DATA MINING FROM TWITTER AND NATURAL LANGUAGE PROCESSING

CIND820 Big Data Analytics Project (Winter 2021)

By: Alireza Rahimnejad Yazdi

1- INTRODUCTION

Social medias are a great source of information because a significant portion of city dwellers share their opinions about various topics on social medias. Before analyzing such data, it needs to be gathered in a process which is known as "data mining." Twitter is a very good choice as a mine of data. Unlike other social platforms, almost every user's tweets are completely public and pullable. This is a huge plus when trying to get a large amount of data to run analytics on. Twitter data is also pretty specific. Twitter's API allows for complex queries like pulling every tweet about a certain topic within the last twenty minutes, or pull a certain user's non-retweeted tweets. Text classification and sentiment analysis can be applied to collected tweets to extract knowledge for various purposes such as product predictions, movie recommendations, etc. In this project, I would like to focus on the problem of public approval of corona virus vaccine. I will be pulling data from twitter about corona virus vaccine and perform sentiment analysis in order to gain insight on the public approval of it in different regions. Tokenization of the tweets will also be performed, and the results will be visualized to gain further insight about the tweets coming from counties with most tweets on this subject.

2- LITERTURE REVIEW

Social medias popularity is high because of their simplicity, speed, and effectiveness in bypassing community rules. Pressures of work environments make it difficult for people to visit their friends in person or call them on the phone. Therefore, people widely turn to social interactions on social networking platforms. As a result, Social medias such as Twitter have become a significant part of people's daily routines in modern societies; People share their opinions and emotions towards various topics on social media. Businesses, researchers, and governments extract data from social media for various financial, scientific, and political purposes. In this literature review, summaries of eight studies that have used tweets as a source of data are provided.

In one study, Twitter was used to analyze the nonmedical use and side effects of methylphenidate. Methylphenidate is a drug that is prescribed for the treatment of attention deficit hyperactivity disorder. This drug is sometimes used for non-medical reasons such as for studying and recreation. Due to active use of social networking services, such non-medical uses of this drug and/or its side effects are likely to be shared on Twitter. The objective of this research was to investigate tweets about the nonmedical use and side effects of methylphenidate via machine learning algorithms. The investigators gathered 34,293 tweets containing "methylphenidate" drug from August 2018 to July 2019 by searching for its name and brand names. The authors randomly selected 20% of the dataset as training dataset and annotated it as positive or negative for two dependent variables:

non-medical use and side effects. Support Vector Machine (SVM) was used as the machine learning algorithm to classify the data. The performance of the model was measured using F scores. F scores of the model for nonmedical use and side effects were 0.547 (precision: 0.926, recall: 0.388, and accuracy: 0.967) and 0.733 (precision: 0.920, recall: 0.609, and accuracy: 0.976), respectively. The authors concluded that the SVM classifiers that were built in their study were precise and accurate and could automatically identify the nonmedical use and side effects of methylphenidate using Twitter [1].

In another study, Lo et al. utilized unsupervised and supervised learning algorithms to classify target audience on Twitter with small annotation efforts. They used Twitter Latent Dirichlet Allocation (LDA) in order to automatically extract topic domains from contents shared by followers of an account owner. The authors trained a Support Vector Machine (SVM) ensemble with contents from different account owners of the various topic domains identified by Twitter LDA. Their results revealed their models were capable of classifying a target audience with high accuracy. Moreover, the result of this study revealed that better SVM ensemble can be constructed via a statistical inference approach such as bootstrapping in over-sampling, compared to random sampling. It was concluded that their model could take advantage of data diversity, which enables real-world applications for differentiating prospective customers from the general audience, translating to business benefits [2].

Alabdullatif et all, attempted to detect the trends in social media that are dependent on identifying relationships between members of a community. Social members share information that does not have a formal data structure and is transmitted in the form of texts, emoticons, and multimedia. The authors got interested in this research because a company wanted to advertise a sport product and had difficulty classifying Arab people on social media who are interested in sports. The company wanted to identify interacting users who share similar interests so that it can send them same advertisements. The company was also interested in finding the best times for sending such messages to potential customers. The authors used Naïve Bayes algorithm for classifications and reached accuracies as high as 90% [3].

Alec et al. from Stanford University, investigated automatic sentiment classification of Twitter messages. They classified the messages as positive or negative based on given query terms. The result of such classification can be of interest for consumers who want to research the sentiment of products before purchasing them, or companies that are interested in monitoring the public sentiment of their products. The authors provide the results of machine learning algorithms for classifying the sentiment of Twitter messages using distant supervision. The training data consists of Twitter messages with emoticons, which are used as noisy labels. Such training data is easily available and can be obtained through automated means. The authors used three machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) for their classifications in this study. They could reach accuracies above 80% when trained with emoticon data. Their paper also describes the preprocessing steps needed in order to achieve high accuracy [4].

Mustaqim et al. conducted a Twitter sentiment analysis on the Indonesian government's response to forest fires in 2019. Their analysis was performed on 6325 Twitter datasets on September 20, 2019 and preprocessing, and automated labeling and classification were performed on the data. A

Valence Aware Dictionary for Sentiment Reasoning (Vader) was used for automatically detecting the negative and positive polarity of each data followed by classification via K-Nearest Neighnors (KNN) algorithm. The result from rapid miner tools showed an accuracy of 79.45% for KNN which is higher than the accuracies that were achieved from decision trees, naïve Bayes and random forests algorithms. The process of sentiment analysis developed in this study can run almost automatically without human touch thanks to automated labeling via Vader [5].

Häberle et al. performed a feature space analysis of of geo-tagged Twitter text messages from the Los Angeles area. They also classified buildings into commercial and residential using a geo-spatial text mining approach. In order to make the feature space, they used broadly accepted word embedding models such as word2vec, fastText and GloVe. They also considered more traditional models based on TF-IDF. The result from the visual analysis of the word embeddings revealed that the two examined classes yielded several word clusters. However, the classification was performed using Naïve Bayes support vector machines, and a convolutional neural network. It was concluded that some tweets offer significant information about the two classes depending on their actual location in the feature space; if they are part of a cluster, they expected to be directly classified but same cannot be expected if they are not part of any cluster [6].

Öztürk et al. looked into the public opinions about the Syrian refugee crisis. They gathered 2381,297 relevant tweets in two languages: Turkish and English. Turkish Tweets were chosen because Turkey hosts the largest number of Syrian refugees and therefore they reflect public perception of a refugee hosting country. Authors conducted a comparative sentiment analysis of gathered Tweets. The results showed a significant difference between the sentiments in Turkish and English tweets. It was revealed that Turkish tweets had slightly more positive sentiments on Syrians and refugees than neutral and negative sentiments. However, the distribution of sentiments was very close among the three major categories. The English tweets on the other hand, had mostly neutral sentiments, followed by the negative sentiments. The percentage of positive sentiments is significantly lower among English tweets with only 12% versus 35% in Turkish tweets [7].

