Building a text classifier using various methods such as Naïve Bayes

Saurabh Jain

Department of Computer Science, University of Southern California,

Los Angeles, California 90089, USA

(Dated: April 8, 2021)

Abstract

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and bio-metrics to systematically identify, extract, quantify, and study affective states and subjective information. This assignment focuses on proposing a model, that can be used to analyze Twitter messages. The primary task is to build a text classifier.

Various models have been used to compute the accuracy of the models and to predict which one is the best model for this particular data set. The task includes building a classifier that can sort the messages into 5 categories - Extremely Negative (0), Negative (1), Neutral (2), Positive (3), and Extremely Positive (4). The various models that were used are: Naïve Bayes & Logistic Regression, and some word embedding models such as Tf-Idf model & Word2Vec model. The best accuracy was found to be that of word2vec model's which was computed to be 0.5582.

I. INTRODUCTION

Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. This assignment focuses on constructing and testing different models, that can be used to analyze Twitter messages. The primary task is to build a text classifier. Various models have been used to compute the accuracy of the models and to predict which one is the best model for this particular data set. The task includes building a classifier that can sort the messages into 5 categories - Extremely Negative (0), Negative (1), Neutral (2), Positive (3), and Extremely Positive (4).

The models used the processed data, which was cleaned to help the models not confuse with similar meaning words. The hyperlinks were deleted from the tweets and the mentions & tags were deleted as well. Finally, the words were lowercased as well to improve the model efficiency. The various models that were used are: Naïve Bayes & Logistic Regression, and some word embedding models such as Tf-Idf model & Word2Vec model. The best accuracy was found to be that of word2vec model's which was computed to be 0.5582.

II. DATA EXPLORATION

The data set was a collection of (labeled) tweets describing or commenting the local Covid situation. The data set contained three files. Each tweet had a corresponding classification - Extremely Negative (0), Negative (1), Neutral (2), Positive (3), and Extremely Positive(4). One file had the strings of labels and the other contained the number associated with it.

Some tweets contained mentions, some had hashtags. A few of the tweets had hyperlinks. Most tweets had punctuation. All of this was dealt using the preprocessing step as discussed in the next section.

III. DATA PREPROCESSING

As mentioned in the previous sections, the tweets included mentions/tags, hyperlinks, uppercase letters, references, etc. The following steps were incorporated in no particular order to clean the tweets so that the model can function well when implemented:

1. Convert text to lowercase

2. Remove numbers

3. Remove punctuation

4. Tokenization: Tokenization is the first step in NLP (Natural Language Processing). It is the process of breaking strings into tokens which in turn are small structures or units. Tokenization involves three steps which are breaking a complex sentence into words, understanding the importance of each word with respect to the sentence and finally produce structural description on an input sentence.

5. Removing Stop words: Stop words are the most common words in a language like "the", "a", "at", "for", "above", "on", "is", "all". These words do not provide any meaning and are usually removed from texts.

6. Removing links

7. Stemming: Stemming usually refers to normalizing words into its base form or root form.

8. Lemmatizing: In simpler terms, it is the process of converting a word to its base form.

The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors. Therefore, lemmatization was used instead of stemming.

IV. MODEL SELECTION

After pre-processing the data, the data was split into train and test using the K-Fold technique so that the efficiency could be optimized. sklearn's KFold module was used for this purpose.

Various models were in consideration and the selected models for this assignment were:

1. Naïve Bayes: sklearn's MultinomialNB() library was used to compute the accuracy of the model by training using the train data set and then testing it using the test data

set. The model was also cross-validated using sklearn's module. Various scores were computed for this:

- Accuracy Score
- Precision score
- Recall Score
- F1-Score

2. Logistic Regression: Basic Logistic regression was also used to train and test the model using the same training and test data set. Accuracy was also computed for the same.

3. TF-IDF model: After the data set was pre-processed, a tf-idf approach was used to compute the Naïve Bayes accuracy score for the model. The amount of features were reduced because the text had too many words.

4. Word2Vec model: Tensorflow was one of the libraries used to compute the epochs required to calculate the accuracy using Word2vec approach. Sequential module was used with LSTM as 32 to keep the computational time less and Dense as 5 for the 5 categories.

