Detecção de Haters nas Redes Sociais

Álisson Rodrigues Teles[1], **Carine Geltrudes Webber** [1]

[1]Área do Conhecimento de Ciências Exatas e Engenharias – Universidade de Caxias do Sul (UCS)
Rua Francisco Getulio Vargas, 1130 – CEP 95070-560 Caxias do Sul – RS – Brasil

{arteles1,cgwebber}@ucs.br

Abstract. *This paper presents a study about Sentiment Analysis, more specifically, about hate speech on Portuguese data texts. To achieve this goal, all necessary stages of sentiment analysis are shown and an experiment with the main algorithms of classifiers was realized. Last on, the paper introduces a demonstration of a tool developed using Twitter's database, Java language and the API of WEKA.*

Resumo. *Esse artigo objetiva apresentar o estudo realizado sobre a Análise de Sentimentos na detecção, mais especificamente, de discursos de ódio em dados textuais na língua portuguesa. Para alcançar o objetivo, são apresentadas todas as etapas necessárias para a análise de sentimentos, são realizados experimentos nos algoritmos mais utilizados para a classificação dos dados e, ao final, é apresentada uma demonstração de ferramenta implementada utilizando base de dados coletada na rede social Twitter, linguagem de programação Java e a API do WEKA.*

1. Introdução

Com a globalização do uso da *internet*, a sociedade aprimorou a sua forma de interagir e desenvolveu novas ferramentas para as relações interpessoais, profissionais e etc. Essas ferramentas são chamadas de redes sociais e são exemplos o *Facebook*[1], o *Twitter*[2] e o *Instagram*[3], entre outras no mercado. Essas ferramentas estão presentes na maioria dos *smartphones* da população, e a cada dia aumenta o número de usuários ativos em suas plataformas.

Segundo pesquisa realizada pelo IBGE, $64,7\%$ da população brasileira, de 10 anos ou mais, utilizou a *internet* e $94,6\%$ das conexões foram realizadas por *smartphones*. Ainda na mesma pesquisa foi verificado que $94,2\%$ da finalidade dessas conexões foi enviar ou receber mensagens de texto, áudio ou imagens por aplicativos diferentes de *e-mail*, o que reforça o crescimento das plataformas de rede social [IBGE 2017].

Com o aumento nas iterações sociais em plataformas digitais um fenômeno desagradável se torna cada vez mais comum, os discursos de ódio, que segundo Chetty e Alathur são consequência da liberdade de expressão e uma tendência para que as pessoas atinjam a popularidade instantânea sem a necessidade de maiores esforços. O discurso de ódio acaba sendo considerado um crime, pois viola os direitos fundamentais dos cidadãos e se opõe à liberdade de expressão, já que é realizado por grupos de ódio que não

[1]https://www.facebook.com
[2]https://twitter.com
[3]https://www.instagram.com

concordam com determinados comportamentos de minorias, tirando-lhes muitos direitos e causando danos, muitas vezes irreparáveis [Chetty and Alathur 2018].

A área de detecção de discurso de ódio estuda ferramentas que auxiliam na automação da busca de discursos realizados por pessoas mal intencionadas, popularmente chamadas de *haters*. Essa área se baseia nas técnicas da Análise de Sentimentos para realizar detecções automáticas e instantâneas desse comportamento. A Análise de Sentimentos tem por objetivo a classificação de um dado textual baseado no seu teor sentimental, de opinião, de avaliação e de atitudes sendo útil na avaliação de determinadas entidades que podem ser eventos, produtos, problemas, entre outros [Bahri et al. 2018].

Esse artigo objetiva abordar as técnicas mais eficientes de classificação textual baseadas em aprendizado de máquina conforme referencial teórico sobre a Análise de Sentimentos, apresentar metodologia para a construção de um dicionário de dados na Língua Portuguesa utilizando a rede social *Twitter* e as técnicas levantadas para a detecção de discursos de ódio, e, após, apresentar ferramenta implementada que será disponibilizada para apoiar em pesquisas científicas e na implementação de *softwares* que tenham o objetivo de detectar discursos de ódio na Língua Portuguesa.

2. Trabalhos Correlatos

Em sua pesquisa, Correa (2017) abordou os conceitos da Análise de Sentimentos aplicados à rede social *Twitter*, analisando as reações dos usuários em relação aos filmes indicados à categoria de "Melhor Filme"do *Oscar 2017*, já que as redes sociais são espaços que possibilitam a divulgação de filmes e também são utilizadas como fonte de opiniões. Diferentes abordagens de classificação foram apresentadas pelo autor. Desde algoritmos de aprendizado supervisionado (*Naive Bayes*) até o aprendizado por supervisão à distância (utilizando a ferramenta *Sentiment140*[4]). Para o desenvolvimento foi utilizado o algoritmo *Naive Bayes* por ser o mais utilizado em casos de bases rotuladas, aplicando a base rotulada no *software Weka* [Correa 2017].

No trabalho proposto por Kwok e Wang (2013) tem-se um estudo sobre o discurso de ódio contra os afro-americanos usuários do *Twitter*. Esses discursos de ódio são especialmente penais dentro da rede social em questão, embora esse efeito não seja percebido em meio aos dados gerados diariamente dentro da plataforma. Para realizar a classificação dos dados, previamente rotulados, entre dados racistas e dados não racistas, foi utilizado o algoritmo *Naive Bayes* já que o mesmo, segundo os autores, funciona tão bem quanto os algoritmos mais sofisticados [Kwok and Wang 2013].

Em seu trabalho, Pelle e Moreira (2017) constataram que as redes sociais são ferramentas que trazem a liberdade para o usuário poder apresentar suas ideias e compartilhar informações. Entretanto, percebem que essa mesma ferramenta está cheia de usuários que promovem o discurso de ódio. A análise desses eventos é inviável de forma manual uma vez que a cada dia muito mais textos são gerados. Sendo assim, os autores verificaram a necessidade de uma ferramenta que realizasse essa análise de forma automática e realizaram o desenvolvimento da mesma para a língua portuguesa, já que atualmente existem poucas abordagens específicas para o idioma. Para realizar o experimento, os autores desenvolveram uma base de dados rotulada específica

[4]http://www.sentiment140.com/

para a língua portuguesa, buscando por discursos de ódio dentro do site *g1.globo.com*[5] (`https://g1.globo.com`) e classificando os comentários entre discursos negativos e discursos positivos [de Pelle and Moreira 2017].

Diante dos trabalhos analisados, verificou-se a importância e a necessidade de um mecanismo de detecção de discursos de ódio que torne possível a detecção automática e específica para a Língua Portuguesa. Sendo possível analisar textos automaticamente e em grande escala sem a necessidade de um especialista humano, agilizando a detecção do discurso de ódio. Com os mesmos trabalhos, foi possível fazer um levantamento inicial das técnicas utilizadas e das abordagens necessárias para desenvolver a ferramenta esperada como objetivo dessa pesquisa.

3. Referencial Teórico

A Análise de Sentimentos tem por objetivo a classificação de um dado textual baseado no seu teor sentimental, de opinião, de avaliação e de atitudes, sendo útil na avaliação de determinadas entidades, que podem ser eventos, produtos, problemas, entre outros [Bahri et al. 2018].