In the last study summarized here, Alkhushayni et al. did an emotion mining project using Twitter data. Emotion mining from text refers to extraction of these emotions via algorithms so that the contents of each tweet can be evaluated. In this research, tweets that contained at least one of the seven basic emotions were pulled. The resulting dataset was a collection of 42,000 tweets with an even distribution of each emotion. A lexicon of roughly 40,000 words, each associated with a weighted vector corresponding to one of the emotions, was made using this dataset. Various algorithms (lexically-based classification, supervised machine learning-based classification and ensemble method involving several multi-class classifiers trained on unigram features of the lexicon) were tested for identifying emotion in these tweets. It was shown that the ensemble method outperformed all other tested classifiers [8].

3- DATASET

In this study, I will not be using an already created dataset. Instead, I will be creating my own dataset by pulling information from Twitter.

First, I had to create a developer Twitter account. After approval of the account, I created an App and stored its credentials in a separate file. For pulling the tweets, I used the tweepy API in this project. Tweepy is an API that can be accessed via Python which can be used to collect tweets. After installing tweepy and some of its libraries, I started collecting data. Tweets about certain subjects can be filtered. Here I used the following search words:

```
search_words = ["coronavirus vaccine", "covid 19 vaccine", "covid vaccine"]
```

In my latest and final attempt, I pulled 1400 tweets on corona virus vaccine from tweeter. The collected tweets initially had the following columns: 'geo', 'text', 'user', 'location' and 'created_at'.

Table 1: Head of the raw data

	geo	text	user	location	created_at
0	None	RT @SandraWeeden: Update to Pfizer and Oxford side effects, Reports: 22 Feb \n\nPfizer: Reactions: 77,207 Blindness: 9 Deaths: 197\nO…	Bigmanalberta	Alberta, Canada	2021-02-28 23:28:51
1	None	RT @ABCWorldNews: Coronavirus vaccines were allegedly stolen and two children were wrongly administered shots in Shelby County, Tennessee,…	BKells8	Chester SC	2021-02-28 23:28:42
2	None	RT @SandraWeeden: Update to Pfizer and Oxford side effects, Reports: 22 Feb \n\nPfizer: Reactions: 77,207 Blindness: 9 Deaths: 197\nO…	Linfield_Fan	Northern Ireland, United Kingd	2021-02-28 23:28:35
3	None	RT @PoliticsForAll: If you are over 60, extremely vulnerable, or are a health/care worker and haven't been contacted for a coronavirus vacc…	w3lshrugby	UK	2021-02-28 23:28:31
4	None	RT @northyorkscc: Even if you've had the #Covid19 vaccine, it's important to continue to follow social distancing guidance, wash your hands…	uk_james	Coalville, England	2021-02-28 23:28:28
5	None	RT @CBSNews: Fauci warns against complacency as COVID-19 cases begin to plateau despite vaccine https://t.co/4ZO0yiuAWb	DnbEndeavors	FL	2021-02-28 23:28:09
6	None	RT @THV11: In the Sunday COVID-19 report, 288 positive cases have been added since Saturday. Gov. Hutchinson said the total number of new c…	shaylateater	Little Rock, AR	2021-02-28 23:27:59
7	None	RT @SandraWeeden: Update to Pfizer and Oxford side effects, Reports: 22 Feb \n\nPfizer: Reactions: 77,207 Blindness: 9 Deaths: 197\nO…	euphrosene	West Sussex	2021-02-28 23:27:51
8	None	RT @CBSNews: Fauci warns against complacency as COVID-19 cases begin to plateau despite vaccine https://t.co/4ZO0yiuAWb	CherylT369	Ontario, Canada	2021-02-28 23:27:18
9	None	RT @NYTHealth: This is how the Johnson & Johnson coronavirus vaccine works https://t.co /4CWRkB5a6y	BKoltunovitch		2021-02-28 23:26:49

Table 2: Tail of the raw data

	geo	text	user	location	created_at
1390	None	RT @SkyNews: A new lockdown has been imposed across the West Bank as Palestinians face a fresh surge of coronavirus cases and a continued w…	jamesdale94	Penzance, England	2021-02-28 19:11:31
1391	None	RT @jongaunt: 20 MILLION Brits have had #vaccine and now we are told that just one jab gives you 90% protection. THIS IS A GOOD NEWS STORY!…	MarkJack67	Tampa, FL us	2021-02-28 19:11:29
1392	None	RT @SkyNewsPolitics: Former French ambassador to the UK Sylvie Bermann says the UK's #COVID19 vaccine rollout has been a "real success", ad…	jamesdale94	Penzance, England	2021-02-28 19:11:28
1393	None	RT @AP: @AP The FDA said J&J's vaccine offers strong protection against what matters most: serious illness, hospitalizations and death. One…	StepinStylePete	Online	2021-02-28 19:11:17
1394	None	RT @Bos_CHIP: .@Bos_CHIP faculty Ben Reis, PhD, @johnbrownstein and project @VaccineFinder in the news.\n\nhttps://t.co/oMS4C7XFpp\n\nhttps://t…	HeartVessTransp	Gulmira Kudaiberdieva EiC	2021-02-28 19:11:02
1395	None	RT @CNN: A third vaccine is poised to join the fight against Covid-19, and its rollout will be quick, officials say https://t.co/T7M4GT2P2F	The_Turnabout		2021-02-28 19:10:56
1396	None	RT @NPR: #Breaking: The FDA has authorized Johnson & Johnson's COVID-19 vaccine -- making it the third vaccine available for emergency use.…	callmemamii	Hades	2021-02-28 19:10:47
1397	None	RT @AP: @AP The FDA said J&J's vaccine offers strong protection against what matters most: serious illness, hospitalizations and death. One…	Angie_RejoinEU	London	2021-02-28 19:10:42
1398	None	FIGURES OF HOPE (28/02/2021)\n\nGB It was confirmed by the @GOVUK today that more than 20 million first doses of a… https://t.co/y4DgeGgsZ8	anasontheair	Scotland	2021-02-28 19:10:38
1399	None	RT @carolina_moon1: There has been a total of 406 deaths reported in the UK shortly after receiving vaccination for Covid-19 up to and incl…	JoLeighHurst1	South Shields	2021-02-28 19:10:37

After installing Pandas package, the data was stored in a DataFrame. Same package was used in later stages for cleaning and processing the data. The head and tail of the raw data are provided in Table 1 and Table 2 respectively.

Before cleaning the data, I wanted to know if these 1400 tweets are relatively evenly distributed among the users or not. By running the code below, it was shown that the 1400 collected tweets are from 1267 different users and therefore the data set is not representing a small but very active group of twitter users.

```
tweet_df.user.value_counts()
```

```
TheVill60772553    13
CoronaUpdateBot    12
Reuters_Health      9
AmandaEverall2      7
jamesdale94         6
                   ..
AlFarr12            1
ONEHEALTHINITI1     1
Cloud_Country       1
olliecorbettt       1
LouieHaro2035       1

Name: user, Length: 1267, dtype: int64
```

I was also interested in the distribution of the locations of the tweets. The results, as shown below, revealed that tweets are from 689 different locations. However, it should be noted that there is inconsistency in how users have defined their locations. As in can be seen in the data head and tail and the following, some users put their country as locations, while others put down their city and province or use just put down the name of their cities. As a result, the location data at its raw form is not suitable for aggregation and needs to be standardized. In later parts of the report, I will demonstrate how I have achieved this goal using Google Maps API.

```
tweet_df.location.value_counts()
```

```
                     370
United States         20
Washington, DC        13
@TheVillage           13
USA                   12
                    ...
Splatsville            1
Berkeley, CA           1
West Hartford, CT      1
Selangor               1
Fitchburg, WI          1
Name: location, Length: 689, dtype: int64
```

4- APPROACH

After extracting tweets about the subject of interest, sentiment analysis will be conducted, and the results will be visualized based on variables such as location and time.

