**Sentiment Analysis of Covid Tweets**

Group 11  
Ausmita Barman | Divya Selvaraj | Mohammad Arifur Rahman

**PROBLEM STATEMENT AND MOTIVATION**

In the past two decades, with the advancement of technology, mostly mobile and computer devices, number of web users have grown exponentially which produces huge volumes of data over the internet. Users of the news forums, discussion groups, blogs, review sites leave their opinion about products, movies, foods, life events or practically everything that generate a large collection of textual information enriched with useful knowledge. Extracting all the actionable information from web reviews is essential in today’s world and sentiment analysis plays an important role in these. The main goal of this project is to introduce a real-time system for sentiment analysis on Twitter data about the coronavirus pandemic. The projected system objects to find the optimal machine learning model that achieves the best performance for coronavirus sentiment analysis prediction and then use the predicted results in real-time which will help healthcare administrations, medical organization, and society monitor current and future covid situation.

**INTRODUCTION**

The COVID-19 or coronavirus pandemic is an ongoing pandemic of the novel coronavirus disease. It was first identified in December 2019 in Wuhan, China and rapidly spread to the rest of the world, making it a pandemic by March 2020. The virus continued to spread and affect the lives of millions of people in several countries. International response to the pandemic varied by country with some imposing complete lockdown, travel restrictions, evacuation of foreign citizens, quarantine, mandatory masks and several other restrictions. As of March 2021, more than 120 million cases have been confirmed, with more than 2.66 million deaths attributed to COVID-19, making it one of the deadliest pandemics in history.

The arrival of a first-in-a-lifetime pandemic created a sudden need for average people to find and process large amounts of complicated and rapidly evolving information. There is also a wide range of changes that people had to accommodate in their everyday life like social distancing, wearing masks, online classes, etc. Many people used social media to express their opinion regarding COVID-19 and shared their experiences in facing this virus. Sentiment Analysis can be performed on these social media posts to study the sentiments on how people react to public health interventions. Since situation like this covid pandemic is unprecedented, people’s perception about the disease and restrictions can be used in future prediction of similar health crisis. With twitter being one of the widely used social media networking sites, it can be considered as a significant data source for sentiment analysis and making predictions.

Sentiment analysis is a new and popular topic in text mining research field which is a Natural Language Processing technique that helps to extract and detect subjective information from an unstructured data. The field of sentiment analysis allow us to categorize the polarity of any given text data set which could be a simple sentence, paragraph, or a whole document. Polarity is a process to group the given text or document into a specific category like positive or negative. Usually, it is categorized in two classes-positive and negative, but one can add up more classes like highly positive, highly negative or neutral. In this project our focus is to classify the data into three classes negative, neutral and positive by implementing machine learning algorithms such as Logistic Regression and Support Vector Machine (SVM). In Data preparation period, data cleaning process is used on the data set, before starting to construct the classification models. The two different feature extraction methods- Bag of Words (BOW) and Term Frequency Inverse Document Frequency (TF-IDF) are applied for both unigram and the combination of unigram bigram features. To evaluate the model and algorithm, we will collect covid related data from tweeter and group our data set into training data and test data. To check the performance, we will run the algorithm on the test data set and evaluate the performance based on the accuracy, precision, recall, and F1 measure.

**DATA**

The twitter dataset used for sentiment analysis has tweets from varying location and dates. Two datasets divided as training dataset and testing dataset has been combined for the sentiment analysis. The dataset has a target feature ‘Sentiment’, which is tagged using 5 classes from ‘Highly Negative’ to ‘Highly Positive’. The target feature is then mapped to 3 target classes as ‘negative’, ’neutral’ and ‘positive’ for analysis.

**Exploratory Data Analysis (EDA)**

Some of the exploratory data analysis performed on the dataset are shown below:

a) Tweets grouped by dates:From the plot, it shows that the maximum number of tweets were done on 20 March 2020 (3500 tweets approx.)

**Chart, line chart

Description automatically generated**

b) Tweets grouped by hashtags:Bar plot shows the top 10 hashtags used for covid related tweets with #coronavirus being the most used.

