CPTS – 575 Data Science Project Report

Twitter Tweet Analysis

Washington State University - Pullman

Team ASK:

Andrei Kondyrev - 011738571
Sahil Shrivastava - 0117717650
Kulpreet Singh - 011771817

1. Abstract:

Unlike any other social media platform, Twitter has impacted today's social realm. It is ubiquitous and mainly utilized for expressing logic-based, political, and scientific opinions and facts shared to all individuals, communities, and government bodies. The essential tool to express these various facts/opinions is via "tweets" / Twitter messages. Tweets are an essential entity used to portray facts. However, the mechanism and algorithmic design are also used to detect possible campaigns, industrial booms/crashes, threats, future predictions, etc.

For this project, we have three consolidated tasks. To understand the meaning and attitude of these tweets (or messages) across various timelines and compare them to recent world events. Furthermore, generate a timeline graph and use multiple Word Cloud analyses of these events. Finally, to thoroughly analyze communication threads and outline them.

In conclusion, we processed the data to obtain multiple Word Clouds and sentiments. The top 50 words were utilized and analyzed for two months. After Exploratory Analysis, we obtained specific patterns and observed these patterns with real-time news during that period; this was done to verify the patterns' accuracy.

2. Introduction:

The main problem faced by many researchers/scientists is handling large unlabeled amounts of data, with the utilization of word clouds as a tool for doing research on sentiments and with the utilization of top 50 words. Also generated additional stopword lists to ensure everything is more accurate and only necessary information is retrieved.

The importance of this type of research is applicable in many fields. It could be applied in information security in controlling the spread of information either political or advertisement campaigns, etc.; data science specifically to understand the evolution in communication patterns, the tonality of the communicating medium, which would aid in efficient assumption analysis; finally in machine learning to maintain an ML Mechanism for sensitive topics.

The approach taken was tweets analysis.

3. Problem Definition:

In general, humans tend to predict false outcomes. However, this changed when we were introduced to the concept of "information." We studied and understood it and developed algorithms and frameworks for handling and dealing with this information. We then understood the difference between wanted and unwanted information. We concluded that the wanted or "desired" information is crucial in prediction and analysis. With this hailed the development of social media, a framework/platform which handles people's information and connects this information to other people.

Retrospective inspection is an excellent analogy to understand and predict future outcomes. Moreover, an appropriate way to retrospect Is by analyzing past information available on social media platforms (in our case – Twitter) and trying to connect the dots to come to a near elegant solution for the future. This social realm will be analyzed via the inferred observations, and conclusions will be drawn based on these outcomes and whether they contribute to future analysis

4. Models:

WordClouds, Stopwords, sentiment analysis followed by extensive exploratory data analysis, regular expression utilized for data cleaning of tweets.

5. Implementation:

- **Data Set Analyzed**: Twitter data of tweets for two months, i.e., April 1, 2020 – May 31, 2020.
- **Data Used for Analysis**: Tweets -> WordClouds of Tweets
- **Experimental Setup**: Google Colab, Python v3.8, Regular Expressions, NLTK, Numpy, Pandas, Matplotlib, WordCloud, BeautifulSoup,
- **Exploratory Data Analysis**: WordCloud Generation and Sentiment Analysis for the tweets, whether it is positive or negative

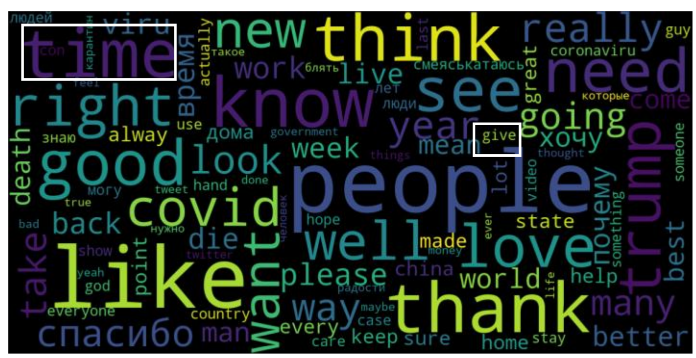
6. Results and Discussions:

From the analysis, three themes/outlines were observed.

