Tweet Sentiment Extraction

Satyanarayana Vinay Achanta
Computer Science
Utah State University
Logan, Utah
a02395874@usu.edu

Saichand Thota
Computer Science
Utah State University
Logan, Utah
a02394497@usu.edu

Hari Chandana Kotnani
Computer Science
Utah State University
Logan, Utah
a02396013@usu.edu

***Abstract*—Twitter is one of the most engaging social networking platforms in the present decade. Many companies, famous personalities, and celebrities share their opinions through this site, making it an exceptional place to have discussions on different topics ranging from global arguments to funny memes. The sentiment of the tweets plays a vital role in analyzing their effect on the community. Tweets by any user have categorized into neutral, positive, and negative, and words in the tweet play a crucial role in determining the sentiment. Many previous reports focused on tweet sentiment prediction, whereas, in this report, our main concentration is on finding out the word or set of words that determine the sentiment of the tweet. In the initial stage, we implemented pre-processing techniques to clean the raw data and then applied different methods to that data to obtain appropriate words from the tweet. In our analysis, we found that neural networking models like LSTM, Bi-LSTM, and RoBERTa performed better when compared with traditional methods. The dataset we examined was categorized into positive, neutral, and negative sentiments.**

I. Introduction

We all need data or information. Whether to improve our personal life, or maintain work-life balance or ease work, we strive to learn new things daily. Sentiment or a way of expressing the story is vital in spreading the information to a longer and deeper stretch. In the olden days, the news was used to pass through mouth talk and later by telegrams and telephones. With the use of the Internet in the present decade, information is spreading in a way that reaches the person who lives on another side of the world within a blink. Social media plays a crucial role in spreading this information throughout the world without much intervention from human beings. Social media like Twitter and Facebook are helping a lot in broadening the posts or the tweets with intensity. People express their emotions or feelings through their posts to increase the depth of the information. Every day, millions of people post millions of tweets on Twitter, generating enormous data with valuable information. Tweet sentiment is categorized into positive, neutral, and negative for finding the sentiment of a tweet. The objective of this project is to find out the word or set of words from the tweet that determines its sentiment.

Numerous methods are available to achieve this task. We implemented various methods such as Naïve Bayes, Support Vector Machine (SVM), K-Means, Sentiment Intensity Analyzer, and also neural networking models like Long-Short-Term Memory (LSTM), Bidirectional LSTM and Robustly Optimized Bidirectional Encoder Representations from Transforms Approach (RoBERTa) to predict the key words of the tweet. Instead of implementing the algorithms from scratch, we imported them from Python packages - Naïve Bayes, SVM, and K-Means from Sci-Kit Learn; LSTM and RoBERTa are TensorFlow models. Pre-processing the tweet data is the essential step in this classification of tweet data which needs to be consistent with the implemented machine learning methods. The background of these methods is discussed in the Related Work section.

This report briefly explains the various approaches and attempts used to clean the raw data. In the Methods section, we elaborated on the architecture and parameters used in training the models.

The traditional methods did not obtain the finest predictions but served as a baseline. LSTM, Bi-LSTM, and RoBERTa models produced more promising results than the initial methods. The findings and our analysis are in the Results section. In a later section, we discussed our conclusions and future work of this project.

II. Ideas Explored and Discussed in Class

So much research has been conducted in the area of "Sentiment Extraction" over the decades by researchers. Used a bag-of-words method for sentiment extraction where the relationships between the words were not weighed up and a document is represented as just a collection of words. The sentiment of each word is determined and all are clubbed with aggregate functions which help in deciding the sentiment of the tweet.

