Classificação de sentimento em tweets sobre política

Arthur Galdino Dangoni[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brazil

arthurdangoni@discente.ufg.br

Abstract. *The ranking of sentiments in tweets about politics is important to obtain a more detailed public opinion. In addition to exercising the knowledge acquired in the field of natural language processing (NLP) in a real problem. This work proposes the creation of machine learning models with the purpose of helping in political decisions and obtaining the first place in the competition. Several machine learning models were tested with several different preprocesssing. Based on the tests performed, the model that obtained the best result was the SGDClassifier.*

Resumo. *A classificação de sentimentos em tweets sobre política é importante para obter uma opinião do público mais detalhada. Além de exercitar os conhecimentos adquiridos na matéria de processamento de linguagem natural (PLN) em um problema real. Este trabalho propõe a criação de modelos de machine learning com o propósito de ajudar em decisões políticas e obter o primeiro lugar na competição. Foram testados diversos modelos de machine learning com diversos pré-processamentos diferentes. Com base nos testes realizados o modelo que obteve o melhor resultado foi o SGDClassifier.*

1. Introdução

A classificação de sentimentos é uma área que procura determinar o tipo de emoção a partir de um texto. Essa emoção normalmente é classificada como positiva, negativa ou neutra porém podem ter outras classificações como raiva e tristeza. Existem muitas aplicações em diversos domíminios diferentes na qual se procura identificar a emoção do público em relação a um certo produto, decisão política, econômica ou até mesmo para verificar o bem-estar social (ALZU'BI *et al*., 2019). Este estudo é específico em realizar a classificação de sentimento de tweets em um contexto político. Portanto permite identificar as emoções das pessoas em relação as leis propostas, ajudando nas tomadas de decisões políticas.

Este artigo propõe a utilização de modelos de machine learning para classificar os sentimentos de tweets em um cenário político. Para obter uma opinião do público mais detalhada além de obter o primeiro lugar na 1ª Competição de processamento de linguagem natural (PLN).

2. Trabalhos Relacionados

Alzu'bi *et al* (2019) propõe a classificação de emoções em tweets na língua árabe. Esse estudo busca fazer essa classificação porque não existem muitos estudos para a língua

árabe. As emoções classificadas são de surpresa, desgosto, raiva, medo, tristeza e alegria. É realizado um pré-processamento dos dados em que é feito a remoção de stop words, hashtags, letras frequentes e por fim faz a normalização dos dados. Os métodos de vetorização utilizados foram o bag of words(BOW) e o term frequency inverse document frequency (TF-IDF). Enquanto os modelos de machine learning implementados foram as Decision Trees , K-nearest neighbor(KNN) e Random Forest. A partir dos testes realizados o melhor modelo foi o Random Forest. Esse estudo é interessante, pois realizada alguns pré-processamentos e técnicas de vetorização aprendidas na disciplica PLN.

Brynielsson *et al.* (2014) propõe a criação de um dataset a partir de tweets em comunicados de alerta em tempos de crise. A partir desse dataset é feito a classificação de emoções em raiva, medo, emoções positivas e outras emoções. Alguns pré-processamentos feitos foram a remoção de stop words e o stemming. Foram utilizados modelos padrões de machine learning como o Naive Bayes (NB) e Support Vector Machine(SVM). Os resultados mostraram que o SVM teve uma acurácia melhor. Por fim é um estudo importante porque mostra como a população em geral se sente em uma crise.

Vora *et al.* (2017) propõe a classificação de emoção em tweets usando o word embedding para a vetorização dos dados. São classificadas em 4 emoções sendo elas felicidade, tristeza, raiva e surpresa. Utiliza um pré-processamento contendo diversas técnicas sendo elas remoção de dados duplicados, urls, menções, 'RT' , hashtags, emojis, repetição de letras, pontuações. Além de converter todos os documentos em minúsculas e utilizar somente os documentos que estão na língua inglesa. Após isso foi feito o word embedding pelo Word2vec e Glove. Por fim esses dados foram classificados pelo modelo Random Forest. Esse modelo mostra que a utilização de word embedding pode ser uma alternativa para a vetorização ao inves de usar somente o BOW e o TF-IDF.

