UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CLEITON DE LIMA PINTO

EXTRAÇÃO DE CARACTERÍSTICAS PARA IDENTIFICAÇÃO DE DISCURSO DE ÓDIO EM DOCUMENTOS

CHAPECÓ
2018

CLEITON DE LIMA PINTO

EXTRAÇÃO DE CARACTERÍSTICAS PARA IDENTIFICAÇÃO DE DISCURSO DE ÓDIO EM DOCUMENTOS

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

CHAPECÓ
2018

de Lima Pinto, Cleiton

Extração de características para identificação de discurso de ódio em documentos / por Cleiton de Lima Pinto. – 2018.
39 f.: il.; 30 cm.

Orientador: Guilherme Dal Bianco
Monografia (Graduação) - Universidade Federal da Fronteira Sul, Ciência da Computação, Curso de Ciência da Computação, RS, 2018.

1. Discurso de Ódio. 2. Aprendizado de Máquina. 3. Classificação de Texto. I. Dal Bianco, Guilherme. II. Título.

E-mail: cleiton.limapin@gmail.com

CLEITON DE LIMA PINTO

EXTRAÇÃO DE CARACTERÍSTICAS PARA IDENTIFICAÇÃO DE DISCURSO DE ÓDIO EM DOCUMENTOS

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

Este trabalho de conclusão de curso foi defendido e aprovado pela banca em: 07/12/18

BANCA EXAMINADORA:

Dr. Guilherme Dal Bianco - UFFS

Dr. Denio Duarte - UFFS

Ma. Andressa Sebben - UFFS

RESUMO

As mídias sociais estão cada vez mais presentes na vida das pessoas, incluindo ferramentas que permitem que usuário colabore com a criação do conteúdo nela exposto. Muitos usuários se aproveitam dessa funcionalidade para disseminar conteúdo ilícito ou criminoso. Caso não seja removido, este conteúdo será visto por cada vez mais pessoas e poderá ser propagado pela internet, atingindo um número maior de vítimas e incentivando a ocorrência de outros crimes. Esse tipo de crime geralmente é voltado aos grupos mais vulneráveis da sociedade, e seus efeitos nocivos podem causar o aumento da exclusão social e da violência praticada contra esses grupos. Este trabalho propõe explorar e extrair características de textos utilizando técnicas de processamento de linguagem natural e aprendizado de máquina para detectar automaticamente discursos de ódio. Os experimentos demonstraram que o método foi capaz de melhorar em até 5% em relação ao método base.

Palavras-chave: Discurso de Ódio. Aprendizado de Máquina. Classificação de Texto.

ABSTRACT

Social media is increasingly present in people's lives, including tools that allow users to collaborate with the creation of the content exposed in it. Many users use this functionality to post texts spreading illicit or criminal content. This offensive content need be removed soon as possible otherwise more and more people we see and can propagate through the internet, reaching a more significant number of victims encouraging the occurrence of other crimes. This work proposes the extraction of characteristics from text using natural language processing techniques and machine learning to detect hate speech automatically. This type of hate crime, in general, is focused on the most vulnerable groups in society, and the harmful effects can lead to increased social exclusion and violence against such groups.

Keywords: Hate speech. Machine Learning. Text Classification.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RSO	Redes Sociais Online
BoW	Bag-of-Words
KNN	K-Nearest-Neighbor
PLN	Processamento de Linguagem Natural
PoS	Part-of-Speech
TF	Term Frequency
IDF	Inverse Document Frequency
TF-IDF	Term Frequency–Inverse Document Frequency
SVM	Support Vector Machine
NB	Naive Bayes

SUMÁRIO

1 INTRODUÇÃO

1.1 Tema

Este trabalho explora a criação de novas abordagens para extração de características utilizadas em modelos de predição que contribuem para a identificação de discurso de ódio em dados textuais.

1.2 Delimitação do Problema

Com o advento das redes sociais online (RSO), cada vez mais pessoas expõem suas ideias e opiniões nestes ambientes. Os usuários exploram aspectos de RSO, como o anonimato e políticas frágeis de publicação de conteúdo, para disseminar mensagens de discurso de ódio, como por exemplo racismo, xenofobia e homofobia, etc (NAKAMURA et al., 2017). O discurso de ódio é comumente definido como qualquer comunicação que deprecie uma pessoa ou um grupo com base em alguma característica como raça, cor, etnia, gênero, orientação sexual, nacionalidade, religião ou outra característica (NOCKLEBY, 2000).

Devido à quantidade de dados que são gerados a cada dia, a auditoria manual de seu conteúdo para identificar discurso de ódio se torna uma tarefa impraticável. Filtros básicos de conteúdo, como expressões regulares ou *blacklist*, que filtram o conteúdo de determinadas palavras, muitas vezes não fornecem uma solução adequada para a classificação (SCHMIDT; WIEGAND, 2017). Com isso a classificação de texto - a atividade de rotular textos de linguagem natural com categorias - está sendo aplicada em muitos contextos, desde a indexação de documentos baseada em um vocabulário controlado até a filtragem de documentos, com geração automatizada de metadados e desambiguação do sentido de palavra (SEBASTIANI, 2002).

Através da classificação de textos e o aprendizado de máquina é possível identificar discurso de ódio em documentos de forma automática. Para tal tarefa, métodos supervisionados de aprendizagem de máquina são aplicados para a criação de modelos que predizem se determinado documento se enquadra como discurso de ódio ou não. Segundo (BATISTA et al., 2003), no aprendizado supervisionado é fornecido ao sistema de aprendizado um conjunto de exemplos $E = \{E_1, E_2, ..., E_n\}$, sendo que cada exemplo $E_i \in E$ possui um rótulo associado. O rótulo determina a qual classe o exemplo pertence. Através de um nova entrada não rotulada, o classificador é capaz de predizer a classe à qual o dado se assemelha. Práticas de aprendizado

de máquina são cada vez mais comuns e se tornam uma fonte de informação para empresas, governos e pesquisadores (NAKAMURA et al., 2017).

Para o funcionamento dos algoritmos de aprendizagem de máquina e classificação de textos, é preciso que os dados possuam determinadas características. Essas características ou atributos nada mais do que informações que descrevem determinado documento. Normalmente esses atributos são exibidos de forma estruturada, como no exemplo da Tabela 1.1, um conjunto de dados que indicam se uma pessoa comprou ou não um carro, onde as colunas $portas, Passageiros, Seguranca, Valor$ representam as informações dos dados e a coluna $Comprou$ representa qual classificação foi atribuída ao dado.

Portas	**Passageiros**	**Segurança**	**Valor**	**Comprou?**
2	4	baixa	10000	Sim
4	4	média	15000	Sim
2	2	alta	50000	Não
4	5	baixa	25000	Não

Tabela 1.1: Exemplo de um conjunto de dados e suas características.

Ao trabalhar com documentos, os dados são somente conteúdo de texto que apresentam dados não estruturados. No entanto, existem métodos para a extração de características dos textos que nos auxiliam na classificação desses dados. Alguns desses métodos serão vistos no Capítulo 2.

Desse modo, a extração de características possibilita montar modelos de classificação que identificam se determinado documento possui ou não um discurso de ódio.

1.3 Objetivos

1.3.1 Objetivo Geral

Explorar e extrair características que ajudem modelos de predição a identificar textos ofensivos em documentos da web.

1.3.2 Objetivos Específicos

- Realizar pesquisas metodológica para analisar trabalhos relacionados que abordam o tema de classificação de texto com o enfoque na identificação de discurso ofensivo;

- Definir uma base de dados para o desenvolvimento deste trabalho onde serão aplicados os métodos para a extração de características;

- Criar características que serão utilizadas neste trabalho com base em trabalhos relacionados na área;

- Relacionar métodos supervisionados de aprendizagem de máquina usados em outros trabalhos para classificar os textos da base de dados;

- Aplicar as métricas que irão medir a eficiência do método proposto;

- Executar diferentes experimentos para identificar a eficiência do método.

1.4 Justificativa

Com a popularização das redes sociais, há um grande volume de dados que se originam através dos conteúdos gerados pelos usuários, expondo suas opiniões e que dificilmente passam por algum tipo de auditoria para a identificação de textos ofensivos. Classificar grandes volumes de documentos de forma manual é uma tarefa que demanda um número expressivo de pessoas para a sua realização.

A classificação de documentos utilizando aprendizado de máquina para resolver esse tipo de problema vem sendo estudado por muitas empresas que sofrem com essa adversidade, dentre as quais destacam-se o *Facebook* e *Twitter* (NOBATA et al., 2016).

