] Fortuna, P., Rocha da Silva, J., Soler-Company, J., Warner, L. and Nunes, S., 2019. **A Hierarchically-Labeled Portuguese Hate Speech Dataset. In: Proceedings of the Third Workshop on Abusive Language Online**. Florence, Italy: Association for Computational Linguistics,pp.94-104.
data: https://b2share.eudat.eu/records/9005efe2d6be4293b63c3cffd4cf193e.

[28] Leite, J.A. & Silva, D.F. & Bontcheva, Kalina & Scarton, Carolina. (2020). **Toxic language detection in social media for Brazilian Portuguese : new dataset and multilingual analysis**.

[29] Alrehili, A. (2019). **Automatic Hate Speech Detection on Social Media: A Brief Survey**. 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)

[30] de Pelle, R. and Moreira, V., 2017. **Offensive Comments in the Brazilian Web: A Dataset and Baseline Results**. In: VI Brazilian Workshop on Social Network Analysis and Mining. SBC.
data: https://github.com/rogersdepelle/OffComBR

[31] João A. Leite, Diego F. Silva, Kalina Bontcheva, Carolina Scarton (2020): **Toxic Language Detection in Social Media for Brazilian Portuguese: New Dataset and Multilingual Analysis.** Published at AACL-IJCNLP 2020.

[32] TING, Kai. **Encyclopedia of machine learning.** 2011. Springer. ISBN 978–0–387–30164–8.

[33] Ribeiro, M.T., Singh, S. and Guestrin, C., 2016, August. " **Why should i trust you?**" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).