

Report- Analytix

TEAM CONVINIA

TEAM MEMBER: VIJAY GARG, SALONI BANSAL



***INDEX***

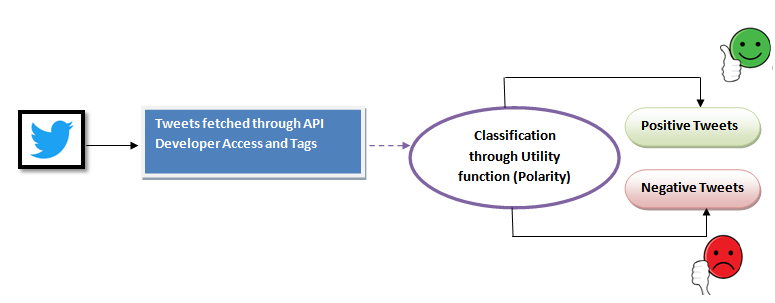
1. Problem statement summary
2. Sentiment Analysis
   1. Data source
   2. Data extraction and pre-processing (R and Python)
   3. Sentiment Analysis (R and Python)
   4. Output and its Analysis
3. Influencer analysis
   1. Data source
   2. Data extraction and pre-processing
   3. Analysis in Python
4. Recent trend
   1. Data extraction
   2. Trends on world level
5. Conclusion

**Sentiment Analysis in Python:**

**Analysis Approach:**

To study public sentiments, Twitter is choosed as a target field. As one of the world’s biggest social network platforms, Twitter hosts abundant user-generated posts, which closely reflects the public’s reaction toward Covid-19 pandemic with low latency.

Using tools like Textblob and Tweepy in Python, we were able to extract and quantify the public sentiments over time.



**1. Data Source:**

* 20,000 tweets with hash tag and keyword of **corona, covid-19, COVID and corona virus** were extracted for analysis.
* The data extracted is in the form of tweets, userid, date , time etc.
* TAGS – Twitter Archiving Google Sheet (<https://tags.hawksey.info/get-tags/>)

|  |  |
| --- | --- |
| Parameters | Tweets |
| Source | API developer Access , TAGS |
| Keyword | Corona, covid-19, COVID , coronavirus |
| Language | All |
| Size | 20,000+ |
| Granularity | Daily |
| Duration | 7 days |

**2. Data Extraction and Pre-Processing**

* Twitter API account allows extracting the tweets. Create one.
* Then, the data is extracted directly from twitter server through developer API access in the programming environment.

Installing packages

pip install tweepy

pip install textblob

Importing libraries

import re

import tweepy

from tweepy import OAuthHandler

from textblob import TextBlob

Authorization through API Access

class TwitterClient(object):

def \_\_init\_\_(self):

consumer\_key = 'AgnlboLoCE8D8mGD306NV52HP'

consumer\_secret = 'zTGF55AeviuW7qiqHD8RWdNXkekknhKiJXTtnlfpLlRtX9qF2O'

access\_token = '1266975421851107328-1Nse3PejLEb6k4b2CA7qmtz1V2XH83'

access\_token\_secret = 'Mx9kOr9MSHVyq239U18PmDeTA8P58rziIAfmVWlawMEkJ'

**3. Data cleaning and Pre-Processing**

Data Extraction: Creating object of Twitter Client Class

api = TwitterClient()

# calling function to get tweets

tweet = api.get\_tweets(query = 'CORONA' or 'COVID' or 'coronavirus' or 'COVID-19', count = 20000)

Data cleaning: Utility function to clean tweet text by removing links, special characters using simple regex statements.

def clean\_tweet(self, tweet):

return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) |(\w+:\/\/\S+)", " ", tweet).split())

Main function to fetch tweets and parse them.

def get\_tweets(self, query, count = 10):

# empty list to store parsed tweets

tweets = []

try:

# call twitter api to fetch tweets

fetched\_tweets = self.api.search(q = query, count = count)

