Classification of Covid-19 Tweets

Hailey Thanki and Tapan Pradyot

***Abstract*— Twitter is becoming one of the most popular sources for mining data for NLP projects. Here we undertook the task of classification of global tweets at a country level. Most of the tweets are in English and contain no explicit information about the location of the user. We perform a descriptive analysis, followed by pre-processing and model implementation. We determine the extent to which the model can correctly classify these tweets due to the challenging nature of classifying tweets with multiple commonalities belonging to English speaking countries.**

I. INTRODUCTION

In the data science community, especially for Natural Language Processing problems, data from various social media platforms is becoming extremely popular. They are the primary sources of data to help understand diverse natural and social phenomena, and this has led to the development of a wide range of computational data mining tools that can extract knowledge and insights from text corresponding to different social media for various kinds of analyses. [2]

This project focuses on inferring tweet geolocation where the origin of the tweet is one of the following countries - UK, USA, Australia, New Zealand, Ireland, Canada. The project is restricted to analyses of tweets from only the 6 aforementioned countries where the native language or the primary language for communication is restricted to English. This cannot be implemented for an unfiltered stream where tweets from any location or country will be observed. We also complement our work by investigating the extent to which a classifier trained on historical tweets can be used effectively on newly harvested tweets.

We found out that determining the location of these tweets based on the text was quite challenging. This was because there is not a lot of noticeable variation in the sentence structures, vocabulary, grammar, etc. that can be translated in a way that a machine learning model could understand. This can be ascertained by the pairwise cosine similarities between the tweets from different countries. This will be discussed later in the report. Moreover, the hashtags used by users from each country were also pretty similar and also did not contain any potential of containing a distinguishing factor for categorization. This can be visually observed and will be presented later.

II. DATA

A. About the Data

The training data contains 240,000 unique tweets about Covid-19 coming from six different English-speaking countries (each country has 40,000 tweets in total). Categorical outcome variable is called 'country'. The test data 60,000 unique tweets for which we will be predicting the 'country' variable.

B. Features in the Dataset

Following are the attributes in the dataset:

- text: The text of tweet (including emojis, htmls, hashtags)
- replytoscreen_name: The Twitter screen name of the user the owner of the tweet is replying to (if any)
- is_quote: A Boolean variable that indicates if the owner of the tweet is quoting someone else's tweet
- is_retweet: A Boolean variable that indicates if the owner of the tweet is retweeting someone else's tweet
- hashtags: A list of hashtags included in the tweet
- country: The label of the country in which the tweet was posted (found only in the training dataset)
- Id: An index number associated with tweets (found only on the test dataset)

C. Descriptive Analysis of the Data and Exploratory Data Analysis before Pre-processing

We wanted to know the mean, median, maximum and minimum word count and character count of tweets (Fig. 1) and analyzing the frequency of words in these tweets (Fig. 2) before the pre-processing steps are performed.

Attribute	Aggregation Type	Value
Character Count	mean	204.98
	median	221
	max	425
	min	1
Word Count	mean	28.83
	median	30
	max	82
	min	1

Fig. 1. Descriptive analysis of the tweets before pre-processing

```
[('the', 238392),
 ('to', 173831),
 ('of', 127650),
 ('#covid19', 124143),
 ('in', 110280),
 ('a', 103720),
 ('and', 99878),
 ('is', 72453),
 ('for', 70530),
 ('on', 52333)]
```

Fig. 2. Top 10 most common words before pre-processing

D. Pre-processing the Data

The following pre-processing steps were carried out to clean the 'text' column in the dataframe which refers to the tweets.

- All the characters in the tweets were transformed into lowercase characters.
- All the links, punctuation, emojis, stopwords like 'covid', 'covid-19', 'coronavirus', 'virus', etc and all words shorter than three characters were removed.
- The tweets were then tokenized and associated with a POS tag.
- These tweets are then wordnet tagged and lemmatized.
- The cleaned and lemmatized tweet data is then added to a separate column in the dataframe.

