INDENG 243: Group 10 Twitter Covid19

Haolan Mai, Yucong Li, Jiaqing Li,Ruoxin Xu, Makie Maekawa, Bennett Cohen

IEOR Department, UC Berkeley

yuli7852@berkeley.edu,hmaibear28@berkeley.edu,rxxu@berkeley.edu

bennettcohen@berkeley.edu,makie.maekawa@berkeley.edu,jiaqing_li@berkeley.edu

May 12, 2023

1 Introduction and Project Objectives

The COVID-19 pandemic has significantly impacted society, prompting varied public reactions. Social media platforms like Twitter offer invaluable insights into public sentiments and discussion topics, which can inform decision-makers' responses, policies, and communication strategies for more effective pandemic management. Our project consists of three modules: Data preprocessing, Model Development and Evaluation, and Analytics Communication and Target Audience.

In Data preprocessing, we validate, process, and clean data.Model Development and Evaluation involves performing Semantic Clustering and Multi-output Classification on COVID-19 tweets, selecting and evaluating suitable models while discussing their rationale and limitations. In Analytics Communication and Target Audience, we identify key stakeholders like public health officials and their decision-making needs, developing an interactive tool to explore sentiment analysis results and effectively communicate decision implications. Our project aims to provide valuable insights into public sentiment and key topics related to COVID-19, allowing informed decision-making in an interactive and accessible format. We hope to contribute to improved pandemic management and better-targeted communication strategies for public health officials.

2 Data Description and Source

In this project, we collected tweets data for 2020 and 2023 from Kaggle and the Twitter API via Python, and we extracted tweets that contained the hashtag `#covid19`. To access the Twitter API, we modified a script found here: https://github.com/gabrielpreda/COVID-19-tweets.

- `COVID2020.csv` contains tweets between July 25th, 2020 and August 29th, 2020 (179,108 rows).
- `COVID2023.csv` contains tweets between January 29th, 2023 and February 8th, 2023 (24,352 rows).

Please be advised that due to the access level of the Twitter developer account, we can only crawl the recent-7-day of tweets, so the size of the 2023 data set is much smaller compared to the 2020 data set.

Column Name	Description
user_name	account user name
user_location	account user location
user_description	account user description
user_created	when the account was created
user_followers	# of followers of the account
user_friends	# of friends of the user
user_favorites	# of favorites of the user
user_verifed	boolean if the user is verified
data	date tweeted
text	text of the tweet
hashtags	list of strings of hashtags
source	source of tweet
is_retweet	boolean if tweet is retweet or original

Table 1: Data source colum descriptions.

Data Limitations: There are two limitations that we see in the data we've collected:

1. User Location/Regional Bias: The data does not cover every country in the world.
2. Time Zone Bias: The time displayed is the PDT after unified conversion.

To mitigate the negative impact of data limitations, we made the following adjustments:

1. We selected a representative number of countries from the data.
2. When we analyzed time related problems, we applied the unit of day rather than specific time.

3 Data Preprocessing

For both of the COVID2020 and COVID2023 tables, we apply the following steps to clean the data, along with a brief rationale.

1. Drop duplicate columns, defined as directly identical tweet text column. We combined the Kaggle data with our own, so there was an overlap.
2. Drop content from users with no follows (i.e. user follower = 0) because a user without followers does not produce content that spreads to other users. Some people may have created accounts simply to complain, so it may not be valuable in understanding the distribution of sentiment.
3. Drop rows with N/A values. Because we have many rows of data, but only a few columns, all of which are vital to describing a tweet, it is fine to drop any N/A rows.
4. Process the tweet text data (text column) using the most common text cleaning methods covered.
    - Convert all letters to lowercase
    - Removed URL
    - Deleted any numbers in the text column
    - Deleted punctuation, stop words from NLTK, emoticons/emojis, and abbreviations.
    - Stripped leading/trailing and extra spaces.
5. Removed emojis from user name, user location, and user description

4 Model Development and Evaluation

Our model was designed based on the following system diagram. By following this system diagram, we aim to develop a model that effectively captures the underlying patterns in the text data and accurately assigns labels based on clustering results.

4.1 Word Embeddings

Word embeddings are one of the popular techniques for digitizing text data used in the field of natural language processing (NLP). Since text data cannot simply be compared, it makes digitized text data, and we can analyze it. To capture the word's meaning, it generates a vector representing each word in the multidimensional space. Word embeddings can be used to compute the semantic similarity of words and perform tasks such as clustering, classification, and searching of text data. We used Word2Vec, among the popular algorithms for word embeddings. This algorithm differs from supervised learning because it learns word meanings from large amounts of text data. We could load in a Word2Vec style word-embedding model to get better vector representations than can be found using TFIDF. However, word embeddings have some limitations. If the same word has multiple meanings, there is only one vector representation of that word, which cannot capture all the meanings of that word. Also, word embeddings are generated from text data and can therefore depend on text data.

4.2 Label Generation

Label Generation uses vectors that are converted by word embeddings for text data.to predict labels for each text. Labels are computed for the semantic similarity of text data, and each text is labeled based on that similarity. By using Label Generation, we can label large amounts of text data efficiently. On the other hand, Label Generation has some limitations. In order to label text data, we need suitable training data. Also, note that word embeddings are dependent on the text data and may result in different models for different kinds of text data.

4.3 KMeans Clustering

For the K-Means Method, we define a function called train_K-Means that takes in three parameters:
1. X: A dataset to train on.
2. n_cluster_values: A list of integers representing the number of clusters to try in K-Means.
3. verbose: A boolean variable to determine whether to output additional logging information.