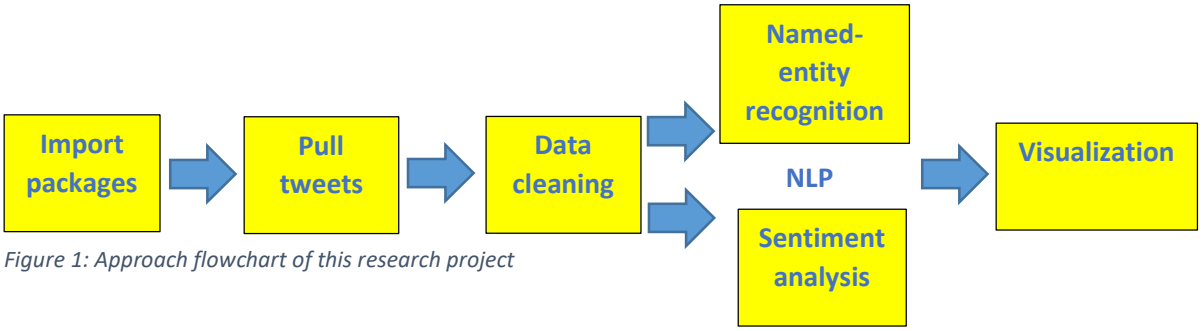

Figure 1: Approach flowchart of this research project

4-1- Import packages

As the first step, the following packages were imported:

drive:

To Connect Google Colab with Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

ConfigParser:

To implement a basic configuration language

```
!pip install ConfigParser
import configparser
config = configparser.RawConfigParser()
config.read('/content/drive/My Drive/Colab Notebooks/twitter.properties')
```

Pandas:

Data manipulation and analysis

```
import pandas as pd
```

tweepy:

To access the Twitter API

```
import tweepy as tw
```

OAuthHandler:

Authentication purposes

```
from tweepy import OAuthHandler
```

os:

Miscellaneous operating system interfaces

```
import os
```

spaCy:

Library for Natural Language Processing

```
import spacy
```

NLTK:

Natural Language Toolkit

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

re:

Regular expression operations

```
import re
```

googlemaps:

To standardize locations

```
!pip install -U googlemaps
import googlemaps
```

Skylearn & CountVectorizer:
To tokenize text

```
import sklearn
from sklearn.feature_extraction.text import CountVectorizer
```

4-2- Pull tweets

Part 1:

After authentication, on 28-Feb-2021, 1400 tweets on were pulled using tweepy package via the following codes:

```
search_words = ["coronavirus vaccine", "covid 19 vaccine", "covid vaccine"
]
tweets = tw.Cursor(api.search, q = search_words, lang = 'en', since = date
_since).items(1400)
```

The following details were gathered:

```
tweet_details = [[tweet.geo, tweet.text, tweet.user.screen_name, tweet.use
r.location, tweet.created_at] for tweet in tweets]
```

4-3- Data cleaning and manipulation

Part 1:

Pandas and re packages were used for data cleaning. First the data was transformed to a DataFrame in Pandas using the following code:

```
tweet_df = pd.DataFrame(data = tweet_details, columns= ('geo','text','user
','location','created_at'))
```

Then, the function below was made for cleaning the data using regular expression commands:

```
def clean_tweets(text):
  text = re.sub("RT @[\w]*:", " ", text)
  text = re.sub("@[\w]*","", text)
  text = re.sub("https?:/[A-Za-z0-9./]*","",text)
  text = re.sub("\n","",text)
  return text
```

The “clean_tweets” function was applied to the “text” attribute of the “tweet_df” DataFrame using the following code:

```
tweet_df["text"] = tweet_df["text"].apply(lambda x: clean_tweets(x))
```

In order to standardize the location and limit the location information to country, googlemaps package that was imported previously was utilized. First, authentication needed to be performed as shown below:

```
gmaps = googlemaps.Client(key = config.get('twitter','googleapikey'))
```

Then the function below was developed for standardizing locations.

```
def get_country (input):
  try:
    output = gmaps.geocode(input)[0]['formatted_address'].split(",")[-
1].strip()
  except:
    output = "Error"
  return output
```

The “get_country” function was applied with the code below in order to create a new column, country, with the standardized country information of each tweet.

```
tweet_df['country'] = tweet_df ['location'].apply (lambda x: "" if (not x.
strip()) else get_country(x))
```

Part 2:

First, the new country column was cleaned using following steps:

- Detection and Removal of Missing Values.

```
null_table_2 = df_tweets_2.isnull()
df_tweets_3 = df_tweets_2.dropna(how='any')
```

- The “Error” and “Europe” rows were filtered out.

```
df_tweets_4 = df_tweets_3[df_tweets_3.country != "Error"]
df_tweets_4 = df_tweets_4[df_tweets_4.country != "Europe"]
```

- Standardizing countries with multiple names.

```
def standardize_countries(country):
  country = country.replace("United Kingdom", "UK")
  country = country.replace("United States","USA")
  country = country.replace ("Dubai - United Arab Emirates","UAE")
  country = country.replace ("- مصفحM-37 - أبـو ظبـي - United Arab
Emirates","UAE")
  country = country.replace("Jeddah Saudi Arabia", "Saudi Arabia")
  return country
```

```
df_tweets_4["country"] = df_tweets_4["country"].apply(lambda x:
standardize_countries(x))
```

Then, the sentiment column was broken down and cleaned:

```
df_tweets_4[['neg','neu','pos','compound']] =
df_tweets_4.sentiment.str.split(",",expand=True,)


def clean_sentiment_scores(score):
    score = re.sub ("\{*['][\w]*['']:*[\s]*", "", score)
    return score


df_tweets_4["pos"] = df_tweets_4["pos"].apply(lambda x:
clean_sentiment_scores(x))
df_tweets_4["neu"] = df_tweets_4["neu"].apply(lambda x:
clean_sentiment_scores(x))
df_tweets_4["neg"] = df_tweets_4["neg"].apply(lambda x:
clean_sentiment_scores(x))
df_tweets_4["compound"] = df_tweets_4["compound"].apply(lambda x:
clean_sentiment_scores(x))


def clean_sentiment_scores_2(score):
    score = re.sub ("\}", "", score)
    return score


df_tweets_4["compound"] = df_tweets_4["compound"].apply(lambda x:
clean_sentiment_scores_2(x))
```

The data types of the scores were corrected via below codes:

```
df_tweets_4["neg"] = pd.to_numeric(df_tweets_4["neg"])
df_tweets_4["neu"] = pd.to_numeric(df_tweets_4["neu"])
df_tweets_4["pos"] = pd.to_numeric(df_tweets_4["pos"])
df_tweets_4["compound"] = pd.to_numeric(df_tweets_4["compound"])
```

Next, the database was grouped by country name and the mean of “pos”, “neg”, “neu” and “compound” scores were obtained and recorded for each country.

```
tweets_country_groups = df_tweets_4.groupby(df_tweets_4['country'])
tweets_country_groups.mean()
```