# parsing tweets one by one

for tweet in fetched\_tweets:

# empty dictionary to store required params of a tweet

parsed\_tweet = {}

# saving text of tweet

parsed\_tweet['text'] = tweet.text

# saving sentiment of tweet

parsed\_tweet['sentiment'] = self.get\_tweet\_sentiment(tweet.text)

# appending parsed tweet to tweets list

if tweet.retweet\_count > 0:

# if tweet has retweets, ensure that it is appended only once

if parsed\_tweet not in tweets:

tweets.append(parsed\_tweet)

else:

tweets.append(parsed\_tweet)

# return parsed tweets

return tweets

except tweepy.TweepError as e:

# print error (if any)

print("Error : " + str(e))

Classification (Polarity): Utility function to classify sentiment of passed tweet using text blob’s sentiment method.

**Polarity:** It helps in making quantitative judgements about the feelings of text. I categorize words from tweets into positive and negative type and give them a score for analysis. The polarity scoring tells us if a chunk of text together is negative, neutral or positive.

If the score is close or equal to 1, the body of text is positive or vice-versa.

def get\_tweet\_sentiment(self, tweet):

# create TextBlob object of passed tweet text

analysis = TextBlob(self.clean\_tweet(tweet))

# set sentiment

if analysis.sentiment.polarity > 0:

return 'positive'

elif analysis.sentiment.polarity == 0:

return 'neutral'

else:

return 'negative'

**4. Sentiment Analysis**

Picking positive and negative tweets and their percentages

# picking positive tweets from tweets

ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']

# percentage of positive tweets

print("Positive tweets percentage: {} %".format(100\*len(ptweets)/len(tweets)))

# picking negative tweets from tweets

ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']

# percentage of negative tweets

print("Negative tweets percentage: {} %".format(100\*len(ntweets)/len(tweets)))

5 top positive tweets and 5 top negative tweets

# printing first 5 positive tweets

print("\n\nPositive tweets:")

for tweet in ptweets[:5]:

print(tweet['text'])

# printing first 5 negative tweets

print("\n\nNegative tweets:")

for tweet in ntweets[:5]:

print(tweet['text'])

**Output obtained and it’s Analysis:**

**In python:**

|  |  |
| --- | --- |
| **Nature of Tweets** | **Percentage** |
| Positive Tweets | 21.95% |
| Negative Tweets | 20.73% |

The percentage of positive and negative tweets is 21% and 20% respectively, showing a slight higher frequency of positive tweets over negative tweets in context of Covid-19.