Tendo em vista o crescimento do número de usuários nas redes sociais, consequentemente o número de produções textuais nesses canais aumentou. Em razão disso, o interesse de grandes empresas em obter pesquisas de opinião à partir de publicações em redes sociais ficou evidente. A Análise de Sentimentos, quando realizada de forma manual por especialistas é dificultada, sendo necessário o uso de recursos computacionais unidos a algoritmos sofisticados realizando análises em grande escala (*big data*) e em menor tempo [Bahri et al. 2018].

O processo de Análise de Sentimentos compreende diversas etapas para que, de forma automática, dados textuais sejam avaliados em termos de sentimentos expressos neles. O processo é composto pelas etapas que são descritas nas seções seguintes: coleta de dados, pré-processamento dos dados, classificação dos dados e análise de resultados.

3.1. Coleta de Dados

A primeira fase da Análise de Sentimentos é a coleta de dados que, posteriormente, farão parte da base de dados que será avaliada pelos algoritmos de classificação de texto. Segundo Amaral, "Dados são fatos coletados e normalmente armazenados. Informação é o dado analisado e com algum significado. O conhecimento é a informação interpretada, entendida e aplicada para um fim.". O trabalho considerou dados eletrônicos digitais, mais precisamente, dados não estruturados (postagens textuais) gerados em redes sociais de forma manual. A informação principal considerada foi a análise léxica baseada em categorias de emoções. Já o conhecimento pretendido foi a detecção do discurso de ódio nos textos coletados [Amaral 2016].

Para a fase de coleta de dados é importante primeiramente definir qual será a fonte de dados e, após, desenvolver ferramentas automáticas para a coleta que se pretende analisar futuramente. Algumas redes sociais disponibilizam APIs (*Application Programming Interface*) para que aplicações sejam desenvolvidas usando o banco de dados dessas plataformas, permitindo tanto o consumo como a geração de dados dentro delas por meio de

[5]Classificado como o site mais acessado no Brasil em 2017, pelo site `http://www.alexa.com/topsites/countries/BR`

postagens de fotos, vídeos, textos e etc. Essas ferramentas conseguem realizar buscas com filtros de data, usuários, palavras-chave, e etc, sendo possível buscar amostras específicas conforme o que se queira pesquisar.

Geralmente as APIs são *web services* (produtos disponibilizados na *internet*) baseados no protocolo HTTP (*Hypertext Transfer Protocol*) e tornam os acessos seguros por meio da geração de *tokens* a cada nova solicitação de inserção, alteração ou exclusão de dados. Um exemplo é a *Graph API*[6] desenvolvida pelo *Facebook*, ela disponibiliza uma interface gráfica chamada *Graph API Explorer*, a ferramenta é gratuita e tem ampla documentação de apoio para o desenvolvimento de aplicações que necessitem de integração com a plataforma.

3.2. Pré-Processamento dos Dados

O pré-processamento realiza a separação das frases em listas de palavras, chamadas de *tokens*, que devem ser processadas até transformar o conteúdo inicial em sua forma reduzida e otimizada [Symeonidis et al. 2018]. É importante considerar que as técnicas de pré-processamento são utilizadas conforme a fonte de dados e também conforme os termos de interesse que se busca, assim como o idioma dos textos que se quer analisar.

As principais técnicas de pré-processamento textual, conforme [Symeonidis et al. 2018]; [Khan et al. 2014] são as seguintes:

- Remoção de *unicode* e ruídos [Symeonidis et al. 2018].
- Substituição de *URL* e menções de usuários usando @ [Khan et al. 2014].
- Substituição de abreviaturas e gírias [Kouloumpis et al. 2011].
- Remoção de conteúdo numérico [He et al. 2011].
- Substituição de pontuação repetida [Balahur 2013].
- Remoção de pontuações [Lin and He 2009].
- Manipulação de palavras em caixa alta [Zhang et al. 2015].
- Padronização de letras minúsculas [dos Santos and Gatti 2014].
- Remoção de *stopwords* [Ferilli et al. 2014].
- Substituição de palavras alongadas [Mohammad et al. 2015].
- Correção ortográfica [Miller 1995].
- *Part-of-Speech* (POS) [Barbosa and Feng 2010].
- Lematização [Guzman and Maalej 2014].
- *Stemming* [Porter 1980].

Após o pré-processamento, os dados estarão mais consistentes e padronizados, contendo as informações mais relevantes para a geração de uma base de treinamento. Existem algumas tratativas específicas para *emojis* e símbolos que também demonstram sentimento. Os mesmos não foram abordados no trabalho por serem de baixa relevância na pesquisa.

3.3. Classificação dos Dados

Existem vários estudos que buscam encontrar os métodos mais eficientes para a etapa de classificação dos dados dentro da Análise de Sentimentos. Tradicionalmente os algoritmos de classificação de Análise de Sentimentos atribuem um sentimento como ”positivo”, ”negativo”e ”neutro”ao texto coletado, existindo algumas abordagens baseadas em

[6] `https://developers.facebook.com/docs/graph-api`

análise de emoção que consideram categorias mais amplas baseadas no estudo de Ekman que podem ser: alegria, tristeza, raiva, medo, nojo e surpresa [Ekman 1992].

Com base nos estudos realizados por Kwok e Wang, e Pelle e Moreira, os principais algoritmos utilizados na etapa de classificação da Análise de Sentimentos são os baseados no Teorema de Bayes (Bayes Net, Naive Bayes, Naive Bayes Multinomial Text, Naive Bayes Updateable) e o algoritmo baseado em máquina de vetor de suporte (SVM) [Kwok and Wang 2013], [de Pelle and Moreira 2017]. Abaixo serão apresentados cada um dos classificadores mencionados.

3.3.1. Algoritmos Bayesianos

O principal algoritmo supervisionado utilizado dentro da classificação de sentimentos é o *Naive Bayes* (NB) por unir simplicidade e eficiência nos cenários mais diversos, especialmente quando aplicado a um conjunto de dados onde seus atributos são independentes [Viegas et al. 2018]. O Naive Bayes é um algoritmo probabilístico baseado no Teorema de Bayes representado na Equação 1. Ela é definida como "a probabilidade de A dado B", ou seja, qual a probabilidade de A estar dentro de um conjunto de evidências assumido como B [Schmitt 2013].

$$P\left(A|B\right) = \frac{P\left(B|A\right)P\left(A\right)}{P\left(B\right)} \tag{1}$$

O algoritmo é considerado ingênuo (*naive*) por assumir que cada atributo é independente, ou seja, a informação de um evento não é informativa sobre nenhum outro. A Figura 1 indica que cada atributo A_i influência a classe C unicamente sem nenhum atributo influenciar qualquer outro [Schmitt 2013].

