Detecção automática de discurso de ódio em comentários online

Peter Dias Paiva[1], Vanecy Matias da Silva[2], Raimundo Santos Moura[1]

[1] Departamento de Computação – Universidade Federal do Piauí (UFPI)

[2] Departamento de Ciências Sociais – Universidade Federal do Piauí (UFPI)

Teresina – Piauí – Brasil

Abstract. *Fighting hate speech on the Internet has become a major challenge. With this in mind, we provide a solution to detect offensive comments that seek to spread hate on the network. The solution used a Bag of Words, which was applied to a database of Portuguese comments taken from a news site. The results showed the feasibility of the proposal, reaching significant values and could be used as the basis for the development of new applications.*

Resumo. *Combater discurso de ódio na Internet tem se tornado um grande desafio. Neste sentido, propõe-se uma solução para detectar comentários ofensivos que buscam disseminar ódio na rede. A proposta usa um Bag of Words, aplicado a uma base de comentários em português retirados de um site de notícias. Os resultados mostram a viabilidade da proposta e que ela pode ser utilizada como base para o desenvolvimento de novas aplicações.*

1. Introdução

A Internet tem sido palco de debates e de manifestações de opiniões pessoais sobre diversos assuntos. Mas como era de se esperar, nem sempre isso ocorre de maneira civilizada e saudável. Muitas pessoas insistem em disseminar ódio através de comentários ofensivos. Casos famosos de ataques em redes sociais, como o ocorrido recentemente com algumas celebridades, evidenciam isso cada vez mais (UOL Notícias, 2019). A ineficiência de barrar atitudes desse tipo na Internet é bastante clara.

Discurso de ódio, segundo Nockleby (2000), é qualquer forma de comunicação que tem o intuito de ofender uma pessoa ou grupo baseando-se em alguma de suas características, como raça, cor, etnia, gênero, nacionalidade, orientação sexual e outros. Detectar discurso de ódio na Web analisando comentários de usuários possibilitará que as redes sociais e os sites em geral alertem o que está sendo postado por seus usuários, registrando casos como o que foi citado anteriormente.

Determinar de forma automática se um comentário é ofensivo ou não, é um problema já conhecido e bastante estudado na Inteligência Artificial (IA), trata-se de uma tarefa de classificação. Algoritmos de Aprendizagem de Máquina têm obtido êxito em cenários desse tipo, devido à capacidade de "aprender" a partir de exemplos reais obtidos do ambiente em que irão atuar, de forma que padrões possam ser reconhecidos (Henke et al, 2011). Utilizar classificação de textos torna possível elucidar problemas complexos, como o explorado neste trabalho, que busca determinar se o autor de um texto tem a intenção de ofender alguém. Dessa forma, tendo como base todos esses conceitos, propõe-se um algoritmo capaz de detectar discurso de ódio utilizando uma base de dados de comentários em português.

Além desta seção introdutória, o restante do trabalho está organizado como segue. A Seção 2 descreve alguns trabalhos sobre detecção de discurso de ódio em redes sociais. A Seção 3 explica a abordagem proposta, descrevendo cada passo, desde o pré-processamento até o uso de *Bag of Words*. A Seção 4 exibe os experimentos realizados e os resultados obtidos. A Seção 5 termina com as considerações finais e conclusões.

2. Trabalhos Relacionados

Poucos trabalhos relacionados com detecção de discurso de ódio abordam a língua portuguesa e os que o fazem, geram sua própria base de dados. Pelle and Moreira (2017) retiraram comentários de um site de notícias brasileiro e realizaram a detecção de discurso de ódio através de métodos de aprendizado supervisionado clássicos como *Support Vector Machine* (SVM) e *Naive Bayes* (NB). Fortuna (2017) coletou *tweets* em português a fim de realizar a detecção de discurso de ódio por meio de um sistema de classificação hierárquica, através de SVM modificado (SVMLinear).

Xiang et al. (2012) aplicou classificadores supervisionados a um conjunto de treinamento composto por *tweets* na língua inglesa. Kwok and Wang (2013) também trabalhou com tweets em inglês, mas com o objetivo de detectar comentários racistas. Mondal et al. (2017) utilizaram o formato de sentença "I <intensity> <userintent> <hatetarget>" para identificar discursos de ódio. Em tal formato, I é referente ao usuário que escreveu o comentário, <intensity> revela o sentimento do usuário (geralmente um verbo), <userintent> está relacionado a palavras que remetem ódio, <hatetarget> corresponde a quem o ódio está sendo dirigido.