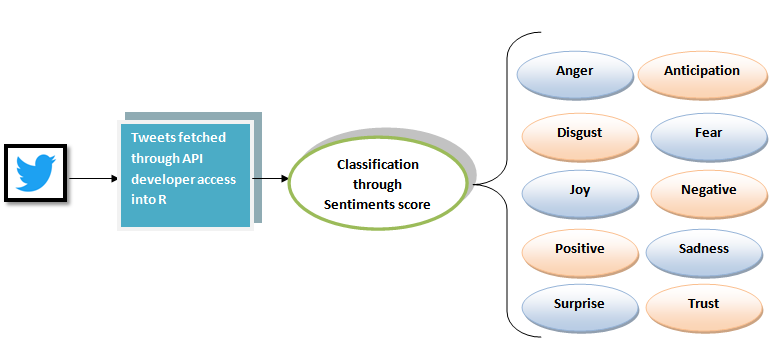
**The Top 5 Positive and Negative tweets are:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Positive tweets:** | **Negative tweets:** |
| **1** | 1.RT @sardesairajdeep: Lovely Sunday read: Bombay, then and now. Long before Corona there was plague: do read.. <https://t.co/9VjLohZDV8> | 1. RT @ishathepoet: Y’all couldn’t impose a curfew, no REAL STRICT stay at home order for Corona but y’all was SPEEDY with this eight pm curfe… |
| **2** | 2. Ayo seseruan  Refreshing Otak  Biar tetap waras meskipun di tengah wabah corona.  TEST IQ ANDA  Klik tautan ini :… <https://t.co/ylNJRQ3bAd> | 2. ना चमगादड़ ना चीन...  corona ले कर आया Alien, How Mean !!  "TV में तो ऐसा ही बता रहे थे"  #garvitguitarcomedy  #GGC… <https://t.co/ZCd4DrPjb5> |
| **3** | 3. RT @the\_only\_ayo: The facts that Corona isn’t the greatest thing we’re battling right now is really scary and sad  RT @YRDeshmukh: The bottom line is: we can not escape from the Corona Virus. Those 80% who will be asymptomatic or with mild symptoms will…  Emergency in Los Angeles, not due to Corona, but Looting, Fire,Confrontation with Police Something never heard of. US IN CHAOS!!!  RT @Shreyaj68170901: Students are the best revenue source, because of their examination fees, the MP government wants to take the exam's.…  @SamKhan999 Exactly. They've already reduced brahmins to 10%  Jihadi mission successful.  But Modi is such a pain in… <https://t.co/qjNhBzsJoX> | 3. RT @Sanjeev80982748: The government closed the colleges when there were only 1-2 patients of covid-19 in the state but they are reopening i…  RT @JerseyEric29: Second wave of Corona, mass unemployment &amp; no checks... |
| **4** | 4. @tjeckholt2015 @BreezerMe @pantomath\_\_ @realDonaldTrump Ok wonderful to hear that. But a majority of them were comp… <https://t.co/IyvTq5AMlP> | 4. Scary hours coming this summer  They already getting fucked over because of the corona. Now looters making it even worse. Hell is hot.  RT @KeyGLOCK: I don’t hear no corona shit now..  RT @latimeralder: Does anyone still think Corona is going to threaten the futire of our civilisation? |
| **5** | 5. @kumarmanish9 Except health workers..rest good works r just on d paper. And they r controlling numbers not Corona.  Students are the best source of revenue for Universities. Now RGPV is conducting exams to collect exam fees by risk… <https://t.co/NqYnl380Ni> | 5. Or is it the hysterical, suicidal ov…  RT @VillageParrot: At this point, Corona is tired of what’s going on in the world.  @sahouraxo Corona be like 😂are you serious I am still alive and killing million people in the USA and you want I fuck the whole USA  RT @yadavtejashwi: At a time of this humanitarian crisis, the governance in Bihar and Corona virus share a strange similarity. They both ar… |

**Sentiment Analysis in R:**

**Analysis Approach:**

Using Libraries like twitteR, we were able to extract and convert the public sentiments into scores over time.



**1. Data Source:**

* 20,000 tweets with hash tag and keyword of **corona, covid-19, COVID and corona virus** were extracted for analysis.

|  |  |
| --- | --- |
| Parameters | Tweets |
| Source | API developer Access |
| Keyword | Corona, covid-19, COVID , coronavirus |
| Language | All |
| Size | 20,000+ |
| Granularity | Daily |
| Duration | Latest at the time of extraction |

**2. Data Extraction**

* Twitter API account allows extracting the tweets. Create one.
* Then, the data is extracted directly from twitter server through developer API access in the programming environment in R.

Importing Libraries:

library('devtools')

library('twitteR')

library('ROAuth')

library('plyr')

library('dplyr')

library('stringr')

library('ggplot2')

library('httr')

library('wordcloud')

library('tm')

library('RCurl')

library('syuzhet')

Data extraction through API Access

consumer\_key = 'AgnlboLoCE8D8mGD306NV52HP'

consumer\_secret = 'zTGF55AeviuW7qiqHD8RWdNXkekknhKiJXTtnlfpLlRtX9qF2O'

access\_token = '1266975421851107328-1Nse3PejLEb6k4b2CA7qmtz1V2XH83'

access\_token\_secret = Mx9kOr9MSHVyq239U18PmDeTA8P58rziIAfmVWlawMEkJ'

setup\_twitter\_oauth(consumerKey, consumerSecret, accesstoken, accesssecret)

some\_tweets = searchTwitter("corona+ covid-19+ COVID+ coronavirus", n=2000, lang= "en")