V. INTERPRETATION

The following were the accuracy scores of the different modules computed:

1. Naïve Bayes
   - Accuracy Score: 0.46
   - Precision score: 0.51
   - Recall Score: 0.46
   - F1-Score: 0.45
2. Logistic Regression
   - Accuracy Score: 0.58

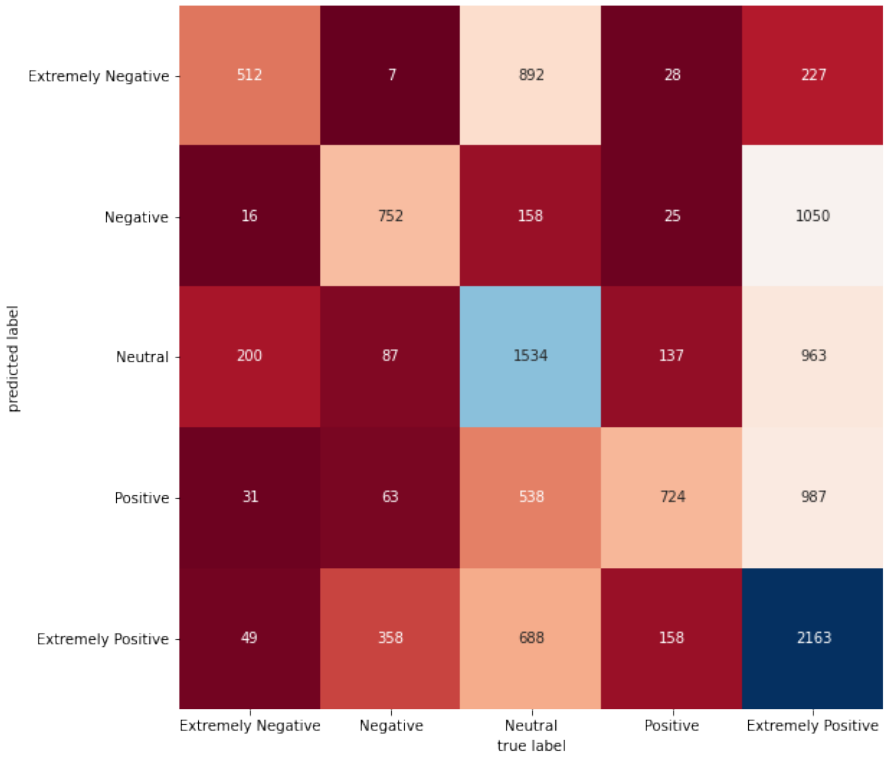

FIG. 1. Confusion Matrix for Naïve Bayes classifier.

3. TF-IDF model

   - Train set score: 0.51
   - Test set score: 0.41

4. Word2Vec model

   - Accuracy Score: 0.56

The scores as mentioned above, most models behave very similarly. Word2vec has a great accuracy with 0.56. This was done using word embedding, just like the tf-idf approach. One-hot encoding was also performed for the purpose of word2vec approach.

For the MultinomialNB(), which is the Naïve bayes theorem, the accuracies can be visualized in Fig. 1

An attempt was made to compute the Naïve Bayes accuracy without pre-processing and the accuracy score diminished significantly which shows the importance of pre-processing and cleaning the data set when tweets are concerned. This can be visualized in Fig. 2.

FIG. 2. Confusion Matrix for Naïve Bayes classifier without pre-processing.

Transfer learning techniques were used such as Word2vec model and the accuracy was also computed. A Bag of words approach was used to make the model perform more efficiently.

VI. CONCLUSIONS

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and bio-metrics to systematically identify, extract, quantify, and study affective states and subjective information. This assignment focuses on proposing a model, that can be used to analyze Twitter tweets. The primary task was to build a text classifier using various models. The models have helped to compute the accuracy and to predict which one

is the best model for this particular data set. The task includes building a classifier that can sort the messages into 5 categories - Extremely Negative (0), Negative (1), Neutral (2), Positive (3), and Extremely Positive (4). The various models that were used are: Naïve Bayes & Logistic Regression, and some word embedding models such as Tf-Idf model & Word2Vec model. The best accuracy was found to be that of word2vec model's which was computed to be 0.5582.

The assignment proved to be important to understand the importance of pre-processing. Various word embedding techniques and transfer learning techniques were part of the learning process as we understand the concepts of Natural Language Processing such as Sentiment Analysis and Text Analysis. A bag of words approach was used to compute the accuracy of Word2vec approach, which proved to be one of the best classifier amongst others such as Naïve Bayes and Logistic Regression.

DATA AVAILABILITY

Data is available at blackboard namely:

1. ps5_tweets_labels_as_numbers.csv
2. ps5_tweets_labels.csv
3. ps5_tweets_text.csv

CODE AVAILABILITY

Code is available at `https://github.com/USC-DSCI-552-Spring2021/dsci552-spring2021-32416d-ps5-sjain681`

- hw5.ipynb
- Tfidf.ipynb
- word2vec.ipynb
- hw5-Copy1.ipynb

ACKNOWLEDGMENTS

I would like to thank professor Kristina Lerman, professor Keith Burghardt, the TAs and the graders of the course DSCI 552 for all their efforts towards this assignment.

Appendix A: References

1. "Google Colaboratory." Google, 2021, colab.research.google.com/github/minsuk-heo/tf2/blob/master/jupyter_notebooks/10.Word2Vec_LSTM.ipynb#scrollTo=915WJfpKp9H_.

2. "Sklearn.Naive_bayes.MultinomialNB — Scikit-Learn 0.24.1 Documentation." Sklearn, 2021, scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html.