UFAM

UNIVERSIDADE FEDERAL DO AMAZONAS

RELATÓRIO FINAL PIBIC/PAIC 2017-2018

FORMULÁRIO PARA RELATÓRIO PARCIAL

1. **Identificação do Projeto**

Título do Projeto PIBIC/PAIC

Detecção de discurso de ódio em comentários online usando embeddings

Orientador

André Luiz da Costa Carvalho

Aluno

Eduardo Carvalho Lopes

2. **Informações de Acesso ao Documento**

2.1 **Este documento é confidencial?**

☐ **SIM** ☐ **NÃO**

2.2 **Este trabalho ocasionará registro de patente?**

☐ **SIM** ☐ **NÃO**

2.3 **Este trabalho pode ser liberado para reprodução?**

☐ **SIM** ☐ **NÃO**

2.4 Em caso de liberação parcial, quais dados podem ser liberados? Especifique.

- **Resumo**
- **Sumário**
- **Introdução**
- **Justificativa**
- **Objetivos (geral e específico)**
- **Revisão bibliográfica**
- **Métodos Utilizados**
- **Resultados e Discussões**
- **Conclusões**
- **Referências**
- **Cronograma de Atividades (discriminando etapas realizadas e a realizar do projeto)**

1. Resumo

O mau uso da internet por certos usuários prejudica e atinge negativamente usuários que necessitam realizar suas tarefas ou simplesmente se divertir. Um desses mau usos é o discurso de ódio presente majoritariamente nas redes sociais. Para impedir esse inconveniente na internet nós propomos um método de detecção de discurso de ódio, seja ele racista, sexista, xenofóbico ou ofensivo. Utilizaremos técnicas de processamento de texto, recuperação de informação e aprendizado de máquina para cumprirmos a tarefa de forma eficiente e precisa.

2. Introdução

Atualmente a quantidade de informação gerada na internet é enorme, por isso é uma tarefa difícil prever ou controlar os comportamentos dos usuários. No Twitter, por exemplo, 500 milhões de *tweets* são enviados por dia, isso é equivalente a 6000 *tweets* por segundo . Já o Facebook possui 1,87 bilhões de usuários ativos por mês e em média 300 milhões de fotos são enviadas por dia pelos usuários. Em meio a tanto conteúdo é comum encontrar postagens com discurso de ódio e discriminação, e a sua detecção ainda é feita majoritariamente manualmente por fiscais. Por conta da grande quantidade de posts denunciados como inapropriados alguns fiscais podem não ser tão rápidos e eficientes para o serviço.

Algumas razões da discriminação na internet são devido à etnia, religião e orientação sexual, por isso os ataques podem ter como alvo um usuário específico ou um grupo. Em [1] é citado um caso de 2013 em que um britânico foi assassinado por dois imigrantes do Oriente Médio. O acontecimento gerou revolta no Reino Unido e os islâmicos foram alvo de discriminação e discurso de ódio no Twitter. Em 2015, a jornalista Maria Júlia Coutinho foi vítima de injúrias racistas por usuários do Facebook. A *tag #SomosTodosMaju* foi propagada pelas redes sociais, assim como o questionamento sobre a facilidade de agressores insultarem usuários sem serem reconhecidos e nem punidos devidamente.

De fato é um inconveniente um usuário sofrer xingamentos e exclusão social por raça, religião ou qualquer outro motivo, e isso se tornou recorrente pela facilidade de ocultamento de perfil nas redes sociais, como o uso de contas *fakes*, e até a falta de penalidades legais aos agressores em comentários na internet. Muitas empresas, de jogos, notícias e redes sociais por exemplo, utilizam método relativamente fracos de detecção de discurso de ódio nos comentários chats,sendo que muitos são ultrapassados e poderiam ser melhorados para evitar transtornos ao usuário.

As técnicas de detecção atualmente usadas são fiscalização manual, o que gera demora e falta de efetividade na detecção além de um custo de mão de obra a mais para as empresas, detecção por *keywords*, que pode ser burlada muito facilmente por exemplo utilizando gírias e abreviações, e reconhecimento com base em palavras recorrentes em textos e comentários deste tipo, podendo gerar resultados não tão eficientes. Neste trabalho será utilizado o uso de aprendizado de máquina para detectar automaticamente textos com linguajar inapropriado e agressivo.

