# Biblioteca para Detecção de Discursos de Ódio

**Franciscone L. A. Junior , Igor E.C Cruz, Victor H. A. Landin**

[1]Universidade Federal de Lavras - UFLA
Lavras - MG
2019

## 1 Introdução

Atualmente, com a popularização das mídias sociais há um grande aumento na conectividade das pessoas e na dispersão de informações, onde pessoas de qualquer lugar do mundo podem se comunicar entre si e gerar todo tipo de conteúdo, sendo que esse conteúdo, geralmente, permanece disponível na internet, como por exemplo o Twitter, que é uma rede social, onde usuários podem postar textos com qualquer conteúdo de sua preferência.

Com essa liberdade que os usuários têm para comentar o que quiserem sobre qualquer assunto, muitas das vezes podem haver comentários ofensivos, os quais infringem leis ou ofendem certos grupos de pessoas. Esses comentários podem ser classificados como discurso de ódio que, de forma genérica, se caracterizam como qualquer ato de comunicação que inferiorize, incite ódio ou busque discriminar uma pessoa ou grupos de pessoas, tendo como base características como etnia, raça, gênero, nacionalidade, religião, orientação sexual ou outros aspectos passíveis de discriminação.

Como consequência do ódio proferido a elas e manifestado na internet, as vitimas desse tipo de discurso muitas das vezes tem sua honra violada, o seu psicológico afetado e podem sofrer violência física no seu cotidiano. É notável que ainda há uma dificuldade expressiva das empresas que controlam essas mídias em identificar e remover conteúdos deste tipo, o que faz com que o conteúdo permaneça exposto, sendo muitas vezes replicado e atingindo inúmeras vítimas.

Com o objetivo de ajudar na detecção de discursos de ódio, é proposto neste trabalho a criação de uma ferramenta computacional para detectar automaticamente estes discursos, utilizando técnicas de Aprendizado de Máquina. Como estes discursos são voltados para populações e grupos vulneráveis e marginalizados dentro da sociedade, seus efeitos podem causar aumento na exclusão social destes grupos, assim, a ferramenta teve como foco a detecção de discursos de ódio direcionados a pessoas negras ou pertencentes a comunidade LGBT em geral.

## 2 Revisão Bibliográfica

A analise de sentimentos e a mineração de opinião são conceitos do Processamento de Línguas Naturais amplamente aplicados e requisitados por empresas de pequeno e médio porte. O *Facebook*, por exemplo, tem usado essas mesmas técnicas para detectar *fakenews* (noticias falsas) e até mesmo conteúdos com discurso de ódio de forma automatizada. O foco desse trabalho é fazer à análise do discurso de ódio para o idioma português, visto que para o inglês e outras línguas existem mais trabalhos sobre o assunto. Para a atingir o objetivo de classificar se os textos contém ou não discurso de ódio foi necessário fazer uma coleta de dados públicos de uma rede social e posteriormente usar técnicas da analise sintática e léxica para rotular os *tweets* de acordo com seu conteúdo.

### 2.1 Análise Léxica

Quando se trata da analise de dados é melhor, a nível computacional, separar todo o processo em etapas, e no neste projeto a primeira delas é a análise léxica, que é responsável por classificar individualmente cada elemento (ou palavra) no texto, definindo seu significado.

### 2.2 Análise Sintática

Já a análise sintática trata-se de analisar a função e a ligação entre os elementos dispostas em uma sentença, extraindo a relação que estabelecem com os demais constituintes da oração, de modo a se formar um todo organizado e harmônico.

### 2.3 Algoritmos de Classificação

Algortimos de classificação são algoritmos responsáveis por categorizar elementos por determinadas características em comum, agrupando estes elementos com intuito de gerar classes distintas.

#### 2.3.1 Logistic Regression

Refere-se a uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de dados fornecidos, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série de variáveis.

