Sentiment Analysis of Tweets During COVID-19

Yating Liang
Department of Analytics
Georgetown University
yl1138@georgetown.edu

Yachen Li
Department of Analytics
Georgetown University
yl1062@georgetown.edu

George Sun
Department of Analytics
Georgetown University
hs1023@georgetown.edu

Abstract

Nowadays, the world is suffering a huge pandemic called COVID-19, which no country can avoid. Through this pandemic, there appears storm not only in reality, but also in social media. As more and more people left, negative emotions gradually spread all areas on the Internet, including songs, messages, videos and so on. As one of the largest social networking platforms, Twitter accordingly gathered proved its significance to help citizens communicate with each other or express their sentiments remotely. Therefore, in order to analyze and even predict the sentiments of the general public towards COVID-19, we decided to treat the messages on twitter as samples to conduct sentiment analysis, based on machine learning models, such as TextBlob, Vader and Random Forest. We gathered data from Kaggle and the result showed that the accuracy of the best model is around 90%.

1 Introduction

Nowadays, internet has already become the most important media through which people can achieve remote communication or share their feelings and attitudes with people that they are even not familiar with. For example, on social media platform such as blogs, Facebook and Twitter, the presidents can immediately communicate with their supporters or detect their attitudes towards specific policy. Then as an unpredictable pandemic called COVID-19 spread all over the world, the importance of these social media applications has been strengthened, for they become the most effective and safe way for the general public globally to immediately get rid of loneliness and anxiety. Since Coronavirus produced a huge impact on both the global economy and personal life, as members who experienced this pandemic, we are interested in the sentiment or attitudes people had towards Coronavirus. One of the best approaches is to analyze the messages left on social media, such as Twitter. Hence, we gathered data about COVID-19 from Kaggle. Indeed, this paper focuses on the sentiment analysis of tweets about COVID-19 and uses sentimental labels to build five models which are capable of predicting the sentiment of a person towards Coronavirus. These models include TextBlob, Vader, Naive Bayes, Logistic Regression, Random Forest, and Neural Network. Besides, in this paper the dataset was transformed into another format, in which there are just three kinds of sentimental labels instead of five. This paper would show the comparison of separate performance of those models on these two distinct sets.

2 Objective

The objective of this paper is to find the distribution of different categories of tweets and compare the performance of each model. As a result, we can obtain a relatively excellent model to predict the label of any tweet in the future as accurately as possible. In this paper, we are going to use both the original dataset and the transformed dataset, so that we can investigate which model performs better than the others and whether less kinds of labels can improve the accuracy of these models.

3 Dataset

3.1 3.1 Data Source

We gathered the dataset from Kaggle: https://www.kaggle.com/datatattle/covid-19-nlp-text-classification.

This dataset was originally used to produce text classification on tweets. The column named "OriginalTweet" contains original messages scraped from Twitter and manual tagging has been done on each tweet to categorize their sentimental labels. Though there are 6 columns in the original dataset, only the "OriginalTweet" and "Sentiment" features are used in this research, and accordingly all the remaining variables in the preprocessing. Furthermore, inherited the insight from our review literatures in which the authors removed neutral sentiments, we also decided to create a new dataset based on the original one in which we removed extra sentimental labels, such as "extremely positive" and "extremely negative" to detect whether the models we chose could generate a better output.

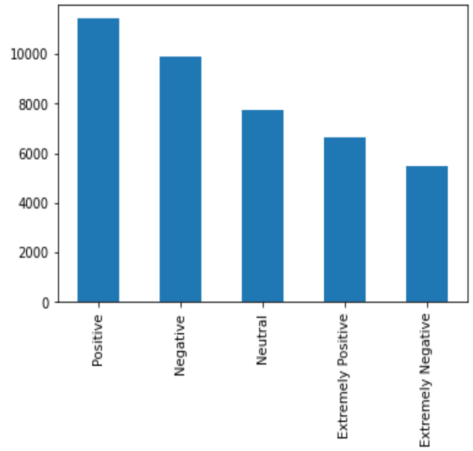

Figure 1: Dataset Label Distribution

3.2 Data Preprocessing

To clean the textual data of tweets sentences, we first removed all the stopwords, URLs, punctuations and numbers, then lowercase and lemmatized the texts. The three labels "negative", "positive", and "neutral" were encoded accordingly. We transformed the texts into count vectorizer for some models such as Logistic regression. We also combined the extra sentimental labels, such as "extremely positive" and "extremely negative", into "positive" and "negative" to detect whether the models could generate a better output.

4 Background

Many researchers have worked on Sentiment Analysis on Twitter social media in literature. Previous work includes research on theoretical comparisons of the state-of-art approaches for Twitter sentiment analysis, which contains machine learning, ensemble approaches and dictionary-based approaches (Alsaeedi & Khan, 2019). Various sentiment-analysis approaches used for Twitter are described including supervised, unsupervised, lexicon, and hybrid approached. Models that were used in their research include MNB, RF, SVM, LR, and etc. The authors also discussed the way of determining the best approach for detecting sentiments by using different WordNet and different measurements. These approaches and combinations they discussed inspired us and were helpful when we explored sentiment analysis on Coronavirus in tweets.