**Chart, bar chart

Description automatically generated**

c) Tweets grouped by Sentiment:Pie chart shows class distribution of tweets (target feature with 3 classes – positive, negative and neutral) It shows that the dataset has an uneven class distribution with highest number of positive tweets followed by negative tweets and least number of neutral sentiment tweets.

**Chart, pie chart

Description automatically generated**

d) Most Used words:Wordclouds with most used words in tweets classified as Positive sentiment Vs Negative Sentiment.

**Text

Description automatically generated**

e) Mentions/Tags:Bar plot below shows the top mentions/tags on covid tweets, with realDonaldTrump tagged the most.

**Chart, bar chart

Description automatically generated**

**Data Preparation**

It is important to preprocess the data before execution of any classification algorithm. Basic preprocessing is applied to each reviews of the dataset which involves of lowercasing, stemming and removing noise characters and stop words. The following data preparation techniques are applied: Data Cleaning, Stemming, Tokenization, Remove Stop Words.

a) Data Cleaning: The raw data is really disorganized which contains excessive punctuation, HTML tags, multiple spaces and numbers. In order to facilitate the data cleaning, we will try to find and remove noise characters. Subsequently, the data are cleaned by removing HTML tags, non-alphabetic characters, punctuation, which have no purpose for sentiment analysis. Most of the data cleaning in this step has been performed by using Regular Expression and BeautifulSoup.

b) Stemming: Stemming is an essential task in natural language processing which reduces a word to its stem or root format. It is a process of replacing a word by its morphological roots by using stemming algorithm which also referred as stemmer. The Porter Stemmer is one of the most popular stemming methods in data mining and information retrieval. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes. The following table is providing an example of words and their morphological root after using the Porter Stemmer algorithm.

|  |  |
| --- | --- |
| Words | Stemmed word |
| Fighting  Fights | Fight |
| Connected  Connecting  Connection  Connections | Connect |

c) Tokenization: Tokenization is the process of breaking or splitting a string, raw text into a list of tokens or small chunks. To prepare the data for feature extraction, first we split the data into individual words by using nltktokenization and then turn each word into lowercase. Here is an example of tokenization.

Input: It is not cool that ping pong is excluded in rio 2016.

Output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| It | is | not | cool | that | ping |
| pong | is | excluded | in | rio | 2016 |

d) Stop words removal: The final step of data preparation is to remove stop words. In natural language, stop words are very frequently used (such as “the”, “a”, “an”, “in” etc., also the auxiliary verbs and adverbs) and low information words which have very little meaning and do not carry any sentiment of the text. During the stopwords removal process, the words included in the “NLTK library stopwords” are removed. Moreover, removing stop words really help to speeding up the models, since we are reducing the amount of data.

Input: It is not cool that ping pong is excluded in rio 2016.

|  |  |
| --- | --- |
| After Tokenization | After Stopwords removal |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | It | is | not | cool | that | ping | | pong | is | excluded | in | rio | 2016 | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | cool | ping | pong | excluded | rio | 2016 | |

A graphical description of the process steps

Start

Twitter Dataset

Classification Model

Data Exploration

Logistic Regression

End

Labelled Reviews

Bag of Words

TF-IDF

N-Gram

Feature Extraction

Stop word removal

Tokenization

Stemming

Data cleaning

Data Preparation

Ensemble

SGD Classifier

SVM

Naïve Bayes

**Feature Extraction**

The feature extraction or vectorization process is an important step for analyzing text which convert text into a matrix (or vector) of features. Since, the machine learning algorithms cannot work on the raw text directly, we need to use some feature extraction techniques that can transform the text data into numeric representations so that the data can be comprehensible by machine learning algorithms. In this paper, we use 3 different feature extraction techniques:

a) Bag of Words: The popular machine learning based feature extraction technique is Bag of Words and it is a fundamental method to transform words into a set of features. In text mining process, the BOW model is grounded on natural language processing (NLP) and information retrieval (IR) that is typically used in document classification where each word is used as a feature for training the classifier. The BOW works as a dictionary which stores distinct words along with a mapping of the words to their frequency of occurrence of that particular word in the dataset regardless their orders.

b) TF-IDF: The TF-IDF is an important feature extraction technique which stands for term frequency-inverse document frequency. TF-IDF is a statistic that highlights a specific issue which holds great importance in a document but might not be too frequent in our corpus. The value of TF-IDF increases proportionally to the number of times a word appears in the document but decreases by the frequency of the word in the corpus.