3. Justificativa

A detecção de discurso de ódio em textos na internet é muito importante para evitar impactos negativos nas vítimas de discriminação ou exclusão, assim como para evitar a impunidade em atos inapropriados na internet. Este problema é de difícil solução uma vez que é um problema adversarial, isto é, sempre estarão tentando burlar a detecção de várias formas. O uso de gírias ou abreviações dificulta muito a técnica de reconhecimento, uma vez que as palavras sem o contexto podem parecer erro de digitação. A elaboração de trechos dependentes também não tem reconhecimento trivial, uma vez que uma frase só tem significado se o sentido da frase anterior for captado. Mensagens subliminares, palavras ambíguas e sarcasmo tornam difícil a extração do sentido do trecho dito, não conseguindo avaliar se é ou não ofensivo.

Alguns estudos na área resolvem alguns dos problemas citados, como o em [2] que tornou possível extrair o sentido de palavras ambíguas, e em [3] é possível ver que palavras de uma “lista negra" podem não ser consideradas ofensivas se estiverem no contexto apropriado. Em [4] é utilizado um método que reconhece o sentido de frases e parágrafos, também chamado de *paragraph2vec*, e em [5] foi utilizado *machine learning* juntamente com *NLP* (*Natural Language Processing*) para detectar se o texto é ofensivo, conseguindo filtrar sarcasmo, ambiguidade, mensagens subliminares e outros problemas citados.

4. Objetivos

Desenvolver um método de detecção de discurso de ódio em comentários online através do uso redes neurais para a geração de embeddings do texto do comentário em adição a outras características.

4.1 Objetivos Específicos

1. Desenvolvimento de métodos de detecção de discurso de ódio baseados unicamente no próprio texto do comentário, com sua avaliação em comentários em inglês.
2. Expansão dos métodos criados para incluir também características extra-texto, como o conteúdo da postagem original, perfil de usuário e outros comentários no mesmo post
3. Avaliação e adaptação dos métodos propostos com comentários feitos na língua portuguesa.

5. Revisão bibliográfica

A solução mais simples para o problema seria utilizar as probabilidades de ocorrências das palavras usando métodos de recuperação de informação tais como *modelo vetorial*, *language models*, *set based model*, e outros. Os experimentos em [6] mostram que apenas a abordagem de *bag of words* é menos eficiente quando comparado a métodos de NLP juntamente com *machine learning*, o uso de métodos envolvendo *TF-IDF* obtiveram uma precisão de apenas 82% em uma base retirada do Twitter.

O problema de categorização de texto também pode ser abordado com análise de sentimentos, também chamado de *opinion mining*, e é um campo que estuda e analisa as opiniões, sentimentos, avaliações, atitudes e emoções para certos produtos, serviços, eventos, tópicos e outros [7]. O uso desse método também foi aprimorado com uso de aprendizado de máquina e recuperação de informação como foi o caso em [8]. Em que foi utilizado uma forma adaptada do modelo vetorial utilizando uma fórmula específica para atribuir

pesos para cada palavra, uma representação em grafo do texto. Os testes foram realizados em uma base referente à críticas de filmes, avaliando se a análise foi positiva ou negativa. O classificador utilizados nos experimentos foi o *SVM* (*Support Vector Machine*), esse classificador é bom para o problema uma vez que é resistente à ruídos de texto e é bom para tarefas de categorização, conforme foi provado em [9], e foi possível alcançar uma precisão de 88%.

Outra forma de realizar a tarefa de detecção de discurso de ódio seria o uso de uma lista de palavras, também chamado de *lexicon approach*. Para isso é feito um estudo prévio das características de cada palavra sendo ela classificadas como positivas, negativas ou neutras. O método é eficiente uma vez que serve como contexto para a frase a ser analisada no entanto não é bom utilizá-lo como único parâmetro para o classificador. As experimentações em [10] avaliam se um comentário é "fortemente ofensivo", "levemente ofensivo" ou se não é ofensivo, em uma base retirada de *blogs* e foi obtido uma precisão de 73%. Este método em redes sociais é viável uma vez que os comentários são em sua maioria curtos e sem muito contexto a ser analisado.