Presented in another article, the research emphasized on people's tweet sentiment regarding coronavirus. The authors (Rajput, Grover, & Rathi, 2020) gathered tweets from both the general public and the WHO account. The authors used the Python built-in package TextBlob to perform sentiment analysis of tweets correlated with the coronavirus outbreak. The polarity values of individual tweets have been computed. The interpretation of these values is as follows: polarity > 0 implies positive; polarity < 0 implies negative; polarity = 0 implies neutral. At the end, the authors removed the neutral tweets and found that positive emotions in the tweets are higher than negative ones, which enlightens us that the contrast between positive and negative tweets might result in further meaningful conclusions.

In another article, the aim of authors (Alanezi, Hewahi, 2020) is to investigate the impact of social distance on people during COVID-19 pandemic using twitter sentiment analysis through a comparison between the k-means clustering and Mini-Batch k-means clustering approaches. In this paper, a comparison between k-means and Mini-Batch k-means is performed to find a pattern. The word frequency shows that there are several words related to the pandemic.
In the recent past, the researchers have shown insights on classifying the emotions using the BERT model on tweets (Singh, Jakhar, & Pandey, 2021). They chose Bi-directional Encoding Representation for a Transformer (BERT) model for emotion classification, where the meaning of a word in a given sentence depends on the other words surrounding it. When preprocessing their

data, the authors converted the training set into respective torch tensors for the model and defined the batch size to create tensors and iterators to fine-tune the BERT model. When evaluating the model performance, they chose MCC validation accuracy, which is based on the Matthews correlation coefficient, a widely used statistical rate that generates high score prediction results. The combination of BERT and MCC inspired us with more feasible approaches to improve the performance of sentiment analysis.

In another article, the authors (Nemes, Kiss, 2021) applied recurrent neural network (RNN), a neural network that is intentionally run multiple times, where parts of each run feed into the next run. In RNN, hidden layers from the previous run provide part of the input to the same hidden layer in the next run. In addition, recurrent neural networks are particularly useful for evaluating sequences, so that the hidden layers can learn from previous runs of the neural network on earlier parts of the sequence. Besides, they treated the TextBlob model as the baseline model and used it to compare with the RNN model.

5 Methodology

5.1 TextBlob

TextBlob is a python built-in library for processing textual data. It provides a simple API for diving into common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification and so on. After applying TextBlob to sentences, it would generate a polarity value ranged from [-1.0, 1.0], where -1.0 means a negative polarity and 1.0 means a positive polarity. This score might also be equal to 0.0, which stands for a neutral sentiment. In this paper, we treated this model as a baseline model, which does not need to be trained on the training set and can be directly applied to the test set to predict the label of each tweet. Using the same interpretation approach adopted by Singh, Jakharand and Rathi, we would use the result of this model to compare with performance of the others.

5.2 Vader

VADER is called Valence Aware Dictionary for Sentiment Reasoning, which is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data. VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text could be obtained by summing up the intensity of each word in the text. The SentimentIntensityAnalyzer() in Vader library takes in a string and returns a dictionary of scores in four categories: negative, neutral, positive and compound(which is computed by normalizing the three scores before). In this paper, we decide to extract the compound value of each sentence from the dictionaries and use a criterion below to categorize each tweet:
If compound >= 0.5, its sentiment is positive;
If compound <= -0.5, its sentiment is negative;
If -0.5 < compound < 0.5, its sentiment is neutral.

5.3 Logistic Regression

Logistic regression is a fundamental classification technique, which adopted the sigmoid function to classify data into different categories. In this paper, we transformed the dataset into count vectorizer and then applied logistic regression to classify our tweets. We set the maximum number of iterations as 1000 taken for the solvers to converge.

5.4 Naive Bayes

Bayes' Theorem provides a way that we could calculate the probability of data belonging to a given class, given some prior knowledge. Naïve Bayes is a classification algorithm for binary and multi-class classification problems. They are probabilistic classifiers, therefore could calculate the probability of each category using Bayes theorem, and the category with the highest probability will be output. In this research, multinomial naïve bayes model were applied to the dataset with count vectorizer.

5.5 Random Forest

Random forests are an ensemble machine learning method that could be used for classification, regression and other tasks. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets.

5.6 Neural Network

The neural network is organized in a series of layers, where the input vector enters at the left side of the network, which is then projected to a "hidden layer." Each unit in the hidden layer is a weighted sum of the values in the first layer. This layer then projects to an output layer, which is where the desired answer appears. In this paper, we adopted Long short-term memory (LSTM). Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as texts or images), but also entire sequences of data (such as speech or video). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. In this research, the final output of the layer is fed through a SoftMax function, which outputs values between 0 and 1, so that the results could be interpreted as probabilities.

