Phase 1 Report

February 22, 2021

1 Utilizing Twitter to Measure the Public's Sentiment on the Pandemic & Its Relevant Protective Measures Over Time

Team 7: Insert Name Here

Members: Lipsa J., Ji K., Yuanfeng L., Yu L.

1.0.1 Background and Motivation

The pandemic has uprooted the lives of every single person in the world. While it began as a minor inconvenience to many people, the harsh reality and severity of the virus were soon realized. In the beginning of enforcing protective measures to protect the public, many people's opinions on the virus, protective rules & procedures, and other topics relating to the pandemic have changed and continually do into 2021.

With such a slow response to protective measures in the U.S compared to other countries globally, we wanted to find out the public's stance on the matter over the period of nearly the entire pandemic.

With this in mind, we want to record and analyze these trends by looking at the metrics such as sentiment, LIWC metrics, and possibly more as we make further discoveries.

Utilizing Twitter, an online social media platform for sharing content and microblogging, we'll be analyzing "tweets" (publically posted messages) from everyday people about how they feel about the pandemic. This procedure will be run on data from January 22 2020, all the way to the most current data being available at this current time of the project (February and March of 2021). We believe this approach will work compared to traditional forms of collecting data on the topic. Twitter specifically has proven itself to be accurate, quick, and better reflect the perspectives of the everyday person since they're the ones whose data we're processing.

We've outlined objectives & research questions we hope to answer through this approach. * **Goal 1**: Find out how many tweets sentiments changed on the regulation or rules about wearing a mask or taking a vaccine for the the year 2020 and current months in 2021 (January - March) * **Goal 2**: Find out the sentiment of tweets relating to the COVID-19 virus for the year 2020 and the current months in 2021 (January - March) * **Stretch Goal 1**: Find out the sentiments across geographical locations within the U.S about either protective measures (Eg. Wearing a mask) and the taking the vaccine. It's been shown throughout various news outlets and social media that different areas in the U.S have had varying responses to these rules. If time & resources allow, we want to run the research experiment at a lower level - focusing on specific areas in the U.S - Perhaps areas with the lowest cases per capita vs. moderate vs. high. * **Stretch goal 2**: Relate our findings to how misinformation & fake news on Twitter changed before and after the election; as well as its

possible consequences on the public's sentiment on the topic of COVID-19 and its related topics (Eg. vaccines, lockdown, social distancing).

Through our efforts, we hope to be able to answer or at least find insight into the following questions as well: * Have specific events affected the public's stance on the pandemic? These could be the presidential election, the presidential candidates debate as well as the vice-presidents debate, Trump getting diagnosed and hospital stayed, and so on. * How have different cities, counties, and states efficacy in containing the virus relate to the public sentiment from the people there? * Is there a positive trend or at least an upward direction for sentiment on social media comparing pre-election to post-election? * What are the trends in Democratic-heavy vs. Republican-heavy vs. Well-mixed (Eg. close to 50/50 Democrat to Republican) States in terms of the pandemic and protective measures?

1.0.2 Our Current Approach

Currently, we're utilizing Twitter for our data collection. Initially, we stated that we'd be utilizing Reddit as well but the caveats of a public forum platform is that it's heavily moderated. With the pandemic being a global crisis, Reddit has emphasized initiatives to remove and censor posts that may be incediary, controversial, promote misinformation and so on. While these things are negative in the grand scheme of society, we actually want to collect this kind of data as well since it shows a sub-population with different views. There is a promising subreddit about Parler, the right-wing social media platform, in which users post the most outrageous or controversial posts they see on the platform to re-post and discuss on reddit. This seemed promising at first but the format mirrors satire and other users have cherry-picked those specific posts and typically upload them as images. It wouldn't accurately reflect the Parler population on specific topics, would be difficult to parse for specific keywords, and may not produce enough data.

We're utilizing Tweepy which is Twitter's API wrapper for Python. It's extremely easy to utilize but one of its caveats is that it will only look at the past week to pull data; which makes sense since many people actually use real-time data for analysis. To get around this issue, we relied on Kaggle and IEEE. Both of them have been data mining the ID number of tweets with keywords relating to the pandemic since near the beginning of 2020. These keywords include identifiers such as "n95", "ppe", "washyourhands", "stayathome", "selfisolating", "social distancing", "covid-19", and so on.