6.1 Theme 1: Covid-19 Amplification

Duration: April 1,2020 – April 15,2020

- COVID-19 Pandemic Spreads across the globe – (*covid*)
- Russia helps US Govt by sending masks to US Hospitals, etc. (*help/give)*
- US Government invokes Defense Production Act -> Manufacture of ventilator machines (*trump)*
- Ford Motor Company suspends vehicle production due to Pandemic in Europe (time)



Corroborated Real-Time News After WordCloud Analysis

ABC7 New York –April 9, 2020

https://www.mytwintiers.com/health/coronavirus/watch-governor-cuomo-gives-an-update-on-covid-19-in-new-york-state/

https://abc7ny.com/coronavirus-new-york-ny-cases-in-news/6090148/

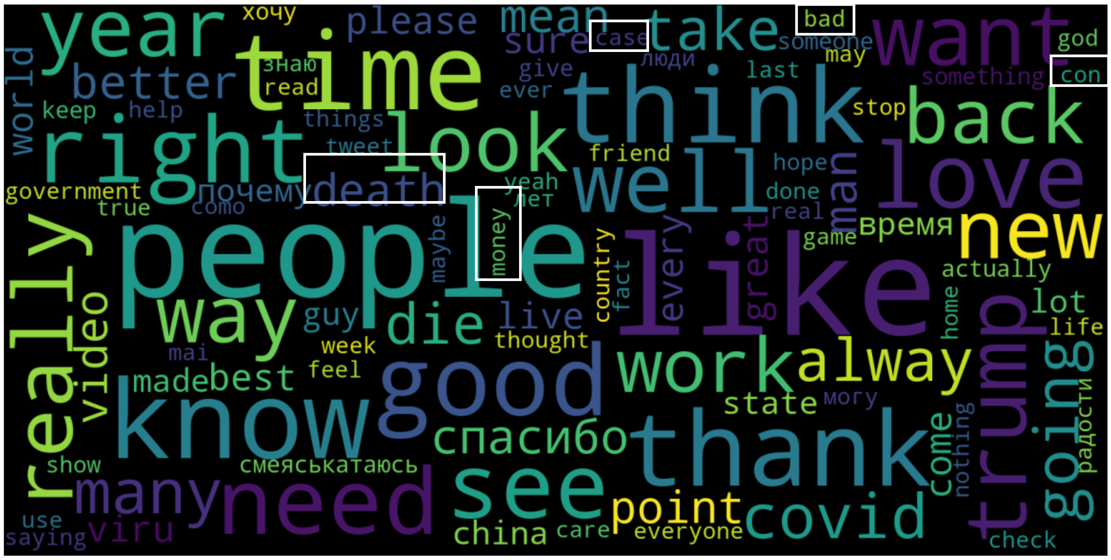
Reuters News –April 14, 2020

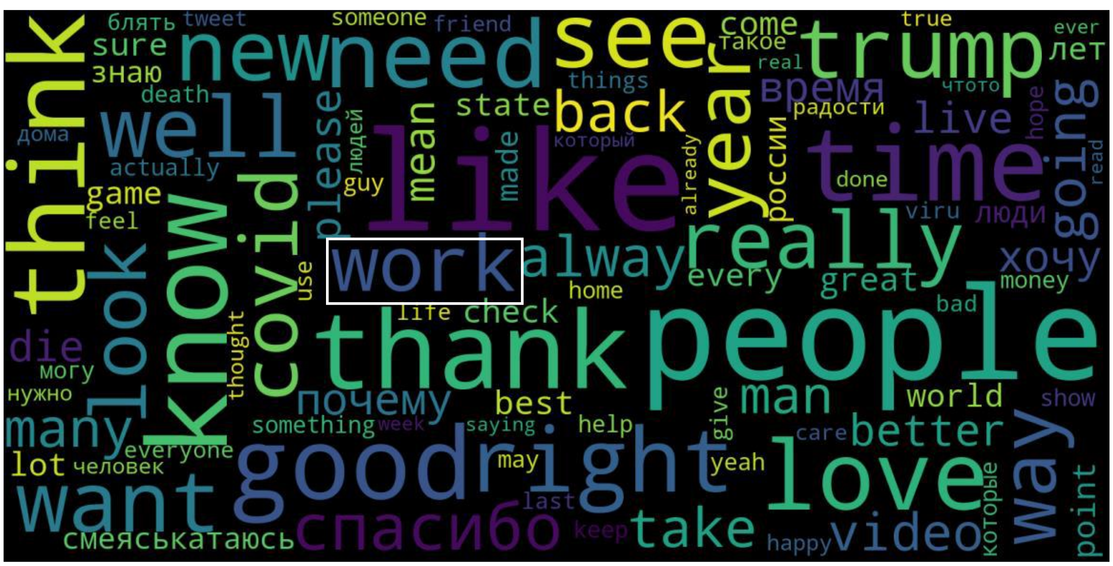