Part 3:

At this stage, I merged all the tweets for each country via following codes:

Columns “text” and “country” were selected;

```
df_tweets_5 = df_tweets_4[['text','country']]
```

A list of all unique countries was obtained;

```
country_list = list(df_tweets_4['country'].unique())
```

An emptry dictionany was created to be filled with each unique country as key and tweets from that country as value;

```
country_tweet_dic = {}
for i in country_list:
    country_tweet_dic[i] =[]


for i in list(df_tweets_5.index):
    for j in country_list:
        if df_tweets_5.loc[i,'country'] == j:
            country_tweet_dic[j].append(df_tweets_5.loc[i,'text'])


for country in country_list:
    str0 = ""
    for element in country_tweet_dic[country]:
        str0 += element
    country_tweet_dic2[country] = str0
```

The Country-tweet dictionary is turned into a Pandas DataFrame.

```
df_tweets_6 = pd.DataFrame (country_tweet_dic2, index = ["text"])
```

Next, the text of tweets will be cleaned before tokenization:

```
df_tweets_7 = df_tweets_6.transpose()

----------------------

def clean_text_round1(text):

    '''Make text lowercase, remove text in square brackets, remove
punctuation and remove words containing numbers.'''

    text = text.lower()

    text = re.sub('\[.*?\]', '', text)

    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)

    text = re.sub('\w*\d\w*', '', text)

    return text


df_tweets_7["text"] = df_tweets_7["text"].apply(lambda x:
clean_text_round1(x))

----------------------

def clean_text_round2(text):

    '''Get rid of some additional punctuation and non-sensical text that
was missed the first time around.'''

    text = re.sub('[‘’“”…]', '', text)

    text = re.sub('\n', '', text)

    return text


df_tweets_7['text'] = df_tweets_7['text'].apply(lambda x:
clean_text_round2(x))

-----------------------

def clean_text_round3(text):

    '''Get rid of emojis and leftover non-alphabetical characters'''

    text = re.sub('[^\w\s]', '', text)

    return text


df_tweets_7['text'] = df_tweets_7['text'].apply(lambda x:
clean_text_round3(x))
```

The data now is ready for tokenization; the tokenized data will be in a Document-Term Matrix which is word counts in matrix format. For many of the techniques we'll be using, the text must be tokenized, meaning broken down into smaller pieces. The most common tokenization technique is to break down text into words. We can do this using scikit-learn's CountVectorizer, where every row will represent a different document and every column will represent a different word. In addition, with CountVectorizer, we can remove stop words. Stop words are common words that add no additional meaning to text such as 'a', 'the', etc.

```
cv = CountVectorizer(stop_words='english')
data_cv = cv.fit_transform(df_tweets_7.text)
data_dtm = pd.DataFrame(data_cv.toarray(), columns=cv.get_feature_names())
data_dtm.index = df_tweets_7.index
data_dtm
```

Part 4:

Finally, exploratory data analysis (EDA) was performed on tokenized text. After importing the previously exported data into the new notebook, some manipulation was performed to make the data easier to work with.

```
data_dtm_2 = data_dtm.set_index('Index_column')
data_dtm_2_T = data_dtm_2.transpose()
```

The top 5 countries with most tweets from the Document_Term Matrix (dtm) were selected.

```
top5_list = ['USA','UK','Canada', 'Australia', 'New Zealand']
data_top5_dtm  = data_dtm_2[data_dtm_2.index.isin(top5_list)]
```

Now, we want to find the most frequently tweeted words in USA, UK, Canada, Australia and New Zealand:

```
data_top5_dtm_T = data_top5_dtm.transpose()

dict_five = {}
for c in data_top5_dtm_T.columns:
    top = data_top5_dtm_T[c].sort_values(ascending=False).head(20)
    dict_five[c]= list(zip(top.index, top.values))
```

```
dict_five_1 = {}
for country, top_words in dict_five.items():
    print(country)
    count_list = []
    for word,count in top_words:
        count_list.append(count)
    print(count_list)
    dict_five_1[country]= count_list
    print('---')


dict_five_2 = {}
for country, top_words in dict_five.items():
    print(country)
    word_list = []
    for word,count in top_words:
        word_list.append(word)
    print(word_list)
    dict_five_2 [country] = word_list
    print('---')
```

The “dict_five” dictionary containing the 20 most frequently tweeted words for each country was converted to a pandas DataFrame using the following code:

```
df_dict_five = pd.DataFrame(dict_five)
```

Finally, Prepare the DataFrame for visualization:

```
five_eyes = list(df_dict_five.columns)


dict_Canada = {"count" : dict_five_1['Canada'], "word" :
dict_five_2['Canada']}
df_Canada = pd.DataFrame(dict_Canada)
--------------------
```

```
dict_USA = {"count" : dict_five_1['USA'], "word" : dict_five_2['USA']}

df_USA = pd.DataFrame(dict_USA)

--------------------

dict_UK = {"count" : dict_five_1['UK'], "word" : dict_five_2['UK']}

df_UK = pd.DataFrame(dict_UK)

--------------------

dict_Australia = {"count" : dict_five_1['Australia'], "word" :
dict_five_2['Australia']}

df_Australia = pd.DataFrame(dict_Australia)

--------------------

dict_NewZealand = {"count" : dict_five_1['New Zealand'], "word" :
dict_five_2['New Zealand']}

df_NewZealand = pd.DataFrame(dict_NewZealand)
```

4-4- Natural Language Processing (NLP)

4-4-1- Named-entity recognition

Part 1:

In order to perform entity recognition, first I loaded “en_core_web_sm” which is an English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, vectors, syntax and entities.

```
nlp = spacy.load('en_core_web_sm')
```

Then I printed the detected entities using the following code:

```
tweet_df['text'].apply(lambda x: [print ("\tText : {}, Entity : {}".format
(ent.text, ent.label_)) if (not ent.text.startswith("#")) else "" for ent
in nlp(x).ents])
        Text : The Pan American Health Organization (PAHO, Entity : ORG
        Text : Connecticut, Entity : GPE
        Text : first, Entity : ORDINAL
        Text : Johnson &amp, Entity : ORG
        Text : Johnson, Entity : PERSON
        Text : Thousands, Entity : CARDINAL
        Text : Sunday, Entity : DATE
        Text : 288, Entity : CARDINAL
        Text : Saturday, Entity : DATE
        Text : Hutchinson, Entity : PERSON
        Text : Pfizer, Entity : ORG
        Text : Oxford, Entity : ORG
        Text : 22, Entity : CARDINAL
```

Using the code below, I added a new column named “entity”, containing detected entities for each tweet, to the DataFrame.

```
tweet_df["entity"] = tweet_df['text'].apply(lambda x: [(ent.text, ent.label_) if (not ent.text.startswith("#")) else "" for ent in nlp(x).ents])
```

4-4-2- Sentiment Analysis

Part 1:

Sentiment Analysis was perform using VADER from NLTK. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. Since VADER is pretrained, you can get results more quickly than with many other analyzers. However, VADER is best suited for language used in social media, like short sentences with some slang and abbreviations. The codes for applying this NLP tool and how to add them to the DataFrame are provided in the following:

```
sid = SentimentIntensityAnalyzer()
```

```
tweet_df['sentiment'] = tweet_df['text'].apply( lambda x: sid.polarity_scores(x))
```

The new column, sentiment, provides each tweet with the scores shown in the example below:

{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

4-5- Visualization

Three main groups of results are visualized using Tableau: number of tweets by country, sentiment scores of countries and most frequently tweeted words.