TF-IDF is combined of two sub-parts, which are Term Frequency (TF) and Inverse Term Frequency (ITF)Term Frequency implies the frequency of the term within the document. It specifies that how many times a word can be find within a document.

The inverse document frequency attempts to find out the rare or the frequent a term across the document in the entire corpus. The higher the value of IDF reflects that those words appear in very few documents across the corpus.

TFIDF can be measured by the product of TF and IDF.

c) N-Gram: In the Bag of Word and TF-IDF vectorization techniques stated above, a unique word is represented in each column of the matrix, that means a single word is being used as the feature to predict the sentiment of the movie review. Though, the features can be expanded to a group of consecutive words which is known as the n-grams technique. For example, n-grams of size 2 also known as bigram will vectorize the data as a matrix with column containing a sequence of 2 consecutive words appeared in the dataset. This specific technique works more efficiently if the phares carries more information for the prediction than individual word. In our report we implemented unigram and unigram bigram combination technique for feature extraction.

CLASSIFICATION MODEL

The main purpose of this project is classification of reviews which is the process of assigning labels to the tweets whose label is unknown. We proceed in two steps to select classification models for this project. Firstly, for this classification task, we implemented our baseline ML models such as Naïve Bayes Classifier, Linear Discriminant Analysis, SVM Classifier and Logistic Regression. Secondly, for preliminary models, our plan is to choose two best ML baseline and add two new models which are SGD Classifier and ensemble methods.

As a baseline model, Naive Bayes algorithm is a widely used algorithm for document classification because of its simplicity and effectiveness. It is a probabilistic machine learning model which based on the idea to estimate the probabilities of categories given a test document by using the joint probabilities of words and categories. Support vector machines, a statistical classification method, also known as SVM is a discriminative classifier which is considered one of the best text classification methods. Logistic regression is a classification technique which can be used for classification problems when thresholds are used on the probabilities predicted for each class. For further evaluation, we approach with SGD classier and ensemble using voting classifier. To build our ensemble model, we combined our previous two models, which are logistic regression and SVM, with SGD classifier and for feature extraction technique we only use TF-IDF for both approaches.

**EXPERIMENTAL RESULTS AND DISCUSSION**

In this section we discuss our experiment, the evaluation criteria, and the results. Our experiment contains two parts: first we consider only unigram or a single word for feature extraction in BOW and TF-IDF process but on the second part we implement unigram and bigram combination during feature extraction. Then we compare the results we get from both the approaches.

a) Evaluation Criteria: Accuracy, Precision, Recall and *F*1 measure are the methods we use in this paper to evaluate the performance of the movie review sentiment analysis.

The *F*1 measure is the harmonic mean of precision and recall as follows:

Here accuracy is the overall accuracy of certain sentiment models. In an ideal scenario, all the experimental results are measured according to the Table below and accuracy, precision and recall as explained below

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Actual | Predicted | | |
| Positive | Negative | Neutral |
| Positive | TP | FN1 | Fnt1 |
| Negative | FP1 | TN | Fnt2 |
| Neutral | FP2 | FN2 | Tnt |

b) Experimental Results: After evaluating our baseline models performance which are Multinomial Naïve Bayes, Linear Discriminant Analysis, Logistic regression and Linear SVM, we decided to remove Multinomial Naïve Bayes and Linear Discriminant Analysis’s accuracy results from comparison table. Since the accuracy result for Multinomial Naïve Bayes is 67.45% and Linear Discriminant Analysis is 77.71%.

