Extração de característica para identificação de discurso de ódio em documentos

Cleiton Lima[1], **Guilherme Dal Bianco**[1]

[1]Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brazil

cleiton.limapin@gmail.com, guilherme.dalbianco@uffs.edu.br

Abstract. *Social media is increasingly present in people's lives, including tools that allow users to collaborate with the creation of the content. Many users utilize these functions to post texts spreading illicit or criminal content. Most works on abusive identification use supervising learning, which demands the feature extraction to achieve good quality. The meta-feature represents a state-of-the-art feature extraction on text classification. In this work, we propose a combination of feature extraction to improve the detecting of offensive speech using meta-features. Our results, on real datasets, show that our proposed combination of features outperforms in around 3.5% the effectiveness of state-of-the-art approaches.*

Resumo. *As mídias sociais estão cada vez mais presentes na vida das pessoas, incluindo ferramentas que permitam que o usuário colabore com a criação do conteúdo nelas 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. Este artigo 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 a qualidade em até 3,5% em relação ao método base.*

1. Introdução

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 and Wiegand 2017]. Com isso a classificação de texto - a atividade de rotular textos de linguagem natural com categorias - vem 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 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. 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.

A classificação de documentos utilizando aprendizado de máquina para resolver esse tipo de problema vem sendo estudada por muitas empresas que sofrem com essa adversidade, dentre as quais destacam-se o *Facebook* e *Twitter* [Nobata et al. 2016]. Para o correto funcionamento dos algoritmos de classificação de textos, é preciso que os dados possuam características informativas. Essas características ou atributos representam informações que descrevem determinado documento. Desse modo, a extração de características possibilita construir modelos de classificação que identificam se determinado documento possui ou não um discurso de ódio.

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. 2016], os meta-atributos são criados através do algoritmo de classificação KNN (*k-nearest neighbors*). O KNN procura K documentos do conjunto de treinamento que estejam mais próximos deste documento com classificação desconhecida, ou seja, que tenham a menor distância. Os experimentos em duas bases de dados demostraram que a proposta obteve um ganho de até 3,5% se comparado ao método *baseline*.

O texto a seguir esta organizado da seguinte forma. Na Seção 2, são apresentados os conceitos de meta-atributos. Seção 3, os trabalhos relacionados são descritos. A proposta é apresentada na Seção 4. Os experimentos são apresentados na Seção 5. Por fim, a conclusão é descrita.

2. Meta-atributos

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)* [Canuto et al. 2016]. Os meta-atributos são capturados usando a vizinhança/similaridade de documentos previamente classificados utilizando o algoritmo de KNN para identificar os K vizinhos próximos. Os meta-atributos baseados em KNN contém os vetores de meta-atributos expressos como a concatenação dos sub-vetores descritos a seguir [Canuto et al. 2013]. Cada vetor de atributos mf é definido para um exemplo $xf \in X$ e categoria $cj \in C$ para $j = 1, 2, ..., m$. Seguem a seguir os três meta-atributos propostos no artigo:

- $\vec{v}_{\vec{x_f}}^{cnt} = [n_j]$: consiste em um vetor unidimensional (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.
- $\vec{v}_{\vec{x_f}}^{ncnt} = [\frac{n_j}{n_{max}}]$: consiste 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.
- $\vec{v}_{\vec{x_f}}^{qrt} = [\cos(\vec{x}_{ej}, \vec{x_f})]$: 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.

3. Trabalhos Relacionados

A detecção de texto abusivo vem sendo explorada com diversas abordagens. Um método bastante simplista é utilizar listas de palavras que remetem a conteúdo abusivo [Sood et al. 2012b]. Tais listas sofrem de uma baixa revocação já que o universo de termos ofensivos é bastante amplo e dinâmico. Para tentar mitigar isto, em [Sood et al. 2012a] é proposto o uso de contribuição colaborativa (ou *crowdsourcing*) para inferir termos ofensivos e dinamicamente identificar novos termos. No entanto, tais abordagens dependem de boas listas de palavras e podem resultar em uma precisão bastante reduzida.