- Number of tweets by country (part 2):

The “df_tweets_4” DataFrame with its country column fully cleaned and standardized was created in previous steps. “value_counts ()” method was use to get the frequency of each country in the data base and the result was converted into the “tweets_per_country” DataFrame. This DataFrame was exported and used in Tableau to create a Symbol Map of tweet frequency per country.

```
tweets_per_country = df_tweets_4['country'].value_counts()
```

```
tweets_per_country = pd.DataFrame(tweets_per_country)
```

- Sentiment scores of countries (part 2):

The "df_tweets_4" DataFrame was grouped by 'country' and the mean of the sentiment scores were stored in the "tweets_country_groups" DataFrame. This DataFrame was exported and used in Tableau to create a Symbol Map of the mean of positive and negative sentiment scores in each country. Treemaps of positive and negative sentiments was also visualized in Tableau.

```
tweets_country_groups = df_tweets_4.groupby(df_tweets_4['country'])
tweets_country_means = tweets_country_groups.mean()
```

- Most frequently tweeted words (part 4):

I used tokenization and bag of words technique to discover the most frequent words used in countries with the greatest number of tweets on corona virus vaccine (US, UK, Canada, Australia and New Zealand). Separate DataFrames containing the topmost frequent words and the number of times they were tweeted were exported as csv files. These csv files were imported in Tableau and then used them to visualize the Word Cloud for each country.

5- RESULTS

A sample of the results from the NLP analysis: named-entity recognition and sentiment analysis are shown in the Table 3. After initial assessment of the results, it was decided to discard the results from named-entity recognition as it did not assist in answering the research question. The sentiment analysis results, on the other hand, were selected for further assessment.

Table 3:

```
tweet_df.head()
```

	geo	text	user	location	created_at	sentiment	entity
0	None	Update to Pfizer and Oxford side effects, Reports: 22 Feb Pfizer: Reactions: 77,207 Blindness: 9 Deaths: 197O…	Bigmanalberta	Alberta, Canada	2021-02-28 23:28:51	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	[(Pfizer, ORG), (Oxford, ORG), (22, CARDINAL), (Feb Pfizer, PERSON), (77,207, CARDINAL), (9, CARDINAL), (197O, CARDINAL)]
1	None	Coronavirus vaccines were allegedly stolen and two children were wrongly administered shots in Shelby County, Tennessee,…	BKells8	Chester SC	2021-02-28 23:28:42	{'neg': 0.176, 'neu': 0.824, 'pos': 0.0, 'compound': -0.4939}	[(Coronavirus, ORG), (two, CARDINAL), (Shelby County, GPE), (Tennessee, GPE)]
2	None	Update to Pfizer and Oxford side effects, Reports: 22 Feb Pfizer: Reactions: 77,207 Blindness: 9 Deaths: 197O…	Linfield_Fan	Northern Ireland, United Kingd	2021-02-28 23:28:35	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	[(Pfizer, ORG), (Oxford, ORG), (22, CARDINAL), (Feb Pfizer, PERSON), (77,207, CARDINAL), (9, CARDINAL), (197O, CARDINAL)]
3	None	If you are over 60, extremely vulnerable, or are a health/care worker and haven't been contacted for a coronavirus vacc…	w3lshrugby	UK	2021-02-28 23:28:31	{'neg': 0.114, 'neu': 0.886, 'pos': 0.0, 'compound': -0.2944}	[(60, DATE)]
4	None	Even if you've had the #Covid19 vaccine, it's important to continue to follow social distancing guidance, wash your hands…	uk_james	Coalville, England	2021-02-28 23:28:28	{'neg': 0.0, 'neu': 0.909, 'pos': 0.091, 'compound': 0.2023}	[]

The frequency of corona virus vaccine related tweets for each country is shown in Figure 3. Since the language selected for gathering the tweets was English, it is not unusual that most tweets are from most populated English-speaking countries: USA (619 tweets), UK (140 tweets), Canada (43 tweets), Australia (22 tweets) , and New Zealand (14 tweets). The population of these countries are as follows; USA: 328 million, UK: 66 million, Canada: 37 million, Australia: 25 million, and New Zealand: 5 million.

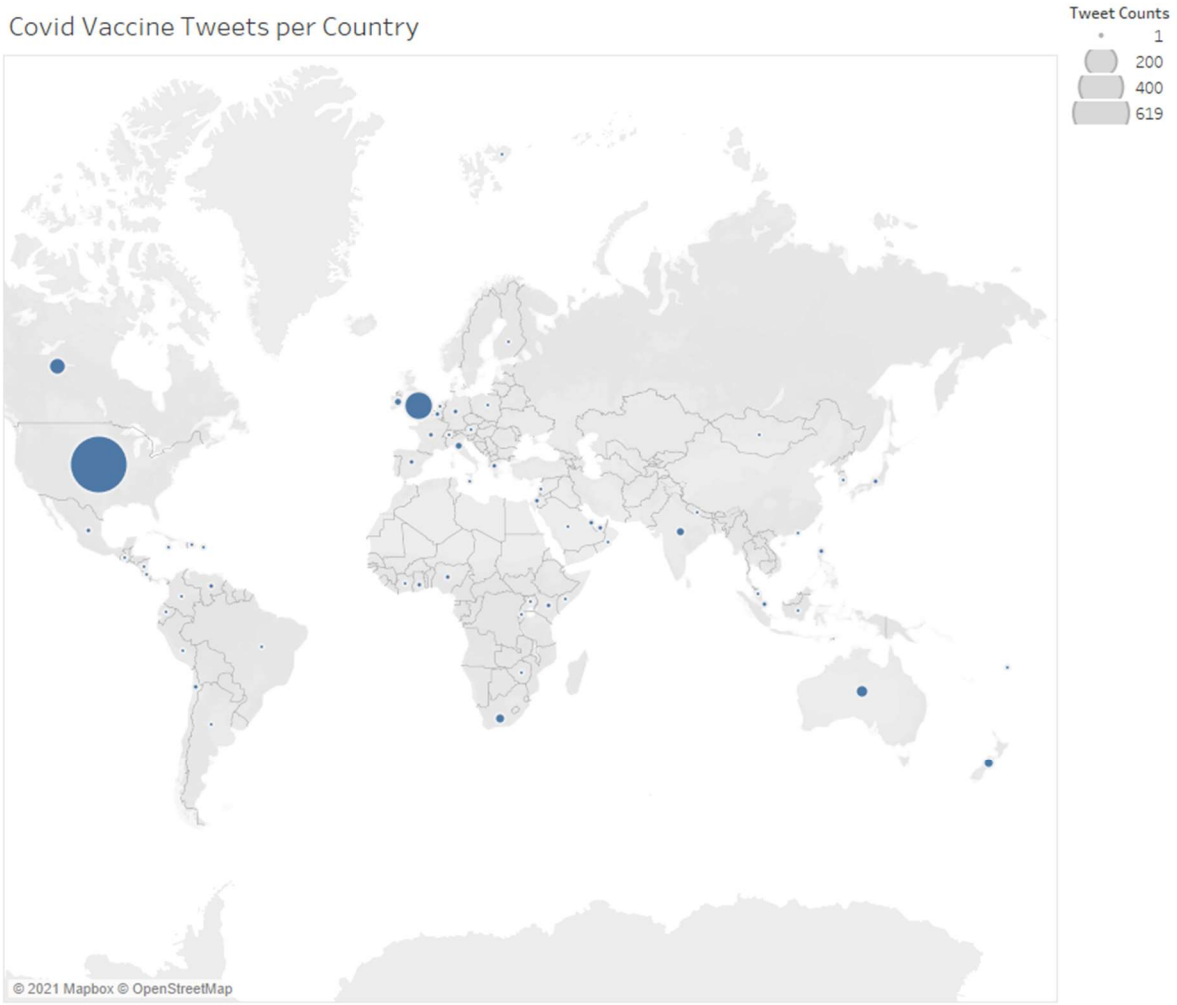

Figure 2: Symbol Map of corona vaccine tweet frequency per country on Feb 28, 2021