3. Metodologia

O conjunto de dados foi disponibilizado pela professora Nádia Félix Felipe da Silva. Esse conjunto de dados é composto por textos do twitter relacionados a política. Além de textos o conjunto de dados também possui a localização do usuário, o nome, o nome visualizado por outros usuários e a contagem de retweets do documento. Esse conjunto é dividido em conjunto de treino que possui 6559 amostras e conjunto de teste que possui 1640 amostras, sendo ao todo 8199 amostras.

Os experimentos foram realizados no Google Colaboratory que é um ambiente de notebooks. Nesse ambiente foram realizados a codificação das etapas para resolver o problema. Todos os modelos seguiram essas etapas que foram a leitura dos dados a partir do dataset de treino, a divisão dos dados em treino e teste, a engenharia de features, o pré-processamento dos dados, treinamento do modelo, predição e por fim a acurácia do modelo. A acurácia dos modelos foram verificadas testando todas as combinações diferentes de pré-processamento.

Após esses experimentos no conjunto de treino. Foram selecionados os melhores modelos e realizadas essas etapas em todo o dataset e a predição dos modelos no conjunto de teste. Por fim essas predições foram salvas em arquivos '.csv' para a submissão no kaggle. O kaggle é um site utilizado para competições de machine learning onde foi realizado a 1ª Competição de PLN da Universidade Federal de Goiás (UFG) para a disciplina de PLN.

4. Experimentos

Nesta seção apresentamos as etapas realizadas para fazer os experimentos. Primeiro foi realizado a engenharia de features onde foram testados métodos para vetorização na seção 4.1. A seguir o pré-processamento no qual é mostrado quais técnicas foram utilizadas para fazer essa etapa na seção 4.2. Em seguida foram utilizados modelos de machine learning para realizar essa classificação na seção 4.3. Após isso se tem os resultados gerados a partir dos modelos implementados na seção 4.4. Por fim as conclusões são mostradas as contribuições deste estudo na seção 4.5.

4.1. Engenharia de Features

Nessa parte foram testados dois métodos sendo eles o BOW e o TF-IDF. Como requisito para fazer essa vetorização é necessário fazer a tokenização de todos os documentos presentes no corpus. Essa tokenização foi feita utilizando o casual tokenize.

O BOW é um modelo no qual se tem todas as sentenças representadas pelas colunas e as linhas são os documentos. Para relacionar as colunas e linhas foi utilizado o term frequency(TF) no qual é contado o número de vezes em que cada sentença está contida no documento. Enquanto o TF-IDF estabelece uma ponderação entre os tokens em um documento e sua relação com todo o corpus. Para fazer essa ponderação é necessário a multiplicação do TF pelo inverse document frequency (IDF) como é mostrado na equação 1 em que 't' representa o termo e 'd' o documento.

O TF calcula a quantidade de vezes que um termo apareceu no documento como é mostrado na equação 2. Enquanto o IDF mede o quanto o termo é frequente em todo o corpus como é mostrado na equação 3. Em que 'N' representa o número total de documentos no corpus dividido pelo o total de documentos que contém 't' em que 'D' representa o corpus.

$$TFIDF(t,d) = TF(t,d) * IDF(t) \tag{1}$$

$$TF(t,d) = freq(t,d) \tag{2}$$

$$IDF = 1 + log(\frac{N}{count(d \in D : t \in d)}) \tag{3}$$

4.2. Pré-Processamento

Na parte de pré-processamentos são utilizadas algumas técnicas para fazer a limpeza dos dados. Neste estudo foram implementados a retirada de stop words, pontuações e links além de tornar todas as sentenças em minúsculas. As stop words são palavras que aparecem diversas vezes por todo o corpus porém não carregam informação útil para caracterizar o documento, essa lista de palavras foi feita fazendo a união das stop words dos idiomas português, ingles e espanhol que a própria biblioteca da 'nltk' fornece. Também foi reduzido o número de letras repetidas e palavras que começam com '@'. Além disso a divisão do conjunto de treino foram realizadas em 80 porcento em treino e 20 porcento em teste.

