Sentiment Analysis on COVID19 Tweets Midterm Report

Xinyi Cai, Sherrie Chen, Sebastian Koh

I. Introduction

The objective of this project is to use natural language processing and machine learning to perform sentiment analysis and classification to COVID-19 related tweets. In this report, we will be covering our approaches to data preprocessing, visualization, extrapolation, feature transformation and model selection. We will also discuss how we will avoid underfitting and overfitting and future work that can be added for the rest of the semester.

II. Data Preprocessing

A. Data Description

Our dataset contains raw information on the location (categorical), date of tweet (ordinal), the main tweet message, and the sentiments (ordinal). It contains approximately 42000 training samples and 4000 test samples.

B. Preprocessing

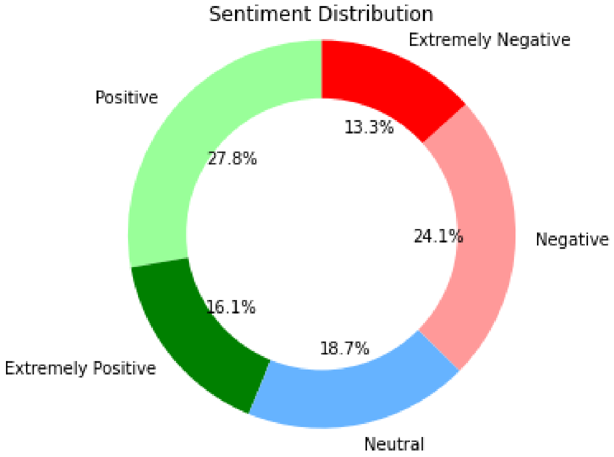
The purpose of preprocessing is to remove information unrelated to semantics so as to ensure consistency for the words across the corpus. Firstly, we drop duplicate data and check for missing values. We discovered that 20% of location data are missing, and all the other data fields contain no missing values. Next, we clean each original tweet by removing https links and urls, hashtags and mentions, punctuation and stop words. Additionally, we converted numerical digits into their equivalent textual representation and set all characters to lowercase. Because some words are in abbreviation, for example, "I'll," we created a dictionary of contractions and mapped abbreviated words into their expanded form.

III. Data Visualization

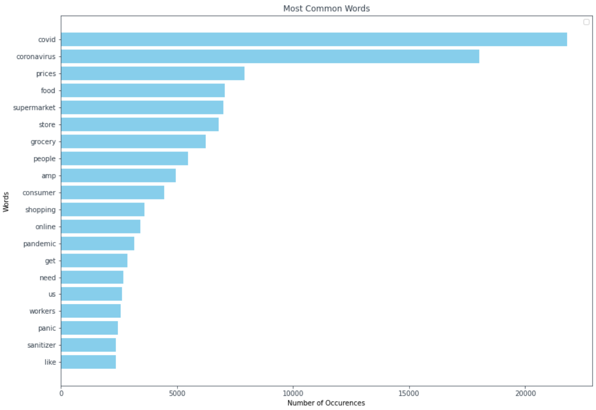
Graph 1 shows an even distribution of samples across different sentiment classes, which favors training of the ML models.

As shown in Graph 2, the most frequent words from the tweets generally include the discussion of the virus; "coronavirus" and "pandemic." People also care about their food supplies and the availability of "grocery" or "sanitizer." The frequency shows that the discussion of COVID19 on Twitter mostly involves basic needs in daily life.

Gambar 1. Distribution of Sentiments



Gambar 2. Distribution of Sentiments

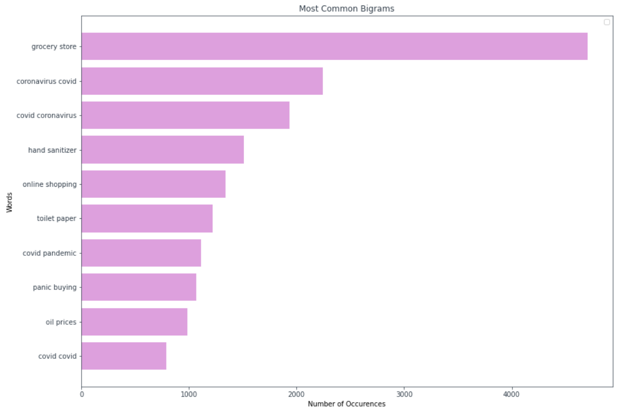


As shown in Graph 3, the most common bigrams, which are adjacent two words in text, demonstrates that many users are concerned about supplies (i.e. toilet paper, hand sanitizer, grocery). The term "panic" appears both in most common unigram and bigram, which indicates that frequent words related to emotions might be helpful for classifying the tweet sentiment.

IV. Model Selection and Outcomes

One key motivation is to be able to provide stakeholders a model that could correctly classify new COVID19 tweets as

Gambar 3. Frequency of the 10 Most Frequent Bigrams from Tweets



extremely negative, negative, positive, or extremely positive so that they can develop follow-up plans to address the public's concerns. Clearly, the output of our prediction model would be the sentiments of the test sample. The first step would be to determine the useful information in the dataset and perform feature transformation so that they are usable for modelling. The second step would involve performing different modelling techniques on the chosen features and compare them with different error metrics.