**3**. **Data cleaning and Pre-Processing**

Data cleaning: Replacement function to clean tweet text by removing links, special characters.

some\_txt = sapply(some\_tweets, function(x) x$getText())

some\_txt1 = gsub("(RT|via)((?:\\b\\W\*@\\w+)+)","",some\_txt)

some\_txt2 = gsub("http[^[:blank:]]+", "", some\_txt1)

some\_txt3 = gsub("@\\w+", "", some\_txt2)

some\_txt5 = gsub("[^[:alnum:]]", " ", some\_txt3)

**4. Analysis**

Evaluating sentiment score and emotion: Use of get\_nrc\_sentiment to generate score for each tweet text that are extracted and pre-processed.

mysentiment <- get\_nrc\_sentiment(some\_txt5)

SentimentScores <- data.frame(colSums(mysentiment[,]))

names(SentimentScores) <- "Score"

SentimentScores <- cbind("sentiment" = rownames(SentimentScores), SentimentScores)

rownames(SentimentScores) <- NULL

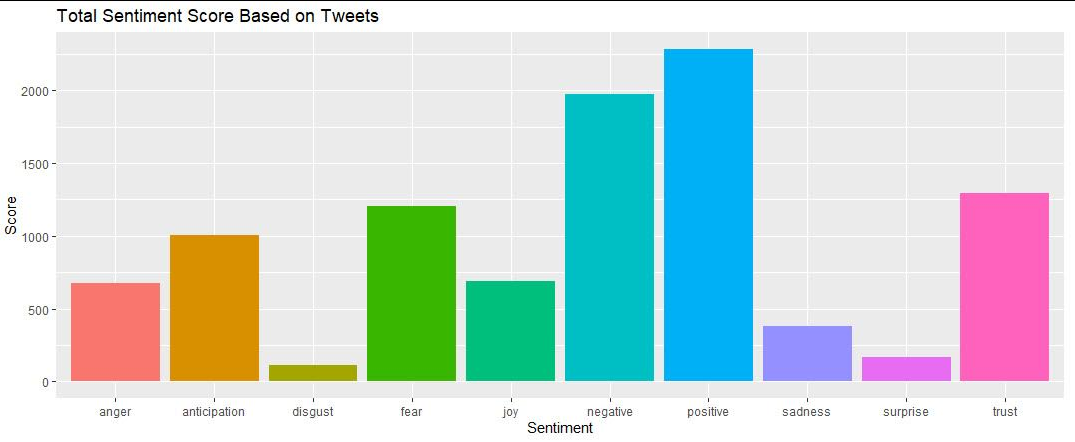
Graphical Representation:

ggplot(data = SentimentScores, aes(x = sentiment, y = Score)) +

geom\_bar(aes(fill = sentiment), stat = "identity") +

theme(legend.position = "none") +

xlab("Sentiment") + ylab("Score") + ggtitle("Total Sentiment Score Based on Tweets")

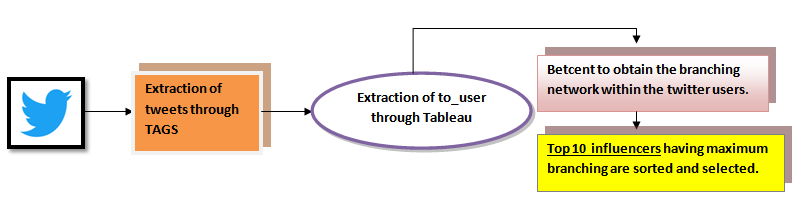


We analysed the sentiments of COVID-19-related tweets all around. The overall trend shows that the public has been more positive during this pandemic time although there is lot of fear and negativity on the social media regarding corona virus.