Utilizing Tweepy and Python, we iterate through these tweet id values to pull the actual tweet status object from Twitter. From there, we extract the following information: * id: ID number of the tweet * username: Username of the person who posted the tweet * text: The literal text content of the tweet * entities: Hashtags the tweet had * retweet_count: Number of times the tweet had been retweeted * favorite_count: Number of times the tweet had been favorited * created_at: Time the tweet had been posted

We're collecting our own data currently with the same parameters and keywords. Taking these datasets, in a csv format, we're running each through LIWC and looking at the following metrics: * Summary variables: Analytical thinking, clout, authentic, and emotional tone * Affect words: Positive emotions, negative emotions, anxiety, anger, sadness * Social words: Family, friends, female referents, male referents * Cognitive Processes: Insight * Biological processes: Body, health/illness * Personal concerns: Work, leisure, money * Informal speech: Swear words

While the biggest contributors will be relating to authenticity, emotions (emotional tone & posi-

tive/negative emotions), we believe the other attributes will aid answering in our research questions and stretch goals.

1.1 Collecting Twitter Data From the Entirety of 2020

As mentioned previously, since we can only directly scrape tweets for the past week, we utilize Kaggle's dataset which is found at https://www.kaggle.com/lopezbec/covid19-tweets-dataset.

Additionally, the IEEE have published a similar dataset with a wider range of keywords which can be found at https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset.

These files contain minimal data in order to save space. Kaggle's has just the tweet ids in a list-like structure, while the IEEE has a similar format but in pairs of tweet id and a sentiment score calculated for the content of the tweet.

Here are a sample of what the Kaggle files look like. We'll be including sized down versions of them in our submission.

[ ]:
```
import numpy as np
kaggle_sample = open('./sample_data/sample_kaggle.txt', 'r')
#removes beginning and ending '[' and  ']'
content = kaggle_sample.read()[1:-1]
#Delimit on comma, convert to int, store into a numpy array for parsing
ids = np.fromstring(content, dtype=int, sep= ',')
print("Number of tweets in sample: {}".format(len(ids)))
print("Sample of the Tweet IDS: {}".format(ids))
```

As mentioned, to save space Kaggle stores these tweet ids as a plaintext, in a list like structure... Eg. [id1, id2, id3, ..., idn]

On the otherhand, IEEE provides theirs as a csv format or a zipped file containing a csv file (to save space) and theirs looks like...

[ ]:
```
import pandas as pd
ieee_sample = pd.read_csv("./sample_data/sample_ieee.csv", header=None)
ieee_sample.columns = ['tweet_id', 'sentiment']
ieee_sample.head()
```

The IEEE data is much easier to work with since we can extract the tweet_id values directly by extracting the column. However, both are eventually in the same format which lets us run our data collector. It should be noted that, unlike the Tweepy API's 7 day period that it can return data for, the get_status() method for getting individual tweets given a tweet id does each call one by one. This takes much longer time but allows us to still get 15,000 records under the rate limit which means we collect data as fast as Tweepy will let us regardless.

1.2 Processing the Tweet IDs

Due to the organizational structure both IEEE and Kaggle have (Eg. each day have their own files, files are grouped by month, etc.), we followed a similar approach. Below is code taken from a previous personal project Ji-Hoon has done just to iterate through directories. The directory

navigation portion isn't entirely important but for breadth and covering bases, we've included this portion if you want to run the file as well.

To perserve the name and directory structure each sub-directory has, it is a recursive method that holds the absolute path of the files (So Python can read in the file's contents) while also preserving the subdirectory path so we can name each file accordingly and know which dataset it came from.