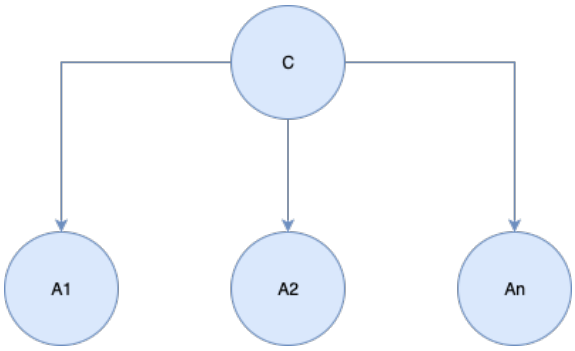

Figura 1. Modelo *Naive Bayes*

O Naive Bayes pode ser usado de forma incremental conforme a necessidade de sua revisão probabilística. Existem várias abordagens sobre os modelos de entrada sendo os principais o modelo binário e o modelo multidimensional. No modelo binário, cada documento é representado por um vetor binário onde a presença, ou a ausência, de um termo é representado, respectivamente, por 1 e 0. Já no modelo multidimensional a representação ocorre por vetores de números inteiros por documento, onde cada índice do vetor representa a quantidade de vezes que um termo acontece no documento [Schmitt 2013].

Diferentemente do Naive Bayes, o algoritmo *Bayes Net* (também conhecido como Rede Bayesiana) trata cada atributo como dependente dos demais atributos para atingir a classificação esperada. O Bayes Net pode ser representado como um grafo acíclico dirigido (Figura 2), onde cada vértice representa uma variável do conjunto de domínio e cada aresta representa as correlações diretas entre as variáveis. Essa representação apresenta a distribuição de probabilidades (que nesse caso são conjuntas e não distintas como na abordagem do Naive Bayes) para o seu domínio, como também auxilia na interpretação humana para o desenvolvimento e manutenção de sistemas que incorporam o modelo probabilístico [Heckerman et al. 1995, Howard and Matheson 2005].

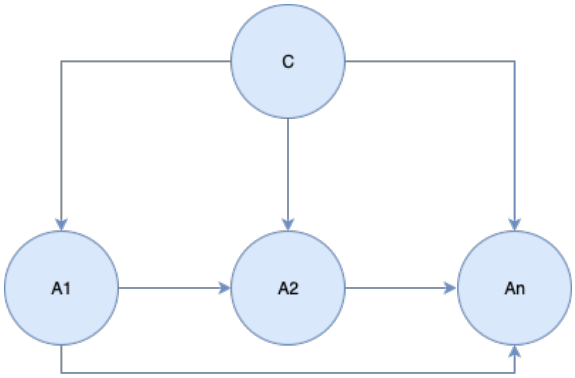

Figura 2. Modelo *Bayes Net*

As redes bayesianas fornecem uma representação otimizada para o aprendizado de máquina e isso faz delas uma ótima opção para cenários de grande incerteza e com muitos ruídos. Por esse motivo são algoritmos extremamente úteis em tarefas de desempenho e de diagnóstico [John and Langley 1995].

Ainda existem outras abordagens que fazem uso do Teorema de Bayes, essas abordagens são específicas para determinados objetivos de pesquisa. Dentre elas estão as específicas para dados textuais (Naive Bayes Multinomial Text), e ainda, as que ajustam o modelo de dados a cada nova alteração (Naive Bayes Updateable). Em seu estudo, McCallum e Nigam apresentam esses recursos na classificação textual e discutem sobre a influência do tamanho dos textos sobre a performance dos classificadores [McCallum and Nigam 1998].

3.4. Máquinas de Vetor de Suporte

As Máquinas de Vetor de Suporte (*Support Vector Machines*) foram desenvolvidas por Vapnik e são amplamente utilizadas para a classificação de textos, imagens, reconhecimento de textos manuscritos e nas ciência em geral [Vapnik 1995]. A técnica utiliza a sua base de treinamento para construir um hiperplano e à partir dele inicia o processo de classificação da base de testes.

Na técnica SVM cada atributo é uma característica do hiperplano. Um conjunto de características dá origem a um vetor de características que, se próximos dos hiperplanos construídos para representar cada classe de domínio, formarão os chamados vetores de suporte [Vapnik 1995].

Mesmo sendo amplamente utilizada no aprendizado de máquina, a técnica de

SVM não se destacou das demais técnicas. Segundo Platt isso se deu por dois motivos principais. O primeiro é que o treinamento das SVMs é um processo muito lento. E o segundo é que os algoritmos de treinamento das SVMs são altamente complexos, sutis e dificultam a implementação [Platt 1998].

Como solução para essas questões Platt desenvolveu o SMO, abreviação de *Sequential Minimal Optimization*. Diferentemente da SVM, que realiza a classificação necessitando de uma solução de problema de programação quadrática, a SMO fragmenta a solução de forma analítica a cada iteração, o que torna a técnica muito mais ágil e com consumo de memória de forma linear [Platt 1998].

3.5. Análise dos Resultados e Métricas

A última etapa da Análise de Sentimentos é a etapa de coleta e avaliação dos seus resultados. A ferramenta mais comum para a análise dos resultados é a matriz de confusão [Provost and Kohavi 1998]. A matriz de confusão contém as informações sobre o estado atual e o estado preditivo dos resultados de uma única classificação. A seguir, é possível verificar a estrutura básica de uma matriz de confusão onde só é possível duas classificações (Verdadeiro ou Falso) e, em que as colunas significam os dados preditos e as linhas significam o estado atual das instâncias que compõem a matriz de confusão:

Tabela 1. Modelo de matriz de confusão

	Verdadeiro	Falso
Verdadeiro	a	b
Falso	c	d

Os termos significam:

a : número de predições corretas onde a instância é verdadeira.
b : número de predições incorretas onde a instância é verdadeira.
c : número de predições incorretas onde a instância é falsa.
d : número de predições corretas onde a instância é falsa.

Tendo essas informações de um método de classificação, é possível extrair outras informações importantes para a verificação da efetividade do método classificador. Abaixo estão alguns dos principais propriedades que se pode extrair de uma matriz de confusão [Provost and Kohavi 1998]:

- *Acurácia*: Corresponde a proporção entre os pontos segmentados corretamente com a soma de todos os pontos segmentados analisados. Sua fórmula é dada por:
  $Acuracia = \frac{VP+VN}{VP+FP+FN+VN}$
- *Revocação*: Corresponde a proporção entre o número de vezes que uma classe foi predita corretamente e o número de vezes que a classe aparece nos dados analisados. Sua fórmula é dada por:
  $Revocacao = \frac{VP}{VP+FN}$
- *Precisão*: Corresponde a proporção entre o número de vezes que uma classe foi predita corretamente e o número de vezes que a classe foi predita. Pode se afirmar que a precisão e a capacidade do método estudado em evitar falsos positivos (FP). Sua fórmula pode ser vista abaixo:
  $Precisao = \frac{VP}{VP+FP}$

- *Medida F*: É a média harmônica entre a precisão e a revocação. Também pode se interpretar como a média de confiabilidade da acurácia. Se a Medida-F for alta, significa que a acurácia obtida é relevante. Sua fórmula é dada por:
  $MedidaF = 2 * \frac{Revocacao * Precisao}{Revocacao + Precisao}$

Onde VP é o número de *Verdadeiros Positivos* (a), TN o número de *Verdadeiros Negativos* (d), FP o número de *Falsos Positivos* (b) e FN os *Falsos Negativos* (c).