A. Feature Transformation

We have chosen the following main groups of data that we can extract features for our models. These features are chosen based on our previous data visualizations that showed the correlation between the available information and our desired output, as well as our sense making hypothesis on the application.

1) Feature 1: Tweet Length: We believed that the length of a tweet message might have some correlation to the severity of the sentiment. For instance, someone who is extremely unsatisfied or extremely satisfied with the COVID situation might have a higher tendency to express his or her comments in a lengthy manner. This length of the tweet message will be used directly as a feature input.

2) Feature 2: Tweet Message: The content within the tweet message will form the most relevant feature for our model. NLP will be performed on the tweet message to extract useful features. For our project, we explore 3 methods for comparison; (1) Bag of words (BOW) Count, (2) Term Frequency–Inverse Document Frequency (TF-IDF), and (3) Word2vec conversion.

1) BOW Count: We simply break down all texts into words and assign unique tokens to unique words. Each token will be assigned as a feature variable and the value of that variable for each tweet sample will signify the frequency count of the unique word in it.

2) TF-IDF: Each tweet is represented by a numerical value that reflects the importance of each word in the tweet to the whole data set. The TF-IDF value increases proportionally to the frequency count of the words in the tweet and is offset by the number of tweets in the training dataset that contains the same words. TF-IDF is more complex than BOW count but it has some element of normalization that re-adjust the feature input value according to how common the same words appear in different samples.

3) In Word2vec, a neural network algorithm is used to perform word embedding that detects synonymous words in the whole training dataset. Semantic similarity between different tweet messages will be analysed and each tweet message will be converted into a vector of values that can be input as feature variables for our model.

In addition, we will also extend these 3 models to n-grams where we believe the occurrences of certain words in certain sequences will help in our prediction model. Specifically, our models will have features expanded to a single word and bigram (2 consecutive words).

3) Feature 3: Location: The location where the tweet was made could provide some correlation in the general sentiments in that area. This might help in the sentiment prediction of a new tweet if there is strong correlation between location and sentiment value. We used the library "Geopy" to translate the location into an object that encodes location information such as coordinates and zip code.

4) Feature 4: Date: The sentiment could be correlated to when the tweet message was posted. Specifically, we believed that a tweet made nearer to the date when the COVID pandemic happened might contain more element of uncertainty and stronger sentimental value, as compared to a tweet that is made months after the pandemic start date.

Currently, our preliminary models will be based on feature 1 and feature 2 using TF-IDF methodology. Subsequently, we will incorporate the remaining features and explore the other NLP methodology as described above.

B. Models

1) Logistic Regression: We only used two features, which are length of tweet and TF-IDF encoded n-grams in our baseline model. Log loss function and 10-fold cross validation was used with the other parameters set to default.

2) Multinomial Naive Bayes: As we only included TF-IDF encoded n-grams as our feature, we could assume that the features are conditionally independent. Since we could easily turn words in tweets into counts, we chose the multinomial distribution to represent the posterior probability.

$$P(C_i|\, x_1, x_2 \dots, x_n) = \left(\prod_{j=1}^{j=n} P(x_j|\, C_i) \right) \cdot \frac{P(C_i)}{P(x_1, x_2 \dots, x_n)} \; for\; 1 < i < k$$

3) Linear Support Vector Classifier (SVC): We chose Linear SVC because it applies a linear kernel function to classification problems and it works well for large datasets. The Linear SVC maximizes the width of the soft margin, and it transforms non-linearly separable data into dimensions that can be separated by a hyperplane.

C. Outcomes

To understand how well our models performed, we used four metrics: accuracy score, f1 score, precision score and recall score. Accuracy score outputs the ratio of the correctly labeled test samples. Precision score tells us about the ratio of false positive samples to the sum of true positive and false positive samples $(fp/(tp + fp))$. Recall score tells us about the ratio of true positives to the sum of false negatives and true positives $(tp/(fn + tp))$. F1 score can be interpreted as a weighted average of precision and recall scores $(F1 = 2 * precision * recall/(precision + recall))$. It reaches its best value at 1 and worst value at 0. We have tabulated all the metrics for three models below.

Metrics	Multinomial Naive Bayes	LVC
train accuracy	0.45	0.53
test accuracy	0.24	0.23
f1 score	0.24	0.21
precision score	0.24	0.21
recall score	0.24	0.23

D. Prevention of Overfitting and Underfitting

Since both the training and testing errors are high, our model currently underfits. We will include more features, as well as replacing TF-IDF with more complex vectorizations such as word and sentence embedding. We will also experiment with more complex ML models such as neural networks. Furthermore, we speculated that the models performed poorly because we directly used training and testing csv files without concatenating them and shuffle. If our model overfits, we would use cross validation (k-fold) and regularization (l1 or l2) to reduce overfitting.

V. Next Step

For our next step, we would like to experiment with word embedding such as Word2Vec. Furthermore, sentence embedding, such as Infersend and Doc2Vec, can add the representation of word vectors and allow us to detect the intent of the tweet. Other than alternating the vectorization of texts, we would like to concatenate word or sentence vectors with location and date features.

Additionally, we will use 10-fold cross-validation to experiment with different training models, for example, ordinal logistic regression and neural networks such as RNN and LSTM. We would also tune the hyperparameters for models such as Linear SVC. We will compare how well these models perform.