[ ]:
```
import pandas as pd
import os
import numpy as np
import tweepy
from collections import defaultdict
import json
import time

max_num_records = 5000

"""
Methodology we covered in class to just load the twitter credentials into␣
 ↪appropriate objects.
This assumes a file with your Twitter developer credentials are in a file named␣
 ↪'twitter.json'
and is in the same directory as the program when it's being run.
"""
def load_keys(key_file):
    with open(key_file) as f:
        key_dict = json.load(f)
    return key_dict['api_key'], key_dict['api_secret'], key_dict['token'],␣
 ↪key_dict['token_secret']


"""
Recursive method to navigate through many directories.
"""
def iterate_files(path, subdir):
    KEY_FILE = "./twitter.json" # Twitter credentials. See def␣
 ↪load_keys(key_file):
    api_key, api_secret, token, token_secret = load_keys(KEY_FILE)
    auth = tweepy.OAuthHandler(api_key, api_secret)
    auth.set_access_token(token, token_secret)
    api = tweepy.API(auth)

    # File recursion portion. Not pertinent to the actual data collection.
    for filename in os.listdir(path):
        filePath = path + "/" + filename
        # If folder, recursively call.
        if filename == 'scraped_data': # Skip folder containing the scraped␣
 ↪data.
```

```
            continue
        if (os.path.isdir(filePath)):
            tempSubdir = ""
            if subdir: tempSubdir = subdir + "/" + filename
            else: tempSubdir = filename
            iterate_files(filePath, tempSubdir)
        # Otherwise, process the file.
        else:
            filekey = subdir
            if subdir: file = subdir + "/" + filename
            else: file = filename
            tweet_content = defaultdict(list)

            """
            TODO: Refactor to detect .txt and .csv to know which dataset it␣
↪came from.
            """
            # So it doesn't read itself or the credentials file
            #if filename not in ['process_ieee.py', 'process_tweets.py',␣
↪'twitter.json']:
            if filename.endswith('.csv'):
                # NOTE: This version of the program is assuming .csv files --␣
↪IEEE data.
                # Samples 5000 records from the data set, takes only the column␣
↪values, then ravels into a np array
                tweet_ids = pd.read_csv(filePath).sample(n=max_num_records).
↪iloc[:,0].values.ravel()
                print("Collecting from file: {}".format(filename))
                # Iterates through each of the tweet ids we sampled.
                for id in tweet_ids:
                    """
                    Must be in a try-catch structure. If a twitter user is␣
↪banned or suspended, the tweet_id
                    refers to data that doesn't exist. The Tweepy api will␣
↪return a 400-level HTTP status code
                    due to the resource not being found - which is considered␣
↪an exception.
                    """
                    try:
                        print(num_records) # Debugging. Just to see that the␣
↪program wasn't stalling.
                        tweet = api.get_status(id) #returns status object
                    except tweepy.RateLimitError:
                        print("Rate Limit hit. Sleeping for 15 minutes.")
                        time.sleep(900)
                        print("Resuming...\n")
```

```
                        continue
                    except Exception as e: # It will throw an exception if␣
↪twitter user has actually been suspended
                        continue
                    if tweet is None:
                        print("Should never be reached. If seen, something went␣
↪wrong.")

                    # Features we're extracting.
                    tweet_content['id'].append(tweet.id)
                    tweet_content['username'].append(tweet.user.name)
                    tweet_content['text'].append(tweet.text)
                    tweet_content['entities'].append(tweet.entities)
                    tweet_content['retweet_count'].append(tweet.retweet_count)
                    tweet_content['favorite_count'].append(tweet.favorite_count)
                    tweet_content['created_at'].append(tweet.created_at)
                result_filename = './scraped_data/' + filename
                """
                We control how many records we want from each day. It'll either␣
↪run through the entire file
                or run based on how many records we want sampled.
                """
                pd.DataFrame(tweet_content).to_csv(result_filename)
                print("Done processing: {}".format(result_filename))


def get_path():
    iterate_files(os.getcwd(), "")

if __name__ == "__main__":
    get_path()
```

An important note to make here is that this Python code wasn't designed to run in a Jupyter Notebook. If you wish to replicate this portion, place this python program in the same directory as the IEEE sample tweet ids along with your twitter.json file.

1.3 Data Collection For Recent Data in 2021

We relied on those datasets to help supplement the earlier twitter data we can't directly retrieve. However, for the current data, we're querying and data mining with a similar approach.

We have taken inspiration from the IEEE Coronavirus (COVID-19) Tweets Dataset, which can be found at https://ieee-dataport.org/open-access/coronavirus-covid-19-tweets-dataset. They have collected tweets relating to a large set of keywords since the very beginning of the pandemic and continually do so. We have taken a scaled down version and taken specific keywords from their larger set - which can be found at https://rlamsal.com.np/keywords.tsv. Note: The link will start the download of a tab-separated file with the keywords but is small in terms of memory size. Just a warning.