https://www.reuters.com/article/us-health-coronavirus-england-casualties/uk-coronavirus-death-toll-could-be-15-higher-than-previously-shown-new-data-idUSKCN21W0

6.2 Theme 2: Economic & Political Hardships

Duration: April 16 – May 3, 2020

- Unrest and riots in Germany due to opposition to lockdown. (*con/case/bad*)
- Oil Industries Businesses suffer losses of 300%(*money*)
- Work from Home Initiated by Major IT Firms, such as Microsoft, Amazon, etc. (*work*)
- Delta Airlines suspends international/domestic flights suffers a loss of $534 million. (*money)*
- Near-Earth Asteroid diameter of 10km makes a close approach. (*death & big*)





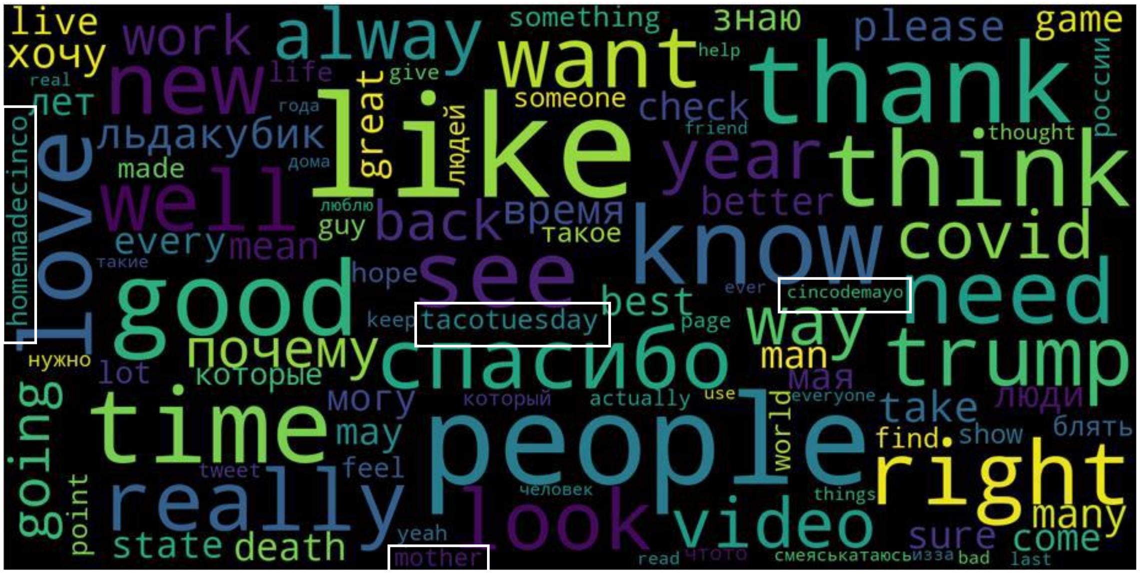
Real-Time news during April 16-May 3rd, 2020:
Reuters News: German police clash with anti-lockdown protesters–April 21, 2020
https://www.reuters.com/world/europe/protesters-gather-germany-debates-covid-19-lockdown-law-2021-04-21/