Outra possível abordagem é a de análise ROC (*Receiver Operating Characteristic*) que foi desenvolvida entre 1950 e 1960 para avaliar a detecção de sinais em radares e sinais na psicologia sensorial [Zou et al. 2011]. Essa abordagem se baseia em duas quantidades da matriz de confusão (ou ainda, tabela de contingência em abordagens médicas), as quantidades de preditivos positivos ($a/(a+b)$) e as quantidades dos preditivos negativos ($d/(d+c)$).

O sistema de coordenadas ROC apresenta como ordenadas os acertos e como abcissas os erros. Ao se projetar as probabilidades, os valores são apresentados linearmente variando entre zero e um, sendo que todas as ROC possíveis estarão limitadas por um quadrado unitário [Zou et al. 2011].

Para uma melhor análise também se utiliza a curva ROC que é definida pelos pares das ordenadas e $1 - abcissas$, resultantes da variação do valor de corte. A curva ROC pode ser considerada como uma representação da capacidade de distinguir dois estados (verdadeiro, falso) no universo especificado dentro de um experimento [Zou et al. 2011].

4. Materiais e Métodos

Com base nos trabalhos correlatos analisados (Seção 2) e, após realizar pesquisa sobre os principais métodos que compõem a Análise de Sentimento, foi possível verificar a necessidade de uma ferramenta que realizasse a mesma análise, especificamente para casos de discurso de ódio para a Língua Portuguesa.

Com esse intuito foi desenvolvida uma *engine* que serve de apoio para projetos de *software* que necessitam da Análise de Sentimentos de ódio expressos em texto. Abaixo são apresentadas as etapas realizadas para a montagem de base rotulada utilizada para a formação da base de treinamento, bem como os métodos de classificação utilizados para o desenvolvimento da aplicação.

4.1. Coleta dos Dados

Como primeira etapa da geração do dicionário de léxicos foi necessário coletar os dados textuais que formaram a base de treinamento dos algoritmos de classificação. Nessa pesquisa foi desenvolvida uma ferramenta de coleta utilizando a API do *Twitter* chamada de *Twitter4J*[7]. Na Figura 3 é possível verificar fluxo de coleta dos dados fazendo uso da ferramenta implementada.

Após a construção do programa de coleta de *tweets*, o próximo passo foi encontrar filtros de consulta que auxiliaram na busca por conteúdos úteis para a geração da base de treinamento. O primeiro critério de seleção foi a busca por perfis de influenciadores digitais (comumente chamados pela versão inglesa *digital influencers*), que são

[7] `http://twitter4j.org/en/index.html`

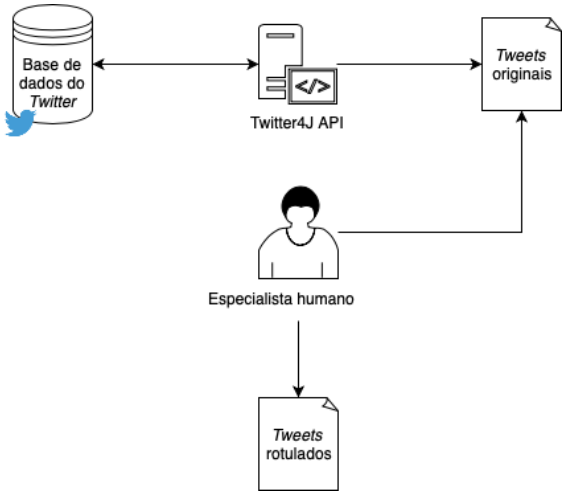

Figura 3. Etapa de coleta dos dados

pessoas que tem grande influência na mídia, tendo vários seguidores e também vários *haters* comentando suas postagens. Os demais filtros foram desenvolvidos arbitrariamente com palavras-chave de assuntos cotidianos do ano de 2018 e 2019 como: política, humor, música, e etc. Na Tabela 2 é possível verificar as quantidades, bem como cada classificação realizada pelo especialista humano na etapa de coleta. O critério para escolha da quantidade de textos coletados foi de escolha do autor e o equilíbrio entre a quantidade de textos neutros e de ódio foi utilizado para que os algoritmos de aprendizado tivessem equilíbrio no aprendizado. Não havendo tendencia a aprender uma única classe entre as duas verificadas.

Tabela 2. Classificação dos *tweets* na etapa e coleta

Classificação	Quantidade de tweets
Tweets Neutros	500 Tweets
Tweets de Ódio	500 Tweets

4.2. Pré-Processamento dos Dados

Após a coleta dos *tweets*, foi necessário realizar a etapa de pré-processamento, afim de diminuir os ruídos e tornar os dados mais determinantes do seu teor neutro ou de ódio. As regras que compunham a etapa de pré-processamento foram: remoção de *unicode* e ruídos, remoção de conteúdo numérico, substituição de pontuação repetida, remoção de pontuações, substituição de palavras alongadas e remoção de *stopwords*. Por convenção todas as palavas restantes foram deixadas em caixa alta (letras maiúsculas).

Para tornar os textos possíveis de interpretação primeiramente foi necessário gerar um arquivo para cada *tweet*, e separar os mesmos em pastas correspondentes a cada uma das possíveis classes. Após esse processo foi realizada a transformação em um vetor de palavras (*weka.filters.unsupervised.attribute.StringToWordVector*) dentro do *WEKA* e depois foi realizado filtro que reduziu o vetor para as palavras mais utilizadas (*weka.filters.supervised.instance.SpreadSubsample*). No final desse processo foi possível gerar arquivo com relação dos atributos a serem utilizados pelos algoritmos de *machine learning* (*Attribute-Relation File Format*) como base de treinamento (Figura 4).

Figura 4. Processo de transformação de frases em vetor de palavras

4.3. Classificação dos Dados

Para realizar essa etapa foi utilizado o *software WEKA*[8] na sua versão 3.8.3. A ferramenta conta com um amplo portfólio de algoritmos de aprendizado de máquina, recebendo atualizações constantes e tendo documentação aberta e amplamente divulgada.

4.3.1. Base de Treinamento

A base de treinamento utilizada foi o resultado das etapas de coleta, pré-processamento e transformação dos dados previamente rotulados por um especialista humano que gerou um arquivo com extensão ".arff"(*Attribute-Relation File Format*). Para realizar os testes de performance dos algoritmos de classificação foi utilizado o mesmo arquivo de treinamento (opção *Use training set* da ferramenta utilizada).

Os mesmos dados foram aplicados a todos os algoritmos. Sendo assim, a base de treinamento é a representação vetorial numérica da base de léxicos mais utilizados tanto em discursos de ódio como neutros da base coletada manualmente e, é baseado nesse resultado que os classificadores foram testados e selecionados para compor a ferramenta implementada.