Para realizar a classificação dos documentos, diversos métodos de processamento de linguagem natural e algoritmos de aprendizagem de máquina supervisionados podem ser empregados.

Reconhecer o contexto dos documentos para identificar quais características serão extraídas dos textos é importante, tendo em vista que, os usuários que publicam os discursos de ódio tendem a disfarçar palavras ofensivas, dificultando ainda mais a extração de informações. Do mesmo modo, destaca-se que tanto os dados quanto a linguagem mudam, sendo necessária a combinação entre métodos e características para auxiliar modelos a realizar predição dos dados, tarefa de grande importância no contexto de classificação de documentos (NOBATA et al., 2016).

1.5 Estrutura do Trabalho

Este trabalho está estruturado em 6 capítulos. No primeiro Capítulo foi apresentada a introdução, definindo o tema de pesquisa, os objetivos e a justificativa. O Capítulo 2 relata a fundamentação teórica. No capítulo 3 são abordados os trabalhos relacionados. O Capítulo 4 apresenta a proposta para este trabalho. No capítulo 5 apresenta-se os experimentos realizados e os resultados obtidos. Por fim, o capítulo que traz a conclusão deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados alguns métodos importantes para o entendimento do trabalho. Tais métodos têm como objetivo a criação de atributos numéricos a partir de um conjunto de palavras (documentos). Alguns métodos destinam-se à criação de atributos como, *Bag-of-words* apresentado na Seção 2.1 e *N-gram* na Seção 2.2. Outros métodos como *TfIdf*, (Seção 2.3) e Stemização (Seção 2.4) tem como objetivo a atribuição de valores às características do documento.

2.1 *Bag-of-Words*

Um modelo *bag-of-words*, ou BoW, é uma maneira de extrair atributos do texto para alimentar os algoritmos de aprendizado de máquina. Na abordagem, olha-se para o histograma das palavras dentro do texto, ou seja, considerando cada contagem de palavras como uma característica (GOLDBERG, 2017).

Bag-of-words é uma representação de texto que descreve a ocorrência de palavras em um documento. Como pode ser visto na Figura 2.1, as palavras dos documentos são separadas de forma distinta montando vetores que representam *palavra* e *frequência*, na qual *palavra* é o termo usado no documento e a *frequência* é o número de vezes que esse termo aparece nos documentos.

Qualquer informação sobre a ordem ou estrutura das palavras no documento é desconsiderada. O modelo só se preocupa com o fato de palavras conhecidas ocorrerem no documento e não com sua localização.

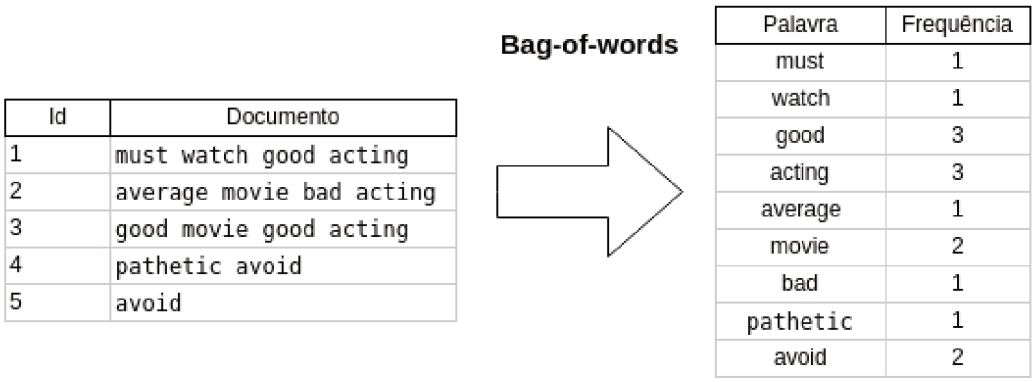

Figura 2.1: Exemplo do uso de BoW.

2.2 *N-gram*

N-gramas de textos são amplamente utilizados em tarefas de mineração de texto e processamento de linguagem natural. N-gram são basicamente uma sequência de termos com o comprimento de N caracteres.

Usando como exemplo a Figura 2.2, ao utilizar N-gram em um determinado documento, cada termo é dividido em N sequências, onde cada um representa uma nova característica. Tendo $N = 1$ (unigram) cada termo é separado individualmente. Para o $N = 2$ (bigram), os termos são divididos de dois em dois e o mesmo serve para o $N = 3$.

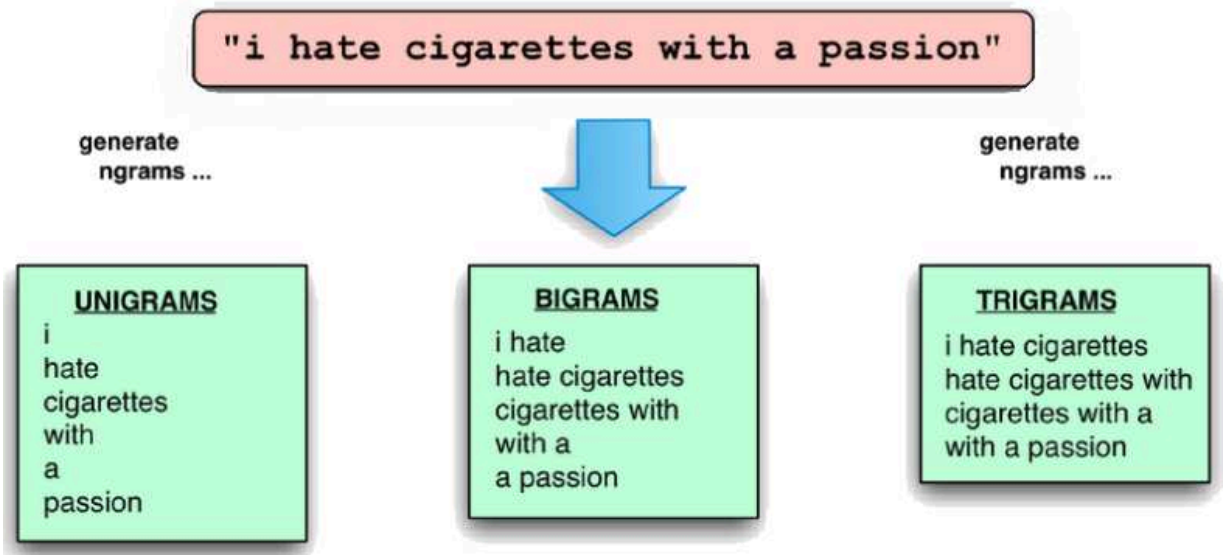

Figura 2.2: Exemplo do uso de N-gram (RESEARCHGATE, 2018).

2.3 TF-IDF

O *TF-IDF* tem como objetivo atribuir um peso para as palavras que aparecem com menos frequência nos documentos. Muitas vezes a importância de uma palavra não se dá somente pela sua frequência.

Documentos maiores, geralmente, são representados por muitos termos. Quando uma grande quantidade de termos é utilizada na representação de documentos, a probabilidade do termo pertencer a um documento é alta e, assim, documentos maiores têm melhores chances de serem relevantes do que documentos menores (MARTINS; MONARD; MATSUBARA, 2003), como é o caso de *bag-of-words*, visto na Seção 2.1.

O *term frequency (tf)* é uma métrica que utiliza o número de ocorrências do termo tj no

documento di. No entanto, quando termos com alta frequência aparecem em todos (ou na maioria) dos documentos da coleção, os mesmos não fornecem informação útil para os atributos nos documentos. Assim, a medida *inverse document frequency (idf)* favorece termos que aparecem em poucos documentos da coleção. As definição do TF e do IDF serão apresentadas a seguir:

$$tf = \frac{tj}{di}$$

onde tf é o número de vezes que o termo tj ocorre em um documento e di é o número de termos do documento.

$$idf = log(1 + \frac{d}{t})$$

onde t é o número de documentos que contém o termo e d é o número total de documentos (*corpus*);

$$tfidf = tf * idf$$

no qual as medidas tf e idf são combinadas.

Na Tabela 2.1 são apresentadas as características dos documentos da Figura 2.1, porém com os pesos de cada característica utilizando *tf-idf*.

	acting	average	avoid	bad	good	movie	must	pathetic	watch
0	0.380406	0.000000	0.000000	0.000000	0.458270	0.000000	0.568014	0.000000	0.568014
1	0.380406	0.568014	0.000000	0.568014	0.000000	0.458270	0.000000	0.000000	0.000000
2	0.348021	0.000000	0.000000	0.000000	0.838514	0.419257	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.627914	0.000000	0.000000	0.000000	0.000000	0.778283	0.000000
4	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Tabela 2.1: Características com *tf-idf*.