The number of tweets for countries and the tweet count per million population for USA, UK, Canada, Australia, and New Zealand are shown in bar charts in Figure 3. As can be seen in the charts, Although the number of the tweets increases with the number of population (at lease in these five English-speaking countries), the average of tweets sent per million people in these countries does not follow the same pattern. For example, New Zealand has the lowest total number of the tweets. However, when looking at the normalized data, it is shown that on average, New Zealanders on average send the most tweets; approximately 3 times more than their Australian neighbors. USA on the other hand, which has the most total number of tweets by far, ranks only 3rd in the normalized chart.

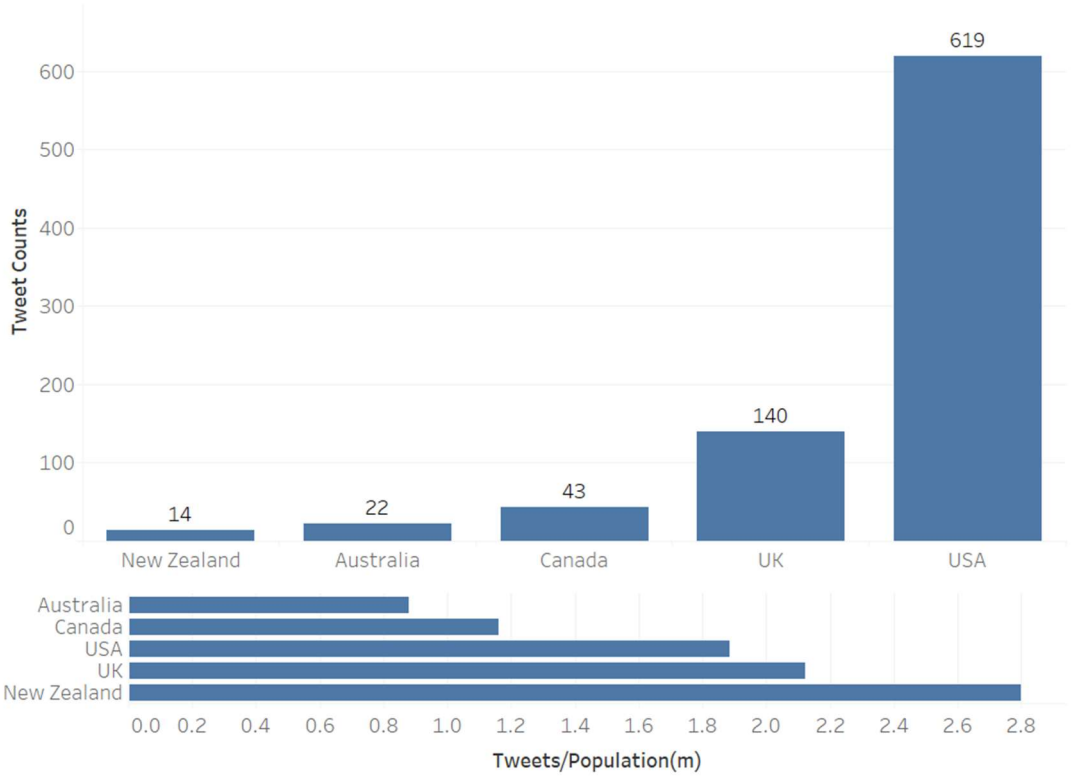

Figure 3: Bar charts of tweet frequency and tweets per million population for USA, UK, Canada, Australia and New Zealand

Figure 4 shows the Symbol Map of Sentiment scores of corona virus vaccine tweets on Feb28, 2021. In the positive sentiment section at the top of Figure 4, the bigger circles in the darker green represent countries with more positive sentiment on the corona virus vaccine while the smaller circles in lighter greens represent countries with less positive sentiment on the subject. In the negative sentiment section at the bottom of Figure 4, the bigger circles in the darker red represent countries with more negative sentiment on corona virus vaccine while the smaller circles in lighter red represent countries with less negative sentiments towards this vaccine.