A abordagem mais utilizada para o problema é a generalização de palavras, que é basicamente a tentativa de prever palavras que apareceriam com base nos dados coletados, *word embedding* é um método de generalização de palavra baseado em redes neurais, para cada palavra existe um vetor com os pesos das palavras mais prováveis a surgirem adjacentes. Em [5] foi utilizado *lexicon approach* juntamente como a generalização de palavras e com manipulação das características linguísticas e sintáticas, também podemos chamar de meta informações, e realizado os testes em uma base de comentários do *Yahoo News* foi obtido uma precisão de 81%. Em [6] é possível perceber a eficiência da generalização de palavras, implementada em *GloVe* foi possível atingir resultados expressivos principalmente utilizando *LSTM* (*Long Short Term Memory*) e *random embedding* como parâmetros, sendo GBDT (*Gradient Boosted Decision Trees)* o classificador obtendo precisão de 93%. O experimento classificou textos de usuários do Twitter marcados como sexistas, racistas ou neutro.

6. Métodos utilizados

O modelo vetorial [11] é um método para calcular a similaridade de um texto com um documento, no caso o texto seria cada entrada a ser avaliada e o documento seria um texto na base de dados já pré avaliado como ofensivo ou não. O modelo vetorial clássico calcula a frequência do termo (TF de *term frequency*), o índice de frequência (IDF de *inverse document frequency*), e com eles é calculado o peso de uma palavra em um documento. Com o TF, IDF e os pesos calculamos a similaridade de um texto com os documentos da base, neste projeto comparamos com duas bases, uma contendo discurso de ódio e outra não, posteriormente comparamos qual base possui a maior similaridade com o texto. Esse método pode ser aprimorado para o uso de bi-gramas ou tri-gramas, ou utilizando outro método também de cálculo de similaridade como o *language models* e outros.

Uma forma de representar a base de dados é com um grafo de ocorrências, em que cada vértice é uma palavra e as arestas a quantidade de vezes que as palavras ocorrem no mesmo documento. Com isso conseguimos analisar o contexto do documento para cada palavra enquanto no modelo vetorial apenas é possível fazer inferências estatísticas. Algumas formas de quantificar a similaridade podem ser utilizadas, como a soma dos pesos das arestas para cada entrada, a soma dos menores caminhos entre as palavras. Podemos também utilizar as cliques do grafo, que são sub grafos que todos os vértices estão ligados, com

isso as palavras que não são relevantes em relação à entrada são removidas, obtendo o grafo somente com as palavras importantes fortemente ligadas.

A melhor forma de solucionar o problema, atualmente, é com o uso de *machine learning*, com isso utilizamos o método *word embedding*, que é baseado em redes neurais. O modelo *word2vec* é um dos mais utilizados, para modelar a base e o *CBOW* como modelo de linguagem neural. No *word2vec* é feito a análise das palavras adjacentes a um pivô que percorre a base inteira, com isso as palavras vizinhas são utilizadas para alimentar a camada escondida da rede e com ela é feito o *CBOW* para prever a palavra baseada no contexto em que foi analisado. No final é obtido quais as possíveis palavras que aparecem juntas com cada uma, esse resultado é usado como parâmetro para o classificador. Uma outra forma de solucionar o problema é com o *paragraph2vec*, que analisa todo o documento ao invés de uma pequena vizinhança para cada palavra, e pelos testes de [6] é possível obter resultados melhores do que o *word2vec*. Uma variação do modelo *word2vec* é o *FastText* que é mais rápida comparada aos outros modelos e alcança uma boa precisão.