The function uses the K-Means algorithm to fit a model to the dataset for each value in n_cluster_values. It computes the silhouette score for each of these models and returns a dictionary containing the silhouette scores for each n_clusters value.

If verbose is set to True, the function will output logging information about the time taken to train the model for each n_clusters value. We will select a quasi-optimal value of n clusters by selecting the one on the "elbow" of the plot.

While most supervised learning algorithms benefit greatly from using a standard K-Fold cross-validation implementation using Scikit-Learn's GridSearchCV object, for K-Means clustering, it's quite common to use the "elbow method." Instead of directly minimizing some evaluation metric like SSE or Inertia, we instead will search over a space of reasonable values of n clusters and plot the corresponding silhouette scores. We search over a space n cluster values from 2 to 9 and use the elbow method to select the value with the highest silhouette score as our final model. We then plot the results below.

We plotted the results of the K-Means exercise by creating a graph of contour scores versus the number of test clusters. We use this code to provide a simple and effective way to visualize the performance of the K-Means algorithm with different cluster numbers, which helps select the best cluster number for a given data set. In line plot, we can see that the optimal clustering of data in 2023 is 2. Through line plot, we find that the optimal clustering of data in 2020 is 8.

4.4 Multilabel Classification

Multilabel classification is a technique for solving the problem of assigning multiple labels to each data point, where data points belong to multiple related categories or attributes. We chose LogisticRegression, LinearDiscriminantAnalysis, Random Forest, Multi-layer Perceptron regressor, and GradientBoostingClassifier as machine learning models for multi-label classification. We trained them and compared their performance.

4.4.1 Hyperparameter Tuning

We used grid search to tune the GradientBoostingClassifier hyperparameters to maximize model performance. We found a suitable combination of hyperparameters by the following procedure.

1. We defined a parameter grid to explore. We decided to experiment with parameters such as learning rate, number of estimators, maximum depth of the tree, the minimum number of samples to split a node, the minimum number of samples required for leaf nodes, and a fraction of sub samples.

2. Initialized the GradientBoostingClassifier with the following MultiOutputClassifier. This class wrapper provides the ability to create a GradientBoostingClassifier for multiple labels for multi-label classification.

3. We used GridSearchCV to perform a grid search. GridSearchCV is a method for finding the optimal combination of hyperparameters by combining a specified parameter grid and cross-validation. We split the training data into 5 folds for cross-validation and used the accuracy rate as the evaluation metric.

4. We got the best model found by GridSearchCV. This yielded the GradientBoostingClassifier model with the best-performing combination of hyperparameters.

4.4.2 Performance Measurement

We load our trained models and compute the performance for each of the models for comparison. Because we are doing a multilabel (not multiclass) classification problem, we must be smart with the metrics we should use. While accuracy is generally fine with binary classification problems, it isn't ideal for this. For instance, suppose a target label is [1, 1, 0, 0, 1], meaning the first two labels and the last label apply. If a model predicts [1, 1, 0, 0, 0], we argue this is a fairly good model because it got 4/5 correct. However, the vectors are not the same, so if accuracy were our primary metric, this would be a score of 0. In short, accuracy is not a detailed enough metric to classify performance. Instead, we chose to use the Hamming Loss, which is a measure representing the proportion of incorrect labels. In the example above, 1 of the 5 labels was incorrect. During hyperparameter tuning, we still decided to leave the scoring metric to be accurate because increasing accuracy will always lead to better Hamming Loss.

4.5 Limitations

- Lack of domain-specific knowledge: GPT is a language model trained on a large corpus of text, but it may not have domain-specific knowledge about Covid19. This could lead to the generation of inaccurate or irrelevant class names.

- Bias in generated class names: GPT may generate class names based on biases present in its training data, which could lead to the generation of class names that are not representative of the true classes.

- Inconsistencies in generated class names: GPT may generate multiple class names for the same concept, leading to inconsistencies and confusion.

- Difficulty in accurately evaluating performance: Without labeled data, it can be difficult to evaluate the accuracy of the generated class names. This makes it challenging to know whether the model is making informed predictions or simply guessing.

- Limited interpretability: Without labeled data, it can be challenging to interpret why the model is making certain predictions. This can make it difficult to identify and address any biases or inaccuracies in the generated class names.

5 Analytics Communication and Target Audience

5.1 Target Audience and Decision Making

As a data analytics service provider for the CDC, our focus is on providing insights into public sentiment, misinformation, and key topics of discussion surrounding COVID-19 in order to assist them in tailoring their communication

strategies and public health campaigns more effectively.

For example, our sentiment classification model can help public health officials create targeted messaging to improve vaccine uptake by understanding people's attitudes toward COVID-19 vaccines. Furthermore, our semantic clustering model can provide retailers with insights into consumers' buying habits and spending patterns during the pandemic, helping them adjust their business models and sales strategies accordingly.

5.2 Communicating Model Results

Our modeling approach could be summarized into four main steps:

1) Converting tweet text into numbers representing different words/tokens. During the word embedding process.
2) Using a clustering model to group the tweets into 8 different categories based on the content of the tweet.
3) Using ChatGPT to sample 50-100 tweets from each cluster and describe the tweets using 5-10 keywords.
4) Training a Multi-output classification model to assign labels to each tweet in our dataset.