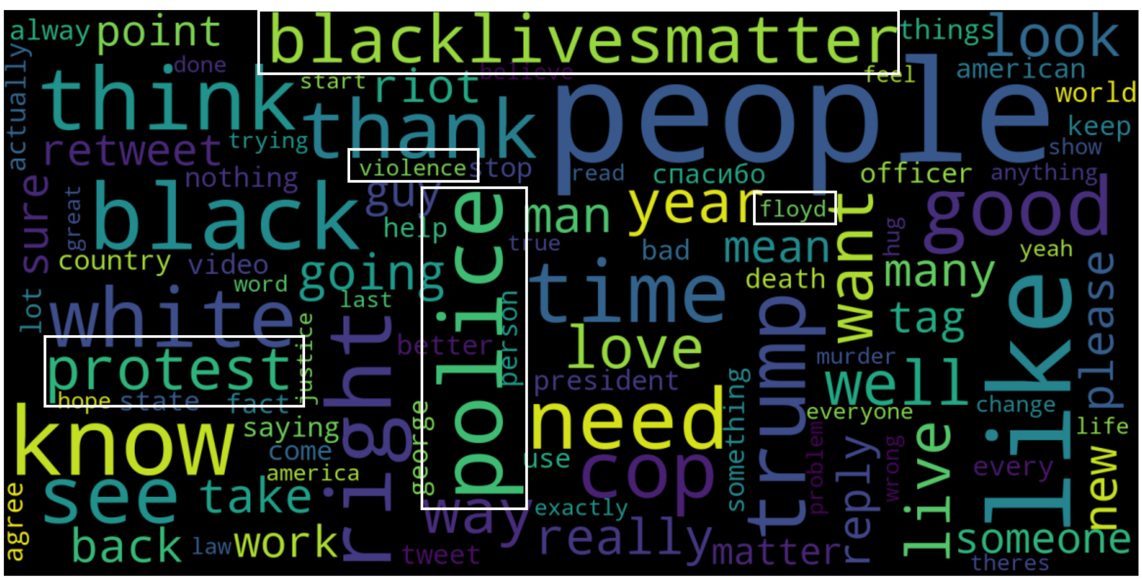
CNBC News Remote Work–April 24, 2020
https://www.cnbc.com/2020/04/24/as-working-from-home-becomes-more-widespread-many-say-they-dont-want-to-go-back.html

6.3 Theme 3: Events and Protests

Duration: May 3 – May 31, 2020

- Cinco De Mayo Celebration. (*Cinco de mayo*)
- Tact Tuesday –National Free Taco Tuesday Across USA. (*taco Tuesday*)
- Mother's Day Celebration. (*mother*)
- Black Lives Matter Movement Started after George Floyd's Fatal Causality (BLM)
- Public Outrage against Police Forces (*riot/violence*)
- Protests across USA Minnesota, Michigan, Georgia, Texas, California, Chicago. (*protest*)
- Implementation of curfew across multiple cities due to riots/protests. (*riot*)





CNN & Reuters -Mothers Day Celebration in COVID–May 9, 2020
https://www.cnn.com/2020/05/10/health/us-coronavirus-sunday/index.html
https://www.reuters.com/news/picture/marking-mothers-day-amid-the-coronavirus-idUSRTX7IBEU

The Guardian –BlackLivesMatter Movement –May 30, 2020, https://www.theguardian.com/us-news/2020/may/30/protest-washington-dc-george-floyd-white-house-trump

6.4 Sentiment Analysis

Once the qualitative analysis was finished, we obtained various sentiments, which we utilized for further analysis purposes.