The approach for collecting recent data has stayed essentially the same as we did in our Lab 3 for the introduction to Tweepy. One difference is that, since we're all constrained with the rate limit, we've separated the main topics to be scraping for. The keywords each person utilized to scrape will be shown at the credits/work distribution section at the end.

Dividing up the number of keywords, we each scraped data looking for our specific delegated keywords.

[ ]:
```
# setting the keywords that I would like to search on
"""
NOTE: This is assuming that the twitter credentials were loaded properly
prior to being run.

This version of code is looking specifically at tweets relating to masks.
Other keywords scraped are shown in the work distribution/credits.
"""
keywords = ["#Wearmask",
            "#Wearmasks",
            "#mask",
            "#masks4all",
            "#n95 mask",
            "#n95 respirator mask"
]



import time
SLEEP_TIME = 60 * 15


def read_write_tweets(search_term, target_page_list):
    """
    it search the serach term in twitter, and write num_items items
    in csv_file

    @parameters:
    search_term: the keyword you use to search, such as 'covid19'
    num_items: how many items for each search
    """
    result_list = []
    page_list = target_page_list
    try:
        for page in tweepy.Cursor(api.search, q=search_term + " -filter:
 ↪retweets", lang
                                  = 'en', tweet_mode="extended").pages():
            if page not in page_list:
                page_list.append(page)
```

```
                for tweet in page:
                    csvWriter.writerow([tweet.id, tweet.user.name, tweet.
 ↪full_text, tweet.entities, tweet.retweet_count,
                                        tweet.favorite_count, tweet.created_at])
    # Rate limit hit. Must sleep for 15 minutes.
    except Exception as e:
        print(e, '; Will Sleep for:',SLEEP_TIME)
        print("now time:", datetime.datetime.now().time())
        # Print out current version of data scraped
        return_df_info(filename)
        # Sleep for 15 minutes
        time.sleep(SLEEP_TIME)
        # Resume scraping
        read_write_tweets(search_term, page_list)

# make filename,
filename = 'ProjectPhase_1_Fv05_' + (datetime.datetime.now().
 ↪strftime("%Y-%m-%d-%H")) + '.csv'
# r+ will not created the file, if it not existed, rest the same

import datetime

time_start = datetime.datetime.now().time()
print("Starting at:", time_start)
with open (filename, 'a+', newline='', encoding="utf-8") as csvFile:
    csvWriter = csv.writer(csvFile)
    # first row is the titles for columns
    csvWriter.writerow(["tweet_id", "username", "text_of_tweet",␣
 ↪"tweet_entities", "retweet_count", "favorite_count", "created_at"])
    # for each keyword, write 350 items in csv_file
    for keyword in keywords:
        read_write_tweets(keyword,[])


time_finish = datetime.datetime.now().time()
print("Finished at:", time_finish)
print("Spendt time:", datetime.datetime.combine(datetime.date.today(),␣
 ↪time_finish) - datetime.datetime.combine(datetime.date.today(), time_start))
```

Since we're all searching for different keywords, it may show that we have duplicates in our set. So we have defined a simple way to see data as we aggregate our findings.