The different training models that we used are logistic regression, linear discriminant analysis, multi-layer perceptrons, random forests, gradient boosting, etc. Then we use *Hamming Loss* to score these models by measuring the proportion of labels that are incorrectly assigned. Linear Discriminant Analysis (LDA) and Logistic Regression (LR) are the best two models due to their lowest hamming loss value.

5.2.1 Interactive EDA

We want to give the CDC the ability to observe macro-trends from the tweet data for the following concepts. Thus, we provide them with an interactive exploratory data analysis demo to interact with. Every plot type can be filtered by *Date Range (07/24/2020-08/30/2020)* and *Location (All Countries, United States, Canada, South Africa, Switzerland, India, United Kingdom)*. The audience could observe:

1) Most common words in this tweets dataset *(Bar Plot of Most Common Words in Tweets and Word Cloud of Most Common Words)*
2) Length of tweets *(Distribution of Length of Tweets)*
3) Number of tweets *(Time-series Plot of Tweet Counts)*

These interactions, provided to the CDC, could have a significant impact on their understanding of macro-trends in tweet data related to COVID-19 thus improving their communication effectiveness and public health campaigns. Additionally, by providing filtering options for date range and location, the CDC can focus on their specific areas of interest and potentially adjust their messaging based on location-specific trends.

5.2.2 Real-Time Inference

Since our dataset was not crawled recently, many more #COVID19 tweets have been created since then with new protocols, vaccine information, regulation, etc. Thus, we built the *Real-Time Inference* demo to let our clients explore new Twitter trends regarding #COVID19 topic.

This demo grants real organizations the ability to choose the model they like when exploring trends. Even if there are similar performances between the two best models, they can give slightly different results. Our clients could compare these two models to balance the decisions that they make. Our client could simply enter COVID-19-related tweets that they are interested in, select different models, and get a group of keywords according to the input tweets. For example, if they would like to make new governmental decisions on vaccination, they could enter vaccination-related tweets under hashtags such as #vaccine or #vaccination, to get public sentiment keywords.

5.2.3 User Tweets Micro-Trends

Our *User Tweets Micro-Trends* demo is based on pre-trained models and machine learning models. Users can input a Twitter username, select a model type, and then get the predicted label distribution of the user's tweets. There are three different types of models to choose from: "KMeans + GPT API," "Logistic Regression," and "Linear

Discriminant Analysis".

Our demo could help health organizations visualize the micro-trends related to COVID-19 by analyzing subtle trends of highly influential Twitter users and gaining insights from them. Our demo will predict this user's high-frequency words and show them in numbers and bar plots. Moreover, this demo could also assist organizations in understanding the opinions and attitudes of specific users toward COVID-19 topics and conducting further comparative analyses.

Analyzing highly influential users like Joe Biden and Shankar Prasad can help the organization make some quarantine plans in advance based on the content of the tweets. For instance, CDC could obtain that the words "recoveries," "testing," "updates," etc. have been discussed by Shankar Prasad frequently. Combining these words with the professional background of Shankar Prasad as the minister of Electronics and Information Technology, CDC would receive effective policy guidance and adjust their decision accordingly. When the organization considers these influential voices and amplifies them, its perceived authority and trustworthiness of its information will be solidified.

6 Conclusion and Future Work

We used Kaggle's 2020 data and obtained our own 2023 data to clean, analyze, and compare them to help us with future sentiment analysis. We repeatedly demonstrated the availability of the data and found that in both the 2020 and 2023 datasets, the most frequent word on Twitter was "COVID". This indicates that there has been a significant amount of discussion about COVID-19 by users over the past three years. Since both our 2020 and 2023 datasets are related to COVID, this positive signal ensures that our future pandemic sentiment analysis remains on-topic.

In addition, we also found some subtle differences between the commonly used vocabulary in 2020 and 2023. In 2020, aside from "COVID-19", the words that were more common on Twitter were "death", "growth", and "lockdown". But by 2023, aside from "COVID-19", the words that were more common on Twitter had shifted to "vaccine" and "health". We noticed that people are shifting from being fearful of COVID-19 to using the vaccine to fight it. As attitudes improve and the death rate decreases with the availability of vaccines, the topics and keywords of discussion have changed.

Although our data has some limitations, such as differences in user location and time zone, and Twitter not covering all users and areas, we gained an in-depth understanding of public perceptions of COVID-19 and their evolution over time, which can enable policymakers to explore this information in a meaningful way.

Next, we will perform sentiment analysis on the COVID-19 tweet dataset based on our exploration and choose and apply appropriate models and methods to achieve our sentiment analysis goals. As a CDC data analytics service provider, our approach involves using machine learning methods such as unsupervised clustering and supervised multi-label classification to categorize COVID-19 related tweets into meaningful topics and identify public trends and areas of concern. Through analysis, we have selected the best two models, linear discriminant analysis (LDA) and logistic regression (LR). We have developed three demos, interactive EDA, real-time inference, and user tweet micro-trends. In the interactive EDA demo, we can understand national sentiment and concerns related to COVID-19 by observing the most common words in tweets, the length of tweets, and the number of tweets over a period of time. By providing filtering options for date ranges and locations, the CDC can focus on specific areas of interest and use these insights to tailor their messaging to address specific issues and promote government decisions. We also built a real-time inference demo that allows customers to explore new Twitter trends on COVID-19 topics by applying models. The CDC can choose the model they want to apply while exploring trends and use them to obtain real-time labels to categorize the latest data. Finally, we presented an inference tool based on pre-trained and machine learning models (User Tweet Micro-Trends). The purpose of this presentation is to gain insights by analyzing subtle trends of influential Twitter users related to COVID-19 and help organizations visualize micro trends related to COVID-19.

