UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Classificação Automática de Discursos de Ódio em Textos do Twitter

Por

Robson Murilo Ferreira do Nascimento

Serra Talhada,
Janeiro/2019

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE SERRA TALHADA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ROBSON MURILO FERREIRA DO NASCIMENTO

Classificação Automática de Discursos de Ódio em Textos do Twitter

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Unidade Acadêmica de Serra Talhada da Universidade Federal Rural de Pernambuco como requisito parcial à obtenção do grau de Bacharel.

Orientador: Prof. Dra. Ellen Polliana Ramos Souza

Serra Talhada,
Janeiro/2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca da UAST, Serra Talhada - PE, Brasil.

N244c Nascimento, Robson Murilo Ferreira do

Classificação automática de discursos de ódio em textos do twitter / Robson Murilo Ferreira do Nascimento. – Serra Talhada, 2019.

46 f.: il.

Orientadora: Ellen Polliana Ramos Souza

Trabalho de Conclusão de Curso (Graduação em Bacharelado em Sistemas de Informação) – Universidade Federal Rural de Pernambuco. Unidade Acadêmica de Serra Talhada, 2019.

Inclui referências, anexos e apêndices.

1.Twitter (Rede social on-line). 2. Discurso de ódio na internet. 3. Mineração de texto. I. Souza, Ellen Polliana Ramos, orient. II. Título.

CDD 004

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE SERRA TALHADA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ROBSON MURILO FERREIRA DO NASCIMENTO

Classificação Automática de Discursos de Ódio em Textos do Twitter

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, defendida e aprovada por unanimidade em 18/01/2019 pela banca examinadora.

Banca Examinadora:

Prof. Dra. Ellen Polliana Ramos Souza
Orientadora
Universidade Federal Rural de Pernambuco

Prof. Arthur Diego de Godoy Barbosa
Universidade Federal Rural de Pernambuco

Prof. Dr. Sérgio de Sá Leitão Paiva Júnior
Universidade Federal Rural de Pernambuco

RESUMO

Discurso do ódio, ou no inglês *Hate Speech*, pode ser definido como qualquer ato de comunicação que inferiorize uma pessoa por sua etnia, raça, religião, orientação sexual, nacionalidade ou outras características. Esse ato está se tornando cada vez mais comum nas redes sociais, onde muitas pessoas confundem liberdade de expressão com intolerância. Os jovens são os principais afetados, pois representam um grupo mais fácil de ser atingido pela ideologia propagada pelos *Haters*, os quais exaltam a violência, adotam ideologias racistas e xenofóbicas, intolerância religiosa e etc. Uma ferramenta capaz de ajudar a combater esse problema, é a Mineração de Texto, que busca extrair regularidades, padrões ou tendências de textos em linguagem natural, assim podendo ser definida como um método de extração de informações relevantes em bases de dados não estruturadas ou semi-estruturadas. Considerando o Twitter como uma das redes sociais mais utilizadas no Brasil, este trabalho tem como objetivo de implementar e avaliar técnicas supervisionadas de aprendizagem de máquina, com intuito de identificar de forma automática discurso de ódio em *tweets*. Para isso, foram utilizados dois corpus, um na língua inglesa, previamente disponibilizado, e outro com a língua português do Brasil,o qual foi montado com texto do *Twitter*, que posteriormente parte dele foi anotado de forma manual, e ambos passaram por um pré-processamento, a fim de criar coleções douradas, utilizadas para construção e avaliação dos modelos supervisionados. Por fim, foi realizada uma análise comparativa dos algoritmos de aprendizagem de máquina: SVM, Naive-Bayse e Regressão Logística, combinados com a técnica de processamento de linguagem natural *stemming*.

Palavras-chave: Discurso de ódio. Mineração de texto. Twitter. Português. Inglês.

ABSTRACT

Hate Speech can be defined as any communication that denigrates a person by their ethnicity, race, religion, sexual orientation, nationality or other characteristics. This behavior is becoming increasingly common in social networks, where many people confuse freedom of expression with intolerance. Young people are the main users affected since they represent a portion which might be easier to be influenced by the ideology propagated by haters, which in turn spread violence, racism, xenophobia, religious intolerance, etc. The tool that might help to handle this issue is Text Mining, which is capable of capture patterns or trends of texts in natural language. This task can be defined as a method of extracting relevant information in unstructured databases or semi-structured. Given that the Twitter is one of the most used social networks in Brazil, this work aims to implement and evaluate supervised machine learning techniques in order to automatically identify hate speech in tweets. With that in mind, we build a corpus with data collected from Twitter and part of it is manually annotated and subsequently preprocessed so we can obtain the ground truth collection used for training and evaluation of the supervised models. Finally, we conduct a comparison between machine learning algorithms, namely the SVM, Naive-Bayes and Logistic Regression. Later, we identify the best model under the described domain.

Keywords: Hate speech, Text mining, Twitter, Portuguese. English.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SVM	Support Vector Machine
NB	Naive-Bayes
API	Application Programming Interface
NLTK	Natural Language Tookit
KNN	K-Nearest Neighbor
TF-IDF	Term Frequency–Inverse Document Frequency

SUMÁRIO

1 Introdução

Neste capítulo é apresentada uma prévia do contexto que envolve este trabalho. A Seção 1.2 retrata a motivação que levou a elaboração do mesmo. Enquanto que a Seção 1.3 apresenta as justificavas. Na Seção 1.4, demarcam-se os objetivos gerais e específicos. A Seção 1.5 fornece uma visão sobre a organização dos capítulos.

1.1 Contextualização

Com notório crescimento das redes sociais nos últimos tempos, também cresceu o número de usuários presentes no mundo online, os quais expressam suas opiniões diariamente de forma livre sobre os mais diversos tipo de assuntos. Tanta facilidade vem fazendo com que parte dessas pessoas percam a noção do que devem postar para o público, assim confundindo liberdade de expressão com intolerância, e como consequência surgem os discursos de ódio ou no inglês *Hate Speech*, que pode ser definido como qualquer ato de comunicação que inferiorize uma pessoa por sua etnia, raça, religião, orientação sexual, nacionalidade ou outras características (NOCKLEBY, 2000).

O discurso do ódio pode entrar em conflito com o direito à liberdade de expressão, que é garantido por diversos documentos e legislações internacionais, os quais são de essencial importância para a democracia (ARTIGO-19, 2014). Como exemplo, a Declaração Universal dos Direitos Humanos, adotada pela Assembleia Geral das Nações Unidas em 10 de dezembro de 1948, em seu Artigo 19 garante que *"toda pessoa tem direito à liberdade de opinião e expressão; este direito inclui a liberdade de, sem interferência, ter opiniões e de procurar, receber e transmitir informações e idéias por quaisquer meios e independentemente de fronteiras"*.

Porém, por outro lado, esse direito não é absoluto, ou seja, ele é limitado por outros direitos consagrados, como o direito à imagem, à intimidade, à honra, etc. Diante disso, são impostas responsabilidades e restrições ao exercício da liberdade de expressão (ARTIGO-19, 2014).

Segundo Chau e Xu (2007) os jovens são os principais alvos dos *haters*, nome dado aos usuários que deliberam ódio na Internet, já que representam um grupo mais fácil de serem

afetados pelos ideais propagados. As autoridades pregam que esses grupos devem ser analisados, com intuito de monitorar suas atividades potencialmente prejudiciais a sociedade (MONDAL; SILVA; BENEVENUTO, 2017). Sendo assim, a identificação automática de discurso de ódio nas redes sociais representa uma importante ferramenta para controle destes grupos (ALMEIDA et al., 2017).

1.2 Motivação

Durante três meses, no Brasil, um estudo foi realizado pelo projeto "Comunica que Muda", iniciativa da agência nova/sb, o qual monitorava redes sociais como Facebook, Twitter, Instagram, entre outros vários blogs e sites, em buscas de menções relacionadas a temas delicados como racismo, posicionamento político e homofobia, foram 542.781 menções coletadas, e segundo Nova/SB, 84% dos comentários sobre esses assuntos continham uma abordagem negativa.

Dentre as várias redes sociais existentes, pode-se destacar o Twitter a qual é uma das principais em relação as opiniões pessoais, pelo fato da própria rede estimular os usuários a expressarem seus pensamentos (TELES, 2016). O microblog gera mais de 500 milhões de *tweets* por dia (INTERNETLIVESTATS, 2018), e, em 2016, o Brasil era o sexto país com maior número de usuários ativos, cerca de 17.97 milhões (STATISTA, 2016). Uma enorme gama de dados é gerada diariamente, dos quais também contém comentários ofensivos, que infelizmente não são fáceis de serem identificados, uma vez que os autores tendem a disfarçar as palavras originais inserindo asteriscos, espaços ou substituir por caracteres que tenham sons semelhantes. Assim, apenas verificar a existência de um termo que está em uma lista pré-computada de ofensas, perderia muitos desses comentários (PELLE; MOREIRA, 2017). Diante do exposto, este estudo trata dessa problemática com técnicas de mineração de texto e aprendizagem de máquina supervisionado, a fim de identificar um modelo capaz a identificar automaticamente comentários ofensivos nos textos do Twitter.

1.3 Justificativa

Classificar discursos de ódio em comentários do Twitter, que é uma das principais redes sociais utilizadas no Brasil, não é uma tarefa simples pois o grande número de dados gerados constantemente tornam a análise manual inviável, custosa e difícil de ser visualizada com recursos computacionais limitados, além de que, usuários que publicam tais sentenças tendem a disfarçar palavras ofensivas inserindo espaços, asteriscos, caracteres especiais ou substituindo certas letras por outras com sons equivalentes.