2.4 Stemização

Na criação da tabela atributo-valor utilizada por algoritmos de Aprendizado de Máquina, cada termo que aparece no documento pode ser mapeado como uma coluna na tabela. Assim, o número de dimensões do conjunto de atributos pode ser grande, gerando um problema que deve ser minimizado.

Vários métodos podem ser utilizados a fim de reduzir a quantidade de atributos visando uma melhor representatividade e melhor desempenho do processo de predição. Entre outros, a transformação de cada termo para o radical que o originou, por meio de algoritmos de stemização (*stemming*), é um método amplamente utilizado e difundido, conforme (MARTINS; MONARD; MATSUBARA, 2003).

Segundo (JIVANI et al., 2011), a stemização é uma etapa de pré-processamento na mineração de textos e recuperação de informação, bem como um requisito muito comum de funções de processamento de linguagem natural (PNL). Pode-se dizer que o objetivo da stemização é reduzir palavras para uma forma mais básica e comum de escrita. A ideia principal é melhorar o manuseio automático de terminações de palavras, reduzindo as mesmas às suas raízes de palavras. Geralmente, a stemização é feita removendo quaisquer sufixos e prefixos (afixos) anexados nas palavras, dado que o radical de um termo representa um conceito mais amplo do que o termo original. Na Tabela 2.2 é exemplificada a aplicação da stemização para a redução das variações do radical *quilometr*.

Na stemização, a conversão de formas morfológicas de uma palavra ao seu tronco é feita supondo que cada um é semanticamente relacionado. O radical não precisa ser uma palavra existente no conjunto de atributo-valor, mas todas as suas variantes devem ser mapeadas para este radical. Há dois pontos a serem considerados ao usar o método de stemização:

- se as formas morfológicas de uma palavra têm o mesmo significado básico, devem ser mapeadas para o mesmo radical;
- palavras que não têm o mesmo significado devem ser mantidas separadamente.

Essas duas regras são boas o suficiente, desde que os valores resultantes sejam úteis para a mineração de texto ou PNL. Segundo (JIVANI et al., 2011), a utilização da stemização é geralmente considerada como um dispositivo de melhoria de revocação (frequência em que um classificador encontra os exemplos de uma classe). Para linguagens com morfologia relativamente simples, a influência da stemização é menor do que para aquelas com morfologia mais complexa.

Um dos algoritmos de stemização mais conhecidos é o algoritmo de Porter que remove sufixos de termos em inglês (PORTER, 2001).

Palavra	Stem
quilométricas	quilometr
quilométricos	quilometr
quilômetro	quilometr
quilômetros	quilometr

Tabela 2.2: Exemplo de Stemização

2.5 KNN

O KNN (*K-Nearest-Neighbor*, em português, K-vizinho mais próximo) é um classificador que procura K documentos do conjunto de treinamento que estejam mais próximos deste documento com classificação desconhecida, ou seja, que tenham a menor distância.

Estes K documentos são chamados de K-vizinhos mais próximos. Verifica-se quais são as classes desses K vizinhos e a classe mais frequente será atribuída à classe do documento desconhecido.

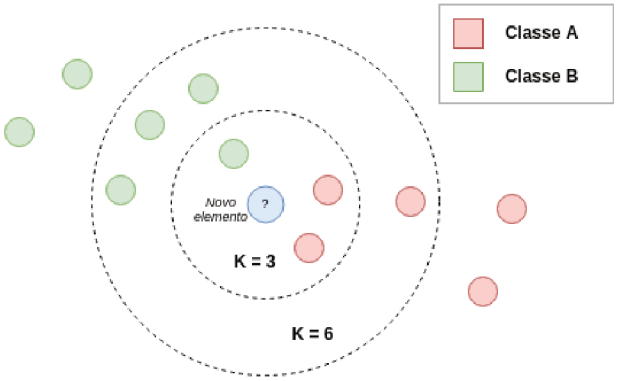

Figura 2.3: Exemplo de execução do KNN.

Como pode ser observado no exemplo da Figura 2.3 a classificação de um novo elemento (em azul) para o $K = 3$. Todos os vizinhos mais próximos desse novo elemento são classificados com a classe A, portanto, o novo elemento será classificado com a classe A. Para o $K = 6$, a maioria dos vizinhos mais próximos são de classe B, então o novo elemento é classificado como sendo de classe B.

A métrica mais comum que calcula a distância é a distância Euclidiana (SANTOS et al., 2009). Seja $X = (x_1, x_2, ..., x_n)$ e $Y = (y_1, y_2, ..., y_n)$ dois pontos de R^n, a distância Euclidiana entre X e Y é dada por:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}$$

3 TRABALHOS RELACIONADOS

Neste capítulo serão descritos os principais trabalhos relacionados a este projeto de conclusão de curso. O capítulo será dividido em duas seções: a primeira delas constituída por alguns trabalhos que identificam discurso de ódio através de modelos de predição, métodos para a criação de novos atributos que ajudam na predição dos dados, bem como os resultados de seus modelos (Seção 3.1). Na segunda seção é realizada uma análise sobre a extração de meta-atributos (atributos que possuem relação com os atributos originais), utilizando *KNN* para obter informações relevantes de documentos similares, bem como uma análise sobre os seus resultados (Seção 3.3).

3.1 Identificação de discurso de ódio

3.1.1 Detecção automatizada de discurso de ódio e o problema da linguagem ofensiva

O trabalho de (DAVIDSON et al., 2017) tem por objetivo a classificação em três classes de textos (documentos): discurso de ódio, discurso ofensivo e não ofensivo. O discurso de ódio é definido, conforme (NOCKLEBY, 2000), como qualquer comunicação que deprecie uma pessoa ou um grupo com base em alguma característica como raça, cor, etnia, gênero, orientação sexual, nacionalidade, religião ou outras. Já discurso ofensivo é categorizado, segundo (DAVIDSON et al., 2017), como documentos que têm linguagem ofensiva mas que não discrimina as pessoas ou um grupo de pessoas.

O discurso de ódio pode ser usado de diferentes maneiras:

- Pode ser enviado diretamente para uma pessoa ou grupo de pessoas;
- Pode ser direcionado para ninguém em particular (sarcasmo, ironia);
- Em conversas entre pessoas (comentários de redes sociais, fóruns de discussão, sala de bate-papos, etc).

Neste trabalho é proposta a divisão entre discurso de ódio e discurso ofensivo por alguns motivos. Em muitos casos, textos apresentam palavras ofensivas, mas que de fato não expressam ódio contra outras pessoas, por exemplo, o uso do sarcasmo entre amigos, muitas vezes considerado como brincadeira. Esse tipo de linguagem é comum nas mídias sociais (KWOK;

WANG, 2013), tornando a tarefa de classificação crucial para qualquer sistema que tente detectar discurso de ódio em documentos.

Para a construção dos dados foram utilizadas palavras e frases (*tokens*) de ódio retiradas de um repositório *on-line*[1] de discurso de ódio. Com esse conjunto de palavras e frases, foi feita uma seleção de *tweets* que continham esses *tokens*, resultando em 33.458 de usuários do twitter e 85,4 milhões de documentos (todos em inglês). Desses dados foram selecionados aleatoriamente 25.000 documentos para realização do experimento.

A classificação desses documentos foi realizada por pessoas (garantindo uma classificação mais correta) que tinham que escolher/votar em uma das três categorias que o documento pertencesse: discurso de ódio, discurso ofensivo e não ofensivo. Os classificadores foram instruídos que a presença de uma determinada palavra, embora ofensiva, não indica necessariamente que um documento é discurso de ódio. Cada documento foi classificado três ou mais vezes, sendo enquadrado na categoria que obteve mais votos. Depois dessa classificação manual, resultaram 24.802 documentos rotulados, alguns desconsiderados pela falta de votos.

Apenas 5% dos documentos foram rotulados como discurso de ódio pela maioria dos classificadores e apenas 1,3% foram classificados de forma unânime. A maioria dos documentos foi considerada discurso ofensivo (76% classificados com 2/3 votos, 53% classificados com 3/3 votos) e o restante foi considerado não ofensivo (16,6% classificado de 2/3, 11,8% em 3/3 vezes).

