Classificando discurso de ódio em um dataset com tweets com palavras e frases ofensivas

Giancarlo Vanoni Ruggiero[a]

[a]*Engenharia da Computação, INSPER*

Prof. Tiago Fernandes Tavares

1. Dataset

Os dados foram coletados por uma equipe composta pela *Cornell University* em conjunto com o *Qatar Computing Research Institute*. O dataset contém 25k tweets que contém frases e palavras de discurso de ódio da *Hatebase.org*. Os dados foram coletados através da API do twitter e selecionados aleatoriamente do total. Foram classificados manualmente por trabalhadores da *CrowdFlower*, separando em: discurso de ódio, ofensivo e nenhum dos dois.

O trabalho desenvolvido pela equipe tem como intuito identificar o discurso de ódio e diferenciá-lo de uma linguagem ofensiva, algo que muitas vezes é confundido. Apesar de nos EUA o discurso de ódio ser protegido pela primeira emenda sobre a justificativa da liberdade de expressão, em alguns países, como Canadá, França e Reino Unido a pessoa pode até ser presa caso seja condenada. [1]

O dataset é desbalancedo. Pois a quantidade de tweets com linguagem ofensiva é consideravelmente maior que a de outras categorias.

2. Classification pipeline

O pipeline de classificação foi dividido em 3 etapas: a primeira em que ocorre a limpeza dos dados, a segunda a vetorização, utilizando o *TfidfVectorizer*[6] e por fim a etapa de treinamento, em que foram utilizados dois modelos: *LogisticRegressor* e *RandomForest*.

Para a etapa do pré-processamento, foram removidas as urls, menções, caracteres especiais, a sigla RT, números e *stopwords*, pois por não possuírem um significado no contexto do tema, iriam prejudicar a análise. Afinal algo que aparece múltiplas vezes mas não possui valor informativo pode causar interpretações equívocadas e diminuir o desempenho do modelo.

Após a limpeza dos dados foi feita a lematização, que consiste em mapear as palavras flexionadas, como plurais ou conjugadas e reduzi-las para sua forma canônica. Por exemplo: transformaria correu, corria, correria em correr. Diferente do *stemming* que corta as palavras para obter suas raízes, criando palavras que não possuem um significado real, a lematização preserva o sentido original da palavra.

Após a limpeza dos dados e a lematização, é feita a vetorização utilizando o *TfidfVectorizer*[6]. Essa técnica é muito similar ao *Bag of Words(BoW)*[3] com a diferença de que, enquanto o BoW representa cada documento como uma matriz de contagem de palavras, o TFIDF utiliza não somente a frequência de cada termo no documento mas também o inverso da frequência de documentos. Isso permite que se tenha uma visão da coleção como um todo.

Entretanto ao utilizar modelos que avaliam pela frequência de palavras, como o BoW, ele não entende o contexto. Por exemplo, no caso de discurso de ódio, palavras como *b*tch* aparecem com muita frequência, logo serão classificadas como tal, porém também são utilizadas para citar letras de rap, o que não necessariamente é um discurso de ódio. Algo que poderia atrapalhar o classificador, seria utilizar sinônimos ou variações da palavra de uma maneira que seja interpretada como uma palavra nova, como ao colocar números no local de algumas letras.

E por fim chega a etapa de classificação. Foram utilizados dois modelos para se fazer uma comparação em qual performa melhor, sendo eles: *LogisticRegressor* [4] e *RandomForest* [5].

3. Evaluation

A validação de cada modelo foi feita da seguinte forma: para cada classificador o dataset foi embaralhado 30 vezes e treinado novamente em cada uma delas. Após isso foi calculada a acurácia utilizando o método *balance accuracy score* [2] do Scikit Learn, que é utilizado para calcular a acurácia em datasets desbalanceados. Após isso foi tirada uma média simples de todas as iterações.

Para o modelo de Regressão Logística [4], apenas o paramêtro *class weight* foi modificado para *balanced*, que modifica os pesos de cada classe inversamente proporcionais a sua frequência. Ao executar o modelo foi possível obter uma acurácia média de 0.81.

Enquanto para o modelo de *Random Forest*[5] foi obtida uma acurácia muito similar de 0.81. Para esse modelo foi modificado o parâmetro *min sample leaf* que suaviza o modelo e novamente o parâmetro *class weight*, só que para *balanced subsample*, que atua de forma similar, com a diferença que eles passam a ser calculados na inicialização de cada árvore. Embora não influencie na acurácia do treinamento, o parâmetro *n jobs* foi modificado para -1 para utilizar todos os núcleos do processador.

As palavras que foram mais proeminentes para Regressão Logística[4]: *fa*got, ni*ger, ni*ga*, e para *Random Forest*[5]: *bi*ch, h*e, fa*got*. Essas palavras são utilizadas, na maioria das vezes, de maneira pejorativa para ofender minorias, o que faz sentido, afinal o dataset contém tweets com discursos ofensivos e de ódio e para classificá-los é natural que essas palavras apareçam com mais frequência e tenham mais importância.

4. Dataset size

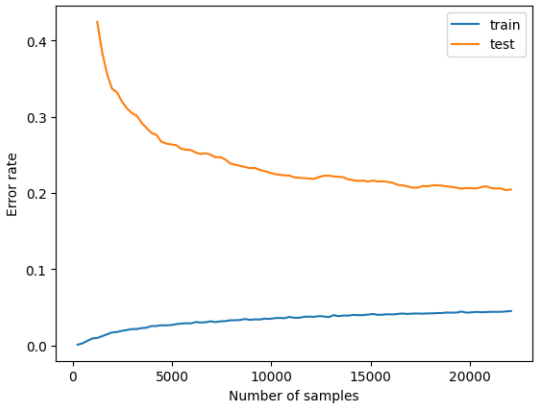

Figure 1. Taxa de erro de acordo com o tamanho do dataset para treino e teste

Como é possível observar a Figura 1, ao chegar entre 10k e 15k amostras o ganho é muito pequeno, logo pode-se afirmar que o dataset por possuir 25k é maior que o necessário. Entretanto, como para formar o dataset foram coletados 85.4 milhões de tweets [1], o dataset teria como crescer muito, embora só gastaria armazenamento e processamento e não traria um ganho real.

5. Topic analysis

Ao utilizar um classificador de tópicos, separando em 3 tópicos, ficou com uma acurácia média um pouco menor que os outros classificadores, sendo 0.67, enquanto os outros ficaram com 0.8. Ao olhar

cada tópico individualmente, o primeiro tópico ficou com 0.58, o segundo 0.67 e o terceiro 0.76, indicando que o terceiro tópico apresenta uma maior facilidade para o classificador identificá-lo.

References

[1] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language", in *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ser. ICWSM '17, Montreal, Canada, 2017, pp. 512–515.

[2] 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html.

[3] W. Contributors, *Bag-of-words model*, Aug. 2024. [Online]. Available: https://en.wikipedia.org/wiki/Bag-of-words_model.

[4] 2024. [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html.

[5] 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[6] 2024. [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.