The sentiments obtained are positive/negative/neutral.

Tweets that exhibited a negative tonality -> *negative sentiments*

Tweets that exhibited a positive tonality -> *positive sentiment*

Tweets that exhibit neither positive/negative tonality -> *neutral sentiments*

	Text	Sentiments
5	This is what we call talent pic.twitter.com/z3...	pos
18	Four left feet have never danced this well. S...	pos
23	Coronavirus special: Angelina Jolie moves to h...	pos
26	400 golf days we paid for + over 130 rallies. ...	pos
28	And it's time! Join us NOW for #CalltheMidwife...	pos
29	he looks like someone named john cena more tha...	pos
36	The internet is full of examples of unchecked ...	pos
37	i want it	pos
39	...aaand we're pressing play..... NOW! #CallTh...	pos
41	thank you so much!!!!! i love him too!!!!	pos
48	Depends on what he is calling essential. Some ...	pos
50	Further scientific evidence that when cats kno...	pos
71	Same. I used to get the automated reply but no...	pos
76	The expressions you chose for everyone were ju...	pos
91	C'est le problème d'avoir une justice gangréné...	pos
96	#SlobberingServility PENCE EMBARRASSING HIMS...	pos
109	you look very cute	pos
116	It's 9pm. Time to sleep! Have sweet QiYao dre...	pos
118	AJ took over our home in November. He is 10. S...	pos
126	Oh yes, please!	pos

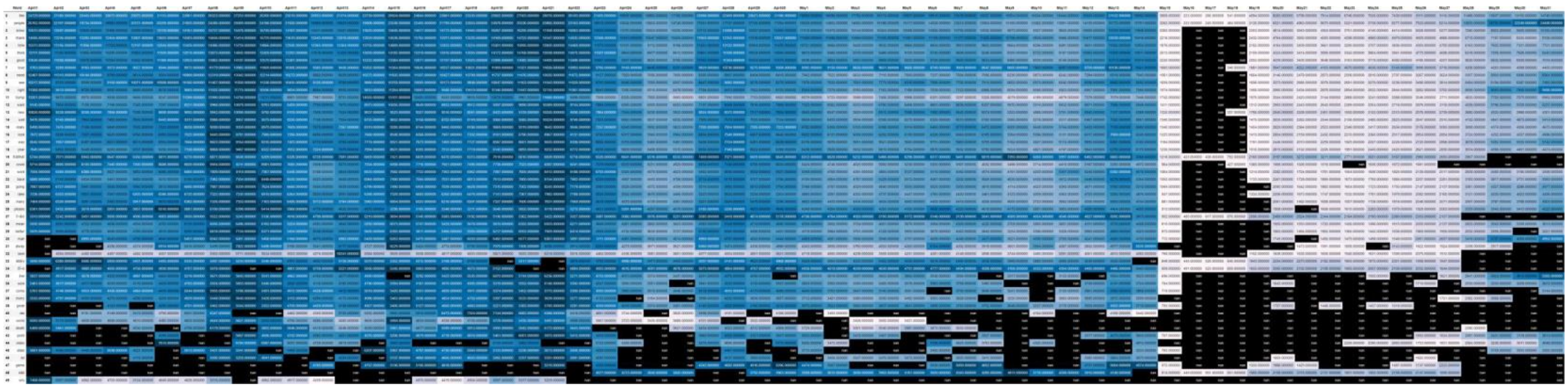
	Text	Sentiments
12	Without sounding rude, I prefer the real thing...	neg
24	I'm hard	neg
53	really yaar. attacking docs is not ok. . ... w...	neg
58	The problem is now with f2p they just keep mak...	neg
60	I kid if you can't catch my rebuttal of humor	neg
61	What even is the point of Twitter if not calli...	neg
64	Back...where the hell did i go??? pic.twitter....	neg
67	What a wonderful example of cherry picking you...	neg
80	i am so fucking bored i will not last until june	neg
110	No, I'm saying people got outraged that he fli...	neg
135	Russian hybrid forces launched 14 attacks with...	neg
138	Areas with low population density have very fe...	neg
146	I got a lot of that idea from Kyle Kulinski. B...	neg
156	I'm confused. In these crazy times, wouldn't o...	neg
165	Younger adults in New York City are being hosp...	neg
166	please explain how im sick	neg
167	November 2029, Peru conquered Colombia territo...	neg
170	I believe he's most known for having an ugly a...	neg
174	Sorry...	neg
175	Took him long enough. This dithering and pande...	neg