Com isso, de acordo com diversos autores como Pelle e Moreira (2017), Almeida et al. (2017) e Mondal, Silva e Benevenuto (2017), uma das melhores soluções é utilizar mineração de texto, que é a descoberta automática de padrões em dados textuais.

Diante da problemática, para o desenvolvimento da aplicação que lida com textos de cunho pejorativo, foi utilizada a linguagem de programação Python, versão 3.5.2, que segundo Souza et al. (2016), é uma das linguagens mais utilizadas no processamento de texto em português, juntamente com a biblioteca NLTK que fornece uma série de recursos para o processamento de texto na língua portuguesa, tais como: classes básicas para representar dados importantes ao processamento de linguagem natural; interface padrão para executar tarefas como o *part-of-speech tagging* (classificação gramatical), análise sintática e classificação textual; bem como implementações definidas para cada tarefa, que podem ser combinadas para resolução de problemas complexos (STEVEN; KLEIN; LOPER, 2009).

Ainda de acordo com Souza et al. (2016), os classificadores Bayesianos e *Support Vector Machines* (SVM) são os mais utilizados para mineração de opinião com a língua portuguesa, e se mostraram bastante eficientes em estudos realizados por Pelle e Moreira (2017), os quais analisaram comentários ofensivos na *Web* brasileira. Em outros estudos como por exemplo o de Davidson et al. (2017) também pode-se observar que a Regressão Logística obteve bons resultados, assim sendo, o presente trabalho realiza análises e execuções dos algoritmos citados, com intuito de avaliar dentre eles qual obtém o melhor desempenho no domínio proposto.

1.4 Objetivos

1.4.1 Objetivo Geral

Este trabalho tem por objetivo geral comparar técnicas supervisionadas de aprendizagem de máquina que visam classificar *tweets* que contenham discursos de ódio.

1.4.2 Objetivos Específicos

- Construir um corpus para análise de discurso de ódio para o Português brasileiro.
- Realizar análise comparativa entre técnicas de pré-processamento e processamento de texto.
- Construir e validar uma aplicação de mineração de texto utilizando um corpora das línguas inglesa e portuguesa.

1.5 Organização do Trabalho

O restante do trabalho está organizado em:

Capítulo 2 que apresenta o referencial teórico desse estudo, fornecendo uma breve explanação sobre discurso de ódio e mineração de texto. Em seguida, são apresentados os tipos de aprendizado de máquina, as técnicas de aprendizado supervisionado que são utilizadas neste trabalho e as formas de validação. E por fim, são apresentados os trabalhos relacionados

No capítulo 3, são apresentados o método e as tecnologias utilizadas para o desenvolvimento deste projeto.

No capítulo 4, descreve-se acerca do processo de experimentação, detalhando as configurações executadas e discute sobre os resultados alcançados.

No capítulo 5, é apresentado as conclusões do trabalho, e destacado as limitações e ameaças, as contribuições deste trabalho, as considerações finais e as propostas de trabalhos futuros.

2 Referencial Teórico

Neste Capítulo é apresentada uma explanação sobre conteúdos utilizados neste trabalho. Na Seção 2.1, é apresentado a definição sobre Discurso de Ódio. Na Seção 2.2, é apresentado conceito de mineração de texto. A Seção 2.3 explana sobre aprendizado de máquina. Na Seção 2.4, são apresentados os trabalhos relacionados.

2.1 Discurso de Ódio

O discurso de ódio, ou no inglês *Hate Speech*, pode ser definido como qualquer ato de comunicação que inferiorize uma pessoa por sua etnia, raça, religião, orientação sexual, nacionalidade entre outras características (NOCKLEBY, 2000). Tais discursos têm sido utilizados para insultar, perseguir e justificar a privação dos direitos humanos, e, em casos mais extremos, para dar razão a homicídios (KAYE, 2015).

Segundo Silva et al. (2011), o "discurso de ódio" caracteriza-se pelo conteúdo segregacionista, fundado na dicotomia da superioridade do emissor e na inferioridade do atingido (a discriminação), e pela externalidade, ou seja, existirá apenas quando for dado a conhecer a outrem que não o próprio emissor. Por outro lado, enfatizando a discriminação, Brugger (2007) fala que discurso do ódio refere-se à palavras que tendem a insultar, intimidar ou assediar pessoas em virtude de sua raça, cor, etnicidade, nacionalidade, sexo ou religião, ou que tem a capacidade de instigar violência, ódio ou discriminação contra tais pessoas.

Dessa forma, o discurso de ódio deve ser aquele que se enquadre dentro dos padrões definidos pelos tratados internacionais e deve seguir os parâmetros da jurisprudência das cortes internacionais, como exemplo pode-se citar o Artigo 20 do Pacto Internacional sobre Direitos Civis e Políticos, o qual determina que:

1. Será proibido por lei qualquer propaganda em favor de guerra.

2. Será proibida por lei qualquer apologia do ódio nacional, radical, racial ou religioso que constitua incitamento à discriminação, à hostilidade ou à violência.

Os ataques odioso acontecem contra os mais diversos tipos de pessoas, como exemplo

o meio-campista da Seleção Brasileira que sofreu diversas ofensas no *Twitter*, onde usuários postaram comentários como: “Fernandinho macaco preto”, dizia uma das mensagens. “Vai se f*, negro macaco filho da p*. E se não gostou da publicação, vem e dá em mim”, disparou outro internauta. “Isso é culpa do macaco do Fernandinho”, completou um terceiro seguidor. Claramente tais usuários expressão o ódio racial de forma agressiva contra o jogador. Tal exemplo demonstra que os *haters* não se importam com quem são suas vitimas, eles simplesmente deliberam o ódio contra qualquer pessoa.

2.1.1 Combate ao Discurso de Ódio

A forma mais comum de detecção do discurso de ódio utilizada em diversas redes sociais, como o *Twitter*, é o sistema de denúncia, onde os usuários avaliam o conteúdo gerado por outros usuários, verificando se esse conteúdo fere os termos de uso da rede ou de alguma legislação. Além desse método, os usuários podem usar campanhas para expor a um número grande de pessoas o conteúdo contendo discurso de ódio, com objetivo de mostrar a atitude negativa e conscientizar outras pessoas de que aquilo é errado.

Outra iniciativa que busca manter a harmonia nas redes, é a organização não governamental Safernet[1], que recebe denúncias de diversos crimes cibernéticos em seu *website* e ainda oferecem ajuda psicológica para as vítimas. Por exemplo, o emissor do discurso de ódio utiliza diversos artifício para validar suas afirmações e influenciar novas pessoas a compactuarem de seus ideias. Um desses artifícios é a criação de conteúdo falso, geralmente composto por sites com noticias falsas que incentivam o público a odiar e insultar determinados grupos. Porém a Safernet ajuda a remover esse tipo de conteúdo da internet, tentando manter a neutralidade da rede.

2.2 Mineração de Texto

Segundo Feldman (2013), a mineração de textos pode ser definida como um método de extração de informações relevantes em bases de dados não estruturadas, ou semi-estruturadas. É uma área ampla que abrangem conhecimentos de informática, estatística, linguística e ciência

[1] Safernet <http://new.safernet.org.br/>

cognitiva. Ela tem se mostrado interessante pelo fato de conseguir tratar elevado volumes de dados, e com o passar do tempo o grande aumento das mídias sociais, vem disponibilizando a indivíduos e organizações conteúdo de opinião diversificado e em grandes quantidades. Usuários da *Web* têm a oportunidade de registrar e divulgar suas ideias e opiniões através de comentários em fóruns de discussão, blogs, redes sociais, entre outros. Este comportamento online representa novas e mensuráveis fontes de informações com muitas aplicações práticas (BECKER; TUMITAN, 2013).

A análise dos textos podem ocorrer em diferentes granularidades, porém, costuma-se realizar em três diferentes níveis (FELDMAN, 2013): a nível de documento que consiste em classificar se o documento, como um todo, expressa um sentimento negativo, positivo ou neutro, por exemplo. A nível de sentença faz a mesma coisa porém determina o sentimento de uma sentença específica de um documento. Já o nível de entidade ou aspecto, visa todas as expressões presentes em um determinado documento, bem como os aspectos a que ela se refere.

As abordagens para classificação de texto frequentemente utilizadas são as de aprendizado de máquina supervisionado e as baseadas em léxicos, mas também há abordagens híbridas, que usam tanto aprendizado de máquina quanto léxicos (PANG B.; LEE, 2008; MEDHAT; HASSAN; KORASHY, 2014; RAVI; RAVI, 2015). Os métodos de aprendizado de máquina supervisionado aplicam algoritmos de classificação para aprender padrões subjacentes a partir de dados de exemplo com o objetivo de classificar novos dados de entrada não rotulados (BALAZS; VELÁSQUEZ, 2016). Para esses métodos, são necessários dois conjuntos de dados anotados: um para treinamento e outro para teste. Os algoritmos de aprendizado de máquina supervisionado têm sido utilizado frequentemente para classificar textos. dos quais se destacam o Naive-Bayes e *Support Vector Machines* (SVM), que têm alcançado grande sucesso nesta área (RAVI; RAVI, 2015; SOUZA et al., 2016).

Para o presente trabalho, será levada em consideração a análise a nível de documento pois esta granularidade é mais adequada quando o documento trata de uma única entidade.