Para a construção dos atributos, cada documento foi formatado em letras minúsculas para evitar divergências. Também foi utilizado o algoritmo de *Porter stemmer* para a remoção dos sufixos das palavras e o *N-gram* para a construção dos atributos, com os valores de $N = 1, 2, 3$. A atribuição dos pesos para os atributos foi feita *TF-IDF*. (veja na Seção 2.3.). Para capturar informações sintáticas dos textos foi usado *Part-of-Speech (POS)* juntamente com *N-gram* para gerar novos atributos que caracterizam informações da estrutura sintática dos documentos. Para capturar a qualidade de cada documento foi usado o cálculo de *Flesch Reading Ease*, que indica se o texto é compreensível para a leitura, onde os valores vão de 0 a 100; quanto maior a pontuação, mais fácil é a leitura.

Outras características aplicadas no trabalho, levando em consideração os documentos utilizados, foram valores quantitativos extraídos dos textos como a contagem de *hashtags*, menções, *retweets* e *URLs*, bem como recursos para o número de caracteres, palavras e sílabas em

[1] https://www.hatebase.org/

cada *tweet*.

Para criação do modelo de predição foram testados uma variedade de modelos: regressão Logística, *Naive Bayes*, Árvores de Decisão, Florestas Aleatórias e o *SVM*. Cada modelo foi testado usando validação cruzada de 5 vezes, mantendo 10% da amostra para avaliação. Os resultados do (DAVIDSON et al., 2017) mostram que Regressão Logística e *SVM* tendem a ter um desempenho significativamente melhor do que outros modelos.

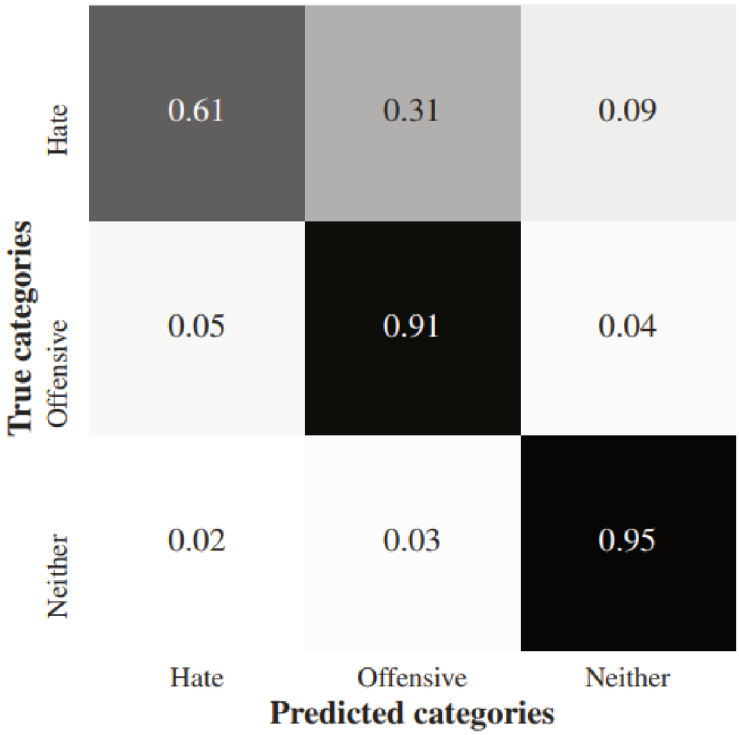

Figura 3.1: Matriz de confusão dos resultados com *F1-Score* (DAVIDSON et al., 2017).

Segundo o artigo, o classificador com melhor desempenho teve uma precisão geral de 91%, revocação de 90% e pontuação de F1 de 90%. No entanto, observando a Figura 3.1 percebe-se que quase 40% dos discursos de ódio são classificados incorretamente: os resultados de precisão e revocação para a classe de ódio são 44% e 61%, respectivamente. A maior parte da classificação incorreta ocorre no triângulo superior da matriz, sugerindo que o modelo é favorável a classificar os documentos como menos ódio ou ofensivos do que os codificadores humanos. Um menor número de documentos são classificados como mais ofensivos ou odiosos do que sua verdadeira categoria, aproximadamente 5% de documentos ofensivos e 2% de não-ofensivo foram erroneamente classificados como discurso de ódio.

3.1.2 Detecção de Linguagem Abusiva em Conteúdos Online

Em (NOBATA et al., 2016) são utilizados vários métodos de Processamento de Linguagem Natural (PLN) para criação de atributos. Tal trabalho combina algumas características do trabalho apresentado na Seção 3.1.1 e propõe alguns novos atributos para aprimorar os resultados. Diversos métodos de PLN foram usados em trabalhos anteriores até então estudados, mas esses recursos nunca foram combinados ou avaliados uns contra os outros para identificar se existe ganho de informação com a combinação. No estudo foi proposto exatamente uma junção de vários métodos para a criação e seleção dos atributos que identificam discurso de ódio em documentos.

No trabalho, a classificação contém as categorias de texto *abusivo* e *não abusivo*. Textos abusivos se referem ao discurso de ódio (comunicação que deprecie uma pessoa ou um grupo de pessoas), ao passo que os não abusivos não contêm discurso de ódio.

Os dados utilizados no trabalho para a detecção de linguagem abusiva são todos de comentários do site do *Yahoo*, mais especificamente, comentários pertencentes as categorias de finanças e notícias. Comentários abusivos da categoria de finanças representam 7% e 16,4% na categoria de notícias.

Para construção dos atributos, foram usadas características que podem ser divididas em quatro classes: *N-grams*, características linguística, características sintáticas e distribuição semântica. Em características linguísticas, são usadas informações do texto para a criação de atributos, como em alguns exemplos:

- Tamanho dos comentários em *tokens*;
- Tamanho médio das palavras;
- Número de pontuações do documento;
- Quantidade de letras capitalizadas;
- Quantidade de palavras desconhecidas do dicionário (em inglês).

Já as características sintáticas se preocupam em classificar os termos do documento. Um dos métodos para realizar esse tipo de operação, *Part-of-Speech* (PoS), é o processo de marcação de uma palavra em um texto como correspondente a uma parte específica de discurso. A marcação é feita através de *tags* que identificam em que parte específica do discurso a palavra se encontra, como por exemplo, marcar palavras que são substantivos, verbos, nomes próprios, entre outros.

Distribuição semântica é um subcampo do PLN que aprende o significado dos usos das palavras. Alguns métodos que trabalham com características semânticas foram referenciados no artigo. Um exemplo é o método *Word2vec*, que são redes neurais superficiais de duas camadas treinadas para reconstruir contextos linguísticos de palavras através da coleção de documentos, assim podendo gerar novos atributos.

O modelo foi treinado e testado utilizando o conjunto de dados para ambos os domínios (finanças e notícias). Para cada domínio, foi usado 80% para treinamento e 20% para teste. A Figura 3.2 mostra a tabela dos resultados de cada domínio quando um modelo treinou com um único tipo de recurso e com todos os recursos combinados. Para ambos os domínios, a combinação de todos os recursos gera o melhor desempenho (79,5% para finanças e 81,7% para notícias). As notícias têm uma ligeira vantagem de desempenho, devido ao conjunto de treinamento maior disponível para esse domínio.

Features	**Finance**	**News**
Lexicon	0.539	0.522
Trained Lexicon	0.656	0.669
Linguistic	0.558	0.601
Token N-grams	0.722	0.740
Character N-grams	0.726	0.769
Syntactic	0.689	0.748
word2vec	0.653	0.698
pretrained	0.602	0.649
comment2vec	0.680	0.758
All Features	**0.795**	**0.817**

Figura 3.2: Tabela de resultados para os atributos utilizados do modelo de predição com *F1-Score* (NOBATA et al., 2016).

3.2 Comentários Ofensivos na Web Brasileira

No trabalho (PELLE; MOREIRA, 2017) é apresentado um conjunto de dados com comentários ofensivos (e não ofensivos) coletados na web brasileira. Juntamente com os dados, são apresentados resultados de algoritmos de classificação que servem como base para demais trabalhos futuros.

Os dados foram extraídos de um site brasileiro de notícias. Dentre eles foram selecionados 1.250 comentários aleatoriamente. Seguindo o padrão adotado para a classificação dos dados, cada comentário foi anotado por três pessoas com o objetivo de dizer se o comentário é ofensivo ou não.

Foram criadas duas base de dados. A primeira, chamada de *OffComBR-2*, possui todos os 1.250 registros (419 classificados como ofensivos) e a classificação atribuída a cada comentário foi determinada por pelo menos duas pessoas. Já a segunda base de dados, *OffComBR-3*, possui 1.033 registros (202 ofensivos) e cada comentário foi classificado pelas três pessoas.