[2]:
```
# showing informations
import pandas as pd

def return_df_info(target_file):
    df = pd.read_csv(target_file)
```

```
    print("shape:", df.shape)
    dup = df.duplicated().sum()
    print("duplicated rows:", dup)
    print("not duplicated data:", df.shape[0] - dup)
    print("df head(): \n", df.head())
    print("df tail(): \n", df.tail())
    print("-----------------------------------------------------------")
    print()

# return_df_info("ProjectPhase_test012021-02-20-13.csv")

return_df_info("./scraped_data/LIWC2015 Results (march_31_2020).csv")
```

```
shape: (3388, 101)
duplicated rows: 0
not duplicated data: 3388
df head():
    Source (A)           Source (B)       Source (C)  \
0         NaN                   id         username
1         0.0  1245016268060622849           angel
2         1.0  1245147924289503232              Rui
3         2.0  1244894426872324096     Zeno Protect
4         3.0  1244868044498841605  destiny's niece

                                        Source (D)  \
0                                             text
1       RT @DailyDoseOfKia: IF "IDGAF" Was A State
2                                       Corona time
3    Schools are empty due to the Corona Crises…
4  i thought ansel elgort got corona bc he was tr…

                                        Source (E)     Source (F)  \
0                                         entities  retweet_count
1  {'hashtags': [], 'symbols': [], 'user_mentions…           4830
2  {'hashtags': [], 'symbols': [], 'user_mentions…              0
3  {'hashtags': [], 'symbols': [], 'user_mentions…              0
4  {'hashtags': [], 'symbols': [], 'user_mentions…              0

       Source (G)           Source (H)  WC  Analytic  …  Comma  Colon  \
0  favorite_count           created_at   1     93.26  …    0.0   0.00
1               0  2020-03-31 15:53:04   7     29.30  …    0.0  14.29
2               1  2020-04-01 00:36:13   2     93.26  …    0.0   0.00
3               0  2020-03-31 07:48:55  25     99.00  …    0.0   0.00
4               2  2020-03-31 06:04:04  18      4.69  …    0.0   0.00

   SemiC  QMark  Exclam  Dash  Quote  Apostro  Parenth  OtherP
0    0.0    0.0     0.0   0.0   0.00      0.0      0.0    0.00
```

```
1    0.0    0.0     0.0   0.0  28.57      0.0      0.0   14.29
2    0.0    0.0     0.0   0.0   0.00      0.0      0.0    0.00
3    0.0    0.0     0.0   0.0   0.00      0.0      0.0    8.00
4    0.0    0.0     0.0   0.0   0.00      0.0      0.0    0.00

[5 rows x 101 columns]
df tail():
       Source (A)           Source (B)             Source (C)  \
3383      3382.0  1244896269539016705                  Amaan
3384      3383.0  1244946151020933122  Engr. H A Waheed Butt
3385      3384.0  1244900194598051840           Rezaul haque
3386      3385.0  1245090887555678208  dobrik / datalie stan
3387      3386.0  1244905249128644614            Sarath Nair

                                              Source (D)  \
3383  RT @LOLrakshak: An Indian virologist has devel…
3384  RT @kdastgirkhan: "Keys to resolve the Corona …
3385  RT @AunindyoC: Religious congregation on March…
3386  sad that joe and annelise have the corona. i r…
3387  RT @oommen: @sushantsareen "All the corona pat…

                                              Source (E) Source (F) Source (G)  \
3383  {'hashtags': [], 'symbols': [], 'user_mentions…       1327          0
3384  {'hashtags': [], 'symbols': [], 'user_mentions…         38          0
3385  {'hashtags': [], 'symbols': [], 'user_mentions…        589          0
3386  {'hashtags': [], 'symbols': [], 'user_mentions…          0          0
3387  {'hashtags': [], 'symbols': [], 'user_mentions…         14          0

              Source (H)  WC  Analytic  …  Comma  Colon  SemiC  QMark  \
3383  2020-03-31 07:56:14  25     63.83  …   4.00   4.00    0.0   0.00
3384  2020-03-31 11:14:27  21     99.00  …   9.52   9.52    0.0   0.00
3385  2020-03-31 08:11:50  21     93.26  …   4.76   4.76    0.0   4.76
3386  2020-03-31 20:49:34  14     29.30  …   0.00   0.00    0.0   0.00
3387  2020-03-31 08:31:55  25     30.73  …   0.00   4.00    0.0   0.00

      Exclam  Dash  Quote  Apostro  Parenth  OtherP
3383     0.0   0.0   0.00      0.0      0.0    4.00
3384     0.0   0.0   4.76      0.0      0.0    4.76
3385     0.0   0.0   0.00      0.0      0.0    4.76
3386     0.0   0.0   0.00      0.0      0.0    0.00
3387     0.0   0.0   8.00      0.0      0.0    8.00

[5 rows x 101 columns]
----------------------------------------------------
```

1.4 Processing the Data

We collected all the data in the same fashion, ran the text fields through LIWC, and separated files to organize based on the time period they represent. We ran these files through LIWC's program which outputs a copy of the same file but with new columns pertaining to the metrics and corresponding values LIWC has produced.