2.3 Aprendizado de Máquina

O aprendizado de máquina pode ser considerado como qualquer método que absorva informação a partir de dados previamente treinados de um classificador, emprega aprendizagem (TELES, 2016). Assim, as técnicas de aprendizado tem como objetivo principal encontrar de

forma automática regras em grandes volumes de dados, que permitam extrair informações implicitamente representadas (BECKER; TUMITAN, 2013).

2.3.1 Tipos de Aprendizado de Máquina

De acordo com Duda, Hart e Stork (2012), os tipos de aprendizado de máquina podem ser divididos em três: aprendizado supervisionado, não-supervisionado e aprendizado por reforço.

O aprendizado supervisionado, é assim chamado por necessitar de um guia, deve ensinar o algoritmo sobre quais conclusões serão tomadas. Ele requer que as possíveis saídas já sejam conhecidas e que os dados usados para treiná-lo já estejam rotulados (CASTLE, 2017). Desse modo, o problema de classificação segue basicamente dois passos: a) treinar um modelo de classificação fornecendo a ele um corpus de treino previamente rotulado com as classes específicas (e.g. positivo, negativo, neutro); e b) com base no modelo resultante, prever a polaridade de novas entradas (BECKER; TUMITAN, 2013). Dentre os diversos algorítimos existentes para a execução dessa atividade, pode-se destacar o *Support Vector Machine* (SVM), o *Naive Bayes*, e o Regressão Logística.

Já nos métodos não-supervisionados, o algoritmo não recebe dados rotulados na etapa de treinamento, fazendo com que a máquina os aborde de forma diferente, tentando criar grupos que são de alguma forma semelhantes ou relacionadas por diferentes variáveis, pois as classes também não são previamente fornecidas ao algoritmo (HAN; PEI; KAMBER, 2011).

No caso de aprendizado por reforço, o classificador recebe apenas um *feedback*, geralmente binário, informando se o resultado está certo ou errado, não sendo previamente fornecido nenhum sinal de categoria, ou classe, correta para ele (HAN; PEI; KAMBER, 2011).

2.3.2 Técnicas de aprendizado supervisionado

2.3.2.1 *Multinomial Naive Bayes* (MNB)

O *Naive Bayes* é um classificador probabilístico. Classificação Bayesiana baseia-se Teorema de Bayes, que utiliza a probabilidade condicional de classificar os dados em classes pré-determinadas. A abordagem é chamada de “ingênua” (naïve) porque ela assume a independência

entre os diversos valores de atributos (KUMARI, 2014).

Essa “ingenuidade” permite que o algoritmo construa facilmente classificações de grandes conjuntos de dados sem recorrer a esquemas complicados de estimativa de parâmetros iterativos. A probabilidade de uma característica específica nos dados é dada como parte do conjunto de probabilidades, e é derivada do calculo da frequência de cada valor da característica dentro de um conjunto de classes dos dados de treinamento. O conjunto de dados de treinamento é um subconjunto usado para treinar um algoritmo classificador usando valores que lhes são fornecidos para prever valores futuros e desconhecidos. O algoritmo usa o teorema de Bayes e assume todos atributos para ser independente, dado o valor da classe variável (DIMITOGLOU; ADAMS; JIM, 2012).

A probabilidade de um documento *d* estar na classe *c* é calculado de acordo com a equação:

$$P(c \vee d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k \vee c)$$

onde $P(t_k \vee c)$ é a probabilidade condicional de t_k termos ocorrerem em um documento da classe c. Interpretamos $P(t_k \vee c)$ como uma medida da quantidade de provas que t_k contribui para que c seja a classe correta. $P(c)$ é a probabilidade a priori de um documento que ocorre na classe *c*. Se os termos de um documento não fornecem evidência clara de uma classe contra outra, escolhemos aquele que tem uma probabilidade anterior superior. $< t_1, t_2, ..., t_{nd} >$ são os *tokens* no d que fazem parte do vocabulário que usamos para a classificação e n_d é o número de tais *tokens* em c (SCHÜTZE; MANNING; RAGHAVAN, 2008).

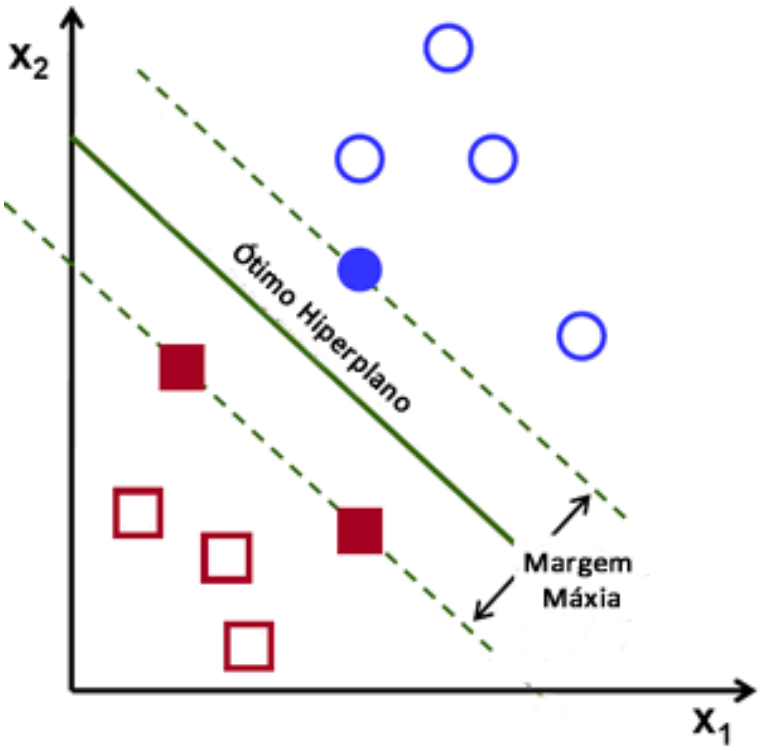
2.3.2.2 *Support Vector Machines* (SVM)

SVM é uma técnica de aprendizado de máquina supervisionado desenvolvida por Cortes e Vapnik (1995), a qual tenta efetuar classificações binárias, pegando as entradas de treinamento, mapeando-as em um espaço multidimensional e usando a regressão para encontrar um hiperplano, que é uma superfície no espaço n-dimensional, o qual é separado em dois outros espaços, visando efetuar a classificação das duas classes de entradas. Uma vez que o SVM tenha sido treinado, ele é capaz de avaliar novas entradas em relação ao hiperplano de separação e classificá-las em uma das duas categorias (READHEAD, 2012).

Ainda de acordo com READHEAD (2012) a melhor maneira de conceituar como o SVM funciona, é considerando o caso bidimensional. O qual tenha duas entradas e retorne uma

única saída, que classifique o dado como pertencente a uma das duas categorias. A Figura 2.1 mostra como é realizada a reparação do hiperplano. O seu objetivo é encontrar a margem de separação, que é o ponto máximo de separação dos pontos.

Figura 2.1 – Representação de um hiperplano



Fonte: READHEAD (2012)

Para Feldman (2013) a definição formal do hiperplano é dada por: $f(x) = \beta_0 + \beta^T x$, onde β beta é conhecido como o vetor de peso e β_0 como o viés. O hiperplano ideal pode ser representado em um número infinito de maneiras diferentes pela escala de β e β_0. Por uma questão de convenção, entre todas as possíveis representações do hiperplano, ele a define como:

$$|\beta_0 + \beta^T x| = 1$$

onde x simboliza os exemplos de treinamento mais próximos ao hiperplano. Em geral, os exemplos de treinamento mais próximos do hiperplano são chamados de vetores de suporte.

Trazendo o SVM para análise de sentimentos em texto, é necessária a conversão do texto para objetos que uma máquina possa entender. Uma das técnicas comumente usada é transformar as sentenças em vetores de frequência, em que cada número representa o número de vezes que uma determinada palavra, letra ou símbolo aparece. Dessa forma um classificador que utiliza fórmulas matemáticas pode aprender e executar classificações textuais (TELES, 2016).

2.3.2.3 Regressão Logística

A regressão logística é um classificador probabilístico linear. É parametrizado por uma matriz de peso *W* e um vetor de polarização *b*. A classificação é feita projetando um vetor de entrada em um conjunto de hiperplanos, cada um dos quais corresponde a uma classe. A distância da entrada a um hiperplano reflete a probabilidade de que a entrada seja um membro da classe correspondente (DEEPLEARNING.NET, 2017).

A probabilidade de um vetor de entrada x ser um membro da classe i, que, por sua vez, é um valor da variável estocástica Y, pode ser escrito como:

$$P(Y = i|x, W, b) = softmax_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

deste modo, a predição y_{pred} será a classe que obtiver a maior probabilidade, como descrito por DEEPLEARNING.NET (2017), na equação:

$$y_{pred} = argmax_i P(Y = i|x, W, b)$$

2.4 Trabalhos Relacionados

Com objetivo de detectar ódio, ofensas, ou citações regulares em comentários curtos, Almeida et al. (2017) propõem uma abordagem para representação de dados baseada em quantificadores de Teoria da Informação (entropia e divergência). Para verificar a validação dos resultados gerados por meio da classificação, utilizou-se a técnica de validação cruzada *10-fold*. Já para classificação dos textos foram utilizados métodos de aprendizagem de máquina supervisionado, com os seguintes classificadores: *Multinomial Naive Bayes*, *K-Nearest Neighbor (K=5)* e *Maximum Entropy*, para o classificador *K-Nearest Neighbor*, utilizou-se empiricamente o melhor número de vizinhos considerados para a inferência. A base de dados utilizada foi a rotulada e proposta por Davidson et al. (2017). Ela é composta por 14.442 *tweets* escritos em inglês provenientes da rede social Twitter. Cada texto foi rotulado de acordo com uma das seguintes classes: “discursos de ódio”, “ofensivo” e “regular”. Para verificar a eficácia dos classificadores utilizou-se das meretricias: precisão, revocação e F1. Dentre diversas configurações utilizadas, o algorítimo que teve melhor eficácia foi o KNN.