Para a classificação dos textos foram criados alguns conjuntos para testes, utilizando métodos PLN, totalizando 24 conjuntos, 12 para cada base de dados, como pode ser visualizado na Figura 3.3, na qual é possível ver o nome de cada conjunto e o número de características. Segue abaixo os métodos utilizados para formação dos conjuntos:

- Redução do texto: nesse método, são criados conjuntos de dados com o texto original (*original*) e outros com o texto em caixa baixa (*lower*).

- Tokenização: o texto é dividido em *tokens*, onde cada *token* é formado por *n-grams* e cada *unigram* ($1G$), *unigram* e *bigram* ($1G + 2G$) e *unigram*, *bigram* e *trigram* ($1G + 2G + 3G$) são combinados. Cada *token* representa como uma nova característica no conjunto a ser classificado.

- Ganho de informação: são selecionadas somente características que têm correlação positiva com a classe a ser predita (FS), assim tendo uma redução na quantidade de atributos.

Os algoritmos usado para realizar a classificação do texto foram SVM (*Support Vector Machine*) e NB (*Naive Bayes*). Como mostra na Figura 3.4, SVM obteve um melhor resultado em ambas as bases de dados. A base de dados *OffComBR-3* teve um melhor resultado, pois a pré-classificação de texto ofensivo está mais definida, e o texto ofensivo foi classificado pelas três pessoas, ao contrário da *OffComBR-2*.

OFFCOMBR-2		OFFCOMBR-3	
File	**#features**	**file**	**#features**
original_1G	4,980	original_1G	4,348
original_1G_FS	241	original_1G_FS	119
original_1G+2G	17,374	original_1G+2G	15,085
original_1G+2G_FS	396	original_1G+2G_FS	164
original_1G+2G+3G	30,711	original_1G+2G+3G	26,600
original_1G+2G+3G_FS	448	original_1G+2G+3G_FS	182
lower_1G	4,123	lower_1G	3,647
lower_1G_FS	250	lower_1G_FS	122
lower_1G+2G	15,899	lower_1G+2G	12,385
lower_1G+2G_FS	426	lower_1G+2G_FS	103
lower_1G+2G+3G	29,126	lower_1G+2G+3G	25,303
lower_1G+2G+3G_FS	489	lower_1G+2G+3G_FS	196

Figura 3.3: Estatística dos conjuntos de dados (PELLE; MOREIRA, 2017).

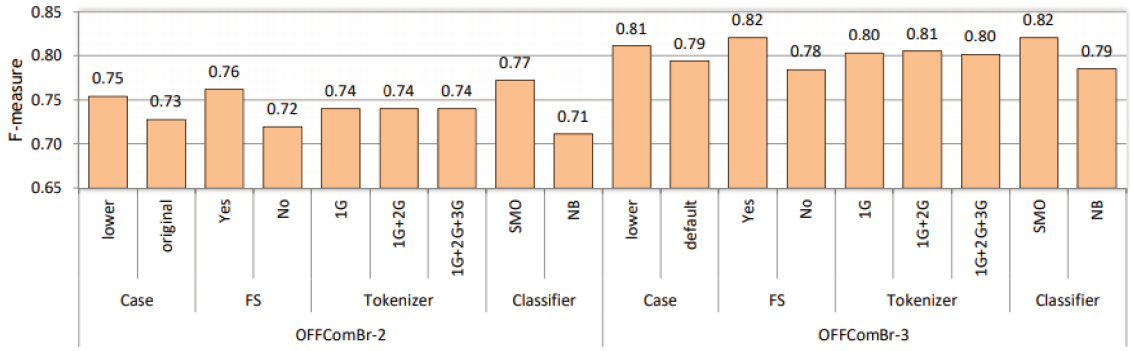

Figura 3.4: Resultados da Classificação dos textos para cada conjunto de características (PELLE; MOREIRA, 2017).

3.3 Extração de meta-atributos

O trabalho (CANUTO et al., 2013) apresenta um estudo sobre extração de meta-atributos para classificação de textos. Meta-atributos são, em geral, manualmente projetados e extraídos de outros atributos no qual o conjunto de treinamento já é rotulado, e capturam relações fundamentais entre o par *(documento, classe)* ou a tripla *(documento, classe, algoritmo)*.

Os meta-atributos são capturados usando a vizinhança/similaridade de documentos previamente classificados utilizando o algoritmo de KNN para identificar os K vizinhos próximos. O trabalho apresenta um estudo comparativo entre utilizar somente meta-atributos, somente os atributos originais da classificação de textos e combinar meta-atributos com os atributos originais.

Os meta-atributos baseados em KNN ($M = mf$) propostos contém os vetores de meta-

atributos expressos como a concatenação dos sub-vetores descritos a seguir. Cada vetor de atributos mf é definido para um exemplo $xf \in X$ e categoria $cj \in C$ para $j = 1, 2, ..., m$. Segue abaixo os três meta-atributos propostos no artigo:

- $\vec{v}^{cnt}_{\vec{x_f}} = [n_j]$: consiste em um vetor de uma dimensão (tamanho 1) dado pela contagem dos n_j vizinhos (entre os k vizinhos) de $\vec{x_f}$ que são exemplos de treino associados à determinada categoria c_j.

- $\vec{v}^{ncnt}_{\vec{x_f}} = [\frac{n_j}{n_{max}}]$: consistem em um vetor unidimensional dado pelo número n_j de vizinhos (entre os k vizinhos) de $\vec{x_f}$. O valor de n_{max} corresponde ao número de exemplos associados à classe com o maior número de exemplos dentre os vizinhos mais próximos.

- $\vec{v}^{qrt}_{\vec{x_f}} = [\cos(\vec{x}_{ej}, \vec{x_f})]$: um vetor de dimensão 5 produzido ao considerar cinco pontos que caracterizam a distribuição de distâncias de $\vec{x_f}$ para seus j vizinhos de dada categoria. As distâncias entre dois vetores $\vec{a}$ e $\vec{b}$ são computadas por similaridade do cosseno, denotada como $\cos(\vec{a}, \vec{b})$. Entre todos os pontos de distância entre $\vec{x_f}$ e seus j vizinhos de dada categoria, os cinco pontos selecionados $\cos(\vec{x}_{1j}, \vec{x_f}), \cos(\vec{x}_{2j}, \vec{x_f}), ..., \cos(\vec{x}_{5j}, \vec{x_f})$ correspondem, respectivamente, à menor distância, à maior distância, à distância média, o quartil inferior (valor que delimita os 25% dos menores pontos) e o quartil superior (valor que delimita os 25% dos maiores pontos).

Os meta-atributos descritos acima têm uma dimensão de 7 por categoria. Esse pequeno conjunto de meta-atributos é capaz de capturar informação do conjunto rotulado de três diferentes formas (conforme descrito nos itens acima). A primeira simplesmente conta o número de exemplos rotulados de cada categoria entre os k mais similares exemplos rotulados. A segunda divide o número de vizinhos em cada classe pelo número de vizinhos da classe com maior número de vizinhos, com objetivo de capturar a relação entre a classe escolhida pelo KNN (a classe com maior número de vizinhos) e as outras classes. A última informação fornecida com os meta-atributos propostos é baseada em uma análise das distâncias e distribuição das classes observada na vizinhança do exemplo. Os pontos que caracterizam essas informações são: a menor distância, a maior distância, a mediana, o quartil inferior e o quartil superior.

Os experimentos realizados são em cinco coleções de textos amplamente utilizadas na literatura de classificação automática de texto:

1. *4 Universities (4UNI)*. Essa coleção contém páginas da Web coletadas de departamentos

de Ciência da Computação de quatro universidades. Existem ao todo 8.277 páginas Web, classificadas em 7 categorias;

2. *Reuters (REUT)*. Coleção de textos classificados, com artigos de notícias coletados e anotados. Foram considerados 8.184 artigos, classificados em 8 categorias;

3. *ACM-DL (ACM)*. Um subconjunto da Biblioteca Digital da ACM, com 24.897 documentos contendo artigos relacionados à ciência da computação. Foi considerado apenas o primeiro nível de taxonomia adotado pela ACM, onde cada documento é associado a uma das 11 classes;

4. *20 Newsgroups (20NG)*. É uma coleção composta de aproximadamente 18.805 documentos de grupo de notícias particionados (aproximadamente) de forma uniforme entre 20 diferentes categorias;

5. *Spambase (SPAM)*. Uma coleção com diferentes tipos de e-mail. Há um total de 4.601 e-mails classificados como spam ou não-spam;

Os meta-atributos foram avaliados usando duas medidas padrão de classificação de textos: *micro averaged* F1 (MicroF1) que mede a eficácia da classificação sobre todas as decisões, e a *macro averaged* F1 (MacroF1) que mede a eficácia da classificação para cada classe individualmente e obtém a média. Todos os experimentos foram feitos utilizando validação cruzada com 5 partições.

