CSE 6363 - Machine Learning

Project Review 3

Group 10

Love Jeswani (1001754568)

Navya Sogi (1001753085)

Sushmitha Srinath Kenkare (1001738893)

Project Topic: Sentiment Analysis for Movie Reviews using Natural Language Processing.

Data Preprocessing:

The reviews dataset has train and test datasets, each with 25,000 reviews that includes both positive and negative reviews.

The reviews that were present in multiple text files have been merged together as 'full_-train.txt' and 'full_test.txt' files.

The data has been cleaned by removing punctuation marks, HTML tags and the words are converted into lower-case letters for easy processing.

Stop words have been removed as mentioned in the paper cited in references.

Extra work apart from that given in reference paper:

Vectorization has been implemented to make sure the algorithm works on multiple values at a time instead of working on one value.

We have implemented normalization techniques like stemming and lemmatization for data pre-processing.

Stemming: Stemming involves removal of affixes (beginning or the end of a word) to extract the base form of the word. For example, consider the words, 'eating', 'eaten' and 'eats'. The stem for the above-mentioned words is 'eat'. This is obtained by removing the last few letters of the words. There are multiple algorithms available to determine how many letters need to be chopped off at the end. But the algorithms do not know the meaning of the word. Search engines use stemming for indexing the words to reduce the size of the index and to increase the word retrieval rate.

Lemmatization: Lemmatization is similar to stemming. The output of this technique is referred to as a 'lemma'. But the difference is that the algorithms have the knowledge of meaning of words in lemmatization. We can say that the algorithms refer to a dictionary to understand the meaning of the word before reducing it to a root word or lemma. For example, the algorithm would identify that the word 'better' is derived from the lemma 'good'.

Feature extraction methods like Bag of Words, N-Gram, Word Counts and TF-IDF are used.

Bag of Words: This model calculates the frequency of the word occurrences in a text file. The order of the words does not hold importance in Bag of Words, but the model only cares about what words appear in the text.

N-Gram: it is a sequence of N-words in a sentence. N is an integer which stands for the number of words in the sequence. When N = 1, N=2, N=3, it is referred to as uni-gram, bi-gram and tri-gram respectively. N-gram is used because unlike in bag of words, the order in which the words appear is important. For example, it is a good idea to consider bigrams like "New York" instead of splitting them into individual words like "New" and "York".

Word Counts: We can just note the number of times a particular a given word appears in a text file instead of finding if a word appears in the file or not. This gives a model much more predictive power.

Term Frequency, Inverse Document Frequency (TF-IDF): This is the most popular way to represent documents as feature vectors. TF-IDF measures how important a word is with respect to a specific text file or document. Term Frequency measures the counts of each word in a document out of all the words in the same document. Inverse Document Frequency measures how important a word is by considering the frequency of the word in the entire corpus.

Accuracy levels of models such as logistic regression, Naïve Bayes, SVM, Decision Tree, Linear SVC are calculated for the above-mentioned feature extraction methods.

- **Did you do the same that was done in the references?**

  Removed punctuation marks, converted text into lower-case and removed stop words as mentioned in the paper cited in references.

- **Any enhancement to the work that was done in references?**

  Added normalization techniques such as vectorization, stemming and lemmatization for data pre-processing. Implemented word count feature extraction method and SVM and Decision tree modeling. Implemented regularization for better results. We are also working towards implementing KNN algorithm as well to better analyze the different models for our dataset.

- **How are you planning to evaluate your results?**

  We are planning to evaluate our results based on Accuracy, F1-Measure and Confusion Matrix. We have been able to calculate accuracy for each model as of now and will be working on the other two result evaluation criteria and will also try to implement toxic comment detection before the final submission.

Research paper reference:

https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf

Other References:

- https://www.researchgate.net/publication/321843804_Sentiment_Analysis_of_Movie_Reviews_using_Machine_Learning_Techniques
- https://www.andrew.cmu.edu/user/angli2/li2019sentiment.pdf
- https://www.lexalytics.com/lexablog/sentiment-accuracy-baseline-testing

Files attached:

IMDBSentimentAnalysis.ipynb - implementation code in Python

Note: Use Jupyter Notebooks to run this file.