4.3.2. Algoritmos de Classificação

A Análise de Sentimentos é, tipicamente, um problema de classificação textual (problemas onde se busca saber a classe do objeto avaliado levando em consideração uma base preditiva, aqui chamado de base rotulada, gerada à partir de coleta de dados e classificação manual. Para escolher os algoritmos de aprendizado de máquina (aqui chamados de classificadores) foram utilizados como critérios os trabalhos correlatos (Sessão 2), seus resultados e também análise experimental sobre os algoritmos.

Para realizar a analise experimental foi utilizado o recurso *Experimenter* do *WEKA*. É um ambiente gráfico no qual é possível realizar análises sobre uma lista prédefinida de base de treinamento e de algoritmos classificadores. Como base de trei-

[8] `https://www.cs.waikato.ac.nz/ml/weka/`

namento. Foi utilizado o arquivo resultante das etapas de coleta, pré-processamento e transformação dos dados, foi utilizado o sistema operacional MacOS 10.14.4, Intel Core i5 1,4 GHz e validação cruzada com 10 partições.

Os algoritmos de classificação selecionados foram os seguintes:

- *Bayes* – BayesNet, Naive Bayes, Naive Bayes Multinomial Text, Naive Bayes Multinomial Updatable.
- *Functions* – SMO (Sequential Minimal Optimization).

Para realizar os testes, foram selecionados os parâmetros padrão de cada um dos algoritmos. E a mesma base de treinamento foi utilizada.

4.3.3. Método Ensemble

Após verificar os classificadores com melhor desempenho é possível fazer uso de algoritmos que combinam outros modelos na busca de uma otimização dos resultados gerados pelos mesmos. A esses algoritmos damos o nome de *Ensemble*. Esses métodos tem a capacidade de unir o melhor resultado de vários modelos de classificação na busca da geração de um resultado mais efetivo do que o que se conseguiria apenas com um modelo de classificação. [Witten and Frank 2005]

Os algoritmos *ensemble* se classificam entre: *bagging*, *boosting* e *stacking*. É possível verificar descrição introdutória de cada um dos métodos na obra de Witten e Frank (*Data Mining: Practical Machine Learning Tools and Techniques*).

Para o desenvolvimento da ferramenta proposta, foi selecionado o algoritmo de votação (*vote* dentro da plataforma *WEKA*). Esse algoritmo é classificado como *Bagging*. Ele analisa as varias saídas de cada classificador utilizado e após verificar o melhor resultado (um exemplo pode ser visto na Figura 5), com base no atributo escolhido (acurácia, TP, F-measure e etc), gera uma única saída como resultado. No desenvolvimento foi considerado o maior valor de *precisão* e cada classificador utilizará a mesma base de treinamento.

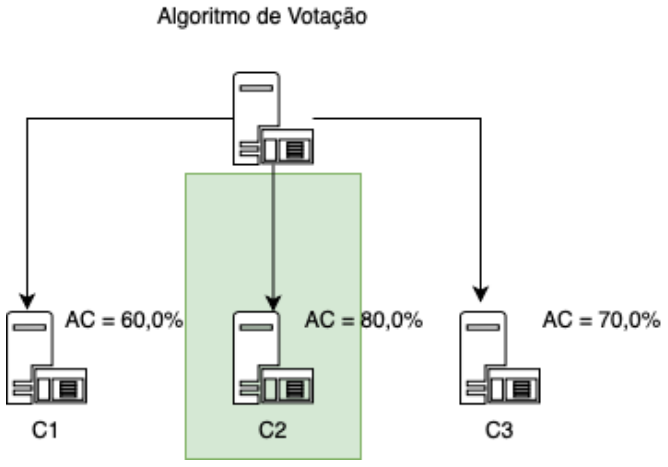

Figura 5. Mecanismo de seleção do algoritmo de votação

4.4. Análise de Performance

Para realizar a análise de performance dos algoritmos classificadores foi considerada a matriz de confusão de cada um dos mesmos. Onde VP é o número de *Verdadeiros Positivos*, VN o número de *Verdadeiros Negativos*, FP o número de *Falsos Positivos* e FN os *Falsos Negativos*.

O primeiro índice a ser avaliado foi o de acurácia dado por:

$Acuracia = \frac{VP+VN}{VP+FP+FN+VN}$

Em seguida, foi considerado o índice de *recall* (também conhecido como revocação), precisão e de *f-measure* (conhecido como medida-f), dados respectivamente pelas fórmulas:

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

$Precisao = \frac{VP}{VP+FP}$

$FMeasure = 2 * \frac{Recall*Precisao}{Recall+Precisao}$

Tendo cada um dos índices para cada um dos algoritmos classificadores é possível verificar precisão e efetividade de cada um deles. Num primeiro momento serão comparados os índices de *F-Measure* já que a mesma está diretamente ligada a acurácia de cada um dos classificadores. Após isso, também será levado em conta a precisão dos algoritmos e os três melhores serão selecionados para compor o método *ensemble* que levará em conta o algoritmo que resultar na maior probabilidade de classificação correta das classes.

5. Resultados

Nessa seção serão apresentados os resultados da etapa de coleta de dados utilizando API do *Twitter*, pré-processamento, classificação utilizando a técnica *ensemble* e análise de performance dos principais algoritmos de classificação textual na plataforma WEKA. Ao final será apresentada a *engine* que foi desenvolvida com base nos experimentos realizados com a base de dados do *Twitter* para contribuir nos projetos que necessitem detectar discursos de ódio na língua portuguesa. A ferramenta está disponível no *GitHub* e pode ser baixada pelo link: `https://github.com/AlissonRTeles/SIO.git`.

5.1. Coleta dos Dados

Após utilizar a API *Twitter Developer Platform* foi possível coletar 489 *tweets* neutros e 497 *tweets* de discurso de ódio, totalizando 986 *tweets* para formar a base de dados rotulada. A seguir é possível verificar exemplos de textos neutros:

”ai esse cara nem é tao bonito assim nao sei pra que tanto biscoito”

”o dia que bolo de bolacha se chamar bolo de biscoito, vc fala comigo, beijos”

”eu amo essa música hahahhaha”

E a seguir é possível verificar exemplos de discurso de ódio coletados:

”que isso sua oferecida”

”gordo maldito, eu tô amaldiçoando vc sua bola de banha”

”exterminar logo essa raça nojenta”

Com esse material já coletado foi possível dar início a próxima etapa necessária para gerar o dicionário de léxicos esperado.

5.2. Pré-Processamento dos Dados

Na etapa de pré-processamento dos dados foram aplicados filtros textuais afim de eliminar elementos que não contribuíam com a análise. No exemplo abaixo é possível observar os filtros (uniformização de letras para maiúsculas, remoção de conteúdo numérico e caracteres especiais, além da remoção das *stopwords*) aplicados nos textos coletados. A frase utilizada para o exemplo será:

$F_n =$”Nossa!! Eu tenho só 3 maçãs!”

A primeira transformação sofrida foi a de remoção de caracteres especiais sendo substituídos por caracteres alfanuméricos equivalentes:

$F_n =$ ”Nossa Eu tenho so 3 macas”

Após, uma lista de *stopwords* foram comparadas afim de remover palavras neutras que não apresentaram teor relevante de sentimento.