Para a execução dos experimentos e realização da predição dos textos, foi utilizado o algoritmo *SVM*. O tamanho da vizinhança para o algoritmo *KNN* responsável pela geração dos meta-atributos foi definido como sendo $K = 30$ (valor tradicionalmente adotado em classificação de texto) para todos os experimentos.

4 PROPOSTA

Conforme apresentado neste trabalho, métodos de aprendizado de máquina são usados para identificar discurso de ódio em textos. Para tais métodos, utilizar boas características (*features*) ajuda a obter melhores resultados durante a classificação.

Identificar discurso de ódio em documentos com a utilização de aprendizado de máquina requer a aplicação de alguns métodos de PLN (apresentados no Capítulo 3 dos Trabalhos Relacionados). Os trabalhos de (DAVIDSON et al., 2017) e de (NOBATA et al., 2016) (Seção 3.1.2 e Seção 3.1.1, respectivamente) utilizam métodos de pré-processamento de textos para criação de suas características, nos quais são usadas combinações de atributos que ajudam o modelo de predição a ter um melhor desempenho. Já no trabalho de (CANUTO et al., 2013) (descrito na Seção 3.3) é proposta a criação de meta-atributos a partir de dados previamente rotulados, onde o objetivo é capturar informações relevantes sobre uma distribuição de dados desconhecida que relaciona os padrões observados às suas respectivas categorias.

A proposta deste trabalho é extrair novas características a partir de meta-atributos gerados através de informações retiradas da vizinhança de cada documento. Inspirado no trabalho de (CANUTO et al., 2013), os meta-atributos são encontrados através do algoritmo de classificação KNN, descrito na Seção 2.5. A base de dados a ser utilizada é a mesma abordada por (PELLE; MOREIRA, 2017) juntamente com os seus conjuntos para teste, apresentados na Seção 5.1.

4.1 Método Proposto

Esta seção tem por objetivo descrever como funciona o método proposto neste trabalho.

Para aplicação do método, inicialmente é usado o algoritmo de classificação KNN para encontrar a vizinhança mais próxima dos documentos previamente rotulados. A distância usada para determinar a proximidade dos vizinhos ao documento, é definida pela similaridade do cosseno.

Com as informações da vizinhança para cada documento, é feita a criação de novas características a partir das mesmas. A proposta das novas características é capturar informação do conjunto já rotulado de três diferentes formas:

1. Contagem de exemplos rotulados, da mesma maneira que faz o método KNN ao realizar

a classificação;

2. Capturar a relação entre a classe escolhida pelo kNN (a classe com maior número de vizinhos) com as outras classes;

3. Análise da distribuição das distâncias para cada classe.

Na primeira são criadas duas características, que são a contagem do número de exemplos rotulados de cada categoria entre os vizinhos. Por exemplo, se 10 dos vizinhos estão classificados como discurso de ódio e os outros 20 como sendo de outra categoria, as duas novas características seriam com os valores 10 e 20.

A segunda abordagem para a criação dos meta-atributos, consiste na normalização do conjunto de meta-atributos anterior. Para tal, é feita a divisão do número de vizinhos em cada classe pela quantidade de vizinhos da classe com maior número de vizinhos. Seguindo o exemplo anterior, se 10 dos vizinhos estão classificados como discurso de ódio e os outros 20 como sendo de outra categoria, as novas características seriam os valores $10/20$ (0.5) e $20/20$ (1).

Por último, a informação fornecida com os meta-atributos propostos é baseada em uma análise das distâncias para cada classe observada na vizinhança. Para tal, foram escolhidos diferentes pontos que podem caracterizar a informação contida na distribuição das distâncias, totalizando cinco novas características por classe, sendo elas:

- **Menor distância**: Dentre todos os vizinhos, foi escolhido o vizinho mais próximo ao documento;

- **Maior distância**: De todos os vizinhos, foi escolhido o vizinho mais distante ao elemento;

- **Distância média** : De todos os vizinhos, é definida a distância média entre os mesmos.

- **Quartil inferior**: Valor que delimita os 25% das menores distâncias.

- **Quartil superior**: Valor que delimita os 25% das maiores distâncias.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos na experimentação. Serão detalhados os algoritmos que foram utilizados para a classificação dos dados, conforme descritos na proposta deste trabalho, Seção 4. A Seção 5.1 tem como objetivo mostrar a base de dados que foi utilizada para realização dos teste e também a descrição de cada experimento proposto. Já na Seção 5.2 e 5.3 são apresentadas as métricas utilizadas para cada experimento e os seus resultados.

Vale ressaltar que, os experimentos e resultados foram construídos e avaliados com ferramentas diferentes ao trabalho de (PELLE; MOREIRA, 2017). Para o trabalho relacionado, os resultados foram obtidos pela ferramenta *Weka*, neste trabalho, são usados outros recursos descritos na Seção 5.5. Parâmetros obrigatórios de algoritmos foram os mesmos utilizados no *Weka*. É notável uma diferença na quantidade de características e em alguns resultados obtidos, porém, ambas não comprometem o resultado final.

5.1 Base de Dados

A base de dados utilizada para a realização dos experimentos foi a utilizada no trabalho (PELLE; MOREIRA, 2017) (Seção 3.2) e que pode ser obtida por *download* [2].

Como visto anteriormente, a base de dados é composta por duas partes denominadas *OffComBR-2* e *OffComBR-3*. As duas contêm os textos (comentários da web) juntamente com o rótulo de classificação, o qual indica se o texto representa discurso de ódio (classificação positiva) ou não. Na primeira parte, composta por 1.250 comentários, 419 destes são considerados discurso de ódio, representando aproximadamente de 33,5% e cada rótulo foi classificado por pelo menos duas pessoas. Já na segunda, são 1.033 comentários, 202 dos quais são considerados discurso de ódio, o que representa aproximadamente 19,55% dos dados e a classificação foi atribuída por três pessoas. Ambos os dados podem ser visto na Tabela 5.1.

Base de Dados	Total de Comentários	Classificação Positiva
OffComBR-2	1.250	419
OffComBR-3	1.033	202

Tabela 5.1: Estatísticas das Bases de Dados.

[2] https://github.com/rogersdepelle/OffComBR

Para a aplicação do método, foram criados alguns conjuntos de experimentos, os mesmos propostos em (PELLE; MOREIRA, 2017). Para cada experimento foram geradas determinadas características utilizando alguns de PLN. Como é apresentado na Tabela 5.2, nos experimentos com o prefixo *original* foram mantidos os textos com a forma original do comentário. Já nos experimentos com prefixo *lower*, o texto foi transformado em caixa baixa, diminuindo assim a dimensionalidade das características. Alguns experimentos possuem combinações de *N-gram* (*1G*, *2G* e *3G*) e outros possuem as melhores características utilizando o ganho de informação que são apresentados com o sufixo *FS*. A coluna *LIMA* indica a quantidade de características do método proposto para cada experimento, em ambas as bases de dados *OffComBR-2* e *OffComBR-3*. As colunas *BR-2* e *BR-3* mostram o total de características referentes ao trabalho de (PELLE; MOREIRA, 2017) para cada base de dados *OffComBR-2* e *OffComBR-3* respectivamente. Colunas *BR-2 + LIMA* e *BR-3 + LIMA* indicam a quantidade de características da combinação dos experimentos originais com o método LIMA.

Experimento	**BR-2**	**BR-3**	**LIMA**	**BR-2 + LIMA**	**BR-3 + LIMA**
original_1G	4.979	4.347	14	4.993	4.361
original_1G_FS	261	148	14	275	162
original_1G_2G	17.373	15.084	14	17.387	15.098
original_1G_2G_FS	263	146	14	277	160
original_1G_2G_3G	30.710	26.599	14	30.724	26.613
original_1G_2G_3G_FS	260	151	14	274	165
lower_1G	4.122	3.646	14	4.136	3.660
lower_1G_FS	259	144	14	273	158
lower_1G_2G	15.898	13.881	14	15.912	13.895
lower_1G_2G_FS	263	142	14	277	156
lower_1G_2G_3G	29.125	25.302	14	29.139	25.316
lower_1G_2G_3G_FS	268	146	14	282	160

Tabela 5.2: Experimentos e suas características.