### 2.4 Hatebase

O Hatebase é uma plataforma de software criada para ajudar organizações e comunidades on-line a detectar e monitorar o discurso de ódio. Os algoritmos desenvolvidos pelos criadores da base analisam conversas públicas usando um vocabulário amplo com base na nacionalidade, etnia, religião, gênero, orientação sexual, deficiência e classe, com dados em mais de 90 idiomas e mais de 175 países.[1]

## 3 Métodos

Com base nas técnicas e ferramentas supracitadas, o objetivo deste trabalho é analisar um corpus retirado do Twitter, à princípio classificando-o como discurso de ódio ou não e utilizando o corpus já classificado para treinar um algoritmo de aprendizado de máquina, utilizando o modelo obtido para identificar se uma sentença contém ou não discurso de ódio explícito.

### 3.1 Obtenção do Corpus

Um corpus trata-se de um conjunto de dados de um determinado assunto. A fim de obtermos o corpus que utilizamos neste projeto com auxilio da API do Twitter que a partir de um dado perfil, ou uma lista de perfis, é possível através de seus seguidores minerar os *tweets* destes seguidores de forma anônima protegendo a identidade destes usuários pela política de privacidade.

### 3.2 Normalização do corpus

Os dados extraídos do twitter e salvos na tabela .cvs forma parte do nosso corpus, pois se utilizássemos desta forma poderia gerar resultados não tão satisfatórios. Para a obtenção de um melhor resultado é necessário trabalhar em cima destes dados afim de normaliza-los, técnica que consiste em deixar o texto de forma mais clara possível.

Para normalização de nosso corpus criamos uma função chamada *normalize()*, esta função recebe por parâmetro cada *tweet* e através de uma rotina de tratamento composta por mais seis métodos resulta em um texto trabalhado e organizado. Os metodos de normalização são os seguintes:

- *remove_username()*: este método percorre o *tweet* trabalhado e por meio de uma expressão regular remove do *tweet* as citações de outro usuário que basicamente trata-se um @ seguido de caracteres ate um espaço em branco.
- *remove_end_line()*: para evitr linhas em branco, como o próprio nome sugere remove as quebras linha do texto.
- *remove_duplicate_letters()*: é muito comum que as pessoas se expressem virtualmente com o uso de letras repetidas em uma única palavra como por exemplo: ”adooooorooooo”. Este tipo de uso pode fazer com que o algoritmo nao entenda o significado da palavra restringindo seu entendimento. Devido isso com uso de uma expressão regular é removido estas letras repetidas dos *tweets*.
- *lowercase()* coloca todas os caracteres do *tweet* em letras minusculas.
- remove_punctuation() remove os sinais de pontuação dos tweets, uma vez que para a análise em questão não vimos necessidade de mante-los.
- *tokenize_words()* tranforma todo o tweet em uma lista de tokens com uso do *spaCy*.

Após a normalização dos *tweet* tornar-se possível o preprocessamentos dos *tweets*, utilização e treinamento dos dados a partir deles.

### 3.3 Preprocessamento dos Tweets

A princípio foi utilizado o *hatebase* que é um dicionário de palavras ofensivas, porém não houve bons resultados, fazendo com que desenvolvêssemos outra solução na mesma linha de pensamento, com isso foi levantado pelos autores quatro listas, que chamamos de *hate lists*, diferentes de cunho ofensivo ao público alvo, classificados como hate_certeiras, hate_verb, hate_adj e hate_words, sendo o primeiro composto por palavras de cunho ofensivo muito alto onde sua presença já caracteriza um discurso de ódio explícito, o segundo composto por verbos ofensivos que expressam ações de violência e ódio, o terceiro composto por adjetivos que expressam ofensas ao público alvo do trabalho e quarto composto por outras palavras de cunho ofensivo com papel sintático diferente dos mencionados anteriormente.

Este preprocessamento gerado de forma extrativa e exaustiva dos *tweets*, consistindo simplesmente em analisar cada *tweet* e identificar a existência das palavras contidas nas *hate lists*, caso a palavra exista no *tweet* e esta em papel sintático que identificamos como sendo característico de cunho ofensivo o tweet em questão é *taggeado* como 1, para *hate* detectado ou 0, para *hate* não detectado.