4.3. Algoritmos de Classificação Avaliados

Os algoritmos de classificação avaliados foram o Logistic Regression, SVM, Multinomial Naive Bayes, KNN, Decision Trees, SGDClassifier e Ensemble dos modelos KNN, SVM, Decision Tree e Logistic Regression. Em todos esses modelos foram feitos a otimização de alguns hiperparâmetros pelo método Grid Search. Entretanto por causa do modelo SGDClassifier ter muitos hiperparâmetros a otimização foi mais demorada e somente foi possível testar o modelo otimizado no melhor cenário de pré-processamento que é a utilização do casual tokenize sem nenhum parâmetro adicional e a remoção de stop words. Esses modelos foram avaliados no conjunto de treino utilizando a métrica de acurácia no qual é calculado de acordo com a equação 4. Em que 'NC' é o número de predições corretas e 'NP' é número total de predições.

$$Acurácia = \frac{NC}{NP} \tag{4}$$

4.4. Resultados

Foi criado uma tabela composta pelos modelos e seus respectivos pré-processamentos e as acurácias obtidas no conjunto de treino, no resultado público do kaggle e no privado como é mostrado na tabela 1. Nessa tabela foi considerado somente o BOW, pois estava obtendo melhores resultados comparado com o TF-IDF. Os modelos que foram submetidos foram os que obtiveram os melhores resultados no conjunto de treino.

Modelo + Pré-Processamentos	Conjunto de Treino	Privado	Público
SVM	0.95960	0.96112	0.97865
Logistic Regression + LSPMR	0.95731	0.95426	0.98475
KNN + LSPMR	0.95731	0.94207	0.96951
SVM + LSPMR	0.95426	0.96265	0.98475
Decision Tree + LSPMR	0.95121	0.96036	0.97256
Ensemble + LSPMR	0.95121	0.96265	0.98170
SGDClassifier + SPMR	**0.96036**	0.96493	0.98170
SGDClassifier + LSPMR	0.95151	0.96570	0.98170
SGDClassifier + S	0.95884	0.96112	0.98780
SGDClassifier Otimizado + S	0.95579	**0.96646**	**0.99085**
SVM + S	0.95807	0.96036	**0.99085**

Table 1. L: Remoção de links, S: Stop words, P: Pontuação, M: Transformação em todas sentenças em minúsculas, R: Redução de letras repetidas em palavras e remoção de palavras iniciadas com '@'

A partir da tabela podemos chegar a conclusão que uma baixa acurácia no conjunto de treino não pode sigfinicar um resultado ruim no conjunto de teste, pois no modelo "SGDClassifier + S" obteve um resultado abaixo de outros modelos no conjunto de treino e um bom no conjunto de teste. O resultado público e privado do kaggle nessa competição pode ser muito diferente, porque o resultado público foi avaliado levando em conta somente 20 por cento do conjunto de teste enquanto o privado contabilizou 80 por cento. Portanto o modelo pode ter "sorte" nos primeiros dados do conjunto de teste e ter uma

acurácia elevada, porém considerando uma maior parte dele que é o resultado privado, a acurácia pode cair. Por fim levando em consideração que o resultado privado foi o resultado definitivo da competição, o melhor modelo obtido foi o SGDClassifier com a otimização dos hiperparâmetros.

4.5. Conclusões

Este trabalho apresentou diversos modelos e pré-processamentos que poderiam ser usados para solucionar o problema de classificar emoções em tweets sobre política com o objetivo de ajudar em decisões políticas. Além de obter o primeiro lugar na 1ª Competição de PLN da UFG para a disciplina de PLN. Neste trabalho a partir dos testes realizados foi obtido o 8ª lugar na competição com o modelo SGDClassifier que obteve a acurácia de 96.6 porcento. Por fim foram aplicados todos os conhecimentos obtidos na disciplina até o momento e pode servir como um guia inicial para pessoas que estão iniciando na área de PLN.

Para trabalhos futuros é proposto implementar uma representação das palavras por meio de word embedding e implementar arquiteturas transformer para verificar uma possível melhoria nos resultados.

5. Referências

[Alzu'bi et al. 2019] [Brynielsson et al. 2014] [Vora et al. 2017]

References

Alzu'bi, S., Badarneh, O., Hawashin, B., Al-Ayyoub, M., Alhindawi, N., and Jararweh, Y. (2019). Multi-label emotion classification for arabic tweets. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 499–504. IEEE.

Brynielsson, J., Johansson, F., Jonsson, C., and Westling, A. (2014). Emotion classification of social media posts for estimating people's reactions to communicated alert messages during crises. *Security Informatics*, 3(1):1–11.

Vora, P., Khara, M., and Kelkar, K. (2017). Classification of tweets based on emotions using word embedding and random forest classifiers. *International Journal of Computer Applications*, 178(3):1–7.