Report: Tweet Sentiment Analysis

Youssef Mostafa, Ahmed Tamer

May 9, 2023

1 Introduction

In this report we are going to discuss our NLP Project deliverable for milestone 1. Our project is a sentiment classification model that targets tweets, as twitter has attracted many users in recent years and has proved to be one of the most engaging platforms for discussions; it makes a good target for an NLP project and with all different kinds of people on the internet, a diverse set of data will be found. We used sentiment140 as our data set, which contains around 1.6M tweets.Then, we prepossessed it and analyzed it thoroughly and this will be discussed in the upcoming sections.

2 Data Analysis

Data analysis is crucial for making sense of the vast amounts of data generated on social media platforms like Twitter. With billions of tweets being posted every day, data analysis can help extract valuable insights from this massive volume of data. By analyzing a Twitter data set, it's possible to gain a deep understanding of the trends, patterns and sentiments of the users, which can be used for various purposes such as: understanding customer preferences, tracking brand reputation and monitoring public opinion on a particular topic. In the case of this project, we first analyzed the skewness of data, by checking the amount of tweets per sentiment and our data set was nearly symmetrical as can be seen in the following figure:

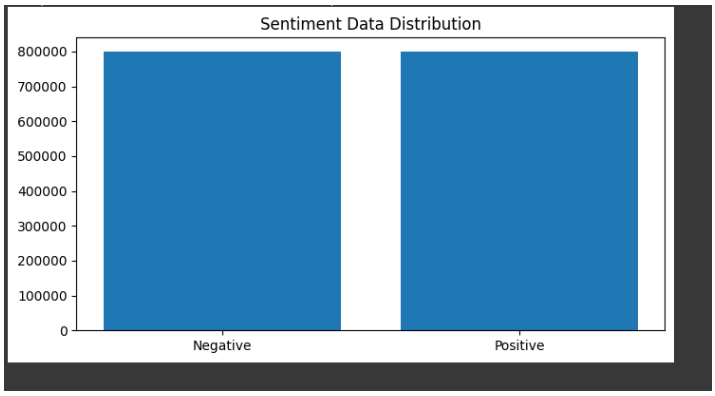

Figure 1: Per sentiment sentence frequency

In addition to this, words used within tweets on a per-sentiment basis were analysed and represented in the form of a word-cloud, shown in the figure below:

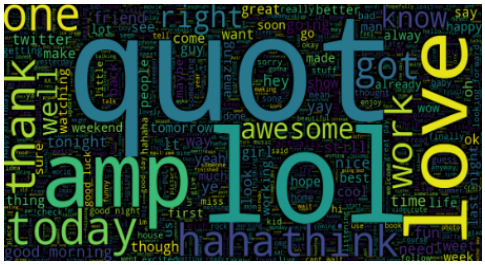

Figure 2: Word cloud for negative keywords

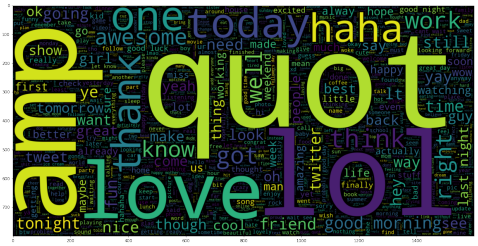

Figure 3: Word cloud for positive keywords

The wordcloud representation is helpful to identify how the different sentiments twitter users portray conform with different words and sentences, as the results from this analysis will be matched and checked with the training results of the model later on while testing in order to measure the accuracy and performance of the model.

Following that, the most common words in each sentiment were analyzed to give a better idea of what words carry the most weight in each, while also checking for any common occuring that are present outside of their expected sentiment text.

In addition to that, as Twitter is filled with gibberish and often lots of misspelled words, so we had to use Pyspellchecker to get the amount of misspelled words and their frequencies in order to statistically correct them with the library itself.

Misspelled Data can create a large amount of noise which is terrible when training a machine learning model, it will also falsely inflate the size of the model's vocabulary creating multiple different representations for each word.

If not handled well, the points mentioned above can drastically reduce the ability of the model to understand the training data.