In summary, the CDC has conducted large-scale data analysis of COVID-19 using machine learning methods and the latest technology tools such as interactive EDA, real-time inference, and user tweet micro-trends, and gained meaningful insights related to the disease. These tools can not only help the CDC better understand public attitudes and concerns about COVID-19 but also provide real-time data analysis and predictions to help make better decisions on public health issues.

7 Appendix

	user_followers	user_friends	user_favourites
count	179108.00	179108.00	179108.00
mean	109055.53	2121.70	14444.11
std	841467.00	9162.55	44522.70
min	0.00	0.00	0.00
25%	172.00	148.00	206.00
50%	992.00	542.00	1791.00
75%	5284.00	1725.25	9388.00
max	49442559.00	497363.00	2047197.00

Table 2: Distribution of COVID2020 numerical features.

	user_followers	user_friends	user_favourites
count	24352.00	24352.00	24352.00
mean	70776.39	2506.54	20891.31
std	715123.12	18074.44	56820.15
min	0.00	0.00	0.00
25%	208.00	127.00	289.00
50%	1125.00	577.00	2975.00
75%	3892.00	1870.00	14439.75
max	16216538.00	477573.00	1342610.00

Table 3: Distribution of COVID2023 numerical features.

	Total	Percent	Data Types
user_name	0	0.0	object
user_location	36771	20.5300	object
user_description	10286	5.7429	object
user_created	0	0.0	object
user_followers	0	0.0	int64
user_friends	0	0.0	int64
user_favourites	0	0.0	int64
user_verified	0	0.0	bool
date	0	0.0	object
text	0	0.0	object
hashtags	51334	28.660	object
source	77	0.0429	object
is_retweet	0	0.0	bool

Table 4: Distribution of missing values in COVID2020

	Total	Percent	Data Types
user_name	0	0.0	object
user_location	6152	25.262812	object
user_description	1256	5.157687	object
user_created	0	0.0	object
user_followers	0	0.0	int64
user_friends	0	0.0	int64
user_favourites	0	0.0	int64
user_verified	0	0.0	bool
date	0	0.0	object
text	0	0.0	object
hashtags	6724	27.611695	object
source	0	0.0	object
is_retweet	0	0.0	bool

Table 5: Distribution of missing values in `COVID2023`

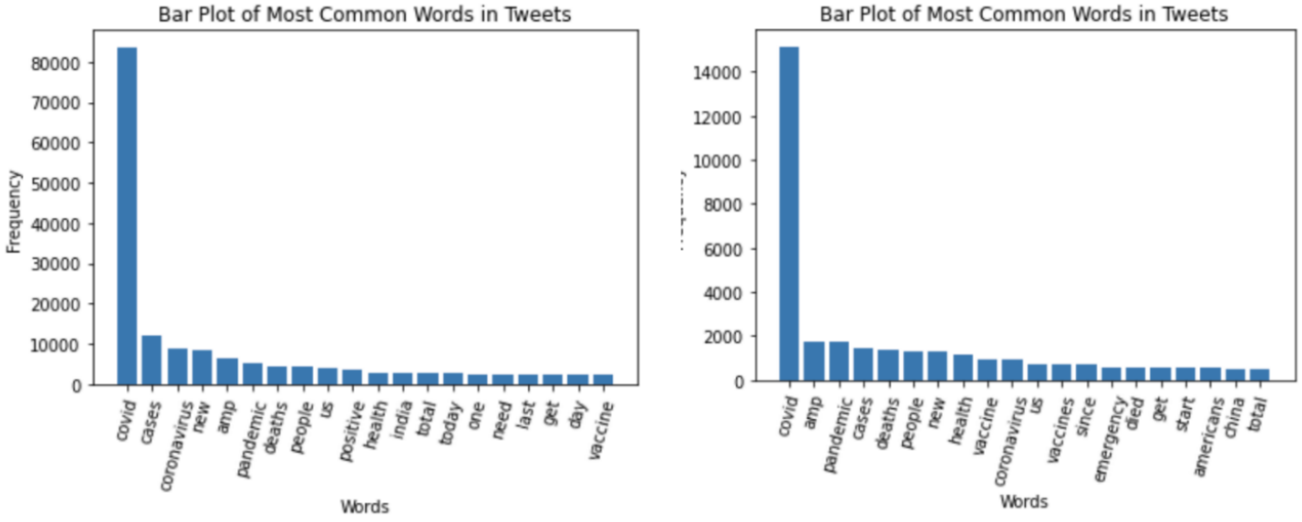

Figure 1: Bar plots of the frequency of 12 of the 20 most common words in the text column of our dataset