3. Abordagem proposta

A abordagem consiste em analisar um comentário e classificá-lo como sendo discurso de ódio ou não. Para isso, iremos treinar o modelo com um conjunto de dados anotados e, depois, iremos classificar novos comentários de acordo com o modelo treinado. A Figura 1 esquematiza a ideia geral da proposta.

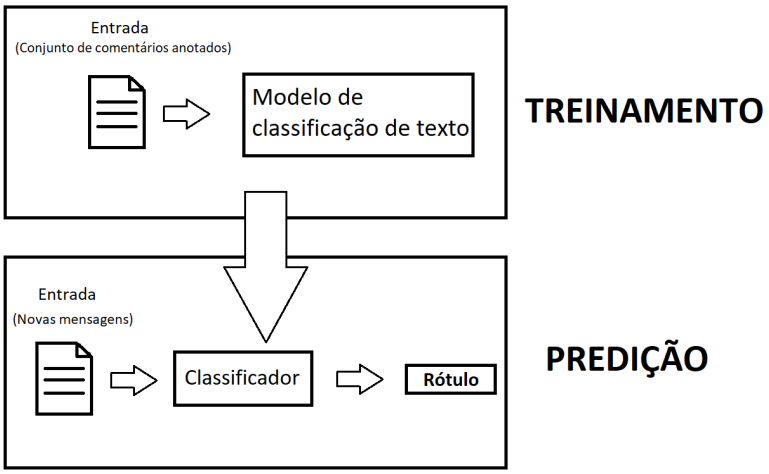

Figura 1. Esquema Geral da abordagem proposta.

O modelo de classificação de texto é dividido em três fases: pré-processamento dos dados, extração de *features* básicas e uso de *Bag of Words*. Antes de descrever cada um deles, o conjunto de dados utilizado no trabalho é apresentado.

3.1. Conjunto de dados

O conjunto de dados utilizado para treinamento e validação, consistiu de uma versão modificada do OffComBR, um *dataset* criado por Pelle and Moreira (2017) que contém comentários ofensivos e não-ofensivos, retirados de um portal de notícias brasileiro.

O OffComBR possui duas versões: o OffComBR-2 e o OffComBR-3. O primeiro possui ao todo 1250 comentários, sendo 831 não-ofensivos e 419 ofensivos, que segundo os autores, foram avaliados por pelo menos dois juízes. Já o OffComBR-3, possui 1033 comentários, com 831 não-ofensivos e 202 ofensivos, determinado por três juízes. Apesar de ter sido originalmente dividido em categorias (racismo, sexismo, homofobia, xenofobia, intolerância religiosa, entre outras), o conjunto disponibilizado pelos autores está classificado apenas de forma binária (ofensivo e não-ofensivo).

Por haver discordância em alguns pontos da classificação original, realizou-se uma análise manual do conjunto com dois juízes, o que resultou na mudança de categoria de alguns comentários. Além disso, uma parte dos dados (escolhida de forma aleatória) foi deixada de lado, a fim de balancear as duas classes. Com isso, a versão derivada do OffComBR possui 406 comentários ofensivos e 406 não-ofensivos.

3.2. Pré-processamento

Para extrair informações mais precisas dos dados, foi necessário prepará-los, excluindo ruídos e informações consideradas desnecessárias, tudo isso feito de forma algorítmica. A Figura 2 abaixo ilustra o passo-a-passo desta fase através de um exemplo.

Texto original
"Que coisa relevanteee a mulher dele e VOLUNTARIA ahahahah"

1. Conversão do texto para minúsculo
"que coisa relevanteee a mulher dele e voluntaria ahahahah"

2. Exclusão de "risos"
"que coisa relevanteee a mulher dele e voluntaria"

3. Exclusão de letras repetidas sequencialmente
"que coisa relevante a mulher dele e voluntaria"

4. Tokenização
["que", "coisa", "relevante", "a", "mulher", "dele", "e", "voluntaria"]

5. Exclusão de *stopwords*
["coisa", "relevante", "mulher", "voluntaria"]

Figura 2. Pré-processamento aplicado em um comentário.