6.5 Tabular Analysis

This table displays the frequency of the top50 words(cumulative) in the dataset daily; this shows us how certain words were used much often on certain days, which might indicate the start of some trend or a movement in general.

The words have been shaded according to their frequencies over the two months. This will also help us see a pattern for these tweets; the darker the shade, the higher the frequency of these words.

We can see a significant drop in the frequencies of specific words later on into the month, we believe this could be the cause of the tweets for those particular days being corrupted, and meaningful data cannot be extracted from them



	Word	April1	April2	April3	April4	April5	April6	April7
0	like	24729.000000	21186.000000	20443.000000	20075.000000	20075.000000	21113.000000	22861.000000
1	people	26762.000000	22197.000000	19734.000000	18883.000000	20721.000000	20209.000000	21803.000000
2	know	15913.000000	13597.000000	12605.000000	12466.000000	12599.000000	13119.000000	14161.000000
3	thank	14595.000000	13236.000000	12205.000000	12454.000000	12807.000000	13603.000000	14303.000000
4	time	15373.000000	13156.000000	11956.000000	11725.000000	12197.000000	12044.000000	13459.000000
5	think	13111.000000	11150.000000	10566.000000	10295.000000	10883.000000	11509.000000	12019.000000
6	good	12626.000000	11058.000000	10675.000000	10384.000000	10422.000000	11389.000000	12053.000000
7	love	9763.000000	9289.000000	9163.000000	9373.000000	9057.000000	8944.000000	9972.000000
8	need	12487.000000	11313.000000	10144.000000	9789.000000	9614.000000	9504.000000	10989.000000
9	see	10217.000000	8703.000000	8099.000000	9150.000000	10071.000000	10508.000000	10160.000000
10	right	11268.000000	9513.000000	8726.000000	8590.000000	8485.000000	8618.000000	9869.000000

May23	May24	May25	May26	May27	May28	May29	May30	May31
6748.000000	7026.000000	7430.000000	9111.000000	9116.000000	11486.000000	13607.000000	13150.000000	14746.000000
5221.000000	5739.000000	6277.000000	7533.000000	8214.000000	13829.000000	18755.000000	22249.000000	24498.000000
3751.000000	4140.000000	4414.000000	5028.000000	5277.000000	7280.000000	8922.000000	9055.000000	10160.000000
4298.000000	4980.000000	4979.000000	4929.000000	4888.000000	6715.000000	7187.000000	6525.000000	8106.000000
3290.000000	3707.000000	3836.000000	4393.000000	4705.000000	6163.000000	7602.000000	7331.000000	7721.000000
3342.000000	3804.000000	3941.000000	4775.000000	4698.000000	6216.000000	7556.000000	7626.000000	8440.000000
3163.000000	3719.000000	4092.000000	4166.000000	4257.000000	5620.000000	6832.000000	6559.000000	7016.000000
4070.000000	4036.000000	5140.000000	3896.000000	3716.000000	4206.000000	4263.000000	4398.000000	4450.000000
2508.000000	2816.000000	2797.000000	3267.000000	3624.000000	5357.000000	6903.000000	6996.000000	8361.000000
2691.000000	3275.000000	3238.000000	3496.000000	3814.000000	5454.000000	6194.000000	6387.000000	7460.000000
2694.000000	2632.000000	2844.000000	3262.000000	3702.000000	5695.000000	8053.000000	7868.000000	9586.000000
2888.000000	3241.000000	3388.000000	3046.000000	2711.000000	4833.000000	7703.000000	7075.000000	6962.000000
2289.000000	2504.000000	2718.000000	2876.000000	3118.000000	4526.000000	5796.000000	5558.000000	6257.000000
1930.000000	2102.000000	2315.000000	2576.000000	2739.000000	3680.000000	4533.000000	3988.000000	4301.000000
2156.000000	2480.000000	2823.000000	2847.000000	2910.000000	4152.000000	4841.000000	4873.000000	5414.000000