A snippet of the outputted file can be shown here:

[3]:
```
import pandas as pd
sample = pd.read_csv('./scraped_data/LIWC2015_feb.csv')
print(sample.head())

print("Column Names: {}".format(list(sample.columns)))
```

```
   standard            id            username  \
0         0  1.363275e+18        Fire Is Born
1         1  1.363263e+18   MyFrenchDietitian
2         2  1.363260e+18  healingcolorsmusic
3         3  1.363260e+18  healingcolorsmusic
4         4  1.363260e+18  healingcolorsmusic

                                                text  \
0  @iamungit I've been in one of them in San Fran…
1  Enjoy the #weekend, go #outdoor, reconnect wit…
2  #healingcolorsmusic #art #music …there is #S…
3  #healingcolorsmusic #art #music …there is #S…
4  #healingcolorsmusic #art #music …there is #S…

                                            entities retweet_count  \
0  {'hashtags': [{'text': 'WearMask', 'indices': …              1
1  {'hashtags': [{'text': 'weekend', 'indices': […              0
2  {'hashtags': [{'text': 'healingcolorsmusic', '…              0
3  {'hashtags': [{'text': 'healingcolorsmusic', '…              0
4  {'hashtags': [{'text': 'healingcolorsmusic', '…              0

  favorite_count           created_at  WC Analytic  … Quote Apostro Parenth  \
0              1  2021-02-20 23:52:00  25    93.26  …   0.0     4.0     0.0
1              1  2021-02-20 23:01:59  24    93.26  …   0.0     0.0     0.0
2              1  2021-02-20 22:52:27  19    93.26  …   0.0     0.0     0.0
3              1  2021-02-20 22:50:59  19    93.26  …   0.0     0.0     0.0
4              1  2021-02-20 22:49:58  19    93.26  …   0.0     0.0     0.0

  OtherP  Unnamed: 101  Unnamed: 102  Unnamed: 103  Unnamed: 104  \
0  16.00           NaN           NaN           NaN           NaN
1  29.17           NaN           NaN           NaN           NaN
2  52.63           NaN           NaN           NaN           NaN
3  52.63           NaN           NaN           NaN           NaN
4  52.63           NaN           NaN           NaN           NaN
```

```
   Unnamed: 105  Unnamed: 106
0           NaN           NaN
1           NaN           NaN
2           NaN           NaN
3           NaN           NaN
4           NaN           NaN

[5 rows x 107 columns]
Column Names: ['standard', 'id', 'username', 'text', 'entities',
'retweet_count', 'favorite_count', 'created_at', 'WC', 'Analytic', 'Clout',
'Authentic', 'Tone', 'WPS', 'Sixltr', 'Dic', 'function', 'pronoun', 'ppron',
'i', 'we', 'you', 'shehe', 'they', 'ipron', 'article', 'prep', 'auxverb',
'adverb', 'conj', 'negate', 'verb', 'adj', 'compare', 'interrog', 'number',
'quant', 'affect', 'posemo', 'negemo', 'anx', 'anger', 'sad', 'social',
'family', 'friend', 'female', 'male', 'cogproc', 'insight', 'cause', 'discrep',
'tentat', 'certain', 'differ', 'percept', 'see', 'hear', 'feel', 'bio', 'body',
'health', 'sexual', 'ingest', 'drives', 'affiliation', 'achieve', 'power',
'reward', 'risk', 'focuspast', 'focuspresent', 'focusfuture', 'relativ',
'motion', 'space', 'time', 'work', 'leisure', 'home', 'money', 'relig', 'death',
'informal', 'swear', 'netspeak', 'assent', 'nonflu', 'filler', 'AllPunc',
'Period', 'Comma', 'Colon', 'SemiC', 'QMark', 'Exclam', 'Dash', 'Quote',
'Apostro', 'Parenth', 'OtherP', 'Unnamed: 101', 'Unnamed: 102', 'Unnamed: 103',
'Unnamed: 104', 'Unnamed: 105', 'Unnamed: 106']

/home/jihk/.local/lib/python3.8/site-
packages/IPython/core/interactiveshell.py:3155: DtypeWarning: Columns
(10,11,12,13) have mixed types.Specify dtype option on import or set
low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

We have noticed some problems across different operating systems for handling csv files. We initially ran into problems while sharing datasets with each other across Debian, Windows, and Mac and have resolved most of them since. One example is that some empty columns will show themselves as "Unnamed" columns with empty or NaN values in them. We ignore these values.

[4]:
```
sample.nlargest(5, ['posemo'])
```

[4]:
```
       standard           id                                    username  \
15037     40342  1.363335e+18                     King Jamison Fawkes
15994     43474  1.363335e+18                     King Jamison Fawkes
9187      21521  1.363309e+18  FA_eye(formally Accureye) #CyberPunk2077
12488     32070  1.363325e+18                                  TEA POt
14506     38373  1.363325e+18                                  TEA POt