E. Descriptive Analysis of Pre-processed Data

It might be useful to now check the mean, median, maximum and minimum word count and character count of tweets (Fig. 3) and the hashtags (Fig. 4) included in these tweets and the analyzing the frequency of words in these tweets (Fig. 5) after the pre-processing steps are completed. So we performed another descriptive analysis of the clean data. The results were as follows:

Attribute	Aggregation Type	Value
Character Count	mean	112.13
	median	116
	max	316
	min	2
Word Count	mean	14.86
	median	15
	max	44
	min	1

Fig. 3. Descriptive analysis of the tweets after pre-processing

Attribute	Aggregation Type	Value
Character Count	mean	16.5
	median	11
	max	124
	min	1
Word Count	mean	1.81
	median	1
	max	20
	min	1

Fig. 4. Descriptive analysis of the hashtags

```
[('amp', 45165),
 ('case', 25477),
 ('new', 24486),
 ('people', 22965),
 ('test', 19252),
 ('death', 18444),
 ('day', 14883),
 ('health', 13505),
 ('trump', 13151),
 ('need', 12933)]
```

Fig. 5. Top 10 most frequently occurring words in the cleaned tweets

F. Exploratory Data Analysis

1) Most frequently occurring hashtags in tweets by country: It might be useful to know if there is a distinguishing factor in the hashtags used by users from different countries. So we created a stacked bar chart which which shows the distribution of these ten most commonly used hashtags for each country. As can be observed from the plot below that there are no major differences in the distribution of the popularity of the hashtags for different countries.

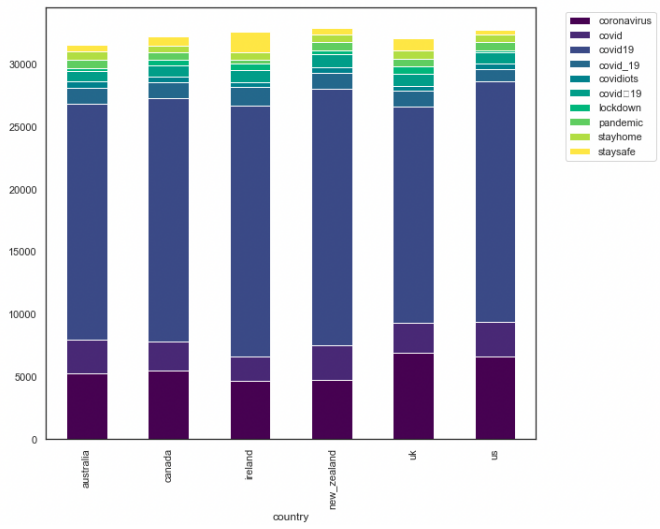

Fig. 6. Top 10 most frequently occurring hashtags

2) Most common words in tweets: A wordcloud helps visualize the most frequently occuring words in the dataset. So we created a wordcloud of the cleaned tweets. It was found that the most frequently occuring word was 'amp' which refers to AMP Rapid Test SARS-CoV-2 Ag which is a rapid immuno-chromatographic test for qualitative detection of SARS-CoV-2 nucleocapsid protein antigen in human nasal or throat swab samples as an aid in quick and efficient diagnosis of Corona Virus Disease 2019 (COVID-19). The other most frequently occurring significant words overall were found to be 'time', 'new case', 'work' and 'need'.

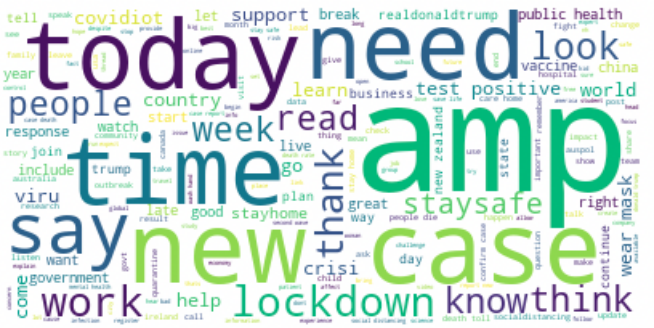

Fig. 7. Wordcloud - clean tweets

3) Most frequently occurring words in tweets by country: The following bar chart denotes the frequency of the most

popular words in the tweets pertaining to each country. Again, there are no observable difference in the distribution of the frequency of these words in tweets from each country.