O estudo de Pelle e Moreira (2017) investiga discursos de ódio em língua portuguesa em comentários provenientes do site de notícias Globo.com. Os autores montaram uma base de dados com 10.336 comentários, dos quais 1.250 foram selecionados de forma aleatória para serem anotados manualmente, seguindo o processo padrão. Assim, três anotadores julgaram os comentários em ofensivos e não-ofensivos, ao final, dois conjuntos de dados foram gerados a partir das anotações. O primeiro chamado de OffComBr-2 contendo todos os 1.250 comentários e segundo OffComBr-3, mais restrito, contendo apenas os comentários que foram julgados de forma igual pelos três anotadores. No pré-processamento, utilizou-se a tokenização com agrupamento de unigramas, bigramas e trigramas. Para a classificação, foi utilizada a ferramenta Weka, com os algorítimos *Naive Bayes* e *Support Vector Machine* (SVM), juntamente com uma abordagem de validação cruzada *10-fold*.

Davidson et al. (2017) elaborou um estudo para detecção automática de discurso de ódio no Twitter, um corpus com 25.000 *tweets* foi montado a partir da extração de 85,4 milhões de cometários coletados de 33.458 usuário, por meio da API do Twitter. A base foi montada de acordo com termos contidos em léxico disponibilizado pela *Hatebase.org*, com isso foi solicitado para um grupo de colaboradores que rotulassem cada *tweet* como uma das três categorias: “discurso de ódio”, “ofensivo ” ou “nenhuma delas”. Alguns *tweets* não foram rotulados, pois não havia classe majoritária. Isso resultou em uma amostra de 24.802 *tweets* anotados manualmente. Para o tratamento das características utilizou-se TF-IDF, *stemmer* e etiquetas unigram, bigram e trigram para *Part-of-Speech*. O algorítimo utilizado para classificação foi a Regressão Logística com L2, que obteve os seguintes resultados: precisão 0,91, *recall* de 0,90 e *F1 score* de 0,90.

2.4.1 Visão Geral dos Trabalhos Relacionados

Pelo bom desempenho apresentado pelos trabalhos relacionados, este estudo tomou como base seus algorítimos e técnicas de pré-processamento de texto, O Quadro 2.1 apresenta as principais características identificadas em cada um dos trabalhos relacionados, dando uma visão geral de todos e facilitando a comparação entre eles.

Quadro 2.1 – Visão geral dos trabalhos relacionados

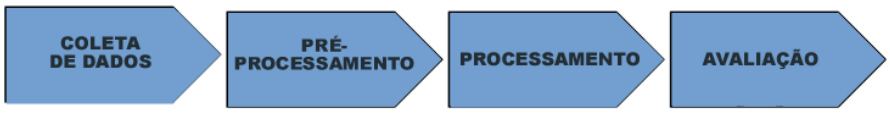
Trabalho	**Mídia**	**Idioma**	**Pré-Processamento**	**Algoritmo**
Almeida et al. (2017)	Twitter	Inglês	Tokenização, filtragem e remoção de *stopwords*.	*K-Nearest Neighbor*, *Naive-Bayes* e *Maximum Entropy*.
Pelle e Moreira (2017)	Conteúdo *Web*	Português	Tokenização, *Case folding* e *Feature selection*.	*Naive Bayes* e SVM.
Davidson et al. (2017)	Twitter	Inglês	Stemming, Part-of-Speech (POS) e padronização dos termos.	Regressão Logística.
Este Trabalho	Twitter	Português e Inglês	Tokenização, filtragem, remoção de *stopwords*, *n-gram*.	SVM, *Naive-Bayes* e Regressão Logística.

Fonte: Elaborado pelo autor

3 Método

Este trabalho utilizou dois corpus, o primeiro é um *dataset* de referência contendo *tweets* apenas em Inglês, disponibilizado por Davidson et al. (2017) e o segundo foi construído com *tweets* na língua portuguesa do Brasil e passou por um processo de quatro etapas conforme apresentado na Figura3.1.

Figura 3.1 – Etapas do método

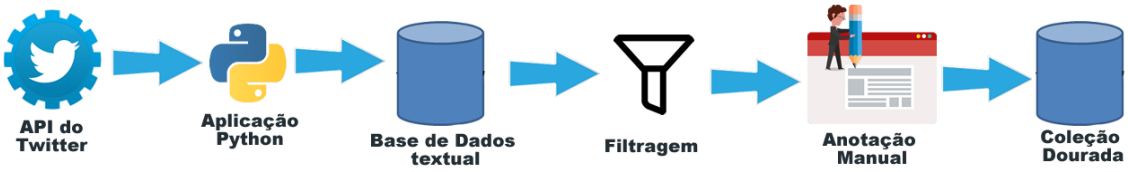


Fonte: Elaborada pelo autor

3.1 Coleta de Dados

A coleta para a língua portuguesa foi a primeira atividade a ser executada, dessa atividade resultou uma base de dados textual, o Corpus. Para isso, foram realizadas três atividades: coleta de *tweets*, filtragem e classificação manual, como ilustrado na Figura 3.2

Figura 3.2 – Processo de coleta de dados



Fonte: Elaborada pelo autor

3.1.1 Extração dos *Tweets*

Para a coleta de *tweets* foi desenvolvida uma aplicação escrita na linguagem de programação Python, versão 3.5.2, utilizando a biblioteca Tweepy, versão 3.7.0, para acessar a API do Twitter, versão 4.0. Essa API disponibiliza os *tweets* postados por usuários dos últimos 7

dias. Foram coletados o total de 2.000 *tweets* entre os meses de Setembro/2018 e Outubro/2018, o critério utilizado para a coleta de um *tweet* foi que esse utilizasse ao menos um dos termos descritos no Quadro 3.1 e que fosse escrito em Português brasileiro.

Quadro 3.1 – Termos pesquisados nos *tweets*

Lista de Termos
preto, preta, branquelo, branquela, macaco, macaca, gay, nordestina, bolsonaro, bolso, haddad, marina, ciro, facista, opressor, opressora, idiota, negro, negra, coxinha, petralha, sapatonona, feminista, puta, terror, terrorista, matar, morrer, morte, gordo, gorda,palito, esquisito, lesbica, burguês, matar, capitalista, socialista, estrangeiro, forasteiro, matuto, preconceito, racista, doente, porca, porco, puto, feio, feia, nordestino, imigrantes

Fonte: Elaborado pelo autor

Parte dos termos citados no Quadro 3.1, são disponibilizados em Hatebase (2018), que monitora termos de discurso de ódio na Internet. Os demais termos foram adicionados de forma relacional ao tema, a fim de coletar um maior número de *tweets* que possivelmente possam expressar ódio ou ofensas.

3.1.2 Filtragem dos *Tweets*

Para manter a integridade dos dados coletados foram removidos *tweets* que não apresentavam informações relevantes ao processo de mineração de texto. Dessa forma, *tweets* que não possuíam conteúdo de texto, como aqueles em que existem somente *links*, *hashtags*, menções, ou *emojis*, foram excluídos. Dessa maneira 200 *tweets* foram removidos da base, restando ao fim do processo 1.800.

3.1.3 Classificação Manual

Os *tweets* restantes, após a filtragem, foram anotados de acordo com a seguinte classificação:

- "Discurso de ódio": *tweets* que contém uma linguagem intencionada a expressar ódio direcionado para um ou mais indivíduos a fim de insultar, depreciar ou humilhar.
- "Ofensivo": discurso ofensivo pode conter palavras depreciativas em relação a uma pessoa ou a um grupo, porém utilizado de uma maneira qualitativa diferente, isto é, o contexto foi

levado em consideração para essa classe (ALMEIDA et al., 2017).

- "Regular": *tweets* que não expressam conteúdo ofensivo ou contenha discursos de ódio.

Três anotadores participaram do processo, no qual cada *tweet* foi classificado por três anotadores. A polaridade final de cada sentença foi definida de acordo com polaridade atribuída pela maioria dos anotadores. Em caso de discordância entre todos, ou seja, se cada um classificou de acordo com uma polaridade diferente da do outro, o *tweet* foi excluído, ao final do processo, o coeficiente de concordância entre os anotadores foi de 90%. O Quadro 3.2 exemplifica como se deu essa classificação.