3 Pre-Processing Phase

Various techniques were utilized in the pre-processing phase, which will be discussed below. The pre-processing phase is very important in any NLP model, as it allows for smooth operation on the given input by the NLP algorithm, due to the model's ability to more easily understand the input when the unnecessary filler details within the text are filtered out. Techniques used consist of:

- Stemming: the action of reducing words to their base or root form, which is called the stem. The goal of stemming is to group together words that have the same root, so that they can be treated as the same word when performing text analysis.
- Lemmatization: the goal of lemmatization is to group together words that have the same root, so that they can be treated as the same word when performing text analysis. Unlike stemming, which simply removes word endings to create a stem, lemmatization considers the context and part of speech of a word to determine its lemma.
- Spelling Correction: the action of correcting commonly found spelling mistakes within sentences and restoring them to their original form, so that the model can handle them with accuracy.
- Train-Test Split: a common technique used in machine learning to evaluate the performance of a model. It involves splitting a dataset into two subsets: a training set and a testing set. The training set is used to train the machine learning model, while the testing set is used to evaluate the performance of the model on unseen data. By evaluating the model on a separate testing set, we can get a better estimate of how well the model will perform on new data that it has not seen before. The train-test split is typically done randomly, with a certain percentage of the data assigned to each subset. For example, an 80-20 train-test split means that 80% of the data is used for training the model, while 20% is used for testing. The exact split ratio depends on the size of the dataset, the complexity of the model, and the specific task at hand.

- Tokenization: tokenization is the process of segmenting text into meaningful units that can be analyzed by a machine learning algorithm. The tokens created through tokenization are usually the basic units of text that a machine learning algorithm processes, such as words, punctuation marks, numbers, and even emojis. Tokenization is typically the first step in many NLP tasks, such as sentiment analysis, text classification, and machine translation.
- Extras: extra techniques were used in addition to the ones listed, including: slang replacement, converting emoji's to short words representing sentiment as well as hyperlink removal.

4 System overview

The current system is made of stacked bidirectional LSTMs following a 1D Convolution layer for features gathering, this allows our model to capture the features from the text and encode them in an interpret-able form to be able to learn from it correctly, the bidirectional LSTMs allow the model to take a look at the input from different directions and better analyze higher order features that aren't easily interpretted.

In addition to that, the text input can't be fed directly into the model without being represented as feature vectors and therefore we resorted to word embedding and use both Glove and word2vec in different iterations of our model

the middle layers in the model did include :

- Dense Layers
- Concatenation layers
- Flatten layer.
- Dropout layer

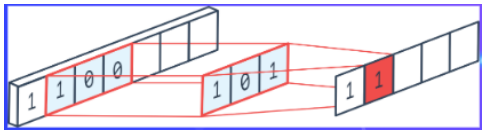

Figure 4: simple convolution

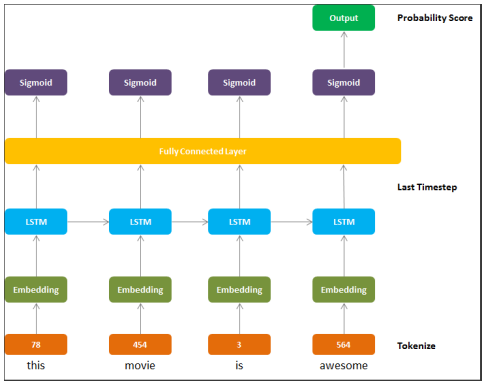

Figure 5: this is an overview of an LSTM model

these layers serve as connections for the data flow in the model. The dense layer adds trainable parameters to our model as well as adjust the size of outputs to match the inputs for the other layers, the Concatenation layer concatenates outputs of different sizes from different layers and outputs them as one single tensor, similarly is the Flatten Layer as it does reduce the dimensions of our tensors to first rank, finally the dropout layer helps to reduce the over fitting when training.

The final layer in our model is a sigmoid layer to output a probability which is divided into intervals to deduce which class does this input belong to.

Finally, different iterations of the models are going to be ensembled together to improve the accuracy of the system as a whole through probability average ensemble.

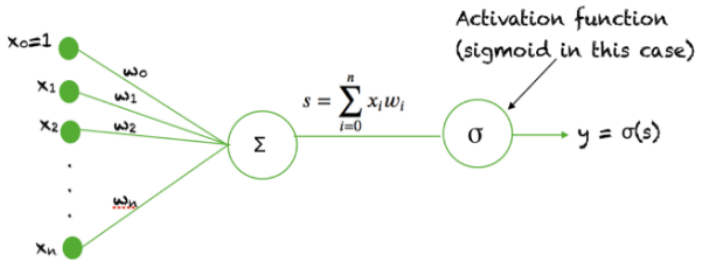

Figure 6: sigmoid diagram

5 Conclusion & Future Work

In conclusion, this report has served as a means of discussion about the importance of sentiment analysis and what it can be used for, as well as portraying how sentiment analysis can be broken done into a series of steps. More specifically: data analysis, pre-processing and detailed operational functionality. Each of those sub-categories were explained thoroughly with listed examples underneath each of them. In the near future, the model that was created will be further fortified with new ideas and technologies, such as: data ensemble, different algorithms (word2vec instead of gloVe for example), as well as new ways to analyze sentiment either the pre-existing sentiment 140 dataset or a newly introduced one, depending on what is discovered to be more suitable.