Inicialmente, o texto é convertido para minúsculo. Em seguida, palavras que simbolizam "risadas" são excluídas tendo como base uma pequena lista (construída de forma empírica) de termos comuns do linguajar da Internet, como por exemplo, "kkk" e "hahah". Depois, são deletadas letras repetidas em palavras, tendo o cuidado de verificar, através do uso de um dicionário (projeto VERO, 2013), se realmente se trata de um erro de grafia, já que o português possui termos com letras duplicadas. Após isso, é realizada a quebra da sentença em unidades significativas, chamada de tokenização. Por fim, as *stopwords*[1] são detectadas e removidas levando em conta a lista de palavras fornecida pela função *stopwords.words('portuguese')* do NLTK (*Natural Language Toolkit*).

[1] *Stopwords* são palavras comuns que não carregam significado, por exemplo, artigos e preposições.

3.3. *Features* básicas

Nesta etapa, após o pré-processamento, as informações pertinentes foram extraídas, onde cada comentário passou a ser representado através de um conjunto de *features* (ver Figura 3), buscando elencar as informações que descrevem o conteúdo de um comentário e baseado nesta representação determinar se ele é ofensivo ou não.

Comentário pré-processado
["coisa", "relevante", "mulher", "voluntaria"]

Comentário representado através de *features*
{'quant_ofensivas': 0, 'quant_palavras': 4, 'tam_medio_palavras': 7.5, 'quant_carac': 30, 'palavras_corretas': 4}

Figura 3. Comentário representado por meio de *features*.

Ao todo foram utilizadas cinco *features* básicas, sendo elas:

- **Quan_ofensivas:** a quantidade de palavras de baixo calão presentes no comentário. Para identificação de tais palavras utilizou-se uma lista criada por Ranzi (2017), que contém palavrões e xingamentos comuns que são utilizados na Internet, como por exemplo, "idiota", "burro", "vadia", entre outras.
- **Quant_palavras:** a quantidade de palavras do comentário;
- **Tam_medio_palavras:** o somatório do tamanho de cada palavra do comentário dividido pela quantidade total de palavras dele;
- **Quant_carac:** a quantidade de caracteres do comentário;
- **Palavras_corretas:** a quantidade de palavras corretas do comentário, considerando o dicionário do projeto VERO (LibreOffice, 2013).

3.4. *Bag of Words*

As features citadas anteriormente, mesmo sendo úteis, não foram consideradas suficientes para identificar a intenção de um comentário (se busca ofender ou não). Por isso optou-se pela utilização de um novo método, o *Bag of Words*.

A premissa do *Bag of Words* é representar cada comentário através de um vetor, onde cada um de seus elementos sinaliza se uma determinada palavra está presente ou não nele (Culotta and Sorensen, 2004). O intuito é determinar o grau de similaridade, já que documentos parecidos terão um certo número de palavras em comum.

Inicialmente, foram consideradas as 500 palavras mais frequentes de toda a base de dados (obtidas empiricamente). Verificou-se quais delas estavam presentes ou não em cada comentário, de modo que estas informações fossem incorporadas ao conjunto de *features* já existente. A Figura 4 mostra o exemplo da Figura 3 após aplicação de *Bag of Words*.

Comentário representado através de *features* e *Bag of Words*
{'quant_ofensivas': 0, 'quant_palavras': 4, 'tam_medio_palavras': 7.5, 'quant_carac': 30, 'palavras_corretas': 4, 'idiota': False, 'vc': False, 'coisa': True, ...}

Figura 4. Comentário representado pelas *features* básicas e por *Bag of Words*.

4. Experimentos e Resultados

Para os experimentos foram selecionados sete classificadores que utilizam Aprendizagem Supervisionada, sendo eles: três do tipo Naive Bayes, Naive Bayes Original, Bernoulli Naive Bayes e Multinomial Naive Bayes; dois do tipo SVM, SVC e Linear SVC; e um do tipo Árvore de Decisão.

Cada algoritmo foi executado 5 vezes, sendo que em cada uma delas o conjunto de dados era embaralhado. Foram reservados 75% da base para treinamento e 25% para testes. O desempenho dos classificadores foi medido de acordo com a acurácia obtida, que demonstra a porcentagem de elementos do conjunto de teste que foram classificados corretamente. Ao final de todas as execuções foi realizada a média aritmética das acurácias, que podem ser conferidos na Tabela 1 abaixo.

Tabela 1. Acurácia média de cada um dos classificadores utilizados.