Quadro 3.2 – Exemplo de classificação

Classe	**Sentença**
Discurso de Ódio	@Irineu é um gay safado e fascista
Ofensivo	@Irineu é muito feio
Regular	@Irineu é meu amigo

Fonte: Elaborado pelo autor

Ao fim do processo de anotação, os *tweets* ficaram divididos da seguinte forma:

- 90 *tweets* classificados como Discurso de ódio.
- 320 *tweets* classificados como Ofensivos.
- 1390 *tweets* classificados como Regular

Tabela 3.1 – Detalhes dos corpus

Corpora	**Balanceamento**	**Ódio**	**Regular**	**Ofensivo**	**Total**
Português	Balanceada	90	90	90	270
	Desbalanceada	90	1390	320	1800

Fonte: Elaborada pelo autor

3.2 Corpus em Inglês

A base de dados em inglês é proveniente de um trabalho de Davidson et al. (2017) os quais a partir de um léxico de discurso de ódio contendo palavras e frases identificadas por usuários da Internet como discurso de ódio, compiladas pelo Hatebase.org, coletaram um conjunto de 85,4 milhões de *tweets*. A partir desse corpus, escolheram uma amostra aleatória de 25k *tweets* contendo termos do léxico e disponibilizaram para o *CrowdFlower*(CF) para que

usuários convidadados pudessem classificar cada *tweet* como uma das três categorias: discurso de ódio, discurso ofensivo, mas não de ódio, nem discurso de ódio nem ofensivo. Eles foram instruídos que a presença de uma determinada palavra ofensiva, não indica necessariamente que um *tweet* é discurso de ódio. Cada *tweet* foi classificado por três ou mais pessoas, ao final alguns *tweets* não tiveram uma classificação precisa, restando 24.802. A concordancia entre os anotadores foi de 92% segundo a CF.

3.3 Pré-processamento do texto

Para que possam ser aplicados os algoritmos de aprendizagem de máquina, é necessário pré-processar a base textual montada na etapa de coleta. Dentre as diversas técnicas existentes o presente trabalho utiliza a Tokenização, *Stemming*, Filtragem e Remoção de *Stopword*.

Nessa etapa foi utilizada a biblioteca NLTK, versão 3.2.4, disponível na linguagem de programação Python. Uma vez que esta é amplamente utilizada em tarefas dessa natureza conforme Ravi e Ravi (2015).

3.3.1 Tokenização

A primeira atividade do pré-processamento foi a tokenização, que consiste em percorrer todo o texto identificando cada palavra entre as sequências de caracteres, ou seja, buscando identificar *tokens* que são as menores unidades de informação presentes no texto e que possuem significado quando analisados de forma isolada. Desse modo, um *token* pode ser uma palavra, sentenças, um número representado por um caractere numérico, um número de telefone, caracteres de pontuação, e etc (WEISS et al., 2005).

Para está etapa, foi utilizado o tokenizador TweetTokenizer, que é capaz de identificar partes específicas de um *tweet* como *emoticons*, *hashtags*, pontuações e etc. A Tabela 3.2 ilustra um exemplo de como se dá esse processo.

Tabela 3.2 – Ilustração do processo de tokenização.

Entrada	[“Não gosto de @Maria #Chata”]
Saída	[‘Não’, ‘gosto’, ‘de’, ‘@Maria’, ‘#Chata’]

Fonte: Elaborado pelo autor

3.3.2 *N-gram*

O *n-gram* é uma sequencia de caracteres de tamanho n extraído de um documento, é um método de verificação de "n" palavras contínuas de uma determinada sequência de texto. Este modelo ajuda a prever o próximo item em uma sequência, em uma análise aprofundada, o modelo *n-gram* ajuda na analisar textual. Com isso, foram utilizados os seguintes agrupamentos: *Unigram* que refere-se a *n-gram* de tamanho 1, *Bigram* que refere-se a *n-gram* de tamanho 2, *Trigram* refere-se a *n-gram* de tamanho 3 (TRIPATHY; AGRAWAL; RATH, 2016).

O método n-gram pode ser demonstrado usando o seguinte exemplo:

- *Unigram*: ('Isso'), ('é'), ('um'), ('exemplo'), ('de'), ('unigram')
- *Bigram*:('Isso', 'é'), ('é', 'um'), ('um', 'exemplo'), ('exemplo', 'de'), ('de', 'bigram')
- *Trigram*: ('Isso', 'é', 'um'), ('é', 'um', 'exemplo'), ('um', 'exemplo', 'de'), ('exemplo', 'de', 'trigram')

3.3.3 *Stemming*

Stemming é o processo de redução das palavras ao seu radical, geralmente removendo cada palavra de seus sufixos derivados e flexionais. Pesquisadores em muitas áreas de linguística computacional e recuperação de informação acham este um passo desejável (LOVINS, 1968).

A utilização de *stemming* como parte da etapa de pré-processamento dos dados, ocorre pelo fato do mesmo conseguir identificar similaridades em função da morfologia das palavras, dessa forma reduzindo o número de atributos de um texto, pois muitas vezes palavras com morfologias semelhantes representam de forma genérica o mesmo conceito. É uma técnica comum usada na pesquisa de mineração de texto, uma vez que reduz a complexidade sem qualquer perda informações para aplicações (MEYER; HORNIK; FEINERER, 2008).

3.3.3.1 Remoção de *Stopwords*

Existem diversas palavras que aparecerem frequentemente nos textos e que podem ser removidas, pelo fato delas não apresentarem nenhum valor semântico, são necessária apenas para compreensão do texto de foma geral, essas palavras são conhecidas na literatura por

Stopwords que normalmente são preposições, artigos ou pronomes. Para remoção das *Stopwords*, foi utilizada uma *stoplist*, exibida no Quadro 3.3, disponível para o português. Também foram removidos todos os URLs, pois não são relevantes para o processo de classificação.

Quadro 3.3 – Exemplo de *stoplist* com *stopwords* do português

Stopwords
a, ao, aos, aquela, aquelas, aquele, aqueles, aquilo, as, até, com, como, da, das, de, dela, delas, dele, deles, depois, do, dos, e, ela, elas, ele, eles, em, entre, era, eram, essa, essas, esse, esses, esta, estamos, estas, estava, estavam, este, esteja, estejam, nossa, nossas, terá, terão, teríamos, teu, teus, teve, tinha, tinham, tive, tivemos, tiver, tivera, tiveram, tiverem, tivermos, tivesse, tivessem, tivéramos, tivéssemos, tu, tua, tuas, tém, tínhamos, um, uma, você, vocês, vos, à, às, éramos, ...

Fonte: Elaborado pelo autor

3.3.4 Modelo VSM

O *Vector Space Model* (VSM) é um modelo algébrico para representar documentos de texto como vetores. Neste modelo, o texto (uma frase ou documento) é representado como um multiconjunto de suas palavras, desconsiderando a estrutura gramatical e até mesmo a ordenação delas, mas mantendo sua multiplicidade. Este modelo é frequentemente utilizado em métodos de classificação de texto para filtragem de informação , recuperação de informação, indexação e rankings de relevância, na qual a frequência de ocorrência de cada palavra é vista como uma característica utilizada para treinar o classificador (SIVIC; ZISSERMAN, 2009).

Para o cálculo da frequência, utilizou-se duas técnicas distintas: a primeira foi a verificação da frequência do termo que indica quantas vezes ele aparece no documento, e a segunda é o *Term Frequency–Inverse Document Frequency* (TF-IDF), que é uma estatística numérica que a qual mensura a importância de uma palavra para um documento em um determinado corpus (ULLMAN, 2011). A equação abaixo demonstra como essa relevância é calculada:

$$IDF(t) = \log(\frac{|D|}{DF(t)})$$

$$W(t) = TF(d,t) * IDF(t)$$

- *Frequência do termo*: resume a frequência com que uma determinada palavra aparece em um documento.
- *Frequência inversa do documento*: reduz as palavras que aparecem em vários documentos.

3.4 Processamento

Após uma revisão sistemática realizada por Brito (2017), constatou-se que em mais de 82% dos estudos (23/28) utilizaram algoritmos de aprendizagem de máquina supervisionado, sendo eles o *Suport Vector Machine* (SVM) e *Naive Bayes*. Assim, para a etapa de processamento, que é a efetiva classificação da polaridade dos *tweets*, foram utilizados três algoritmos de aprendizado de máquina supervisionado: o Linear SVC do *Suport Vector Machine*, *Multinomial Naive Bayes* e Regressão Logística. Esse último foi utilizado pelo fato de ter apresentado bons resultados em um trabalho realizado por Davidson et al. (2017). Todos esses recursos estão disponíveis na biblioteca Scikit-Learn que é uma biblioteca de código aberto de aprendizado de máquina feita na linguagem de programação Python.

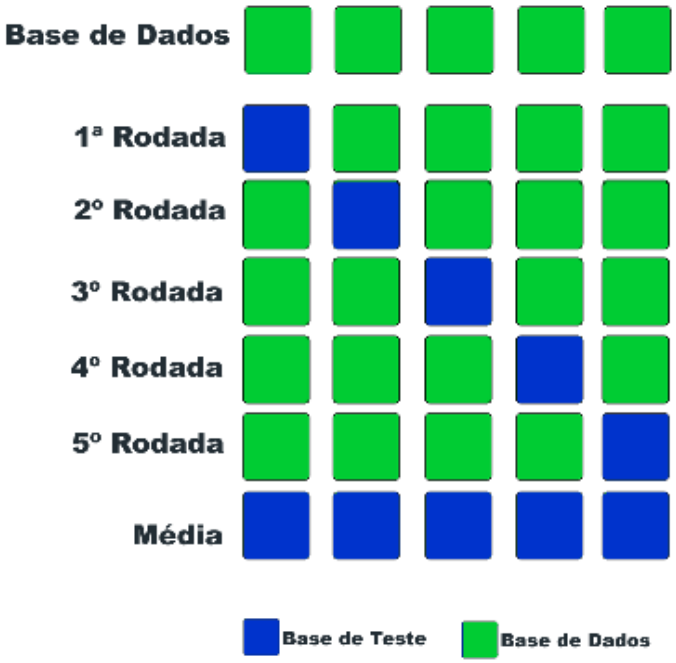
3.5 Avaliação

Para a etapa de avalização, foi escolhido o método de validação cruzada *5-fold*, enquanto que os resultados foram obtidos utilizada a medida de acurácia, precisão e revocação.