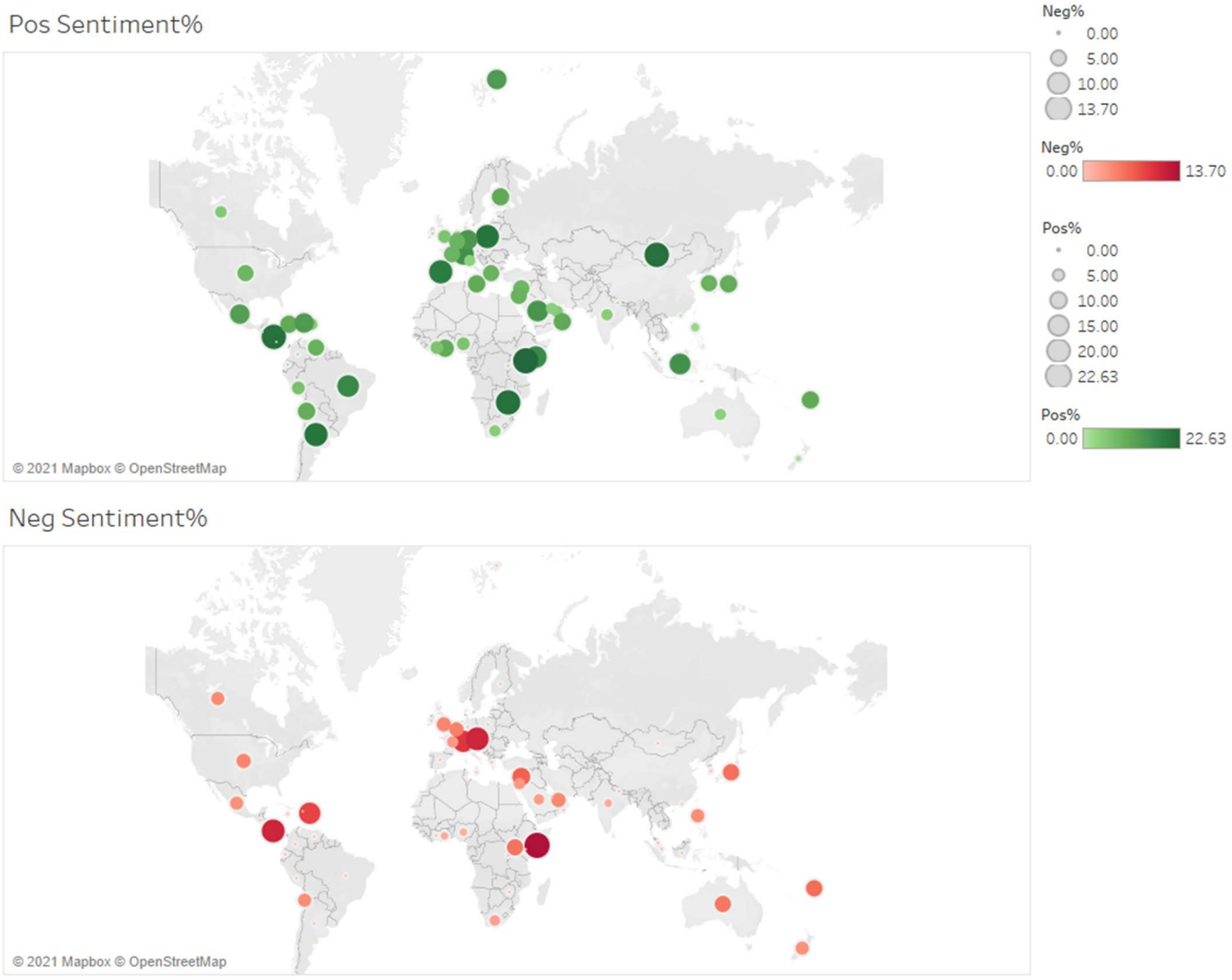

Figure 4: Symbol Map of Sentiment scores of corona virus vaccine tweets on Feb28, 2021

Figure 5 shows the Treemaps of Sentiment scores of corona virus vaccine tweets. In the positive sentiment section at the top of Figure 5, the bigger rectangular boxes in the darker green represent countries with more positive sentiment on the corona virus vaccine while the smaller rectangular boxes in lighter greens represent countries with less positive sentiment on the subject. In the negative sentiment section at the bottom of Figure 5, the bigger rectangular boxes in the darker red represent countries with more negative sentiment on corona virus vaccine while the smaller rectangular boxes in lighter red represent countries with less negative sentiments towards this vaccine. The Treemap charts are used to compliment the Symbol Maps in Figure 4 by providing a more organized visualization which also includes labels for countries with most negative and most positive sentiments.

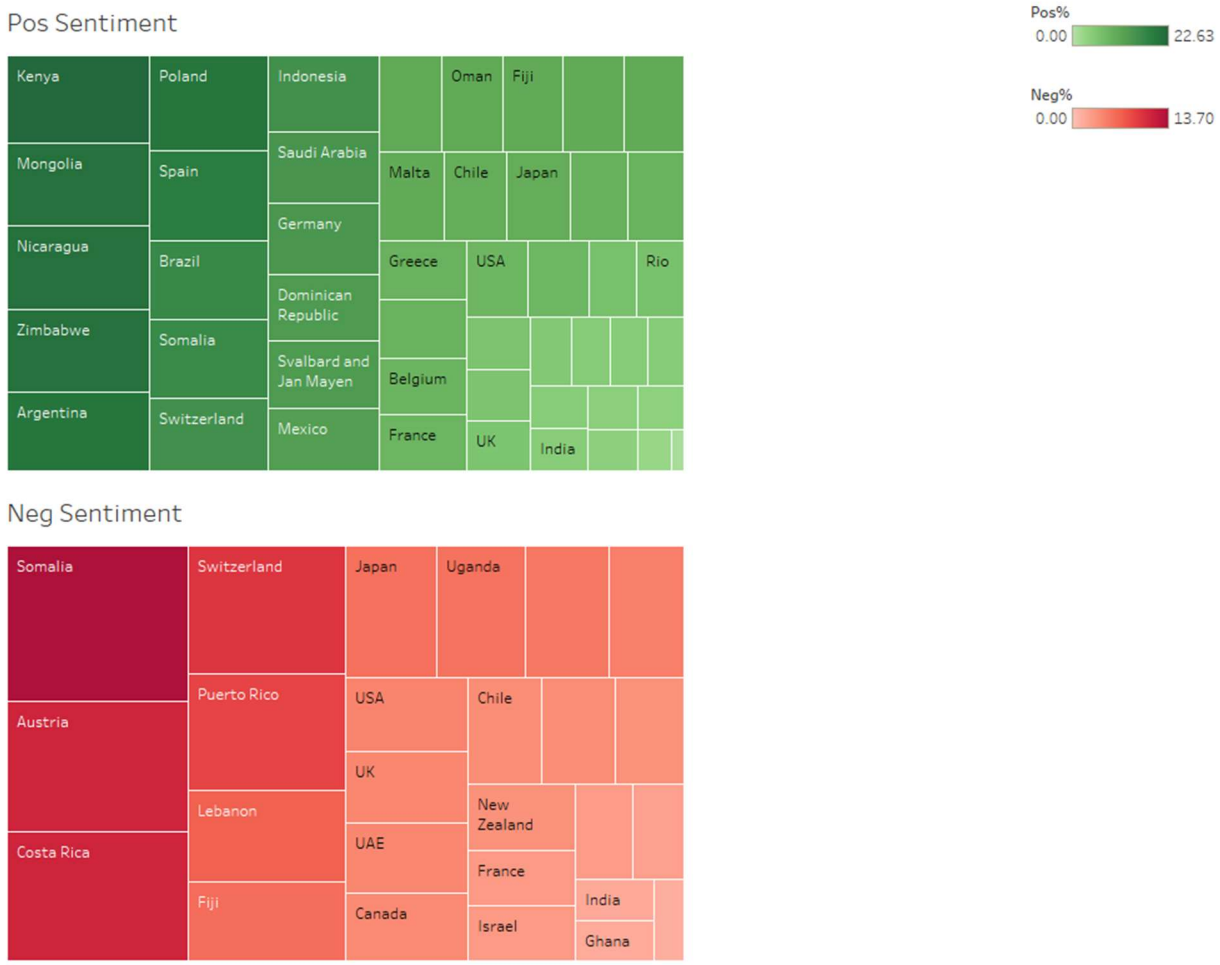

Figure 5: Treemaps of Sentiment scores of corona virus vaccine tweets on Feb28, 2021

Figure 6 shows Word Cloud visualizations of the most frequently words in US, UK, Canada, Australia, and New Zealand. Some insights can be extracted from these Word Clouds. For example, the top 20 words for both Canada and USA included "fauci" (the director of the U.S. National Institute of Allergy and Infectious Diseases and the chief medical advisor to the U.S

president), and "johnson" (an American pharmaceutical producing Corona virus vaccine). However, the top 20 tweeted words by Canadians did not include any Canadian health official, politician, or organization. In UK, "oxford" which refers to University of Oxford which has produced a corona vaccine is seen next to Pfizer, an American corona vaccine producer. The Brits also seem to be concerned with "blindness" when discussing corona virus vaccines. In Australia, the most tweeted vaccine manufacturer is "novavax". In fact, this is the only vaccine manufacturing brand included in the top 20 words tweeted by Australians. Another interesting and unique frequently tweeted word in Australia is "fear"! Finally, in New Zealand, the top tweeted words do included some unique references to New Zealand such as "newshub" (a news program in New Zealand), "auckland" (a major city in north of New Zealand), and "dhb" (which stands for district health board in New Zealand). People of New Zealand also seem concerned about "gps", "lockdown" and "priority" when discussing corona vaccine, and also mention "disgusted" often!

