DESENVOLVIMENTO DE ALGORITMO VOLTADO PARA A AVALIAÇÃO DE OPINIÃO/POSICIONAMENTO EM RELAÇÃO A NOTÍCIAS NA INTERNET

Ricardo Augusto Fernandes Júnior

Luciano Antônio Digiampietri

Escola de Artes, Ciências e Humanidades / Universidade de São Paulo

rafernandesjr@usp.br; digiampietri@usp.br

Objetivos

Com a crescente facilidade em se acessar e compartilhar informações no Brasil [1], é perceptível o aumento da difusão de notícias falsas (*fake news*), algo muito presente nos dias de hoje. Dado esse contexto, o objetivo deste trabalho é desenvolver um algoritmo de mineração de opiniões, de forma a classificar a opinião de comentários de notícias em *concorda* e *discorda*. Em termos de algoritmo, partiu-se do estudo dos códigos que obtiveram os melhores resultados na competição *Fake News Challenge* [2].

Métodos e Procedimentos

A principal linguagem de programação utilizada foi Python. Para a realização dos experimentos, foi montada uma base de dados a partir da coleta de tweets da página do G1, as respostas de cada tweet e os dados da notícia compartilhada. Essa coleta foi feita utilizando o *Tweepy* (uma API disponibilizada pelo Twitter para acessar sua base de tweets), e o *Scrapy* (um framework voltado para a extração de dados de páginas web).

No pré-processamento dos dados foi aplicado um *stemmer* (radicalizador), e as *stopwords* foram removidas. Após a limpeza, restaram 208 notícias e 4032 respostas.

A principal métrica de ranqueamento utilizada foi o *TF*IDF*, seguindo a seguinte fórmula base. Para cada par contendo um trecho *T*, um documento *Z* e um corpus *O*, foi aplicada a equação 1, onde *p* representa cada palavra pertencente a *T*, *TFp* é a frequência relativa de *p* em *Z*, e *IDFp* é o inverso da frequência relativa de *p* no corpus *O* formado por todos os outros documentos de um conjunto (sem o documento *Z*) [3].

1. $$\sum_{p \in T} \log\left(1 + TF_p \times IDF_p\right)$$

Outra representação utilizada foi um modelo *bag-of-words*. Os classificadores utilizados, da biblioteca *scikit-learn* do Python, foram o *Random Forest* e o *Naive Bayes*.

Resultados

Para obtenção dos resultados, foram feitas validações cruzadas usando cinco subconjuntos (folds). O melhor resultado com *Random Forest* (utilizando *TF*IDF*) atingiu uma acurácia média de 65,38%, enquanto o com *Naive Bayes* (utilizando o *bag-of-words*) atingiu uma acurácia média de 65,01%.

Foi feita uma abordagem diferente com *Naive Bayes* utilizando um *TF*IDF* vetorizado como métrica, o que gerou uma acurácia média de 65,07%.

Conclusões

Dado o baseline de 54,22%, os resultados apresentados se mostraram promissores. Uma abordagem futura seria considerar também o peso semântico das palavras, utilizando técnicas como o *Word2Vec*.

Referências Bibliográficas

[1] PNAD Contínua TIC 2017. IGBE (2018).

[2] Fake News Challenge. FNC (2017).

[3] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1):11–21.