5.2 Métricas e Configurações

Para avaliar a eficácia do método proposto foi utilizada a mesma abordagem do trabalho (PELLE; MOREIRA, 2017). A métrica avaliada para os experimentos foi *f-score*, a qual representa a média harmônica entre precisão e revocação, levando sempre em consideração o peso das classes (*f1-weighted*). Para obter uma média mais confiável dos experimentos, foi usada

validação cruzada de dez vezes em cada conjunto de testes e feita a média do *f-score* de todas as execuções.

Os experimentos foram executados com dois algoritmos de classificação, o SVM, com os hiper parâmetros sendo: $kernel = linear$ e $C = 1.0$. NB foi utilizado com os parâmetros originais do classificador. Cada teste foi executado com validação cruzada de dez vezes.

O número de vizinhos utilizados para extração os meta-atributos foi de 30, o qual é a configuração do KNN, $N = 30$.

Para validar as comparações entre os métodos, foi utilizado o teste *Student's T-Test* (t-test) na métrica *f-score*. Esse teste é feito sobre dois conjuntos de dados e o seu resultado é um número, entre 0 e 1, que mede a confiança de uma afirmação. Neste trabalho, as afirmações que passam por validação são os resultados do trabalho (PELLE; MOREIRA, 2017) e da proposta deste trabalho. Caso o resultado do *t-test* for $\alpha > 0,05$, pode-se afirmar que uma proposta foi melhor ou pior que a outra em um determinado aspecto.

5.3 Execução dos Experimentos

Foram conduzidos experimentos para avaliar a eficácia e o poder discriminativo dos meta-atributos descritos anteriormente, bem como dos atributos textuais originais. Esses atributos originais serão referenciados nas tabelas e no texto como *baseline*. O grupo dos meta-atributos, método proposto neste trabalho, serão denominados como *LIMA*.

A seguir serão apresentados os resultados das execuções em ambas as base de dados. As Tabelas 5.3 e 5.4 mostram os resultados das execuções dos algoritmos de classificação SVM e NB e juntamente com o desvio padrão, a indicação de ganho estatístico de cada média representado $\uparrow$, indicação de perda estatística das médias representada por $\downarrow$ e o empate estatístico representado por $\bullet$.

5.3.1 Experimentos com *OffComBr-2*

Na base de dados *OffComBr-2*, após aplicar o método e suas execuções, apresentados na Tabela 5.3, pode-se perceber que em dois casos a média das execuções entre *baseline* e a combinação de *baseline* + *LIMA*, obteve-se um resultado melhor com o classificador SVM, no caso de *lower_1G_FS* teve um ganho de aproximadamente 5,14% e *lower_1G_2G_3G_FS* de 7,7%.

Na execução dos experimentos *LIMA* relação ao *baseline*, o classificador SVM obteve resultados inferiores, no qual, somente quatro experimentos tiveram empate estatístico. Já o algoritmo de NB obteve empate estatístico em dez casos e ganho estatístico em dois, *lower_1G_FS* e *lower_1G_2G_3G_FS*.

	baseline				***LIMA***				***baseline + LIMA***			
Experimento	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**
original_1G	67,12%	0,05	64,20%	0,05	61,59% ↓	0,04	67,00% •	0,04	67,77% •	0,06	64,20% ↓	0,05
original_1G_FS	70,81%	0,06	65,63%	0,03	64,14% ↓	0,05	66,77% •	0,05	72,46% •	0,05	66,14% •	0,04
original_1G_2G	66,47%	0,05	65,81%	0,04	62,09% •	0,05	68,18% •	0,04	66,81% •	0,07	65,81% ↓	0,04
original_1G_2G_FS	70,05%	0,06	64,15%	0,03	63,08% ↓	0,03	64,37% •	0,05	71,23% •	0,05	65,83% •	0,03
original_1G_2G_3G	67,67%	0,06	65,98%	0,04	61,71% ↓	0,05	68,32% •	0,04	66,91% •	0,05	65,98% ↓	0,04
original_1G_2G_3G_FS	70,79%	0,06	66,90%	0,04	62,07% ↓	0,04	64,73% •	0,06	70,82% •	0,05	66,54% •	0,04
lower_1G	71,50%	0,06	65,47%	0,05	66,20% ↓	0,06	67,19% •	0,06	71,43% •	0,05	65,47% ↓	0,05
lower_1G_FS	68,66%	0,06	45,80%	0,08	64,57% ↓	0,05	67,57% ↑	0,05	72,19% ↑	0,05	47,02% •	0,10
lower_1G_2G	70,49%	0,06	67,19%	0,05	67,24% •	0,05	68,53% •	0,06	70,67% •	0,05	67,19% ↓	0,05
lower_1G_2G_FS	69,58%	0,06	63,30%	0,05	65,29% ↓	0,04	65,55% •	0,05	72,46% •	0,04	46,50% ↓	0,09
lower_1G_2G_3G	69,15%	0,06	67,57%	0,05	67,07% •	0,05	69,23% •	0,06	70,99% •	0,05	67,57% ↓	0,05
lower_1G_2G_3G_FS	66,95%	0,05	41,18%	0,11	67,94% •	0,04	67,22% ↑	0,04	72,11% ↑	0,05	43,20% •	0,10

Tabela 5.3: Experimentos com a base de dados *OffComBR-2*.

5.3.2 Experimentos com *OffComBr-3*

Na Tabela 5.4, são apresentados os resultados das execuções com a base de dados *OffComBR-3*. Destaca-se que todos os ganhos são maiores pelo fato de que a classificação dos comentários são mais precisos que a da base de dados *OffComBR-2*.

Os experimentos *original_1G_2G_3G_FS*, *lower_1G_FS*, *lower_1G_2G_FS* e *lower_1G_2G_3G_FS*, tiveram um ganho estatístico com a combinação de *baseline + LIMA* utilizando o classificador SVM de até 5,23%.

Para o classificador NB, com a combinação *baseline + LIMA*, somente o experimento *lower_1G_FS* obteve melhor resultado com um ganho de 3,85%. Resultados somente com o método *LIMA*, tiveram empate estatístico em oito experimentos.

5.4 Discussão sobre os resultados

O objetivo desta seção é apresentar um resumo dos resultados obtidos após experimentos efetuados. Como visto anteriormente, foram realizados doze experimentos para as bases de dados *OffComBR-2* e *OffComBR-3*, totalizando vinte e quatro experimentos.

Analisando os resultados obtidos, os meta-atributos propostos neste trabalho, tiveram

	baseline				*LIMA*				*baseline + LIMA*			
Experimento	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**
original_1G	78,16%	0,03	77,82%	0,07	71,73% ↓	0,00	73,73% •	0,11	78,67% •	0,04	77,82% ↓	0,07
original_1G_FS	80,61%	0,03	81,07%	0,02	78,78% •	0,04	77,80% ↓	0,04	81,42% •	0,04	79,97% •	0,03
original_1G_2G	78,02%	0,03	77,69%	0,05	71,73% ↓	0,00	76,67% •	0,03	77,86% •	0,04	77,69% ↓	0,05
original_1G_2G_FS	79,29%	0,02	81,14%	0,03	79,52% •	0,04	77,22% ↓	0,04	81,71% •	0,04	80,38% •	0,03
original_1G_2G_3G	77,25%	0,03	77,46%	0,05	71,95% ↓	0,01	76,21% •	0,03	77,44% •	0,05	77,46% ↓	0,05
original_1G_2G_3G_FS	80,19%	0,02	78,67%	0,03	80,49% •	0,04	69,69% ↓	0,06	82,63% ↑	0,03	79,04% •	0,03
lower_1G	77,47%	0,02	76,90%	0,07	71,73% ↓	0,00	77,10% •	0,04	77,11% •	0,03	76,90% ↓	0,07
lower_1G_FS	78,86%	0,03	78,56%	0,05	81,46% ↑	0,04	70,09% ↓	0,05	81,90% ↑	0,04	80,72% ↑	0,04
lower_1G_2G	77,62%	0,04	76,69%	0,04	71,73% ↓	0,00	77,26% •	0,03	78,12% •	0,04	76,69% •	0,04
lower_1G_2G_FS	79,91%	0,03	78,96%	0,05	78,16% •	0,04	74,17% •	0,07	82,30% ↑	0,04	77,59% ↓	0,05
lower_1G_2G_3G	77,23%	0,02	76,70%	0,04	72,12% ↓	0,01	76,17% •	0,04	77,91% •	0,04	76,70% ↓	0,04
lower_1G_2G_3G_FS	80,36%	0,02	77,53%	0,03	82,24% •	0,04	73,10% •	0,05	84,57% ↑	0,03	78,51% •	0,05

Tabela 5.4: Experimentos com a base de dados *OffComBR-3*.

um melhor desempenho quando somados com as características do *baseline*. O classificador com melhor desempenho foi o SVM em quase todos os ganhos estatísticos. Com o método LIMA, em alguns casos, apresentou um resultado melhor em até 3,3%. Para o classificador NB na base de dados *OffComBR-2* e *OffComBR-3*, boa parte dos resultados estivem abaixo do *baseline*.