$F_n =$ ”Nossa Eu tenho 3 macas”

Como penúltimo procedimento de pré-processamento, todos os conteúdos numéricos foram removidos do texto.

$F_n =$ ”Nossa Eu tenho macas”

Por fim, todas as letras foram convencionadas para maiúsculas.

$F_n =$ ”NOSSA EU TENHO MACAS”

Esses filtros foram aplicados em todos os textos coletados (986 textos). Os resultados foram salvos em arquivos binários para serem utilizados na etapa de transformação dos dados em base de treinamento rotulada manualmente por um especialista humano.

5.3. Base de Treinamento

Após executar a coleta e a transformação dos dados, foi possível gerar a base de treinamento que foi utilizada pelos algoritmos de classificação. Na Figura 6 é possível verificar as instâncias bem como as suas classificações (rotuladas manualmente por um especialista humano). A mesma base de treinamento foi utilizada em todos os experimentos realizados pela pesquisa.

5.4. Resultados dos Algoritmos de Classificação

Na Tabela 3 é possível verificar o resultado do experimento realizado com os classificadores selecionados na ferramenta já mencionada. Para realizar o experimento, foi parametrizado 10 iterações para cada algoritmo, gerando assim 500 resultados analisados que foram avaliados conforme a *precisão*, *f-measure*, *recall* e *acurácia*.

Com esses resultados foi possível confirmar o desempenho dos classificadores mencionados nos trabalhos correlatos (Seção 2) bem como selecionar os classificadores que compõem o *ensemble* de votação. Os algoritmos selecionados foram: *Bayes Net*, *Naive Bayes* e SMO.

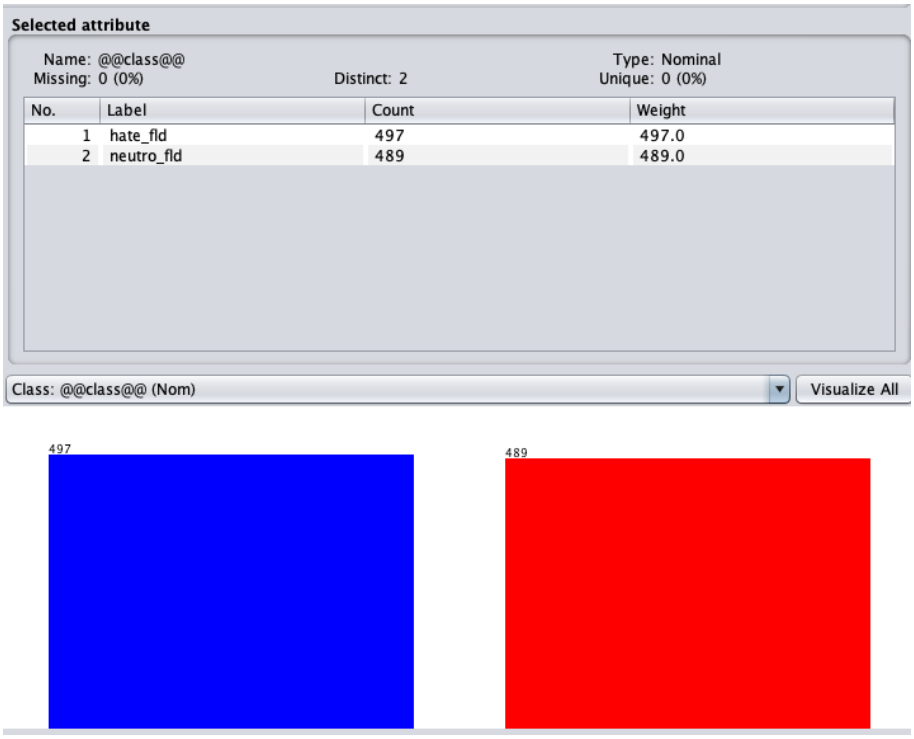

Figura 6. Base de treinamento gerada no WEKA

Tabela 3. Performance dos classificadores

Classificadores	F-Measure	Recall	Precisão	Acurácia
Bayes Net	0,86	1,0	0,75	83,16%
Naive Bayes	0,88	0,98	0,8	86,22%
Naive Bayes Multinomial Text	0,67	1,0	0,5	50,41%
Naive Bayes Updatable	0,88	0,98	0,8	86,22%
SMO	0,89	0,99	0,81	88,00%

5.5. Engine Implementada

Com a intensão de auxiliar no progresso dos estudos sobre a Análise de Sentimentos na língua portuguesa foram desenvolvidas uma série de mecanismos que, encapsulados, formaram a *engine* resultado de toda a pesquisa descrita nesse artigo. A ferramenta foi desenvolvida na linguagem *Java* e está empacotada na extensão *.jar*. A ferramenta pode ser baixada no site `https://github.com/AlissonRTeles/SIO.git`. Os detalhes de cada uma das classes podem ser vistos na Figura 7.

Cada uma das classes disponibilizadas diz respeito a uma etapa da Análise de Sentimentos (coleta, pré-processamento, classificação e resultados), além das classes principais. A seguir são apresentadas cada uma das classes, reproduzidas em uma interface gráfica que implementa as funcionalidades básicas de cada uma das etapas da Análise de Sentimentos.

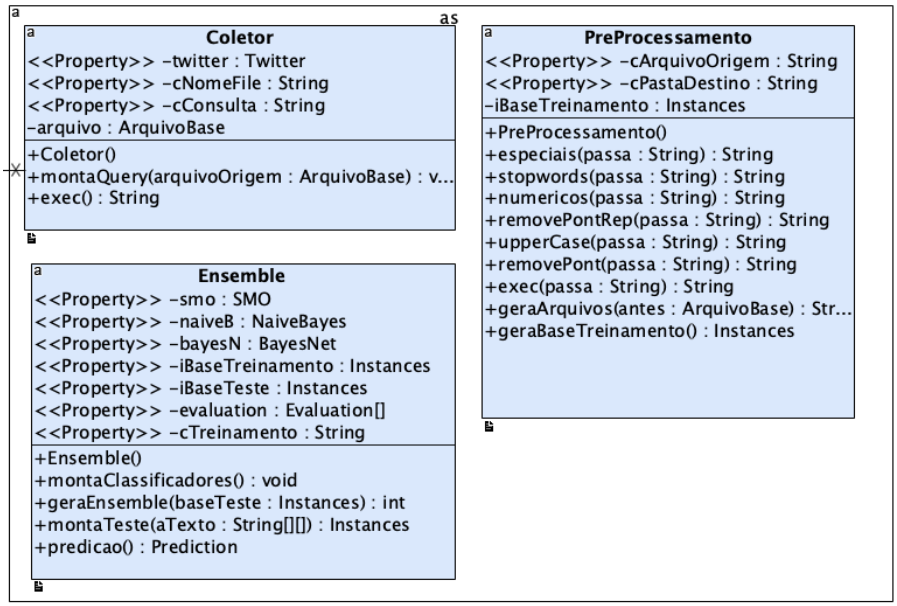

Figura 7. Diagrama de classes da ferramenta implementada

5.5.1. Classe Coletora

A classe *Coletor* é responsável pela coleta de textos. Por padrão essa funcionalidade é integrada com a API do *Twitter* sendo também possível a integração com outras ferramentas de integração. Na Figura 8 é possível verificar essa funcionalidade incorporada a uma interface gráfica com as funcionalidades básicas da *engine*.