                                                   text  \
15037  @WildHogPower "WELL WELL WELL WELL WELL WELL W…
15994  @WildHogPower "WELL WELL WELL WELL WELL WELL W…
```

```
9187            @TheSphereHunter Nice love it great mask
12488  @FailedSoul_ *winning laughs* pretty impressiv…
14506  @FailedSoul_ *winning laughs* pretty impressiv…

                                                 entities retweet_count  \
15037  {'hashtags': [], 'symbols': [], 'user_mentions…              0
15994  {'hashtags': [], 'symbols': [], 'user_mentions…              0
9187   {'hashtags': [], 'symbols': [], 'user_mentions…              0
12488  {'hashtags': [], 'symbols': [], 'user_mentions…              0
14506  {'hashtags': [], 'symbols': [], 'user_mentions…              0

      favorite_count           created_at  WC Analytic  … Quote Apostro  \
15037              0  2021-02-21 03:50:29  26        1  …  3.85     0.0
15994              1  2021-02-21 03:50:29  26        1  …  3.85     0.0
9187               0  2021-02-21 02:05:49   6    62.04  …  0.00     0.0
12488              0  2021-02-21 03:10:45   8    72.69  …  0.00     0.0
14506              1  2021-02-21 03:10:45   8    72.69  …  0.00     0.0

      Parenth OtherP  Unnamed: 101  Unnamed: 102  Unnamed: 103  Unnamed: 104  \
15037     0.0  11.54           NaN           NaN           NaN           NaN
15994     0.0  11.54           NaN           NaN           NaN           NaN
9187      0.0  16.67           NaN           NaN           NaN           NaN
12488     0.0  50.00           NaN           NaN           NaN           NaN
14506     0.0  50.00           NaN           NaN           NaN           NaN

       Unnamed: 105  Unnamed: 106
15037           NaN           NaN
15994           NaN           NaN
9187            NaN           NaN
12488           NaN           NaN
14506           NaN           NaN

[5 rows x 107 columns]
```

The 5 tweets from the most recently collected data that have the highest scores in terms of positive emotions. However, we noticed that even tweets that have a positive sentiment initially can that the overall message is negative. This is why the other metrics are utilized alongside. For comparison, here is the top 5 most negative tweets.

[5]:
```
sample.nlargest(5, ['negemo'])
```

[5]:
```
       standard            id                          username  \
12848     32430  1.363322e+18                        BIG_BOOB$
20535     61597  1.363367e+18                      Sassy | BLM
235         235  1.360839e+18                       calledryan
9540      21874  1.363307e+18                      KingOfSoup
10850     26761  1.363317e+18  President Dr.Jillian(MAGA Bean)
```

```
                                                    text  \
12848                WEAR A FUCKING MASK YOU STUPID FUCK
20535  Goodnight. Fuck racists. Fuck Ted Cuntface Cru…
235                                copernicus was wrong
9540                                     My mask ugly
10850  America is full of fools. Weak mask wearing fo…

                                                entities retweet_count  \
12848  {'hashtags': [], 'symbols': [], 'user_mentions…              0
20535  {'hashtags': [], 'symbols': [], 'user_mentions…              0
235    {'hashtags': [], 'symbols': [], 'user_mentions…              0
9540   {'hashtags': [], 'symbols': [], 'user_mentions…              0
10850  {'hashtags': [], 'symbols': [], 'user_mentions…              1

      favorite_count           created_at  WC Analytic  … Quote Apostro  \
12848              0  2021-02-21 02:59:35   7    93.26  …   0.0     0.0
20535              0  2021-02-21 05:58:21  11    98.34  …   0.0     0.0
235                0  2021-02-14 06:30:15   3    18.82  …   0.0     0.0
9540               0  2021-02-21 01:57:06   3    18.82  …   0.0     0.0
10850              6  2021-02-21 02:39:01   9    93.26  …   0.0     0.0