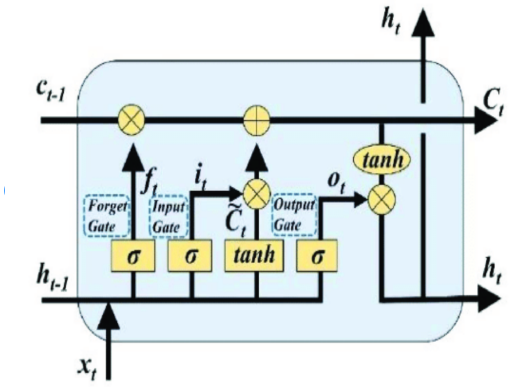

Figure 2: LSTM Illustration

6 Results

Classification Results

Models	Accuracy	Precision	Recall	F1
Logistic Regression	0.600	0.620	0.607	0.610
Naive Bayes	0.432	0.527	0.396	0.411
Random Forest	0.494	0.562	0.479	0.483

Table 1: Validation Results with 5 labels

From this table, we can see that the performance of these models on dataset with 5 categories of labels is not good enough to generalize a useful model to predict the sentiment of further tweets. The naïve bayes and random forest model have accuracy lower than fifty percent. Hence, we decided to substitute all extreme sentiments, such as "extreme positive" and "extreme negative" with simply "positive" and "negative". And we have assumption that after combining to three labels, models will have better performance.

Models	Accuracy	Precision	Recall	F1
TextBlob	0.556	0.534	0.549	0.530
Vader	0.904	0.899	0.894	0.897
Logistic Regression	0.802	0.778	0.782	0.780
Naive Bayes	0.672	0.669	0.570	0.569
Random Forest	0.664	0.646	0.639	0.642
Neural Network	0.757	0.768	0.747	0.757

Table 2: Validation Results with 3 labels

From the Table 2, we can see that the metrices of all above models increase a lot. We treat the TextBlob model as the baseline with an accuracy of 0.556. Vader generates the highest metrics' scores among all the adopted comparative models, with an accuracy of 0.904. Besides, the logistic regression produced the second-best performance, even better than the one resulted by Neural Network. The neural network model with LSTM produced third-highest metric. The naïve bayes and random forest model perform not really well compared with all other models. The metrics from Neural Network might be different each time. However, the difference is not obvious.

7 Discussion

It can be perceived that surprisingly; Vader has the best performance among six models. It might be the season that Vader considers adverbs of degree and negative words into compound score

calculating. It also incorporates numerous lexical features common to sentiment expression in microblogs, including a full list of Western-style emoticons. This could be the reason that it can improve measurements when applying to subjects of online texts or opinions. It also states a fact that complication of machine learning model is not equivalent to better performance.

The baseline model textblob has the lowest metrics among all models. The reasons might be that it takes part-of-speech tagging and it converts a sentence into a list of words with their tags. For example, for this tweet "Please STOP shaming folks who stock up on food and personal care products ahead of #CoronaVirus", the true label is positive. However, the textblob model treats it as negative. This could be due to the list of words in the tweet of "stop" and "shame" with using negative polarity.

Among all the classic machine learning models, Naïve Bayes has a relatively poor performance. It might because of the assumption that the features are strictly independent, which might not apply to sentimental analysis on tweets.

For random forest and LSTM model, we believe the metrics could still have some improvements if we could apply some feature engineering or hyperparameter tuning with a thorough grid search.

8 Conclusion

In this research, we treat the messages on twitter as samples to conduct sentiment analysis, based on machine learning models, such as TextBlob, Vader, Logistic Regression, Naïve Bayes, Random Forest, and Neural Network. We gathered data from Kaggle, and the final result showed that the accuracy of the best model is around 90%, which is the Vader model. Also, though we have designed our methodology bases on previous work, there are still some limitations. For future improvements, we believe the metrics could still have some improvements if we could apply some feature engineering or hyperparameter tuning with a thorough grid search.

References

Nikhil Kumar Rajput, Bhavya Ahuja Grover, & Vipin Kumar Rathi. 2020. "Word frequency and sentiment analysis of twitter messages during coronavirus pandemic". *A Preprint,* 9 April. 2020, https://arxiv.org/pdf/2004.03925.pdf.

Mrityunjay Singh, Amit Kumar Jakhar, & Shivam Pandey. 2021. "Sentiment analysis on the impact of coronavirus in social life using the BERT model". Social Network Analysis and Mining, 15 February. 2021, https://europepmc.org/backend/ptpmcrender.fcgi?accid=PMC7976692&blobtype=pdf.

László Nemes & Attila Kiss. "Social media sentiment analysis based on COVID-19". *Journal of Information and Telecommunication*, 14 Jul. 2020, https://www.tandfonline.com/doi/full/10.1080/24751839.2020.1790793.

M. A. Alanezi and N. M. Hewahi, "Tweets Sentiment Analysis During COVID-19 Pandemic". 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 20 Jan. 2021, https://ieeexplore.ieee.org/document/9325679.

Abdullah Alsaeedi & Mohammad Zubair Khan. "A Study on Sentiment Analysis Techniques of Twitter Data". *International Journal of Advanced Computer Science and Applications Vol. 10, No. 2. 2019,* https://pdfs.semanticscholar.org/6f34/ad869da01abdf8d183a1786b54ff4217aefd.pdf

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.