Os experimentos que obtiveram ganhos estatísticos foram os que utilizaram redução de atributos, na qual, as características mais relevantes são selecionadas. Foi possível perceber que os meta-atributos propostos sempre estiveram presentes nos experimentos com o método de redução de atributos. A Tabela 5.5 apresenta os resultados das duas bases de dados, somente com os experimentos que possuíram características relevantes para realizar a classificação. Levando em consideração o classificador SVM, quase todos os resultados de *baseline + LIMA* obtiveram uma média melhor que ao *baseline*.

	OffComBR-2				*OffComBR-3*			
	baseline		**baseline + LIMA**		**baseline**		**baseline + LIMA**	
Experimento	**SVM**	**NB**	**SVM**	**NB**	**SVM**	**NB**	**SVM**	**NB**
original_1G_FS	70,81%	65,63%	72,46% •	66,14% •	80,61%	81,07%	81,42% •	79,97% •
original_1G_2G_FS	70,05%	64,15%	71,23% •	65,83% •	79,29%	81,14%	81,71% •	80,38% •
original_1G_2G_3G_FS	70,79%	66,90%	70,82% •	66,54% •	80,19%	78,67%	82,63% ↑	79,04% •
lower_1G_FS	68,66%	45,80%	72,19% ↑	47,02% •	78,86%	78,56%	81,90% ↑	80,72% ↑
lower_1G_2G_FS	69,58%	63,30%	72,46% •	46,50% ↓	79,91%	78,96%	82,30% ↑	77,59% ↓
lower_1G_2G_3G_FS	66,95%	41,18%	72,11% ↑	43,20% •	80,36%	77,53%	84,57% ↑	78,51% •

Tabela 5.5: Experimentos com redução de atributos e seus resultados.

A partir desta análise, pode-se concluir que os meta-atributos combinados com outras características, obtiveram um bom resultado para classificação dos textos com o objetivo de

identificar o discurso de ódio.

5.5 Código-Fonte

Esta seção apresenta as ferramentas e bibliotecas que foram utilizadas no desenvolvimento do código fonte. Todas as etapas do processo foram codificadas utilizando a linguagem de programação Python. Para trabalhar com os algoritmos de classificação foi utilizado a biblioteca **scikit-learn**[3]. Esta, possui uma grande quantidade de ferramentas para algoritmos de aprendizado de máquina, além de possuir todas as ferramentas necessárias para fazer a leitura dos arquivos da base de dados. Para realizar o pré-processamento do texto foi utilizado a biblioteca **nltk**[4] (*Natural Language Toolkit*).

Todos os códigos das ferramentas utilizadas para os testes estão disponível publicamente.[5]

[3] https://scikit-learn.org/stable/
[4] https://www.nltk.org/
[5] https://gitlab.com/cleiton-limapin/tcc-final

6 CONCLUSÃO

Esse trabalho de conclusão de curso teve como objetivo explorar e propor novas características para a classificação de texto, com o intuito de identificar o discurso de ódio em documentos. Para tal, foram usados métodos de processamento de linguagem natural e aprendizagem de máquina.

Foi utilizada como fundamentação o método proposto por (CANUTO et al., 2013), que cria meta-atributos a partir da extração de informações sobre a similaridade/vizinhança de cada documento. Tais características foram analisadas de forma isolada e em conjunto com outras características de trabalhos relacionados.

Utilizando a base de dados proposta por (PELLE; MOREIRA, 2017), experimentos foram realizados com diferentes combinações para analisar o uso dos meta-atributos em diferentes cenários. O método proposto obteve bons resultados em alguns casos. Os meta-atributos, combinados com características propostas por (PELLE; MOREIRA, 2017), obtiveram ganhos estatísticos de até 5,24% em comparação com as características originais.

Utilizando o classificador SVM, os meta-atributos, analisados separadamente, obtiveram resultados próximos ao original, mostrando que as novas características são promissoras para melhorar a qualidade da classificação.

6.1 Trabalhos Futuros

Uma forma de complementar esse trabalho é explorar métodos de analise de sentimento, área onde houve bons resultados em trabalhos relacionados, e realizar a combinação com os meta-atributos.

No trabalho de (CANUTO et al., 2013), são citados alguns trabalhos relacionados que propõem outros meta-atributos, levando em consideração outros tipos de informação. Para um trabalho futuro é interessante aplicar os mesmos métodos para identificação de discurso de ódio em textos.

Por fim, outro trabalho importante seria realizar a construção de um modelo de predição com o mesmo conjunto de dados e método proposto neste trabalho, porém explorando outros métodos de classificação e os mesmos com outras configurações, com o objetivo de aprimorar os resultados.

REFERÊNCIAS

BATISTA, G. E. d. A. P. et al. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.

CANUTO, S. et al. Um Estudo sobre Meta-Atributos para Classificaçao Automática de Texto. , [S.l.], 2013.

DAVIDSON, T. et al. Automated hate speech detection and the problem of offensive language. **arXiv preprint arXiv:1703.04009**, [S.l.], 2017.

GOLDBERG, Y. Neural network methods for natural language processing. **Synthesis Lectures on Human Language Technologies**, [S.l.], v.10, n.1, p.1–309, 2017.

JIVANI, A. G. et al. A comparative study of stemming algorithms. **Int. J. Comp. Tech. Appl**, [S.l.], v.2, n.6, p.1930–1938, 2011.

KWOK, I.; WANG, Y. Locate the Hate: detecting tweets against blacks. In: AAAI. **Anais...** [S.l.: s.n.], 2013.

MARTINS, C. A.; MONARD, M. C.; MATSUBARA, E. T. Uma metodologia para auxiliar na seleçao de atributos relevantes usados por algoritmos de aprendizado no processo de classificaçao de textos. In: XXIX CONFERENCIA LATINOAMERICANA DE INFORMATICA-CLEI, LA PAZ, BOLIVIA.(TO BE PUBLISHED). **Anais...** [S.l.: s.n.], 2003. v.38.

NAKAMURA, F. G. et al. Uma Abordagem para Identificar e Monitorar Haters em Redes Sociais Online. , [S.l.], 2017.

NOBATA, C. et al. Abusive language detection in online user content. In: OF THE 25TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB. **Proceedings...** [S.l.: s.n.], 2016. p.145–153.

NOCKLEBY, J. T. Hate speech. **Encyclopedia of the American constitution**, [S.l.], v.3, p.1277–79, 2000.

PELLE, R. P. de; MOREIRA, V. P. Offensive Comments in the Brazilian Web: a dataset and baseline results. In: BRAZILIAN WORKSHOP ON SOCIAL NETWORK ANALYSIS AND MINING (BRASNAM), 6. **Anais...** [S.l.: s.n.], 2017. to appear.

PORTER, M. F. **Snowball**: a language for stemming algorithms. 2001.

RESEARCHGATE. Exemplo de N-gram. Acessado em junho 2018, https://www.researchgate.net/figure/N-gram-text-representation_fig4_256290162.

SANTOS, F. C. et al. **Variações do método kNN e suas aplicações na classificação automática de textos**. 2009. Tese (Doutorado em Ciência da Computação) — Dissertação de Mestrado, Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, Universidade Federal de Goiás, Goiânia, Brasil.

SCHMIDT, A.; WIEGAND, M. A survey on hate speech detection using natural language processing. In: FIFTH INTERNATIONAL WORKSHOP ON NATURAL LANGUAGE PROCESSING FOR SOCIAL MEDIA. **Proceedings...** [S.l.: s.n.], 2017. p.1–10.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM computing surveys (CSUR)**, [S.l.], v.34, n.1, p.1–47, 2002.