3.5.1 Validação Cruzada e *k-fold*

Após estudo dos trabalhamos relacionados, constatou-se uma predominância na utilização do método de validação cruzada *k-fold* (ALMEIDA et al., 2017; PELLE; MOREIRA, 2017; DAVIDSON et al., 2017). E isso levou a escolha dessa técnica para o presente trabalho. Ela consiste em dividir a base de dados em *k* blocos de tamanhos iguais, onde em cada rodada de execução, um deles será utilizado para teste e os restantes serão utilizados para o treinamento do classificador, do qual será medida a acurácia, revocação e precisão, que serviram para avaliar o desempenho dos classificadores junto de suas configurações. A Figura **??** exemplifica utilização do *k-fold* considerando $k = 5$.

Figura 3.3 – Exemplo de um *k-fold* considerando $k = 5$.



Fonte: Elaborada pelo autor

3.5.2 Acurácia

Acurácia é definida como sendo a fração de classificações corretas feitas pelo classificador. Essa é uma medida precisa e efetiva usada frequentemente para avaliar aprendizado de máquina em problemas de classificação (SCHÜTZE; MANNING; RAGHAVAN, 2008).

Neste trabalho, ela foi medida pela razão entre o número de *tweets* classificados corretamente e o número total de *tweets* classificados, como expresso em Tripathy, Agrawal e Rath (2016), dada pela equação:

$$Acuracia = \frac{CC}{CC + CI}$$

onde CC representa o número de *tweets* classificados corretamente e CI representa o número de *tweets* classificados incorretamente.

De acordo com Weiss et al. (2010) um classificador pode conseguir uma acurácia muito alta, por exemplo, acertando todos os dados que são negativos. É, portanto, útil para medir o desempenho de classificação, ignorando os dados negativos que previu corretamente e, em seguida, analisar os tipos de erros cometidos pelo classificador. Outros Dois índices que também foram medidos, alcançaram destaque especial: precisão e revocação. Suas definições são dadas,

respectivamente, nas Equações abaixo:

$$precisão = \frac{número\ de\ predições\ positivas\ corretas}{número\ de\ predições\ positivas}$$

$$revocação = \frac{número\ de\ predições\ positivas\ corretas}{número\ de\ documentos\ de\ classe\ positiva}$$

Ao final de cada ciclo, para cada configuração, foi medida a acurácia do classificador juntamente com sua precisão e revocação. Quando todos os ciclos foram concluídos, essas medidas foram computadas e consideradas como o resultado final da configuração.

4 Resultados e Discussão

Neste capítulo são apresentados os resultados alcançados por cada configuração proposta. Também são levantadas, na Seção 4.1, as combinações de configurações utilizadas. E a Seção 4.3 discute as análises feitas a partir dos resultados obtidos.

4.1 Configurações do Experimento

O objetivo do experimento propostos nesse trabalho é realizar uma análise comparativa de técnicas de pré-processamento e processamento de texto.

As configurações alternam tanto na etapa de pré-processamento quanto no processamento. Nelas, as técnicas de filtragem, limpeza, *n-gram* e tokenização permanecem constantes em todos os experimentos. A única diferença de alternância que houve entre os corpus, foi a utilização das *stopwords* e *stemming*, pois utilizou-se o específico para cada linguagem. A Tabela 4.1 mostra as características de cada corpus.

Tabela 4.1 – Detalhes dos corpus

Corpora	Balanceamento	Ódio	Regular	Ofensivo	Total
Inglês	Balanceada	1430	1430	1430	4290
	Desbalanceada	1430	4153	19133	24717
Português	Balanceada	90	90	90	270
	Desbalanceada	90	1390	320	1800

Fonte: Elaborada pelo autor

Todos as configurações foram executadas com as bases balanceadas e desbalanceadas, portanto cada configuração foi executada com quatro bases distintas. E ainda todas as configurações foram executadas com os três classificadores que estão sendo analisados nesse trabalho: SVM, *Naive-Bayes* e Regressão Logística. A Tabela 4.2 apresenta a configuração dos experimentos que foram testados.

Tabela 4.2 – Combinações de configurações.

Configuração	Classificador	*N-gram*	Vetorização	*Stemming*
Configuração 1	SVM	Uni, Bi, e Trigrama	TF-IDF	Sim
Configuração 2	*Naive Bayes*	Uni, Bi, e Trigrama	TF-IDF	Sim
Configuração 3	Regressão Logística	Uni, Bi, e Trigrama	TF-IDF	Sim
Configuração 4	SVM	Uni, Bi, e Trigrama	TF-IDF	Não
Configuração 5	*Naive Bayes*	Uni, Bi, e Trigrama	TF-IDF	Não
Configuração 6	Regressão Logística	Uni, Bi, e Trigrama	TF-IDF	Não
Configuração 7	SVM	Uni, Bi, e Trigrama	CountVectorize	Sim
Configuração 8	*Naive Bayes*	Uni, Bi, e Trigrama	CountVectorize	Sim
Configuração 9	Regressão Logística	Uni, Bi, e Trigrama	CountVectorize	Sim
Configuração 10	SVM	Uni, Bi, e Trigrama	CountVectorize	Não
Configuração 11	*Naive Bayes*	Uni, Bi, e Trigrama	CountVectorize	Não
Configuração 12	Regressão Logística	Uni, Bi, e Trigrama	CountVectorize	Não

Fonte: Elaborada pelo autor

4.2 Resultados

As Tabelas 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 e 4.10, apresentam os resultados obtidos para cada uma das configurações, nelas são mostradas as acurácias (Acc), Revocação (Rev) e a Precisão (Prec) alcançadas pelos três classificadores: *Suport Vector Machine* (SVM), *Multinomial Naive-Bayes* (MNB), e Regressão Logística (LR).

Tabela 4.3 – Resultados do corpus inglês - desbalanceado com *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**89,1**	67,1	87,8	88,0	68,6	87,3
	Bigrama	88,1	64,8	86,7	85,6	64,4	85,3
	Trigrama	87,4	62,7	85,8	84,2	61,5	83,9
MNB	Unigrama	79,8	38,2	78,3	86,7	55,3	84,8
	Bigrama	79,8	38,6	79,0	87,4	62,7	85,9
	Trigrama	79,5	37,8	78,1	86,0	62,1	84,9
LR	Unigrama	**89,0**	67,4	87,7	88,3	68,6	87,4
	Bigrama	88,1	64,4	86,6	85,8	64,8	85,5
	Trigrama	87,0	62,3	85,4	83,9	60,0	83,5

Fonte: Elaborada pelo autor

Tabela 4.4 – Resultados do corpus inglês - desbalanceado sem *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**88,9**	67,0	87,6	88,2	69,3	87,6
	Bigrama	88,0	64,1	86,4	85,4	64,3	85,1
	Trigrama	87,2	62,3	85,6	84,1	61,9	83,9
MNB	Unigrama	79,7	37,8	77,3	86,3	54,1	84,2
	Bigrama	79,6	37,8	77,1	87,0	61,1	85,3
	Trigrama	79,1	37,4	77,5	85,9	61,8	84,8
LR	Unigrama	**88,8**	66,7	87,5	88,1	68,6	87,4
	Bigrama	88,0	64,3	86,5	85,4	64,5	85,2
	Trigrama	87,1	62,3	85,6	83,9	61,8	83,7

Fonte: Elaborada pelo autor

Tabela 4.5 – Resultados do corpus inglês - balanceado com *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**79,4**	79,4	79,2	78,2	78,2	78,1
	Bigrama	76,5	76,5	76,3	73,2	73,2	73,4
	Trigrama	73,6	73,6	73,4	69,5	69,6	69,5
MNB	Unigrama	72,8	73,0	75,0	74,4	74,4	75,9
	Bigrama	71,0	71,1	72,6	71,9	72,0	73,2
	Trigrama	68,0	68,1	70,0	69,6	69,6	70,0
LR	Unigrama	**78,3**	78,2	78,1	78,3	78,3	78,2
	Bigrama	76,2	76,3	76,1	72,9	72,9	73,0
	Trigrama	74,8	74,9	74,5	70,1	70,1	70,1

Fonte: Elaborada pelo autor

Tabela 4.6 – Resultados do corpus inglês - balanceado sem *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**78,7**	78,7	78,6	77,7	77,8	77,6
	Bigrama	76,0	75,9	75,9	73,1	73,0	73,33
	Trigrama	73,1	73,1	72,9	69,5	69,5	69,4
MNB	Unigrama	73,0	73,1	74,8	73,5	73,6	75,1
	Bigrama	70,3	70,3	72,7	70,6	70,7	72,5
	Trigrama	67,8	67,9	70,6	67,6	67,6	69,9
LR	Unigrama	**78,6**	78,7	78,5	77,6	77,5	77,4
	Bigrama	75,8	75,9	75,7	72,6	72,6	72,8
	Trigrama	73,8	73,7	73,6	70,0	70,1	70,2

Fonte: Elaborada pelo autor

Tabela 4.7 – Resultados do corpus português - desbalanceado com *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**75,2**	36,1	67,38	70,0	37,5	66,8
	Bigrama	74,0	35,2	65,4	66,7	38,0	66,5
	Trigrama	74,6	35,7	66,0	66,9	37,6	65,2
MNB	Unigrama	**77,7**	33,3	60,5	76,9	34,4	68,5
	Bigrama	77,2	33,3	59,6	73,2	36,6	66,1
	Trigrama	77,2	33,3	59,6	70,6	36,4	66,6
LR	Unigrama	**74,8**	35,6	66,2	70,2	36,7	66,9
	Bigrama	74,5	35,4	66,1	68,0	37,7	66,6
	Trigrama	74,8	35,3	66,6	66,2	37,3	66,2