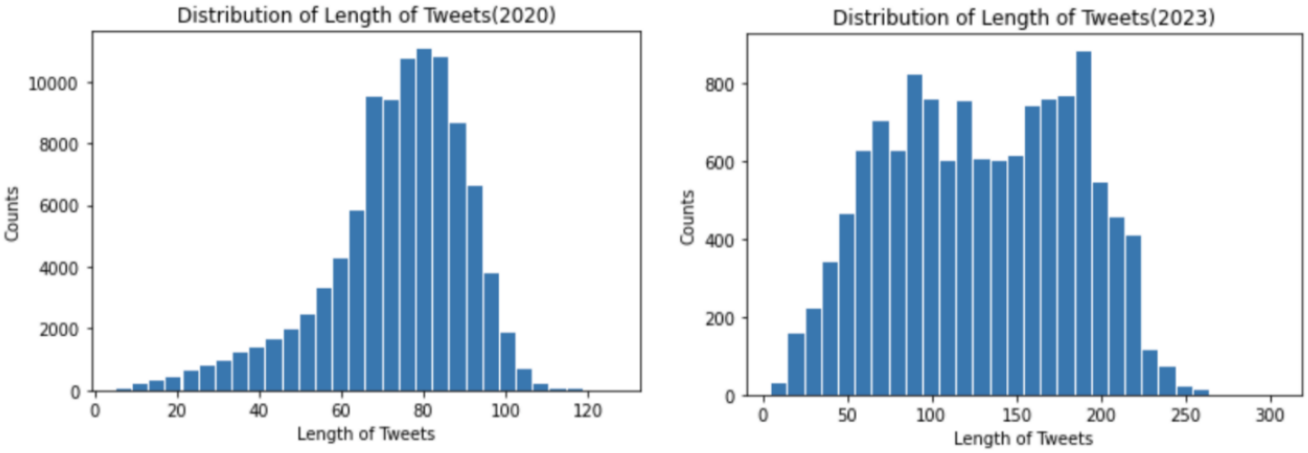

Figure 2: Histogram of the distributions of tweet lengths in 2020 and 2023

User_location for covid2020 User_locationin for covid2023

Figure 3: Word cloud of the tweet location distribution.