6.6 Algorithm Analysis

Language Used: Python 3.8

We used Python as our primary programming language to implement such data processing. We used **pandas** to hold the data, **NumPy** to make calculations better, **nltk** library for various text processing features, **word cloud,** and **matplotlib** libraries to generate word clouds and **re** as a library for regular expressions usage. First of all, we had to separate the whole dataset by day. The whole dataset represents two months of ongoing discussion. It consists of ~46 million tweets, so it would be prolonged to process the whole dataset at once.

```
def separate(twits, i):
  if i < 8:
    twits = twits[(twits['Time']>='2020-04-0{} 00:00:00'.format(i+1)) & (twits['Time']<'2020-04-0{} 00:00:00'.format(i+2))]
  elif i == 8:
    twits = twits[(twits['Time']>='2020-04-0{} 00:00:00'.format(i+1)) & (twits['Time']<'2020-04-{} 00:00:00'.format(i+2))]
  else:
    twits = twits[(twits['Time']>='2020-04-{} 00:00:00'.format(i+1)) & (twits['Time']<'2020-04-{} 00:00:00'.format(i+2))]

  twits = twits.drop(twits.columns[0], axis = 1)
  twits = twits.dropna(axis = 0)
  twits = twits.reset_index(drop = True)

  return twits
```

After separating the dataset by days, we had to clean the tweets from some "junk": punctuations, mentions, links, emoticons, and numbers so that we would be left only with clean text. The next step is to tokenize the words to have many words with the same stem in one token. This would give better veracity in terms of top-used words. Then the idea is to combine all words into one string, separating them with ". "

```
regex_pattern = re.compile(pattern = "["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           "]+", flags = re.UNICODE)


linkpatternH = re.compile(r"http\S+")
linkpatternW = re.compile(r"www.\S+")
emojipattern = re.compile(r"emoji\S+")
linkimage = re.compile(r"pic.twitter.\S+")
mention = '@[A-Za-z0-9_]+'
hashtag = '#[A-Za-z0-9_]+'
punctuation = re.compile(r"[^\w\s]")
numbers = re.compile(r"[\d-]")
```

```python
def clean(t):

    lower_case = t.lower()
    del_pic = re.sub(linkimage, '', lower_case)
    del_linkH = re.sub(linkpatternH, '', del_pic)
    del_linkW = re.sub(linkpatternW, '', del_linkH)
    del_amp = BeautifulSoup(del_linkW, 'lxml')
    del_amp_text = del_amp.get_text()
    del_link_mentions = re.sub(mention, '', del_amp_text)
    del_hashtags = re.sub(mention, '', del_link_mentions)
    del_punctuation = re.sub(punctuation, '', del_hashtags)
    del_numbers = re.sub(numbers, '', del_punctuation)
    del_emoticons = re.sub(regex_pattern, '', del_numbers)
    del_emoji = re.sub(emojipattern, '', del_emoticons)

    words = token.tokenize(del_emoji)
    #words = token.tokenize(del_punctuation)
    result_words = [x for x in words if len(x) > 2]

    return (" ".join(result_words)).strip()
```

After cleaning the tweets, tokenizing the words, and putting them into one string, we started thinking of stop words. The initial stop words list from the **nltk** library is not complete. It has some apparent words like articles, pronouns but we needed more. After many discussions, we made our list of words that appended the standard one. This complete list was used in the word cloud generation and obtained the top 50 words for one specific day.

```python
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

stopwordslist = set(stopwords.words("english"))
ad_words = readFile('drive/MyDrive/WSU/Twitter/stopwords.txt')
stopwordslist.update(ad_words)
stopwordslist.update(stopwords.words("russian"))
```