Fonte: Elaborada pelo autor

Tabela 4.8 – Resultados do corpus português - desbalanceado sem *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**74,8**	34,7	65,8	70,8	36,0	65,9
	Bigrama	74,5	34,8	66,1	68,2	36,1	66,0
	Trigrama	**74,8**	35,3	66,0	66,2	35,8	64,8
MNB	Unigrama	**77,9**	33,3	60,7	76,3	34,0	64,9
	Bigrama	**77,8**	33,3	60,6	73,0	34,6	64,7
	Trigrama	**77,1**	33,3	59,5	70,7	35,4	65,7
LR	Unigrama	**74,8**	34,9	65,6	71,0	36,3	65,5
	Bigrama	74,2	34,7	65,0	67,7	35,7	65,8
	Trigrama	74,6	34,7	65,2	66,7	35,9	65,3

Fonte: Elaborada pelo autor

Tabela 4.9 – Resultados do corpus português - balanceado com *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**45,1**	45,8	46,2	44,5	44,6	45,7
	Bigrama	41,1	42,0	42,3	41,4	42,4	43,9
	Trigrama	40,6	41,3	42,4	40,4	40,6	40,5
MNB	Unigrama	43,3	44,1	45,0	42,8	43,4	42,5
	Bigrama	39,3	40,4	43,0	38,5	38,7	38,7
	Trigrama	38,0	40,0	42,1	38,0	38,4	38,9
LR	Unigrama	**44,3**	44,7	45,4	42,9	43,2	43,7
	Bigrama	42,5	43,0	44,3	40,4	40,9	40,0
	Trigrama	42,0	42,5	43,3	39,3	39,8	40,1

Fonte: Elaborada pelo autor

Tabela 4.10 – Resultados do corpus português - balanceado sem *stemming*.

		TF-IDF			Frequência		
Classificador	***N-gram***	**Acc (%)**	**Rev (%)**	**Prec (%)**	**Acc (%)**	**Rev (%)**	**Prec (%)**
SVM	Unigrama	**45,0**	45,0	45,3	42,0	42,5	42,2
	Bigrama	41,7	42,3	42,2	39,1	39,3	40,3
	Trigrama	40,1	40,9	41,8	38,0	38,4	38,7
MNB	Unigrama	41,2	43,8	46,8	42,3	42,5	43,2
	Bigrama	39,0	39,4	40,8	38,6	38,7	38,8
	Trigrama	38,8	40,1	41,4	38,1	38,3	38,6
LR	Unigrama	**44,3**	44,5	45,5	42,5	42,9	43,2
	Bigrama	42,3	42,6	44,0	39,3	39,4	39,5
	Trigrama	41,2	42,0	42,6	39,0	39,2	39,8

Fonte: Elaborada pelo autor

4.3 Discussão

A presente pesquisa apresenta a discussão dos resultados obtidos no experimento desenvolvido por meio da comparação dos dados apresentados nas tabelas deste estudo. Tal explanação auxilia na visualização dos méritos e limitações deste trabalho ao tornar a exposição e interpretação de dados mais simples e concisa.

Os resultados obtidos com o corpus Inglês desbalanceado podem ser observados nas Tabelas 4.3 e 4.4. Tais tabelas demonstram que os três classificadores tiveram bons resultados, com destaque para o classificador SVM que obteve 89,1% de acurácia, 67,1% de revocação e 87,8% de precisão. A Regressão Logística, por sua vez, alcançou 88,8% de acurácia, 66,7% de revocação e 87,5% de precisão. Isso ocorreu por ambos utilizarem a vetorização TF-IDF e unigrama. Já o *Naive Bayes*, mesmo abaixo dos outros classificadores, obteve o seu melhor resultado com frequência de palavras e bigrama, pois, como já demonstrado na literatura de mineração de textos, o mesmo combina melhor com esse tipo de vetorização pelo fato ter mais chances de melhores probabilidades em seu julgamento. Confirmando tal premissa, o *Naive Bayes* atingiu 87,4% de acurácia, 62,1% de revocação e 85,9% de precisão nesta pesquisa.

O corpus Inglês balanceado das Tabelas 4.5 e 4.6 apresentou resultados menores do que o corpus Inglês desbalanceado, demonstrando, porém, maiores acurácia, revocação e precisão também atingidas pelo classificadores SVM com a utilização de unigramas. No experimento desenvolvido nesta pesquisa, observou-se ainda um aumento da revocação pelo fato do número de *tweets* ofensivos e regulares terem se igualado ao número de *tweets* com discurso de ódio. Os resultados também concluem que a diferença miníma quanto a utilização ou não do *stemming*, bem como quanto ao tipo de vetorização utilizada.

A tabela 4.7 exibe os resultados do corpus em Português desbalanceado, o qual apresentou os melhores resultados com o classificador *Naive Bayes* junto da vetorização TF-IDF com *stemming*, alcançando uma acurácia de 77%, seguida de 33,3% de revocação e 60,5% de precisão, com diferença miníma entre todos os modelos de *n-grans* utilizados. Novamente, a presente pesquisa corrobora premissas observadas na literatura ao explicitar um desempenho superior do *Naive Bayes* neste corpus, com tal algoritmo apresentando melhores resultados em pequenas bases.

Os resultados obtidos nos corpus Português balanceado, por sua vez, demonstrados nas Tabelas 4.9 e 4.10, não foram satisfatórios. Tais resultados ocorreram em razão do tamanho da

base, a qual totalizou apenas 270 *tweets* após o processo de balanceamento.

Os resultados demonstram que com as bases desbalanceadas, a acurácia e precisão são elevadas, mas a revocação é baixa. Isso ocorre devido aos classificadores cometerem alguns falsos negativos na classificação. Esse comportamento se repete em todas as configurações e para todos os classificadores nas bases desbalanceadas devido a pequena quantidade de *tweets* com discurso de ódio em relação a quantidade de *tweets* regulares e ofensivos nos corpus. De modo geral, a utilização ou não da vetorização TF-IDF e do processamento de linguagem natural *stemming* não afetou satisfatoriamente os resultados obtidos em todas as configurações dos três classificadores. Essas duas técnicas tem um custo computacional elevado, que de acordo com os resultados alcançados nestes experimentos, é plausível considerar a não utilização das mesmas para esse tipo de corpus.

Para efeitos de organização dos resultados desta pesquisa e da contribuição de tal estudo para a literatura da área, a Tabela 4.11 apresenta os resultados gerais dos trabalhos relacionados citados nesta discussão e a comparação com este estudo.

Tabela 4.11 – Resultados gerais dos trabalhos relacionados com corpus Inglês.

Trabalho	Fonte	Idioma	Algorítimo	Resultados
Almeida et al. (2017)	Twitter	Inglês	*K-Nearest Neighbor*, *Naive Bayes* e *Maximum Entropy*	Melhor resultado com K-Nearest Neighbor para corpus desbalanceado: Precisão 86% e revocação de 84%.
Davidson et al. (2017)	Twitter	Inglês	Regressão Logística	Para corpus desbalanceados os resultados foram: Precisão 91% e Revocação de 90%
Este Trabalho	**Twitter**	**Inglês**	**SVM, *Naive Bayes* e Regressão Logística.**	**Melhores resultados: 89,1% de acurácia, 68,6% de revocação e 87,5% de precisão com SVM e Regressão Logística, 0.87.**

Fonte: Elaborada pelo autor

Tabela 4.12 – Resultados gerais do trabalho relacionado com corpus Português.

Trabalho	Fonte	Idioma	Algorítimo	Resultados
Pelle e Moreira (2017)	Conteúdo *Web*	Português	*Naive Bayes* e SVM	Melhores resultados com SVM em corpus desbalanceado Revocação: 77% e precisão 82%
Este Trabalho	**Twitter**	**Português**	**SVM, *Naive Bayes* e Regressão Logística**	**Melhores resultados com *Naive Bayes* em corpus desbalanceado Acurácia 77,7% , Revocação 33,3% e Precisão 60,5%.**

Fonte: Elaborada pelo autor

5 Conclusão

Neste capítulo é feito o desfecho conclusivo deste trabalho. Na Seção 5.2, descrevem-se as contribuições desta monografia. As propostas para trabalhos futuros são apresentadas na Seção 5.3. Por fim, na Seção 5.4, são apresentadas limitações na proposta do trabalho e as ameaças à sua validade.

5.1 Considerações Finais

Neste estudo foi desenvolvida e avaliada uma aplicação de mineração de texto, com o objetivo de analisar três técnicas de aprendizado de máquina supervisionado, a saber: 1) Multinomial Naive-Bayes; 2) Suporte Vector Machine; e 3) Regressão Logística. Para tal, foram combinadas diversas configurações de pré-processamento de texto, como a utilização da vetorização *Term Frequency–Inverse Document Frequency* (TF-IDF), remoção de *stop words* e processamento de linguagem natural *stemming*. Utilizou-se também textos extraídos do Twitter e escritos em duas linguagens diferentes: Português e Inglês.