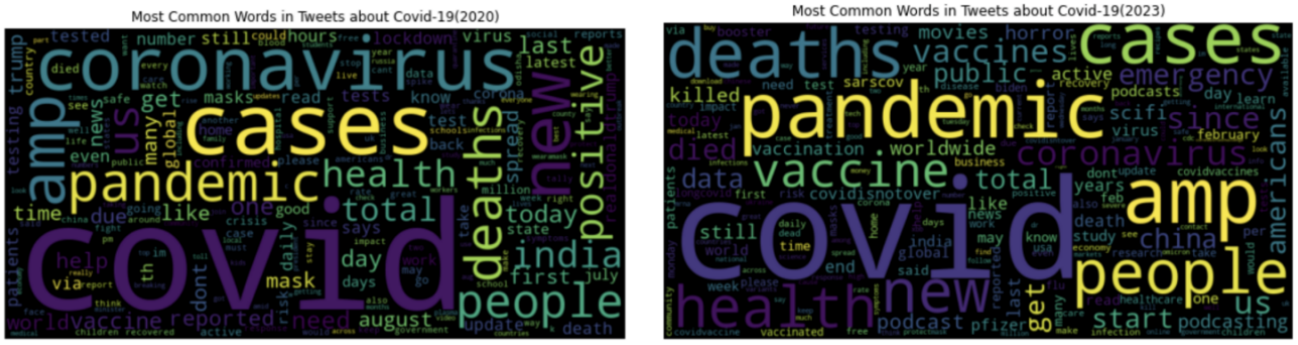

Figure 4: Word cloud of the most common words in all tweets

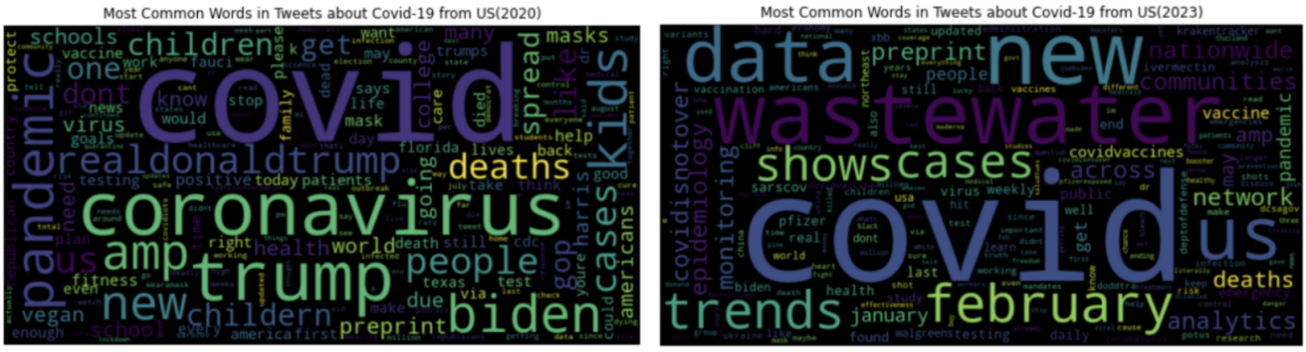

Figure 5: Word cloud of the most common words in tweets from the United States

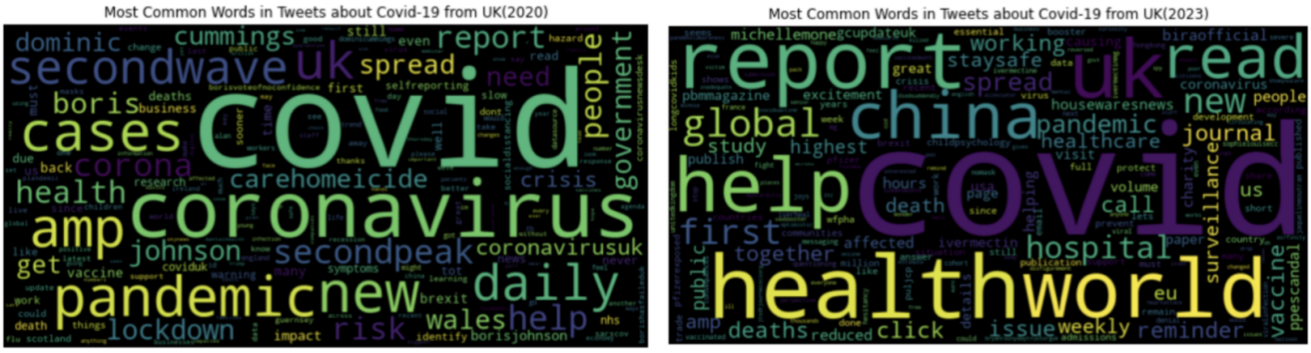

Figure 6: Word cloud of the most common words in tweets from the United Kingdom

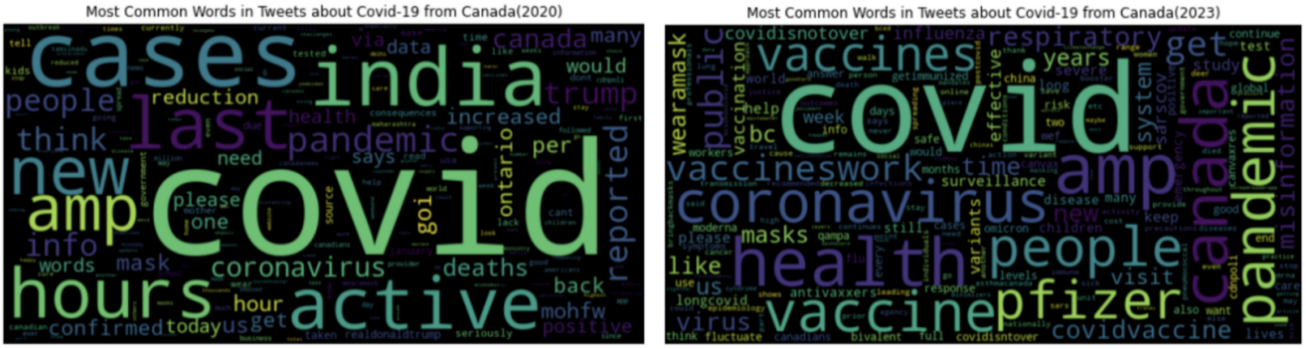

Figure 7: Word cloud of the most common words in tweets from Canada

Figure 8: Word cloud of the most common words in tweets from Switzerland

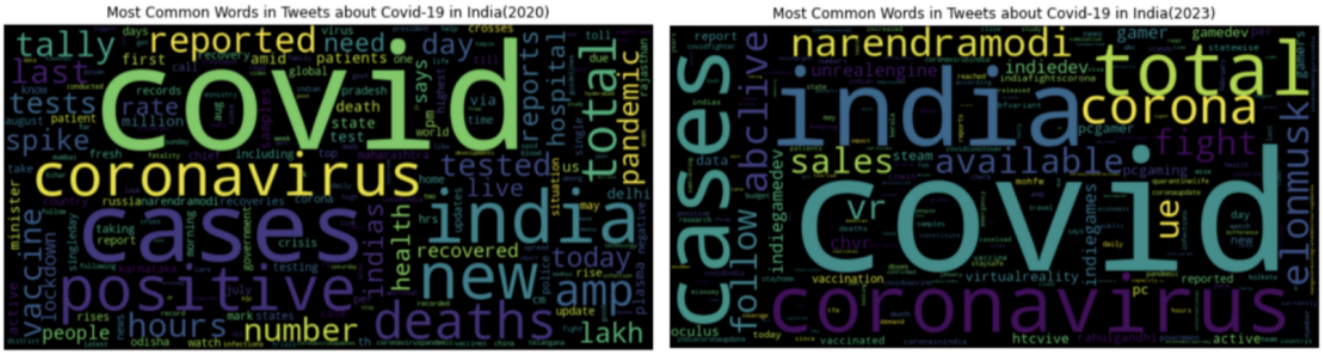

Figure 9: Word cloud of the most common words in tweets from India

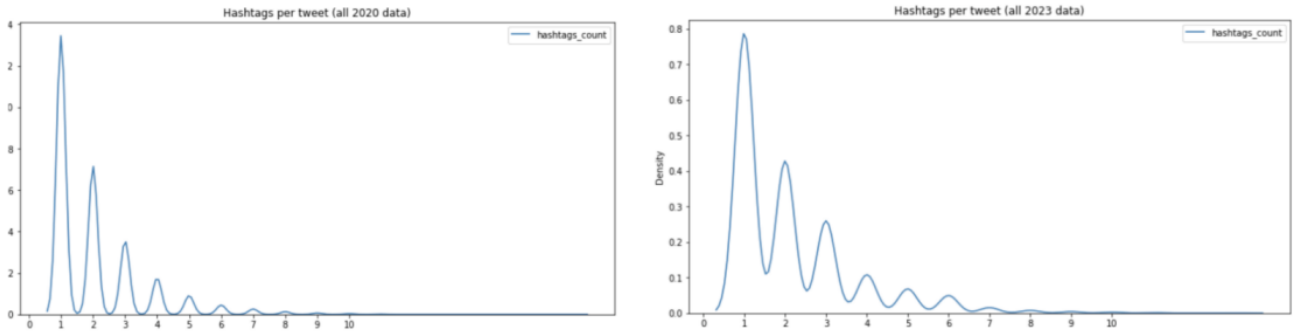

Figure 10: Distribution of the hashtag count in 2020 and 2023

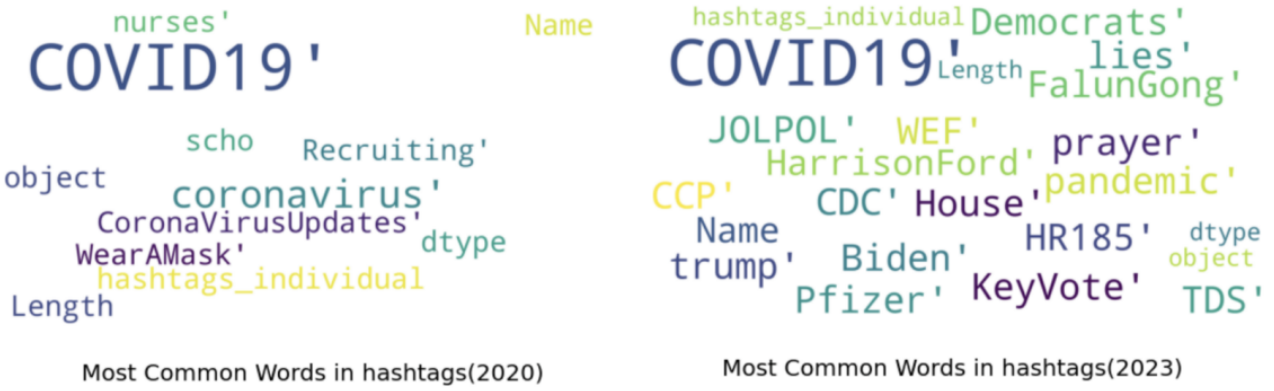