### 3.4 O Treinamento

A etapa de preprocessamento tem a ideia de gerar apenas uma base de forma mais automáticas da classificação de quais *tweets* se encaixam em discursos de ódio, uma vez que fazer essa tarefa de forma manual resultaria em uma enorme demanda de tempo considerando o tamanho do corpus obtido. Contudo esta abordagem de análise não é muito interessante devido ao tempo gasto e podendo gerar consideráveis mal entendidos, com isso partimos para o treinamento do corpus com uso de aprendizado de máquina, mais especificamente com o uso do algoritmo contido na biblioteca do *scikit-learn* o *LogisticRegression*.

Após o taggeamento dos *tweets* em *hate(1)* ou *não hate(0)* iniciamos o preparo para treinamento dividindo o corpus em duas classes, sendo elas treino e testes, a partir disso com uso do *TfidVectorizer* preparamos as classes para que o *LogisticRegression* consiga treinar a classes de treino de forma adequada.

## 4 Resultados

Os resultados foram gerados de forma parcial divididos em algumas etapas e somente na etapa final ocorre criação de um modelo treinado o suficiente para identificação de discursos de ódio.

A primeira etapa baseia-se na execução da API do Twitter descrita no tópico *Métodos* sobre quatro perfis de usuários e a partir dos seguidores deles salvamos em um arquivo .csv os mais variados *tweets* encontrados durante a mineração resultando em um total de 85.181 **tweets**.

A segunda etapa é a fase de normalização dos dados, onde na primeira etapa foi gerado uma planilha contendo as colunas Unnamed, id e text, na segunda etapa esta planilha .cvs foi lida e gerando a partir dela um *dataframe* para que facilitasse a manutenção e visualização dos dados. Criamos uma quarta coluna que conteve o *tweet* normalizados com as rotinas citadas anteriormente onde já se tornou possível uma melhor vizualização dos tweets e as palavras sendo exibidas de forma mais clara.

No preprocessamento, terceira etapa, com uso das *hate lists* conseguimos classificar o corpus apesar exaustivamente mas de forma automática separando os *tweets* em 80% para treino e 20% para testes, onde do corpus total de 85.181 *tweets* 36.993 foram identificados como discursos de ódio e o restante de 48.188 como não. Apenas com essa fase de abordagem extrativa já é possível notar que existe uma grande quantidade de comentários de cunho ofensivo.

Na quarta etapa iniciamos o procedimento de treinamento do corpus. Primeiramente dividimos o corpus de forma aleatória afim de manter os dados homogêneos pois a escolha dos dados podem influenciar no resultado tendendo a um modelo *viciado* que poderá sempre gerar resultados positivos ou negativos. Visto o corpus foi separado em 80% dos dados destinados a treino e 20% dos dados destinados a teste.

Após o treinamento do modelo os resul-

tado obtidos atingiram uma métrica de 0.95037 ou aproximadamente 95% de acurácia de acordo com o medidor do *LinearRegression*. Entendemos que foi um resultado relativamente alto baseando-se na análise feita pelo algorítimo de avaliação da acurácia.

## 5 Considerações Finais

Apesar da acurácia de 95% de acordo com o algoritmo de medição do *LinearRegression* acreditamos que com um corpus taggeado de forma manual ou até mesmo com uma rígida revisão manual dos resultados obtidos pelo nosso algoritmo de taggeamento extrativo possamos obter melhores resultados, pois nos testes de mesa não conseguimos um resultado tão satisfatório quanto gostaríamos, acreditamos que seja devido ao palavras utilizadas em nossas *hate list* ou uma etiquetagem que pode ser melhor trabalhada.

## Referências

[1] Hatebase Inc. *The world's largest structured repository of regionalized, multilingual hate speech*. 2019. Disponível em: `https://hatebase.org//`.