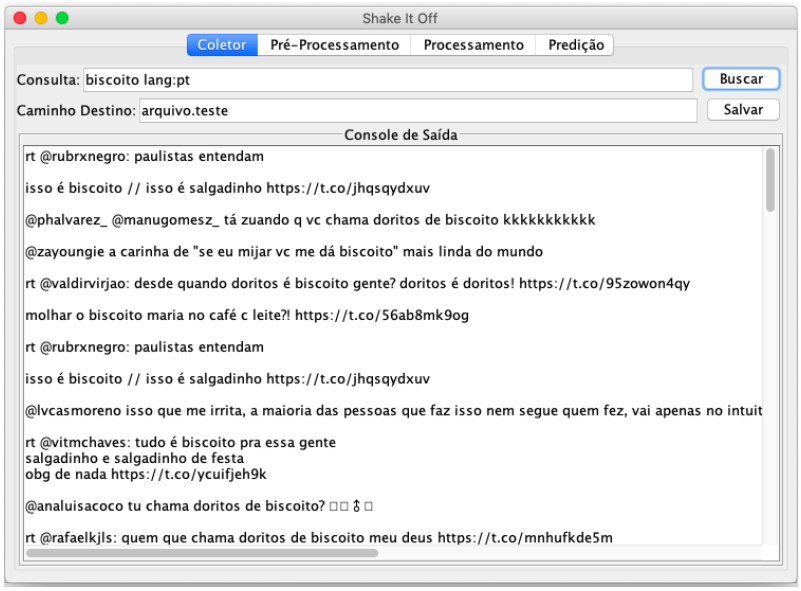

Figura 8. Funcionalidade de coleta de expressões

No exemplo da Figura 8 é apresentada a consulta pela palavra *biscoito* e pelo idioma português (*lang:pt*). Na área de *Console de Saída* é possível verificar o resultado da consulta, sendo *tweets* dos últimos 7 dias. Também é possível salvar o resultado da

consulta preenchendo o campo *Caminho Destino*.

5.5.2. Classe de Pré-Processamento

A classe responsável pelo pré-processamento é a *PreProcessamento*. Com ela é possível extrair as palavras principais de cada uma das frases contidas em qualquer arquivo de coleta (oriundos ou não de coleta gerada pela classe responsável). A classe também desmembra cada uma das frases e um único arquivo. É importante salientar que os arquivos devem ser salvos em pastas que correspondam a sua classe. Na Figura 9 é possível verificar que a classe correspondente dos textos presentes em *teste2.txt* é a *neutro*.

Após gerar os arquivos é possível também gerar a base de treinamento com todos os atributos necessários para, posteriormente, serem analisados pela classe de classificação.

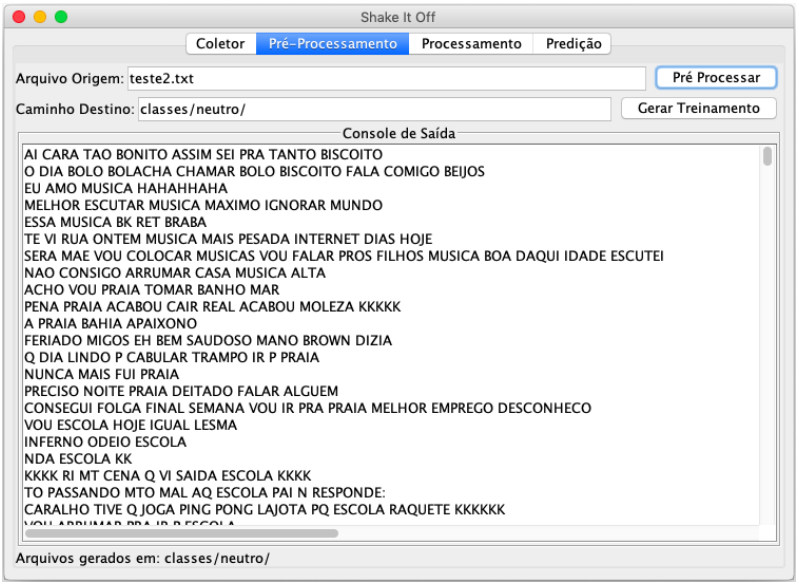

Figura 9. Funcionalidade de pré-processamento

5.5.3. Classe que Implementa Ensemble de Classificadores

A classe responsável pelo aprendizado, ou, classificação da base de treinamento, é a *Ensemble*. Ela realiza testes com a base de treinamento baseada na API do *WEKA* e, baseada nos experimentos realizados faz uso dos algoritmos: *Naive Bayes*, *BayesNet* e *SMO*. Como o *ensemble* implementado utiliza o critério de votação, a cada novo teste é avaliado o algoritmo de melhor *precisão* para que esse seja assumido na fase de predição.

Na Figura 10 é possível verificar a saída básica da classe. São apresentados os dados estatísticos de cada um dos algoritmos classificadores bem como a avaliação de qual algoritmo tem a melhor avaliação pelo *ensemble*. Todas as enumerações e predições são públicas dentro da classe, qualquer alteração nas regras de apresentação dos resultados é possível.

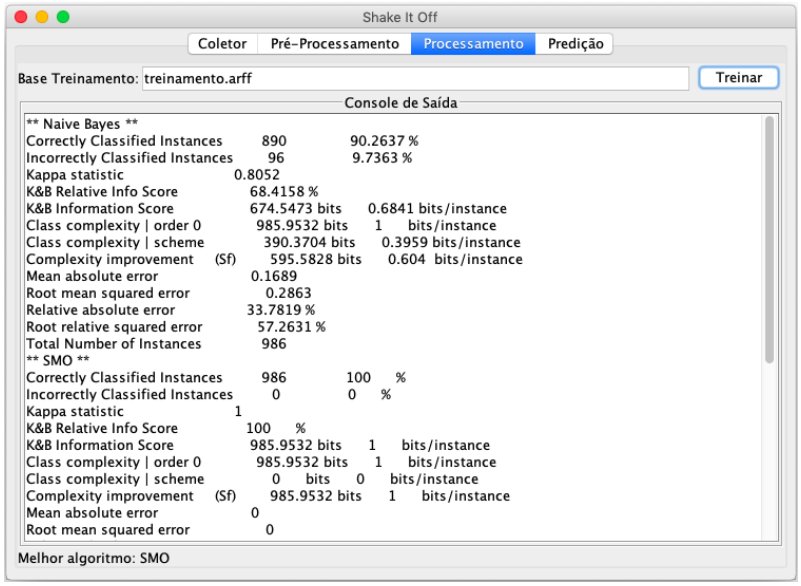

Figura 10. Funcionalidade de classificação

Para realizar uma predição, é necessário gerar um texto de teste. Esse texto deve ser pré classificado como: *neutro*, *hate*, ou ainda, nenhuma classe (aqui representado por *?*). Após esse processo é possível realizar a predição do texto com base no arquivo de treinamento já classificado. Como saída da predição é possível verificar as matrizes de confusão, dados estatísticos como *Recall*, *F-Measure*, *Precision* e etc (Figura 11).