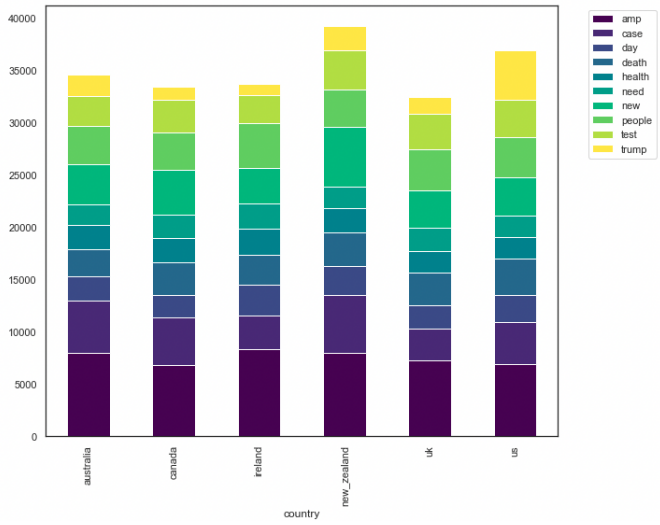

Fig. 8. Top 10 most frequently occurring words in tweets

4) Latent Dirichlet Allocation: With topic modeling, you can cluster words for a set of documents. This is unsupervised learning, because it automatically groups words without a predefined list of labels. If you feed the model data, it will give you different sets of words, and each set of words describes the topic.

Topic modeling is useful, but it's difficult to understand it just by looking at a combination of words and numbers. One of the most effective ways to understand data is through visualization. So we visualized these topics using an LDA model as shown below.

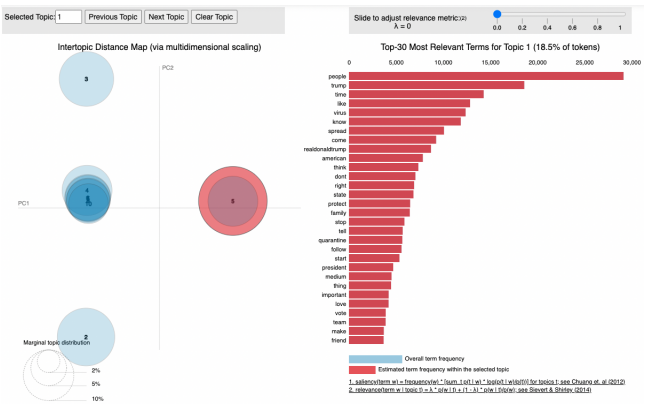

Fig. 9. LDA

PyLDAvis allows us to interpret the topics in a topic model. The left panel, labeled Intertopic Distance Map, circles represent different topics and the distance between them. Similar topics appear closer and the dissimilar topics farther. The relative size of a topic's circle in the plot corresponds to the relative frequency of the topic in the corpus. An individual topic may be selected for closer scrutiny by clicking on its circle, or entering its number in the "selected topic" box in the upper-left. The right panel, include the bar chart of the top 30 terms. When no topic is selected in the plot on the left, the bar chart shows the top-30 most "salient" terms in the corpus. A term's saliency is a measure of both how frequent the term is in the corpus and how "distinctive" it is in distinguishing between different topics. Selecting each topic on the right, modifies the bar chart to show the "relevant" terms for the selected topic. Relevence is defined as in footer 2 and can be tuned by parameter λ , smaller λ gives higher weight to the term's distinctiveness while larger λ s corresponds to probablity of the term occurance per topics.

Therefore, to get a better sense of terms per topic we'll use λ =0.

5) Non-Negative Matrix Factorization (NMF): Non-negative matrix factorization (NMF or NNMF), also non-negative matrix approximation is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into (usually) two matrices W and H, with the property that all three matrices have no negative elements.

Non-Negative Matrix Factorization is a statistical method that helps us to reduce the dimension of the input corpora or corpora. Internally, it uses the factor analysis method to give comparatively less weightage to the words that are having less coherence.

Generalized Kullback–Leibler Divergence: It is a statistical measure that is used to quantify how one distribution is different from another. As the value of the Kullback–Leibler divergence approaches zero, then the closeness of the corresponding words increases, or in other words, the value of divergence is less.

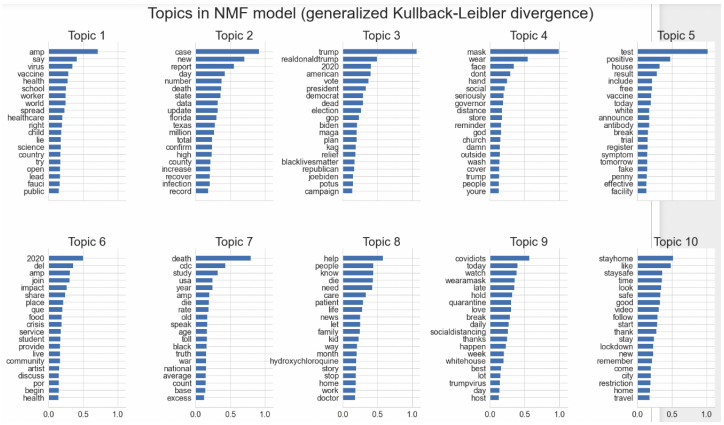

Fig. 10. Kullback–Leibler Divergence

Frobenius Norm: It is another method of performing NMF. It is defined by the square root of the sum of absolute squares of its elements. It is also known as the euclidean norm.