Para classificar os resultados será utilizado o *SVM*, uma vez que ele é resistente à ruídos de texto e é um bom classificador no problema da categorização de textos conforme foi provado em [9]. O classificador supervisionado será capaz de avaliar com base nos parâmetros se um texto contém ou não discurso de ódio. Uma outra alternativa seria o *GBDT* que em [6] consegue os melhores resultados, conforme já mencionado.

7. Resultados e discussões

O modelo vetorial clássico foi implementado com alterações no *tf* e no *idf:*

$$idf(t) = log(\frac{N - Nt + 0{,}5}{Nt + 0{,}5})$$

Sendo N o número de documentos da coleção e N_t a quantidade de vezes que o documento t aparece na coleção.

$$tf(t, d) = 1 + log(ft, d)$$

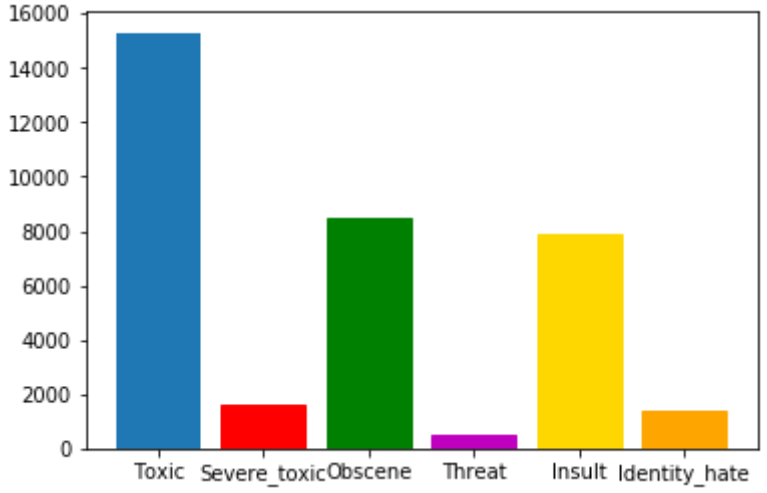
Sendo $f_{t,d}$ a frequência em que o termo t aparece no documento d. A fórmula do peso de cada palavra é idêntica à forma clássica da mesma forma que o cálculo da similaridade. A base foi retirada do *contest Kaggle Detecting Insult in Social Commentary* e foi divida em duas partes sendo uma contendo discurso de ódio e a outra não, e para uma entrada é calculada a similaridade em relação às duas bases. Com as similaridades calculadas é realizada uma média ponderada dos primeiros 25 documentos com mais similaridade, a base que possuir a maior média definirá se a entrada contém ou não discurso de ódio. Dessa forma foi possível alcançar uma precisão de 65%.

O principal problema do modelo vetorial, por ser um modelo estatístico, é o de não analisar o contexto de cada palavra na coleção fazendo com que sejam obtidos muitos resultados errados. Para isso foi utilizado a modelagem em grafo de ocorrências, dessa forma cada vértice é uma palavra e o peso das arestas o número de vezes que ocorrem juntas. Podemos agora manipular o grafo de forma que procuraremos analisar a semelhança de uma entrada com os grafos que representam a base, analisando fortemente o contexto em que as palavras estão envolvidas. Será necessário com a base montar dois grafos de ocorrências, o grafo que possui apenas as entradas da base que possuem discurso de ódio e outro contendo as entradas sem

discurso de ódio. Para cada nova instância a ser avaliada será criado um grafo que será analisado comparando com os grafos de base. Para comparar os grafos é feito a soma dos pesos das arestas que ligam entre si dessa forma é possível quantificar a conectividade dos grafos.