Even though firstly, we consider Logistic regression and Linear SVM as baseline models, because of their better accuracy we use those models as our preliminary models. The following table shows the resultant performance – accuracy, F-measure, precision and recall- of BOW and TF-IDF techniques for unigram and unigram bigram combination by using Logistic Regression and SVM classifiers. It could be noticed that the accuracy for Logistic Regression with BOW is 80.94 percent which is the best result so far comparing with other accuracy results from the table below. But the accuracy for unigram bigram TF-IDF for SVM is almost similar which is 80.68 percent.

Table 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature  Extraction | Performance measure | | Logistic Regression | | SVM Classifier | |
| Unigram | Unigram  Bigram | Unigram | Unigram  Bigram |
| Bag of Words | Accuracy (%) | | 80.94 | 80.42 | 78.37 | 79.76 |
| F-Measure | positive | 0.84 | 0.83 | 0.82 | 0.83 |
| negative | 0.82 | 0.82 | 0.80 | 0.81 |
| neutral | 0.71 | 0.71 | 0.67 | 0.71 |
| Recall | positive | 0.84 | 0.83 | 0.82 | 0.82 |
| negative | 0.82 | 0.81 | 0.79 | 0.79 |
| neutral | 0.71 | 0.72 | 0.69 | 0.75 |
| Precision | positive | 0.83 | 0.83 | 0.82 | 0.83 |
| negative | 0.83 | 0.82 | 0.81 | 0.83 |
| neutral | 0.72 | 0.71 | 0.65 | 0.67 |
| TF-IDF | Accuracy (%) | | 79.12 | 78.61 | 80.68 | 79.80 |
| F-Measure | positive | 0.82 | 0.82 | 0.84 | 0.84 |
| negative | 0.81 | 0.80 | 0.82 | 0.82 |
| neutral | 0.67 | 0.66 | 0.71 | 0.71 |
| Recall | positive | 0.85 | 0.85 | 0.85 | 0.85 |
| negative | 0.81 | 0.81 | 0.82 | 0.82 |
| neutral | 0.60 | 0.58 | 0.69 | 0.69 |
| Precision | positive | 0.79 | 0.79 | 0.83 | 0.83 |
| negative | 0.80 | 0.79 | 0.82 | 0.82 |
| neutral | 0.76 | 0.76 | 0.73 | 0.73 |

In table 2 we discussed the performance results of SGD classier and ensemble model (logistic regression, SGD and LinearSVC) using voting classifier). Comparing the results with Table 1 it is noticeable that, we achieved better accuracy for SGD and ensemble method. The following table is presenting the results for SGD and ensemble method.

Table 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature  Extraction | Performance measure | | SGD Classifier | Ensemble Method |
| TF-IDF | Accuracy (%) | | 82.09 | 82.44 |
| F-Measure | positive | 0.85 | 0.85 |
| negative | 0.83 | 0.83 |
| neutral | 0.74 | 0.74 |
| Recall | positive | 0.85 | 0.86 |
| negative | 0.82 | 0.83 |
| neutral | 0.76 | 0.73 |
| Precision | positive | 0.84 | 0.84 |
| negative | 0.84 | 0.84 |
| neutral | 0.73 | 0.76 |

Chart, box and whisker chart

Description automatically generated

We also tried to apply deep learning model with keras on the dataset and the accuracy we were able to get an accuracy of 73.8 %. This needs further tuning to improve accuracy and will be considered as future work.

CONCLUSION AND FUTURE WORK

In conclusion, by comparing the performance results of all the models and techniques for unigram and combination of unigram and bigram, it is clear that the ensemble classifier with TF-IDF unigram technique (82.44 percent) outperformed all the other approaches.

One of the key improvements that can be integrated as we move ahead in this project is to include deep learning neural network model. Another important point of improvement is removing the negation words (e. g. no, not) from stop words which can change the sentiment of words when considering bigram or more. Also, adding feature selection techniques to reduce the huge number of features we get from our dataset can help to improve the accuracy.