Após a análise dos resultados dos três classificadores utilizados, observou-se que a melhor configuração para corpus inglês balanceado ou desbalanceado e para português balanceado é obtida utilizando-se a técnica de processamento de linguagem natural *stemming* com a vetorização TF-IDF e o classificador SVM. Já para o corpus em Português desbalanceado, o *Naive-Bayes* demonstra ser uma melhor algorítmica que os demais devido ao fato da base de dados ser menor.

Assim, de acordo com os resultados dos experimentos realizados neste estudo, conclui-se que o classificador SVM é de forma geral mais eficaz na identificação de discursos de ódio do que os classificadores de Regressão Logística e o *Naive-Bayes*, este ultimo mesmo apresentando melhores resultados com a língua portuguesa, não foi satisfatório. Conclui-se ainda que a utilização das técnicas de pré-processamento de linguagem natural *stemming* e a vetorização TF-IDF, além de apresentarem alto custo de processamento, não agregam valores significativos no resultados, contribuindo apenas para um maior tempo de execução da aplicação nos corpus testados.

5.2 Contribuições deste Trabalho

Uma contribuição desse trabalho foi a análise de três classificadores de aprendizado de máquina supervisionado (Multinomial Naive-Bayes, *Suporte Vector Machine* e Regressão Logística), juntamente com uma técnica de processamento de linguagem natural *stemming* com a vetorização TF-IDF, em cenários utilizando diferentes linguagens em textos do Twitter.

Uma outra contribuição é a disponibilização do corpus com textos do Twitter na língua portuguesa, desenvolvido para análise desse trabalho. Essa base permite fazer diferentes tipos de análise com duas ou três classes e com bases balanceadas e desbalanceadas.

Outra contribuição foi a construção de uma aplicação capaz executar e avaliar todas as combinações de configurações presentes neste trabalho.

5.3 Proposta para Trabalhos Futuros

Em virtude dos resultados obtidos com o corpus em Português brasileiro, prende-se amplia-ló e testar novas técnicas de processamento de linguagem natural, como o POS-tagger, a fim de alcançar melhores resultados. Outra abordagem testar novos classificadores, bem como a utilização de redes neurais, a fim explorar melhores dos resultados. Um outro trabalho que futuro que pode ser feito, é a verificação da variação linguística entre o Português brasileiro e Português europeu.

5.4 Limitações e Ameaças

Uma das limitações encontradas neste trabalho é o não tratamento de ironia e sarcasmo, pois a identificação dos mesmos ainda apresenta um certo grau de complexidade para os classificadores de aprendizado de máquina, então esse não tratamento pode indicar uma diminuição na taxa de acerto entre os experimento. Outro ponto que contribuiu negativamente para os resultados, foi o tamanho do corpus Português brasileiro, pois o mesmo contia um baixo número de *tweets* de discurso de ódio, afetando diretamente a base de treino do classificador, isso se deu pelo fato das novas políticas do *Twitter*, que proíbem ameaças específicas de violência e desejar a

um indivíduo ou grupo de pessoas danos físicos graves, morte ou doenças, como disposto no (TWITTER, 2019)

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, T. G.; SOUZA, B. À.; NAKAMURA, F. G.; NAKAMURA, E. F. Detecting hate, offensive, and regular speech in short comments. In: ACM. *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*. [S.l.], 2017. p. 225–228.

ARTIGO-19. *Panoramasobre discurso de ódio no Brasil*. 2014. <http://docplayer.com.br/docview/33/16571832/#file=/storage/33/16571832/16571832.pdf>. [Online; Acessado em :14-02-2018].

BALAZS, J. A.; VELÁSQUEZ, J. D. Opinion mining and information fusion: a survey. *Information Fusion*, Elsevier, v. 27, p. 95–110, 2016.

BECKER, K.; TUMITAN, D. Introdução à mineração de opiniões: Conceitos, aplicações e desafios. *Simpósio brasileiro de banco de dados*, 2013.

BRITO, E. M. N. D. Mineração de textos: detecção automática de sentimentos em comentários nas mídias sociais. *Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento*, v. 6, n. 1, 2017.

BRUGGER, W. Proibição ou proteção do discurso do ódio? algumas observações sobre o direito alemão e o americano. *Direito Público*, v. 4, n. 15, 2007.

CASTLE, N. *Supervised vs. Unsupervised Machine Learning*. 2017. <https://www.datascience.com/blog/supervised-and-unsupervised-machine-learning-algorithms>. [Online; Acessado em :20-02-2018].

CHAU, M.; XU, J. Mining communities and their relationships in blogs: A study of online hate groups. *International Journal of Human-Computer Studies*, Elsevier, v. 65, n. 1, p. 57–70, 2007.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

DAVIDSON, T.; WARMSLEY, D.; MACY, M.; WEBER, I. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*, 2017.

DEEPLEARNING.NET. *Classifying MNIST digits using Logistic Regression*. 2017. <http://deeplearning.net/tutorial/logreg.html>. [Online; Acessado em :05-12-2018].

DIMITOGLOU, G.; ADAMS, J. A.; JIM, C. M. Comparison of the c4. 5 and a naïve bayes classifier for the prediction of lung cancer survivability. *arXiv preprint arXiv:1206.1121*, 2012.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012.

FELDMAN, R. Techniques and applications for sentiment analysis. *Communications of the ACM*, ACM, v. 56, n. 4, p. 82–89, 2013.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011.

HATEBASE. *Hatebase Vocabulary*. 2018. <https://hatebase.org/vocabulary_updates>. [Online; Acessado em :18-02-2018].

INTERNETLIVESTATS. *Twitter Usage Statistics*. 2018. <http://www.internetlivestats.com/twitter-statistics/>. [Online; Acessado em :18-02-2018].

KAYE, D. *Report of the Special Rapporteur on the promotion and protection of the right to freedom of opinion and expression*. [S.l.]: Office of the United Nations High Commissioner for Human Rights, 2015.

KUMARI, A. Study on naive bayesian classifier and its relation to information gain. *International Journal on Recent and Innovation Trends in Computing and Communication*, v. 2, n. 3, p. 601–603, 2014.

LOVINS, J. B. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, v. 11, n. 1-2, p. 22–31, 1968.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014.

MEYER, D.; HORNIK, K.; FEINERER, I. Text mining infrastructure in r. *Journal of statistical software*, American Statistical Association, v. 25, n. 5, p. 1–54, 2008.

MONDAL, M.; SILVA, L. A.; BENEVENUTO, F. A measurement study of hate speech in social media. In: ACM. *Proceedings of the 28th ACM Conference on Hypertext and Social Media*. [S.l.], 2017. p. 85–94.

NOCKLEBY, J. T. Hate speech. *Encyclopedia of the American constitution*, Detroit: Macmillan Reference USA, v. 3, p. 1277–79, 2000.

PANG B.; LEE, L. Opinion mining and sentiment analysis. foundations and trends in information retrieval. 2008.

PELLE, R. P. de P.; MOREIRA, V. P. M. Offensive comments in the brazilian web: a dataset and baseline results. In: *Congresso da Sociedade Brasileira de Computação-CSBC*. [S.l.: s.n.], 2017.

RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, Elsevier, v. 89, p. 14–46, 2015.

READHEAD. *MACHINE LEARNING: HOW SUPPORT VECTOR MACHINES CAN BE USED IN TRADIN*. 2012. <https://www.mql5.com/en/articles/584>. [Online; Acessado em :05-12-2018].

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. *Introduction to information retrieval*. [S.l.]: Cambridge University Press, 2008. v. 39.

SILVA, R. L. da; NICHEL, A.; BORCHARDT, C. K.; MARTINS, A. C. L. Discurso de ódio em redes sociais: jurisprudência brasileira. *Revista direito GV*, SciELO Brasil, v. 7, n. 2, p. 445–467, 2011.

SIVIC, J.; ZISSERMAN, A. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 31, n. 4, p. 591–606, 2009.

SOUZA, E.; ALVES, T.; TELES, I.; OLIVEIRA, A. L.; GUSMÃO, C. Topie: An open-source opinion mining pipeline to analyze consumers' sentiment in brazilian portuguese. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2016. p. 95–105.

STATISTA. *Number of active Twitter users in leading markets as of May 2016*. 2016. <https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/>. [Online; Acessado em :18-02-2018].

STEVEN, B.; KLEIN, E.; LOPER, E. Natural language processing with python. *OReilly Media Inc*, 2009.

TELES, I. *Uma análise comparativa de técnicas supervisionadas para mineração de opinião de consumidores brasileiros*. Serra Talhada: [s.n.], 2016.

TRIPATHY, A.; AGRAWAL, A.; RATH, S. K. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, Elsevier, v. 57, p. 117–126, 2016.

TWITTER. *Regras do Twitter e Políticas*. 2019. <https://help.twitter.com/pt/rules-and-policies>. [Online; Acessado em :19-01-2019].

ULLMAN, J. D. *Mining of massive datasets*. [S.l.]: Cambridge University Press, 2011.

WEISS, S.; INDURKHYA, N.; ZHANG, T.; TEXTMINING, F. D. *PredictiveMethodsforAnalysing Unstructured Information*. [S.l.]: Springer, 2005.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. *Text mining: predictive methods for analyzing unstructured information*. [S.l.]: Springer Science & Business Media, 2010.

ANEXO A – Modelo para classificação dos *twittes*

DISCURSO NORMAL = 0 DISCURSO OFENSIVO= 1 DISCURSO DE ÓDIO = -1	CLASSIFICAÇÃO DE TWITTERS

CLASSIFICAÇÃO	TWITTER	ID_TWITTER	ACCOUNT_ID
TIPO DE DISCURSO	TWITTER POSTADO	USUÁRIO	ID DA CONTA