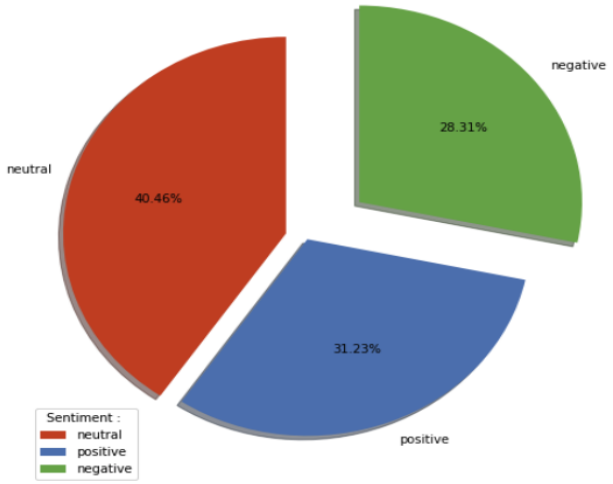
Data pre-processing techniques are quite useful in removing stop words, symbols, and links. Tokenization and Term Frequency – Inverse Document Frequency (TF-IDF), word embedding is used in assigning the values to the words in the tweet. Used one of the popular methods GloVe – global vectors for word representation, uses unsupervised training to learn the same context of words as features[1]. Neural Networking models like LSTM from the class and advanced topics Bi-Directional LSTM and RoBERTa models s significantly improved the predictions

III. Methods

A. Dataset Collection and Exploration

The Dataset obtained from the Kaggle is a set of 27,481 labeled training tweets, and a test set of 3534 raw text tweets,

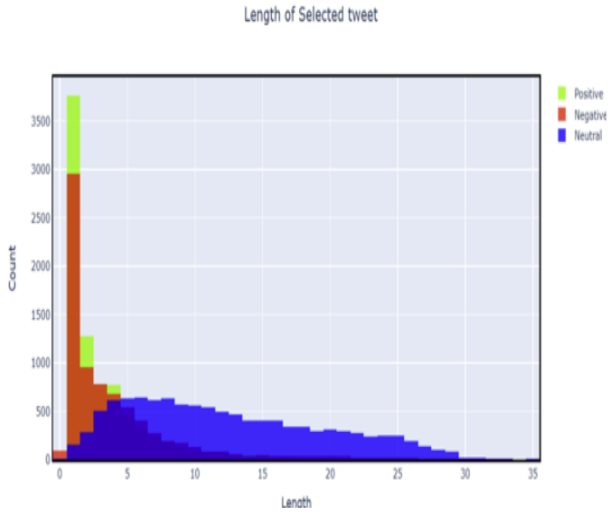
both sets are tagged with sentiments ”positive”, ”negative”, or “neutral”. A single sample of training tweets contains up to 4 columns - a unique identifier for the tweet, the text of the tweet, the keywords that exemplify the sentiment, and the sentiment of the tweet (i.e., neutral). The keyword of the tweet in the test set is the potential to be empty. The aim is to fetch a substring from the tweet that represents the support for the given sentiment. For example: [Sooo SAD I will miss you here in San Diego!!!, negative] would have the output of “Sooo SAD”.



B. Pre-Processing

Most of the unnecessary words are removed using Data pre-processing tricks like

- Cleaning the text – removing URLs, numbers, punctuations, text in square brackets, missing values, tags, special characters, and null values.
- Removing the English Stop Words from sentences - Imported stopwords from Natural Language Toolkit (NLTK) which removes the basic English words like the, an, a, if, and so on from the tweet.
- Splitting the sentence into words - Vectorizing the words



- Assigning the Parts of Speech - Found out the Parts of Speech of words in the tweet and removed the proper and collective nouns from the text
- Tokenization of text

After performing the Data pre-processing techniques we implemented the below models.

C. Models

Naive Bayes

This classifier computes the probability of the class given by the specific feature. Later, tweets are categorized by calculating the probabilities for all three classes and assigned to the label with the highest probability. As the labels of the tweet are already provided, this method finds the word based on the given label. General assumptions for this model are features make equal and independent contributions to the outcome. After pre-processing the data, we implemented the technique - Multinomial Naïve Bayes model, and tuned hypermeter alpha to achieve better performance.