After defining all these functions, we had to implement the main body. We have put the separator, the cleaner, the wordcloud generator, the top50 word qualifier, and a sentiment analyzer there.

```python
for j in range(16):
  sentiments = []
  twitsDays[j] = separate(twits, j)
  print("Cleaning the tweets for day {}...\n".format(j+1))
  twitsCleaned[j] = []
  for i in range(twitsDays[j].shape[0]):
    if( (i+1)%100000 == 0 ):
        print("Tweets {} of {} have ben processed".format(i+1,twitsDays[j].shape[0]))

    x = clean((twitsDays[j].Text[i]))
    twitsCleaned[j].append(x)
    score_f = list(vds.polarity_scores(x).values())
    if (score_f[3] > 0):
      sent = 'pos'
    elif (score_f[3] == 0):
      sent = 'neu'
    else:
      sent = 'neg'
    sentiments.append(sent)

  twitsDays[j]['Sentiments'] = sentiments
  twitsDays[j].to_csv("drive/MyDrive/WSU/Twitter/Andrei/DaysByTweets/April{}.csv".format(j+1))
  string[j] = pd.Series(twitsCleaned[j]).str.cat(sep=' ')
  wordcloud = []
  wcpt = []
  wcpt = WordCloud(width=1600, stopwords=stopwordslist,height=800,max_font_size=200,max_words=100,collocations=False, background_color='black').process_text(string[j])
  top50[j] = dict(sorted(wcpt.items(), key=lambda item: item[1], reverse = True)[:50])
  writeToCSV(top50[j], j)
  wordcloud = WordCloud(width=1600, stopwords=stopwordslist,height=800,max_font_size=200,max_words=100,collocations=False, background_color='black').generate(string[j])
  wordcloud.to_file("drive/MyDrive/WSU/Twitter/Andrei/WordClouds/April{}.png".format(j+1))
```

For the sentiment analysis part of the project, we have used Vader, a sentiment analysis tool directly available from the nltk library. For this part, we needed a tool that could help us detect the polarity of these tweets without needing labeled data. As Vader deals with unlabelled data, it became the perfect choice for us. We calculated the mean sentiment for the top50 words from the tabular analysis using Vader.

7. Conclusion:

From the above research our the main aim was a qualitative analysis of tweets analyzed between April 1, 2020, and May 31, 2020.

After extensive data cleaning of the tweets being analyzed, using regular expression and the NLTK Library along with Beautiful Soup implementation -> word clouds were generated for each day, i.e., 60-word clouds.

We found unique keywords specific to the specified time frames. From the time frame, we noticed certain events.

Furthermore, we observed a transition from events related to the COVID virus to various riots and protests based on movements.

This analysis was corroborated with verified news articles searched for during those specified periods from reputed news channels such as Reuters News, CNN, ABC7, etc.

We obtained various sentiments from these tweets and analyzed their tonality to be positive, negative, or neutral.

In conclusion, a tabular analysis was done to observe the frequency of the top 50 words for each day from the initial to the last date.

8. Bibliography:

1. R. Kaptein, "Using Wordclouds to Navigate and Summarize Twitter Search Results," p. 4.
2. C. Kariya and P. Khodke, "Twitter Sentiment Analysis," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-3, doi: 10.1109/INCET49848.2020.9154143.
3. A. I. Kabir, R. Karim, S. Newaz, and M. I. Hossain, "The Power of Social Media Analytics: Text Analytics Based on Sentiment Analysis and Word Clouds on R," IE, vol. 22, no. 1/2018, pp. 25–38, Mar. 2018, doi: 10.12948/issn14531305/22.1.2018.03.
4. https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664
5. https://ieeexplore.ieee.org/document/9154143