6) Text cleaning and Text Lemmatization: In order for machine learning algorithms to perform better, twitter data must be transformed into a format which can be used by the classification algorithms.

- Removing links and unusual characters to decrease

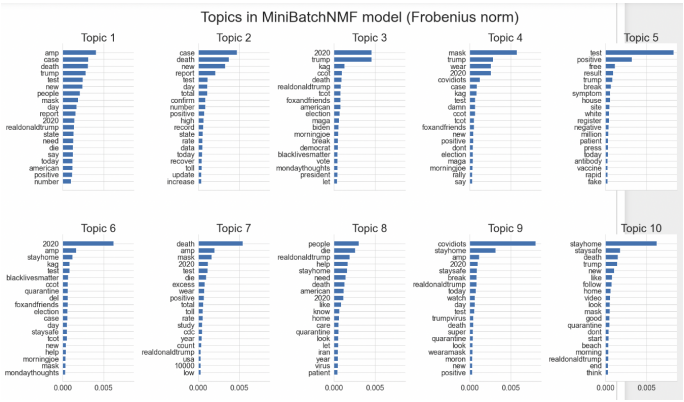

Fig. 11. Frobenius Norm

the noise for exploratory analysis and classification algortihms.

- Removing Stopwords like "we" "are" "the" etc. to reduce computing time since they donot contribute to classification.
- Lemmatization of Text to reduce a word into its root form based on the lexicon of the language with the gola of removing only the inflectional endings. WordNetLemmatizer was used from the NLTK package for the process. This contains a lexical database of over 200 languages.
- Parts of Speech Tagging was done using the NLTK library and used to tag each word and assign grammatical information to each word like Noun, Verb, Adverb etc.

7) Cosine Similarities: Cosine Similarities is a metric used to measure how similar the documents are. It measures the cosine of the angle projected by two vectors in multidimensional space. For this paper we created a small data set of hash-tags captured from the tweet text and categorised on the basis of countries. Count vectorization is used to convert the data into vectors and then into a document term matrix. The Matrices are then processed to find the similarities between the hash-tags of each country.

	US	UK	Australia	Ireland	New Zealand	Canada
US	1.000000	0.396048	0.391410	0.330443	0.395557	0.370646
UK	0.396048	1.000000	0.397894	0.389263	0.406406	0.438726
Australia	0.391410	0.397894	1.000000	0.346508	0.406839	0.406340
Ireland	0.330443	0.389263	0.346508	1.000000	0.362701	0.366429
New Zealand	0.395557	0.406406	0.406839	0.362701	1.000000	0.415081
Canada	0.370646	0.438726	0.406340	0.366429	0.415081	1.000000

Fig. 12. Cosine similarity

III. Methods

1) Logistic Regression with TFIDF: The text vectorizer Term frequency-inverse document frequency converts the text into a useable vector. Term Frequency (TF) and Document Frequency (DF) are combined in this idea (DF). The term frequency refers to the number of times a term appears in a document. It shows how widely used the word is. The weight of a term is determined by the inverse document frequency (IDF), which is used to minimize the weight of a term whose occurrences are dispersed throughout all documents. The higher the TF-IDF score, the more relevant the term is; the lower the TF-IDF value, the less relevant the term is. Logistic Regression is used along with TFID which gives model gives an accuracy of 44 percent. This was used as a Base Model.

```
              precision    recall  f1-score   support

   australia       0.48      0.44      0.46      7956
      canada       0.36      0.34      0.35      7955
     ireland       0.62      0.62      0.62      7999
 new_zealand       0.45      0.40      0.42      8023
          uk       0.33      0.34      0.33      8102
          us       0.41      0.52      0.46      7965

    accuracy                           0.44     48000
   macro avg       0.44      0.44      0.44     48000
weighted avg       0.44      0.44      0.44     48000

Confusion Matrix: [[3465  909  440  792  873 1477]
 [ 940 2696  689  989 1674  967]
 [ 371  617 4987  459  996  569]
 [ 729 1065  518 3185  962 1564]
 [ 813 1417  988  777 2723 1384]
 [ 861  690  407  899  959 4149]]
```