Classificador	Acurácia média com *Bag of Words*	Acurácia média sem *Bag of Words*
Naive Bayes Original	75%	70%
Bernoulli Naive Bayes	75%	73%
Multinomial Naive Bayes	81%	64%
SVC	74%	69%
Linear SVC	76%	73%
NuSVC	75%	70%
Árvore de Decisão	62%	57%

O uso de *Bag of Words* melhorou a acurácia de todos os classificadores, em especial do Multinomial Naive Bayes que obteve melhor desempenho, com uma acurácia média de 81%. Já a Árvore de Decisão apresentou pior resultado com 62%.

Na Figura 5 estão listadas as *features* que foram mais úteis para a classificação. Através delas é possível fazer uma análise do perfil dos dois tipos de comentários. A primeira linha diz respeito a relação entre comentários ofensivos que possuem uma palavra ofensiva e comentários não-ofensivos, na proporção de 20.5 para 1. Um resultado de certa forma já esperado, tendo em vista que comentários ofensivos certamente conterão palavras ofensivas.

A segunda linha aponta que comentários ofensivos são desleixados quanto a grafia, pois a relação entre ofensivos que não possuem nenhuma palavra correta e não-ofensivos segue a proporção de 9.6 para 1. As duas últimas linhas mostram que comentários não-ofensivos possuem maior número de palavras e maior tamanho médio de palavras, explicitando que são mais elaborados e contém mais conteúdo do que comentários ofensivos.

quant_ofensivas = 1	ofensi : nao_of = 20.5 : 1.0
palavras_corretas = 0	ofensi : nao_of = 9.6 : 1.0
tam_medio_palavras = 6.0	nao_of : ofensi = 5.4 : 1.0
quant_palavras = 10	nao_of : ofensi = 5.3 : 1.0

Figura 5. *Features* que foram úteis para a classificação.

5. Conclusão e Trabalhos Futuros

Após analisar os resultados, fica evidente que a abordagem proposta obteve certo êxito, atingindo acurácia superior a 80%. O *Bag of Words*, apesar de simples, influenciou de forma significativa os resultados. O Multinomial *Naive Bayes* produziu os melhores valores, demonstrando uma eficácia superior em relação aos outros classificadores.

Uma quantidade relativamente baixa de comentários foi utilizada, tanto para treinamento como para testes. No entanto, acredita-se que o desempenho da abordagem continue satisfatória mesmo com o aumento do conjunto de dados. Dessa forma, espera-se que futuramente mais informações sejam incorporadas ao conjunto utilizado.

Seria interessante também, como trabalho futuro, utilizar classificação não-binária, categorizando os comentários de acordo com a natureza da ofensa (racismo, homofobia, machismo, entre outras). Métodos mais sofisticados, como por exemplo, CNN (*Convolutional Neural Network*), também podem ser utilizados com o intuito de alcançar melhor precisão.

6. Referências

Culotta, A., and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In Proc. of the 42nd annual meeting on association for computational linguistics (ACL).

Fortuna, P. (2017). Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

Henke, M., Santos, C., Nunan, E., Feitosa, E., dos Santos, E., & Souto, E. (2011). Aprendizagem de máquina para segurança em redes de computadores: Métodos e aplicações. In Livro dos Minicursos do XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBC, ed.).

Kwok, I., and Wang, Y. (2013). Locate the hate: Detecting tweets against blacks. In Twenty-seventh AAAI conference on artificial intelligence.

Mondal, M., Silva, L. A., and Benevenuto, F. (2017). A measurement study of hate speech in social media. In Proc. of the 28th ACM Conference on Hypertext and Social Media.

Nockleby, J. (2000). Hate speech. Encyclopedia of the American constitution, 3(2).

Pelle, R., and Moreira, V. (2017). Offensive Comments in the Brazilian Web: a dataset and baselines results. In Proc. of the 6th Brazilian Workshop on Social Network Analysis and Mining (BraSNAM).

Ranzi, C. (2017) "lista-palavroes-bloqueio.txt". Disponível em: https://pt.scribd.com/document/345921799/lista-palavroes-bloqueio-txt.

UOL Notícias (2019) "De Xuxa a Madonna: Famosas sofrem ataques de ódio por envelhecerem", Disponível em: https://noticiasdatv.uol.com.br/noticia/celebridades/de-xuxa-madonna-artistas-sofrem-ataques-de-odio-por-envelhecerem-26946.

LibreOffice (2013) "VERO". Disponível em: https://pt-br.libreoffice.org/projetos/vero/.

Xiang, G., Fan, B., Wang, L., Hong, J., & Rose, C. (2012). Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In Proc. of the 21st ACM international conference on Information and knowledge management.