Support Vector Machine

Support Vector Machine is to find out the hyperplane in this 3-dimensional labeled dataset (’negative’, ’positive’, ’neutral’). This plane has the maximum margin i.e., lies from the maximum distance from all the labels. This classification is robust to outliers and generalization is also carried out to prevent over-fitting of the output. We imported from sklearn and tuned the hypermeter - alpha for predicting the text.

K Means

Performed TF-IDF and dimensionality reduction techniques like PCA and t-SNE before clustering. Principal Component Analysis (PCA) reduces the dimensionality by representing all data vectors as linear combinations of small eigenvectors. t-SNE reduces dimensionality by preserving local clustering, and it can also handle outliers. After this, clustering was performed with optimal clusters = 100 by importing the MiniBatchKMeans and obtaining the top keywords selected from the clustering. The advantage of this method is to reduce the minimum square error (MSE) - noise filter. Using clusters and labels obtained keywords of the tweets.

Sentiment Intensity Analyzer

The model assigns the polarity score to each word depending on the tweet’s sentiment. Later, choosing these words from the test data to determine the final selected texts.

Long short-term memory (LSTM)

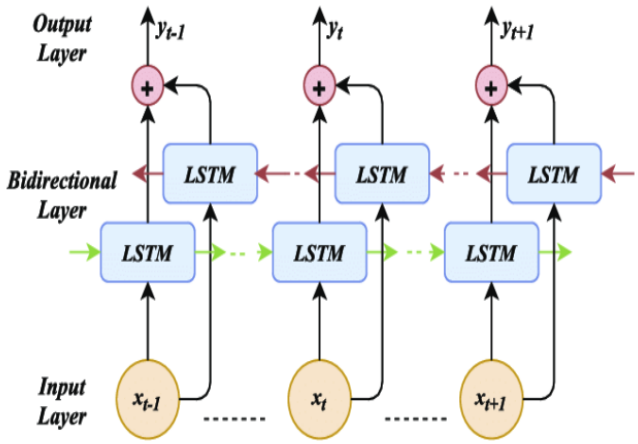
LSTM is a sequential network introduced to handle the long-term dependencies problem. Long short-term memory networks are a type of Recurrent Neural Network that can learn long-term dependencies[4]. LSTM contains special units like a memory cell that keep track of information in memory for a long time.

In this model, we first performed pre-processing techniques, embedding using Glove[2], and tokenization of words. Then,

we did padding based on the longest tweet in our data set. Embedding the layer inputs and performed LSTM (Embedd sequences) on these layers with hidden layers 256 and l2 regularisation with lr = 0.01. Later, applied Timedistibuted layer to every temporal input slice with the activation function ' relu'. Dropping layers with 0.2 drop out on the outputs. Outputs are dense with activation 'sigmoid,' optimizer 'adam' and loss 'binary$_crossentropy'$.

Bi-directional LSTM (Bi-LSTM):

Bidirectional LSTM is an extension to LSTM which makes the input flow in both forward and backward directions to preserve the past and future information in a sequence. It is also useful for modeling sequential dependencies between words and phrases in both directions of the sequence[6]. For this task, we are not doing sequence prediction, the output depends on the entire sequence. This is why a bidirectional network is important. The output features should represent the word in the context of the sequence. The dimensions from the output of the LSTM, Omi, are concatenated with the sentiment of the sequence (positive, negative, neutral).