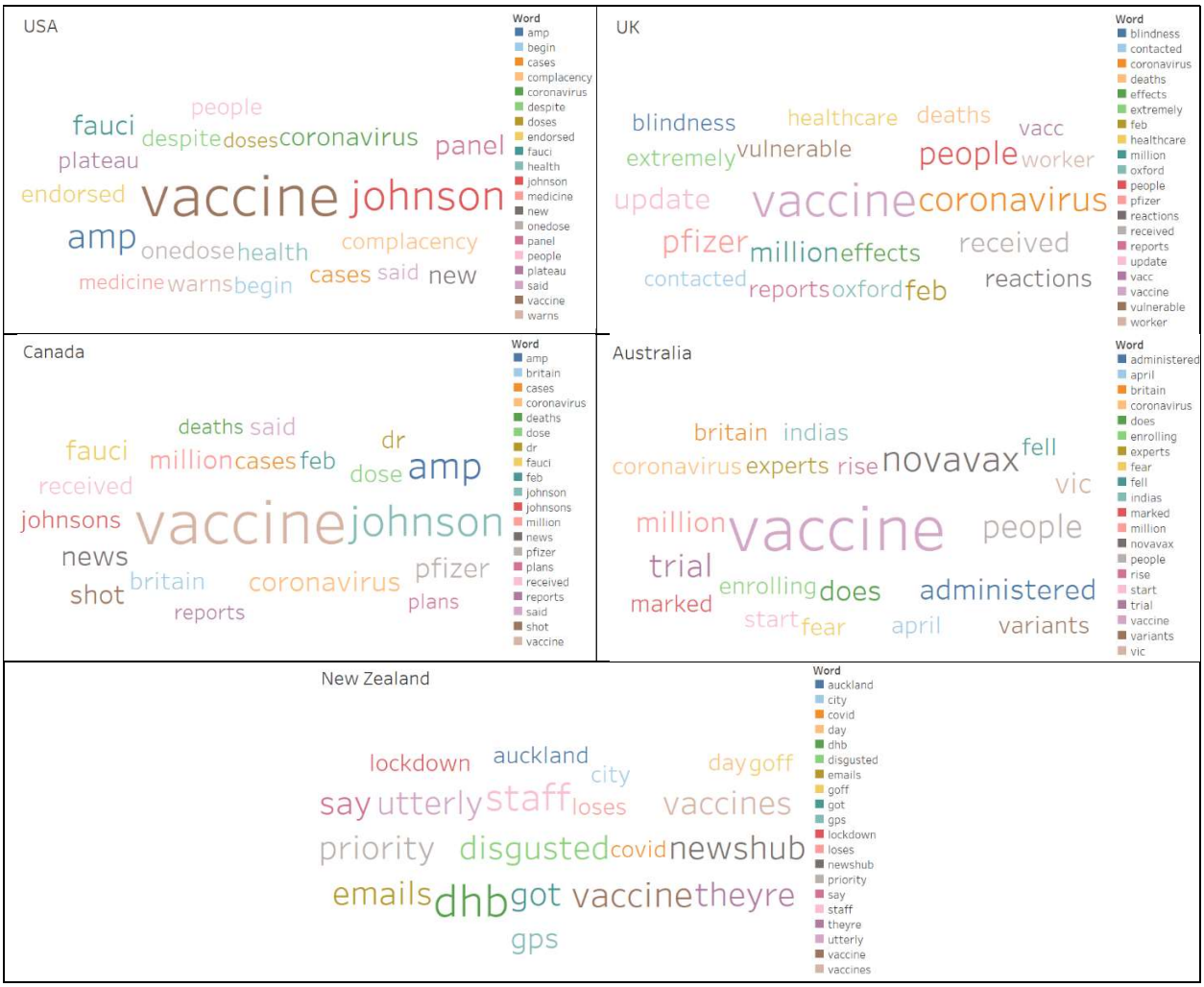

Figure 6: Word Cloud visualization of the most frequently words in US, UK, Canada, Australia and New Zealand

6- CONCLUSION

1400 tweets about corona virus vaccine were pulled from Twitter on February 28, 2021. After multiple stages of data cleaning and manipulation, the number of tweets in each country was determined and this data was normalized by population for USA, UK, Canada, Australia, and New Zealand which had the greatest number of the tweets. Among these five countries, USA had the greatest number of tweets and New Zealand had the least. However, when normalized by population, New Zealand had the highest average of tweets per million in population. Next, natural language processing (NLP) was performed on the data. Two techniques were used: named-entity recognition and sentiment analysis. After initial assessment of the results, the entity recognition results were discarded and those of sentiment analysis were further processed such that they can be visualized using Tableau software. Finally, through further processing, the tweets were merged for each country into a document-term matrix which was used to visualize the most frequently tweeted words in each country using Word Cloud visualization. Some interesting observations were made from the Word Clouds. For example, Canadians tweeting about American health officials instead of their own which might reflect the geographical proximity of the two country and USA influence in Canada. Another interesting observation was that in New Zealand, references unique to that country were observed in the most tweeted words but no mention of a foreign entity (unlike Canada, UK and Australia) which might reflect on their geographical isolation from the rest of the word. Americans also did not have foreign entities in their top tweeted words which in their case might be a result of their superpower status in the world.

References:

1- Myeong Gyu Kim, Jungu Kim, Su Cheol Kim, and Jaegwon Jeong. Twitter Analysis of the Nonmedical Use and Side Effects of Methylphenidate: Machine Learning Study. J Med Internet Res. 2020; 22(2): e16466.

2- Siaw Ling Lo, Raymond Chiong, David Cornforth. Using Support Vector Machine Ensembles for Target Audience Classification on Twitter. PLoS ONE. 2015; 10(4): e0122855. doi:10.1371/journal.pone.0122855

3- Abdullatif Alabdullatif, Basit Shahzad, and Esam Alwagait. Classification of Arabic Twitter Users: A Study Based on User Behaviour and Interests. Hindawi Publishing Corporation. 2016, Article ID 8315281.

4- Go Alec, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford 1.12. 2009.

5- T Mustaqim*, K Umam and M A Muslim. Twitter text mining for sentiment analysis on government's response to forest fires with vader lexicon polarity detection and k-nearest neighbor algorithm. Journal of Physics: Conference Series 1567. 2020; 032024.

6- Matthias Häberle, Martin Werner & Xiao Xiang Zhu. Geo-spatial text-mining from Twitter – a feature space analysis with a view toward building classification in urban regions. European Journal of Remote Sensing. 2019; 52(S2): 2–11.

7- Nazan Öztürk, Serkan Ayvaz. Sentiment analysis on Twitter A text mining approach to the syrian refugee crisis. Telematics and Informatics. 2018; 35: 136–147.

8- Suboh M Alkhushayni , Daniel C Zellmer, Ryan J DeBusk, Du'a Alzaleq. Text emotion mining on Twitter. IOP SciNotes. 2020; 1: 035001