Figure 11: Word cloud of the most common hashtags in tweets in 2020 and 2023

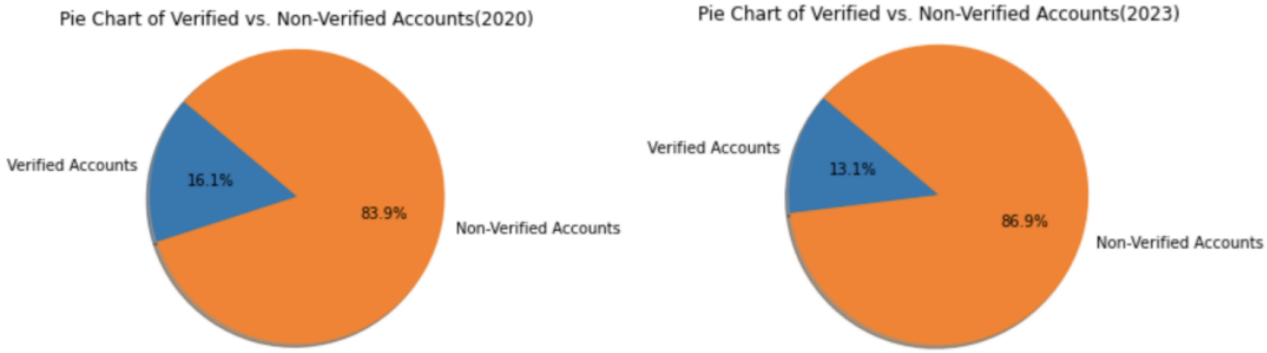

Figure 12: Pie chart of the distribution of verified accounts

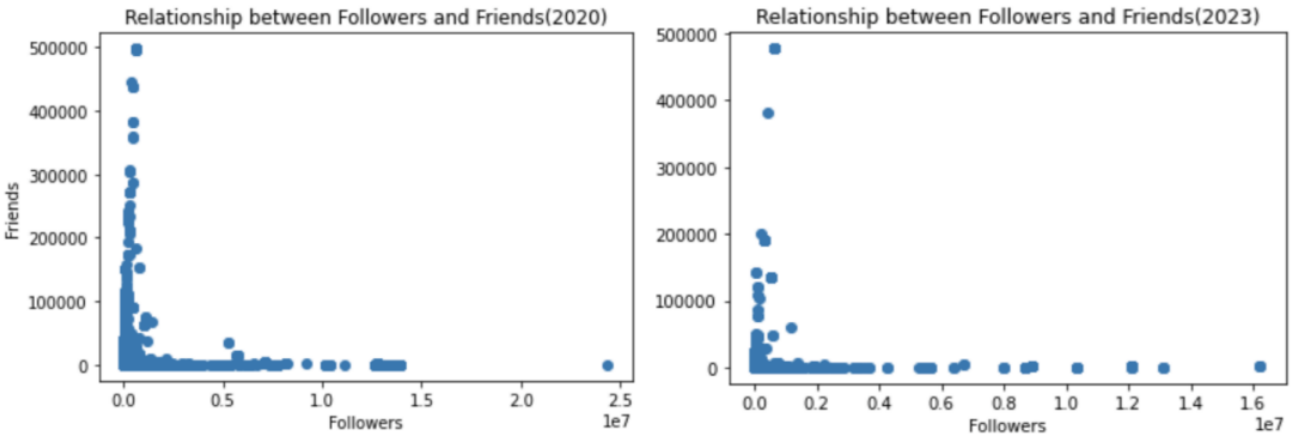

Figure 13: Scatter plot of follower count and friends count

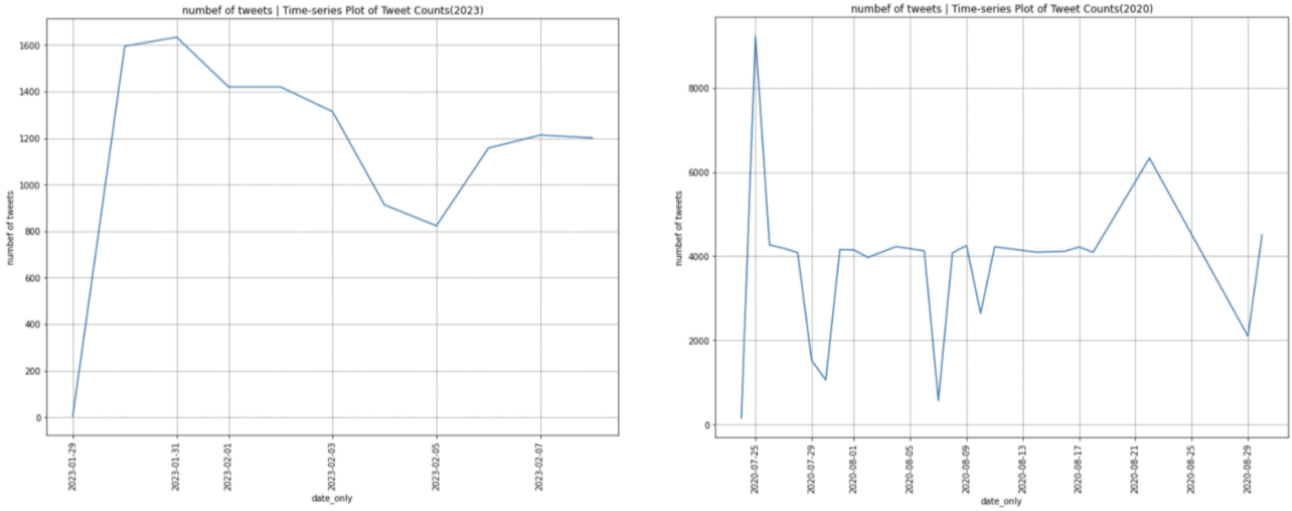

Figure 14: Plot of the number of tweets over time in 2020 and 2023

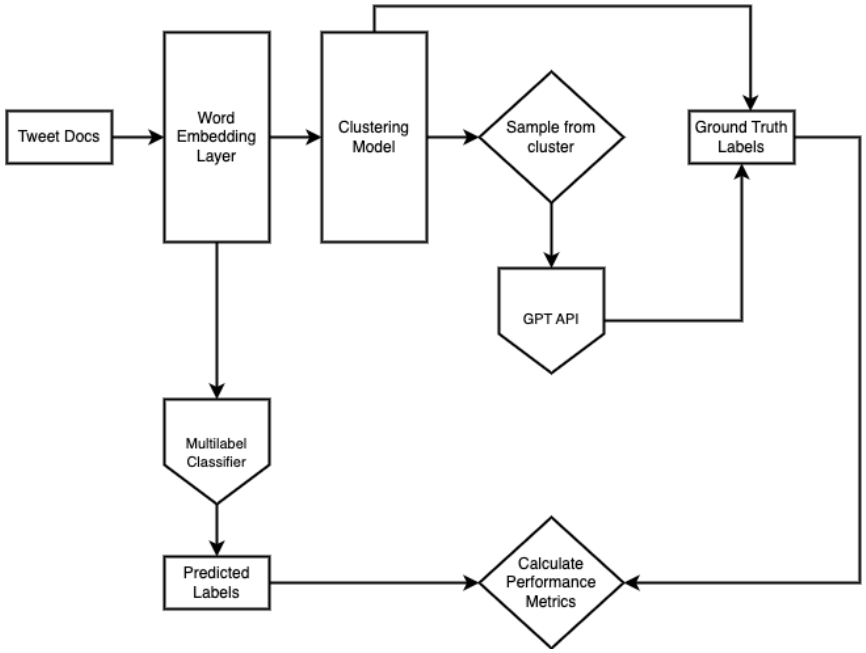

Figure 15: Model Approach Flow Diagram

	0	1	2	3	4	5	6	7	8	9	...	758
0	-0.021892	0.019932	-0.010053	-0.024508	-0.042268	-0.035663	-0.008145	0.004859	0.053373	0.033196	...	-0.004726
1	-0.007098	0.045060	-0.008168	-0.004465	-0.013927	-0.017722	-0.027128	0.008203	0.016782	0.018806	...	0.042896
2	-0.014795	0.072627	-0.024924	-0.014272	0.015880	-0.057928	-0.003245	-0.001590	0.058332	0.043341	...	0.002370
3	-0.046708	-0.003250	-0.022359	-0.033408	-0.017705	-0.051481	-0.003138	0.010111	0.034598	0.012806	...	0.003838
4	-0.033658	0.062035	-0.016353	-0.023915	-0.006017	-0.049146	0.006027	0.013803	0.062163	0.048505	...	0.002227
...	...	...	...	...	...	...	...	...	...	...	...	...
24671	-0.005804	0.095952	-0.006469	-0.014626	0.012864	0.005027	0.023838	0.031686	0.076822	0.008978	...	0.031242
24672	-0.004632	0.041510	-0.017558	-0.012113	0.003798	-0.035611	-0.013221	-0.026403	0.034207	0.041392	...	0.024190

Figure 16: Word Embeddings

	train_hamming_loss	test_hamming_loss	train_accuracy	test_accuracy
LR	0.0426	0.0412	0.6829	0.6913
LDA	0.0285	0.0294	0.8071	0.8033
MLP	0.2582	0.2625	0.0069	0.0053
RF	0.1949	0.1966	0.0046	0.0060
GBC	0.1064	0.1046	0.2143	0.2247

Figure 17: Performance comparison