UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO DE DESENVOLVIMENTO DO MÓDULO DE ANÁLISE DE SENTIMENTOS

Bruno Queiroz Santos
William dos Santos Abreu

LAVRAS - MG
16 de junho de 2019

1. Introdução

Algoritmos que conseguem extrair sentimentos de textos estão sendo muito utilizados em diversas áreas do mercado, devido à sua capacidade de automação no processo de obtenção de feedback dos usuários, o que pode auxiliar na melhoria de produtos online e consequentemente aumentar as vendas.

O algoritmo apresentado neste relatório envolve a análise de sentimentos nos comentários de produtos do website BuscaPé, utilizando a técnica de 'Machine Learning' para ajudar a classificar um bom comentário de um mau comentário. Para isso, foi usado como parâmetro a quantidade de estrelas (score) que as pessoas marcam como nota de avaliação e os comentários deixados sobre cada produto. O score varia de 0 a 5, sendo zero um sentimento mais negativo e cinco o mais positivo.

A partir do dataset de avaliações de produtos do BuscaPé, o objetivo deste trabalho é construir um modelo capaz de classificar sentenças quanto ao sentimento expresso por elas, tendo como medida um valor em escala inteira de 0 a 5, conforme o score de cada dado do dataset.

2. Método

A seguir, está descrito sequencialmente a metodologia utilizada na construção do modelo classificador de sentimentos.

2.1. Carregamento do dataset

O dataset (diretório "trainset") é composto por arquivos XML, cujos atributos de interesse são as tags "stars", "pros", "cons" e "opinion". A primeira tarefa é fazer o parse dos arquivos e carregar os dados de interesse em um dicionário Python, estrutura de dados ideal para a tarefa de carregamento devido sua versatilidade de manipulação de um conjunto de dados extenso.

2.2. Normalização das sentenças

Após o carregamento dos arquivos XML em dicionário, deve ser aplicada uma rotina de normalização de texto, para deixar as sentenças somente com aquilo que é realmente necessário quanto ao conteúdo como também manter a forma padronizada. Dessa forma, são seguidas os seguintes procedimentos:

- Converter todos os textos para caixa baixa
- Remover todos os sinais de pontuação, quebras de linha e etc
- Remover stopwords

Vale ressaltar que, por padrão, o NLTK considera a palavra “não” como stopword, porém ela tem sentido quanto à tarefa de análise de sentimentos. Por isso, “não” não pode ser considerado stopword e não pode ser removida dos textos no processo de normalização.

2.3. Divisão entre trainset e testset

Para se ter uma medida de acurácia do modelo construído, o dataset é dividido em dois conjuntos disjuntos, chamado de trainset aquele que contém os dados utilizados no treinamento do algoritmo de machine learning e de testset aquele com os dados utilizados para calcular a acurácia do modelo treinado, uma vez que já possuem seu valor conhecido, funcionando como gabarito no processo de teste.

A abordagem é dividir aleatoriamente 75% dos dados dataset para o trainset e 25% para o testset, a fim de obter um valor mais confiável para o teste de acurácia do modelo criado por treinamento supervisionado.

2.4. Vetor TF-IDF

Para criação do modelo classificador, é necessário mapear as entradas (neste caso, texto) em representações numéricas. A abordagem é utilizar do ***term frequency–inverse document frequency***, que representa estatisticamente o quão importante é uma palavra dentro do documento, fazendo assim o mapeamento texto-número. A vantagem dessa abordagem é que ela coloca peso em termos de maior frequência, que acabam determinando o sentimento de uma sentença.

Apesar de ter carregado os “pros” e “cons”, eles não foram utilizados no modelo, somente foi usado a “opinion” como entrada para ser mapeada.

2.5. Regressão Logística

Os dados transformados em números, servem de entrada para a construção do modelo classificador utilizando regressão logística. A regressão logística é um modelo estatístico de classificação, não um modelo de aprendizado de máquina. Sua escolha se deve

ao fato de, durante teste de desenvolvimento, o modelo estatístico apresentar maior acurácia na tarefa.

3. Resultados

O ‘trainset’ fornecido para a realização da tarefa trouxe bons resultados no treinamento do modelo classificador, sendo que o algoritmo ‘Logistic Regression’ foi o que trouxe as melhores análises de sentimento nos poucos textos que foram testados.

Primeiramente, o processo de normalização estava removendo o “não” das sentenças. Concomitante a isso, a partição dos dados de treino e de teste estava meio-a-meio. Tudo isso em conjunto estava dando uma acurácia por volta de 50%.

Notamos que quanto mais dados de treinamento foram fornecidos para o algoritmo, maior acurácia obtivemos. Assim foi feita a modificação na partição dos conjuntos, deixando com aproximadamente 75% de dados para treinamento e 25% de dados para teste. Também a biblioteca de normalização em ***nlputils*** foi adaptada para não excluir “não” na remoção de stopwords. Conseguimos uma acurácia de 56,7% no último teste feito após essas modificações.