O uso do grafo juntamente com o modelo vetorial como parâmetros para o *SVM* foram classificados utilizando o RBF *Kernel*, o que proporcionou 75% de precisão

Também foram realizadas experimentações na base do *contest Kaggle Toxic Comment Classification*, com 153164 instâncias de treino e a validação realizada via submissão no site. O problema é de classificação multi-classe, sendo as seguintes classes de comentários: *Toxic, Severe Toxic, Obscene, Threat, Insult, Identity Hate* ou não ser nenhuma das citadas, que chamaremos de classe *None.* A base é desbalanceada, sendo sua maior parte as instâncias da classe *None*. Para analisarmos melhor as outras classes, removemos a classe *None* e obtemos o seguinte gráfico:



O primeiro teste foi realizado utilizando o método *FastText* com dimensão de tamanho 100, tamanho da janela de contexto sendo 2, softmax como função de perda e realizando 50 épocas de treino, com isso foi obtido uma precisão segundo o site de 92%. Uma forma de melhorar esse resultado foi utilizando os vetores já treinados da *Wikipedia* disponibilizados no site do *FastText*. Como os vetores já treinados tinham uma dimensão de tamanho 300, foi necessário mudar esse parâmetro para se adequar ao treino, e com isso a precisão obtida pelo site é de 93%.

A fim de descobrir se o modelo se comporta bem com os comentários com mensagens subliminares e ironias, que detectamos como as instâncias que o modelo mais errava, testamos o modelo na base *Sarcasm on Reddit.* A base contém um milhão de comentários do *Reddit* e foi divida dois terços para treino e um terço para teste. A melhor precisão obtida utilizando o modelo *FastText* foi de 61% com os parâmetros similares aos utilizados anteriormente na base do *contest Kaggle Toxic Comment Classification* .

8. Conclusões

O problema da detecção de discurso de ódio é complicado por vários aspectos. As poucas bases existentes não são numerosas a ponto de validar completamente o experimento, devido à rápida

transformação do linguajar é difícil identificar novas formas de insulto, principalmente nas redes sociais. O fato do problema ser adversarial dificulta bastante a aprendizagem dos modelos principalmente no que constatamos em mensagens subliminares e ironia. A ausência de uma base brasileira significativa para experimentações não permitiu os estudos com os modelos treinados para comentários em português. Das soluções estudadas e utilizadas o método *FastText* se destaca por obter ótimos resultados além de ser muito eficiente em relação ao tempo de execução. O fato dos modelos de *embedding* analisarem apenas as ocorrências de palavras próximas impede um entendimento da frase como um todo, o que gera muitos falsos negativos.

9. Cronograma

Nº	Descrição	Ago 2017	Set	Out	Nov	Dez	Jan 2018	Fev	Mar	Abr	Mai	Jun	Jul
1	Revisão Bibliográfica	X	X	X									
2	Download e processamento da Base de dados de referência		X										
3	Desenvolvimento de um método de detecção de discurso de ódio baseado unicamente no texto do comentário e do post original			X	X	X	X	X					
4	Desenvolvimento de um método de detecção de discurso de ódio baseado também em características extra-texto						X	X	X				
5	Avaliação de resultados				X	X			X	X			
6	Coleta de comentários de sites nacionais												
7	Avaliação da adequação dos métodos propostos para conteúdo em português												
8	Elaboração do relatório final												X

10. Referências

[1] Awan, Imran. "Islamophobia and Twitter: A typology of online hate against Muslims on social media." *Policy & Internet* 6.2 (2014): 133-150.

[2] Yarowsky, David. "Unsupervised word sense disambiguation rivaling supervised methods." *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995.

[3] Sood, Sara Owsley, Judd Antin, and Elizabeth F. Churchill. "Using Crowdsourcing to Improve Profanity Detection." *AAAI Spring Symposium: Wisdom of the Crowd*. Vol. 12. 2012.

[4] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." *International Conference on Machine Learning*. 2014.

[5] Nobata, Chikashi, et al. "Abusive language detection in online user content." *Proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2016.

[6] Badjatiya, Pinkesh, et al. "Deep learning for hate speech detection in tweets." *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017.

[7] Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.

[8] Martineau, Justin, and Tim Finin. "Delta TFIDF: An Improved Feature Space for Sentiment Analysis." *Icwsm* 9 (2009): 106.

[9] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." *European conference on machine learning*. Springer, Berlin, Heidelberg, 1998.

[10] Gitari, Njagi Dennis, et al. "A lexicon-based approach for hate speech detection." *International Journal of Multimedia and Ubiquitous Engineering* 10.4 (2015): 215-230.

[11] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.