Figura 11. Demonstração de saída de predição com texto de teste

5.6. Resultados da Engine Implementada

Após o desenvolvimento da *engine* foi realizado teste para validar a precisão da implementação, bem como validar a precisão da base de treinamento gerada durante as etapas de coleta e pré-processamento dos dados. Para realizar o teste foram coletados 40 textos do *Twitter*, pré-rotulados e compostos por 20 textos neutros e 20 textos que foram devidamente preparados para compor a base de testes utilizada para verificar a precisão da ferramenta implementada.

Ao final do teste aplicado foi possível verificar que a acurácia da implementação foi de 90% (Tabela 4) sendo que, dentre os 40 textos, 4 textos neutros foram classificados incorretamente. O resultado apresentado foi satisfatório para a pesquisa e também motiva

o aprimoramento da ferramenta em trabalhos futuros onde novas metodologias poderão ser desenvolvidas e também onde a base de treinamento poderá ser aprimorada.

Tabela 4. Resultado do teste realizado na ferramenta implementada

Classe	F-Measure	Recall	Precisão
Neutro	0,89	0,80	1,0
Ódio	0,91	1,00	0,83
Total	0,90	0,90	0,92

6. Conclusões

Após realizar uma pesquisa de trabalhos correlatos sobre a *Análise de Sentimentos* e também sobre os *Discursos de Ódio* foi possível verificar o atual cenário dos casos de violência digital e também suas consequências, e ainda, foi possível apurar as melhores técnicas de aprendizado de máquina utilizadas para detectar sentimentos em dados textuais. Os trabalhos correlatos são encontrados na Seção 2.

Para fundamentar o objetivo da pesquisa foi realizado estudo sobre cada etapa e as principais técnicas de Análise de Sentimentos sendo apresentadas as melhores práticas e também representações gráficas de cada abordagem. Uma breve análise dos algoritmos bayesianos e de SVM foram realizados durante a pesquisa, já que as mesmas foram utilizadas nos trabalhos correlatos. Tendo o referencial teórico (Seção 3) foi possível estruturar os procedimentos necessários para realizar a classificação textual e a geração da base de dados utilizada no processo de treinamento de cada algoritmo de classificação.

Após a fase de pesquisa foi possível estruturar os procedimentos necessários para desenvolver uma ferramenta que detectasse discursos de ódio. A descrição desses procedimentos é apresentada na Seção 4, onde são mencionadas as *APIs*, linguagem de programação, técnicas de coleta, transformação e análise de performance utilizadas para o desenvolvimento de uma ferramenta capaz de detectar discursos de ódio na língua portuguesa.

O resultado final da pesquisa foi uma *engine*, desenvolvida na linguagem de programação *Java*, que realiza a detecção de discurso de ódio com base em dados textuais. A ferramenta disponibiliza recursos independentes para as etapas de coleta de dados, pré-processamento, classificação e apresentação de resultados e tem o objetivo de auxiliar no desenvolvimento do tema que, ainda hoje, é pouco explorado na língua portuguesa. Após realizado teste com 40 textos coletados da base de dados do *Twitter* e pré rotulados em 20 textos neutros e 20 textos de ódio foi possível verificar que a implementação tem 90% de acurácia e uma precisão de 92%.

Com o resultado do teste aplicado foi possível concluir que a ferramenta implementada poderá contribuir nas pesquisas com a temática de Análise de Sentimentos, bem como implementações que necessitem detectar discursos de ódio na língua portuguesa. Como a ferramenta foi desenvolvida em uma linguagem de programação multiplataforma também se conclui que novos trabalhos poderão explorar outras plataformas, tanto para a coleta dos dados como para a etapa de classificação e predição.

Referências

Amaral, F. (2016). *Introdução à Ciência de Dados. Mineração de Dados e Big Data (Em Português do Brasil)*. Alta Books.

Bahri, S., Bahri, P., and Lal, S. (2018). A novel approach of sentiment classification using emoticons. *Procedia Computer Science*, 132:669 – 678. International Conference on Computational Intelligence and Data Science.

Balahur, A. (2013). Sentiment analysis in social media texts. In *WASSA@NAACL-HLT*.

Barbosa, L. and Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chetty, N. and Alathur, S. (2018). Hate speech review in the context of online social networks. *Aggression and Violent Behavior*, 40:108–118.

Correa, I. T. (2017). Análise dos sentimentos expressos na rede social twitter em relação aos filmes indicados ao oscar 2017. Universidade Federal de Uberlândia.

de Pelle, R. P. and Moreira, V. P. (2017). Offensive comments in the brazilian web: a dataset and baseline results. In *6th Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78. ACL.

Ekman, P. (1992). An argument for basic emotions. volume 6, pages 169–200. Informa UK Limited.

Ferilli, S., Esposito, F., and Grieco, D. (2014). Automatic learning of linguistic resources for stopword removal and stemming from text. *Procedia Computer Science*, 38:116–123.

Guzman, E. and Maalej, W. (2014). How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. IEEE.

He, Y., Lin, C., and Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 123–131, Stroudsburg, PA, USA. Association for Computational Linguistics.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Howard, R. A. and Matheson, J. E. (2005). Influence diagrams. *Decision Analysis*, 2(3):127–143.

IBGE (2017). *Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal: 2016*. Centro de Documentação e Disseminação de Informações. Fundação Intituto Brasileiro de Geografia e Estatística, Rio de Janeiro, 1 edition.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Khan, F. H., Bashir, S., and Qamar, U. (2014). TOM: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems*, 57:245–257.

Kouloumpis, E., Wilson, T., and Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg!

Kwok, I. and Wang, Y. (2013). Locate the hate: Detecting tweets against blacks.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM 09*. ACM Press.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*, pages 41–48.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Mohammad, S. M., Zhu, X., Kiritchenko, S., and Martin, J. (2015). Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Provost, F. and Kohavi, R. (1998). On applied research in machine learning. In *Machine learning*, pages 127–132.

Schmitt, V. F. (2013). Uma análise comparativa de técnicas de aprendizagem de máquina para prever a popularidade de postagens no facebook.

Symeonidis, S., Effrosynidis, D., and Arampatzis, A. (2018). A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, 110:298 – 310.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.

Viegas, F., Rocha, L., Resende, E., Salles, T., Martins, W., e Freitas, M. F., and Gonçalves, M. A. (2018). Exploiting efficient and effective lazy semi-bayesian strategies for text classification. *Neurocomputing*, 307:153 – 171.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann.

Zhang, Z., Wu, G., and Lan, M. (2015). Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. pages 561–567.

Zou, K. H., Liu, A., Bandos, A. I., Ohno-Machado, L., and Rockette, H. E. (2011). *Statistical Evaluation of Diagnostic Performance: Topics in ROC Analysis (Chapman & Hall/CRC Biostatistics Series)*. Chapman and Hall/CRC.