Fig. 13. Logistic Regression with TFIDF

2) Multinomial Naive Bayes: It is a classification strategy based on Bayes' Theorem and the assumption of predictor independence. A Naive Bayes classifier, in simple terms, posits that the existence of one feature in a class is unrelated to the presence of any other feature. Naive Bayes is broadly used in email spam detection, we built a pipeline to first vectorize the word and then Naive Bayes is used to classify. The accuracy of the model is 45 percent and shows a slight improvement on the logistic regression model. We some improvement in the individual subcategory as well.

```
              precision    recall  f1-score   support

   australia       0.54      0.41      0.47      7956
      canada       0.41      0.29      0.34      7955
     ireland       0.57      0.65      0.60      7999
 new_zealand       0.43      0.40      0.41      8023
          uk       0.35      0.34      0.35      8102
          us       0.41      0.62      0.49      7965

    accuracy                           0.45     48000
   macro avg       0.45      0.45      0.44     48000
weighted avg       0.45      0.45      0.44     48000

Confusion Matrix: [[3262  645  621  904  743 1781]
 [ 744 2304  942 1141 1717 1107]
 [ 286  390 5175  571  883  694]
 [ 539  781  672 3223  914 1894]
 [ 633 1015 1268  865 2728 1593]
 [ 571  461  454  854  708 4917]]
```

Fig. 14. Naive Bayes

3) Long Short-Term Memory networks (Best performing): LSTM network is a RNN that is used for sequential data. The basic operation of LSTM is to hold the required information and discard the the information that is not required for further prediction. For our model we used Bidirectional-LSTM, in bi-directional we can make the input flow in both directions to preserve the future and the past information. While performing text modelling and preprocessing task and modelling task focuses on creating data sequentially for example POS tagging, stopwords elimination and sequencing of the text. The feature of elimination of unused information and memorizing the sequence of the information makes the LSTM a powerful tool for performing text classification or other text-based tasks.

```
Epoch 1/2
7500/7500 [==============================] - 168s 22ms/step - loss: 1.4600 - accuracy: 0.4126
Epoch 2/2
7500/7500 [==============================] - 162s 22ms/step - loss: 0.9042 - accuracy: 0.6753
```

Fig. 15. LSTM

IV. Results

From Figures we can see that LSTM has the highest accuracy for the overall prediction. Although from the test set we can see that there is an issue of over-training where the accuracy drops to 49.8%

We believe the problem here is that there is a duplication of data within the cleaned tweets which is the cause of the over-training. The next step would be to calculate a similarity matrix of the tweets and drop out similar tweets and use stratified sampling to reduce the data size and run LSTM. We require more computing power for this operation as the estimate on our machine was approximately 3 days.

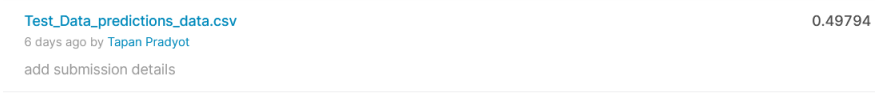

Fig. 16. LSTM

V. Appendix: Tables, Visualizations, etc

References

[1] Determine the User Country of a Tweet Han van der Veen, Djoerd Hiemstra, Tijs van den Broek, Michel Ehrenhard, and Ariana Need University of Twente

[2] Towards Real-Time, Country-Level Location Classification of World-wide Tweets Arkaitz Zubiaga1, Alex Voss2, Rob Procter1, Maria Liakata1, Bo Wang1, Adam Tsakalidis1

[3] https://www.nltk.org/api/nltk.tag.html

[4] https://www.tensorflow.org/guide/gpu

[5] https://stackoverflow.com/questions/41768196/python-convert-dataframe-into-a-list-with-string-items-inside-list

[6] https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

[7] https://stackoverflow.com/questions/15586721/wordnet-lemmatization-and-pos-tagging-in-python

[8] https://towardsdatascience.com/building-a-text-normalizer-using-nltk-ft-pos-tagger-e713e611db8

[9] https://www.machinelearningplus.com/nlp/lemmatization-examples-python/

[10] https://docs.python.org/3/howto/regex.html

[11] https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/

[12] https://stackoverflow.com/questions/34293875/how-to-remove-punctuation-marks-from-a-string-in-python-3-x-using-translate

[13] https://www.tensorflow.org/text/tutorials/text_classification_rnn