      Parenth OtherP  Unnamed: 101  Unnamed: 102  Unnamed: 103  Unnamed: 104  \
12848     0.0    0.0           NaN           NaN           NaN           NaN
20535     0.0    0.0           NaN           NaN           NaN           NaN
235       0.0    0.0           NaN           NaN           NaN           NaN
9540      0.0    0.0           NaN           NaN           NaN           NaN
10850     0.0    0.0           NaN           NaN           NaN           NaN

       Unnamed: 105  Unnamed: 106
12848           NaN           NaN
20535           NaN           NaN
235             NaN           NaN
9540            NaN           NaN
10850           NaN           NaN

[5 rows x 107 columns]
```

A similar situation happens. We can see that some of these tweets have negative emotions over the fact not enough people are wearing masks while others have negative emotions *because* of masks.

1.4.1 Next Steps

After collecting and processing the actual data, some thoughts have come up. 1. How will we define overall positive sentiment a tweet has against protective public initiatives such as masks, social distancing, vaccines, and so on? It's shown that a tweet can be considered extremely negative but due to not enough people following those public health guidelines. Negativity or positivity of a message doesn't equate to their own feelings about those topics. This will primarily be looking at

how LIWC will define sentiment - perhaps combining with either TextBlob or Vader.

2. Sizing DOWN our data. Since we're doing a trend over nearly the entirety of the pandemic, we need to be able to define how much data we'll be collecting from each day/week/month/etc. To put things in perspective, over 1 billion tweets have been collected based on keywords relating to the COVID-19 Pandemic. Size, computational, and time-wise it's not feasible to process over 1 billion. Currently, we're thinking about roughly 3000-5000 tweets per week in 2020.
3. Explore with clustering on the data. Possibly utilizing Expectation Maximization algorithm to confidently define how many clusters there are. Then within each cluster, gain a sense of what the group represents looking at the matching keywords and LIWC scores.
4. For tweets in 2020, must divide up the larger datasets based on the keywords so we can then cross compare the various keywords/topics over time. We can then start to compare recent data about specific keywords with data in 2020 about the same keywords, then do this for the entire time range we have.

1.4.2 Concerns & Notes

1. We have noticed a few things after collecting data. Tweets, by default, are limited to 140 characters by default but can get up to 280 by setting "tweet_mode" to "extended". Since this affects rate limit and the average tweet length is around 30 characters, we've decided not to.
2. Some of the tweets may be in different languages. While primarily querying for U.S, not everyone's primary language in the U.S is English; and consequently, they speak, read, interact, etc. in their most comfortabel dialect. We believe this actually won't be a problem due to the small number of these kinds of observations.

1.5 Credit Listing

Lipsa Jena * Code for scraping current data (2021) * Collected tweets relating to vaccines (Eg. keywords = vaccines, covid vaccine, moderna, pfizer, etc.)

Ji Kang * Collected past twitter using the IEEE and Kaggle tweet-id datasets * Ran data through LIWC to generate LIWC metrics.

Yuanfeng Li * Code for scraping current data (2021) * Collected tweets relating to masks (Eg. keywords = wearmask, wearamask, masks4all, n95, respirator, etc.)

Yu Ling * Code for scraping current data (2021) * Collected tweets relating to preventative measures (Eg. keywords = social distancing, self isolation, quarantine, socialdistancing now, etc.)

Note: For the submission, we've included the following files: Files prepended by LIWC2015 denote files that have already been processed and enriched with the LIWC 2015 dictionary model metrics. 1. files in /sample_data folder are sized down versions of the tweet IDs both Kaggle and IEEE gives. Named accordingly. 2. files in /scraped_Data folder contain samples of collected tweets from 2020 using tweet id as well as recent tweets collected specific keywords. march_31_2020 are tweets collected using tweet id for that day based on all the keywords. 'data_social_distancing' is **recent** (Past week) data collected for keywords related to social distancing. Lastly, 'masks' is **recent** data collected for keywords relating to masks.