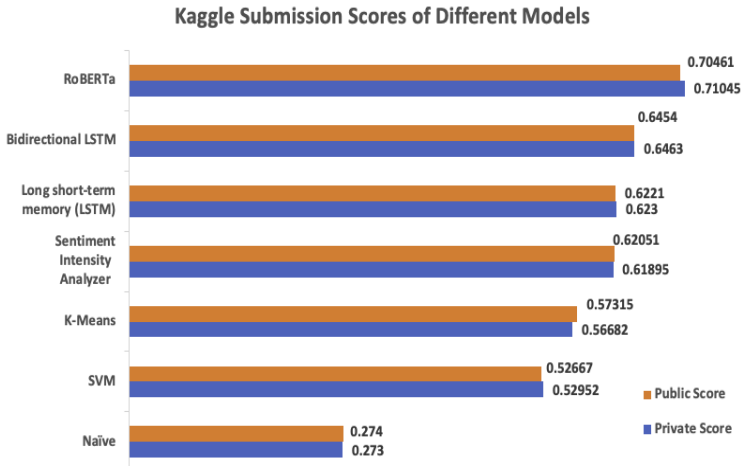
Similar to LSTM, we did embedding and Tokenization of all words, padding to max length, Implementing the NN model with epochs – 5, activation function –'relu', batch size – 32, optimizer – 'Adam', and Dropout 0.6. The network model used consists of 2 recurrent layers and a hidden layer dimension of 64 (32 for each direction).

RoBERTa:

The pre-trained model we applied is RoBERTa-base. RoBERTa is a robustly optimized BERT pre-training approach presented by Liu et al [5], it has the almost same architecture as BERT. To improve the training time and result on BERT, RoBERTa removes the Next Sentence Prediction objective, trains with a bigger batch size longer sequence, and dynamically changes the masking pattern.

Pretrained model approach on which we performed the tokenization of data and build the model with 2 Conv1D layers, dropout, flatten, activation function – 'Softmax', Optimizer – 'Adam'; trained with 5k Folds.

IV. RESULTS:



V. EVALUATION:

Using sequence representations to predict set membership, we were able to obtain a comparable validation Jaccard similarity score with the Twitter dataset

Models	Jaccard Score
Naïve	0.272
SVM	0.52663
K-Means	0.57312
Sentiment Intensity Analyzer	0.62045
Long short-term memory (LSTM)	0.61497
Bidirectional LSTM	0.6451
RoBERTa	0.70455

VI. CONCLUSION AND FUTURE WORK

We started with pre-processing the data using traditional cleaning techniques like stopwords, vectorization, parts of speech, and tokenization and implemented baseline models – Naïve, Support Vector Machine, and K-Means. Next, we tried Sentiment Intensity Analyzer which produces the outputs based on the polarity scores. Later, we introduced three neural networking models – LSTM, Bi-LSTM, and RoBERTa which gave us better outcomes of 0.6191, 0.6463 0.7104 private scores. The RoBERTa proved to be the most performance in the training process. In the future, we will research on implementing these models by tuning hypermeters with optimized code and test the data from other network models – DistilBERT and ALBERT to obtain different results.

REFERENCES

[1] Pennington, J., Socher, R., amp; Manning, C. (2014). "Glove: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)". https://doi.org/10.3115/v1/d14-1162.

[2] “George, D. W. (2020, June 14). Glove.6b.100d.TXT. Kaggle. Retrieved May 2, 2022, from https://www.kaggle.com/danielwillgeorge/glove6b100dtxt ”.
[3] B. Baharudin, L. H. Lee, and K. Khan, “A review of machine learning algorithms for text-documents classification,” Journal of Advances in Information Technology, vol. 1, no. 1, 2010.
[4] Bai, X. (2018). Text classification based on LSTM and attention. 2018 Thirteenth International Conference on Digital Information Management (ICDIM). https://doi.org/10.1109/icdim.2018.8847061.
[5] J.Devlin,M.-W.Chang,K.Lee,andK.Toutanova,“Bert:Pre-training of deep bidirectional Transformers for language understanding,“ arXiv.org, 24-May-2019. [Online]. Available: https://arxiv.org/abs/1810.04805. [Accessed: 01-May-2022].
[6] Y. Verma, “Hands-on guide to Bi-LSTM with attention,” Analytics India Magazine, 22-Aug-2021. [Online]. Available: https://analyticsindiamag.com/hands-on-guide-to-bi-lstm-with-attention/.