O uso de modelos de predição (métodos supervisionados) surge como uma alternativa para possibilitar uma evolução na capacidade de identificação de textos ofensivos. Um dos primeiros trabalhos, tem como enfoque a identificação de textos ofensivos usando o método supervisionado *Support Vector Machines* (SVMs). No entanto, como os métodos supervisionados dependem do mapeamento de texto para valores numéricos, a extração de características (*features*) informativas é fundamental para alcançar uma alta qualidade. Em [Chen et al. 2012], por exemplo, usa a combinação de n-grams[1], lista de

[1]N-gram são uma sequência de termos com o comprimento de N caracteres.

termos abusivos, e manualmente constrói expressões regulares. Em [Nobata et al. 2016] são utilizados vários métodos de Processamento de Linguagem Natural (PLN) para criação de atributos. Tal trabalho propõe alguns novos atributos para aprimorar os resultados como tamanho médio de palavras, número de pontuações no documento, letras capitalizadas, entre outros.

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.

4. Proposta

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.

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 função de 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

Nesta seção, 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.

5.1. Configurações

A base de dados utilizada para a realização dos experimentos foi a utilizada no trabalho [de Pelle and Moreira 2017] e que pode ser obtida por *download* [2]. 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.

Para a aplicação do método, foram criados alguns conjuntos de experimentos, os mesmos propostos em [de Pelle and Moreira 2017]. Para cada experimento foram geradas determinadas características. Como é apresentado na Tabela 1, 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 [de Pelle and 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.

Para avaliar a eficácia do método proposto foi utilizada a mesma abordagem do trabalho [de Pelle and 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*). 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 $C = 1.0$. E o Naïve Bayes (NB)

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

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 1. Experimentos e suas características.

que foi utilizado com os parâmetros originais do classificador. Outras combinações de configurações serão exploradas nos próximos trabalhos. Cada teste foi executado com validação cruzada de dez vezes. O número de vizinhos utilizados para extração dos meta-atributos do algoritmo KNN foi definido como 30 $N = 30$, conforme sugestão de [Canuto et al. 2016].

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 [de Pelle and 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.

5.2. 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 2 e 3 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 $\bullet$.

Na base de dados *OffComBr-2*, após aplicar o método e suas execuções, apresentados na Tabela 2, 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 clas-

sificador 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* em 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 2. Experimentos com a base de dados *OffComBR-2*.

Na Tabela 3, 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.

	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 3. Experimentos com a base de dados *OffComBR-3*.

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

veram 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. Se compararmos o melhor caso do *baseline* com o melhor caso *baseline+LIMA*, podemos afirmar que o método proposto obteve ganho estatístico na base de dados OffComBR-3 e empate na base na base OffComBR-2. A Tabela 4 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 4. 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.

6. Conclusão

Esse artigo 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 utilizado 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 [de Pelle and 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 [de Pelle and 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.

Uma forma de complementar esse trabalho é explorar métodos de análise de sentimento, área onde houve bons resultados em trabalhos relacionados, e realizar a combinação com os meta-atributos. Novos experimentos serão realizados com intuito de avaliar configurações do classificador SVM, assim como, testes estatísticos complementares.

Referências

Batista, G. E. d. A. P. et al. (2003). *Pré-processamento de dados em aprendizado de máquina supervisionado*. PhD thesis, Universidade de São Paulo.

Canuto, S., Gonçalves, L. F., Salles, T., and Gonçalves, M. A. (2013). Um estudo sobre meta-atributos para classificaçao automática de texto.

Canuto, S., Gonçalves, M. A., and Benevenuto, F. (2016). Exploiting new sentiment-based meta-level features for effective sentiment analysis. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 53–62. ACM.

Chen, Y., Zhou, Y., Zhu, S., and Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

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)*. to appear.

Nakamura, F. G. et al. (2017). Uma abordagem para identificar e monitorar haters em redes sociais online.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Nockleby, J. T. (2000). Hate speech. *Encyclopedia of the American constitution*, 3:1277–79.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Sood, S. O., Antin, J., and Churchill, E. (2012a). Using crowdsourcing to improve profanity detection. In *2012 AAAI Spring Symposium Series*.

Sood, S. O., Churchill, E. F., and Antin, J. (2012b). Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2):270–285.