**2. Influencer Analysis:**

Influencer Analysis helps in understanding Social Networking Potential (SNP) of individual Twitter users. It takes into account the number of followers, of individual interactors, retweets, mentions and the total amount of tweets.

**Analysis Process:** Through TAGS, 50,000+ tweets are fetched in Archieve file. Theto\_user are extracted from tweet text attribute containing @object through the use of Tableau. Then the Betcent is used to obtain the branching network within “to\_user” and “from\_user”. Depending upon the maximum branching shared, the users are sorted and top 10 users are selected as top 10 influencer.



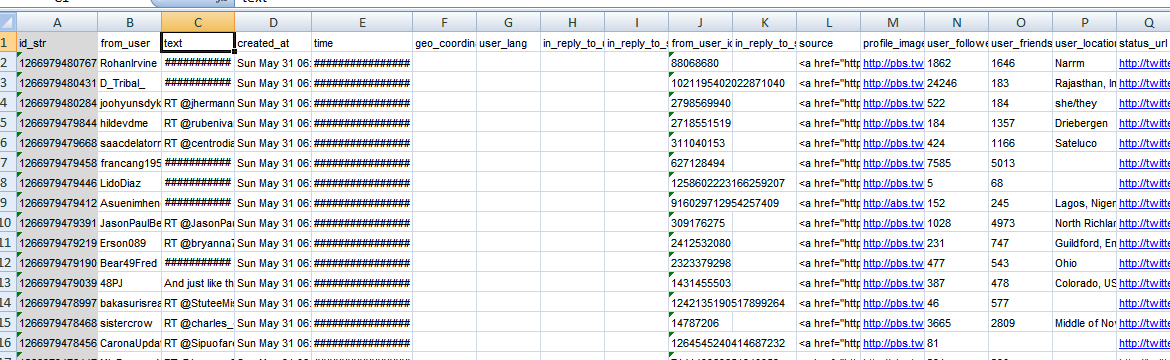
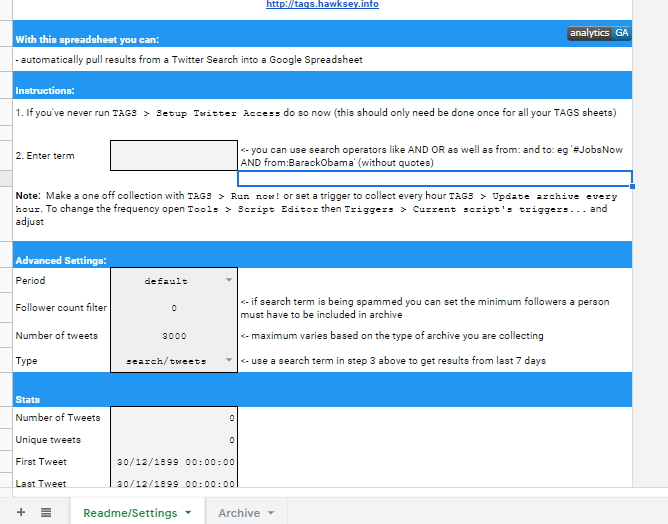
Data Source, Extraction and Pre-Processing:

* Tweets are extracted through TAGS (<https://tags.hawksey.info/get-tags/>).
* By inputting the keywords: corona, covid-19, COVID and coronavirus, a copy of TAGS is created.

|  |  |
| --- | --- |
| Keywords | Corona, covid-19,coronavirus,COVID |
| Number of tweets | 50,000 |
| Type of tweets | Search/tweets (result from last 7 days) |

* After running Tags, you need to update archive every hour to start a continuous collection of data. These fields are filled as per the data requirements
* A Google sheet in the corresponding tab (named Archieve) generates consisting desired outputs.
* The Google Sheet contain attributes like id\_str , from\_user,text, created\_at ,time, geo\_cordinates,from\_user\_id\_str,source, profile, image\_url, user\_followers\_count, user\_friends\_count, user\_location, status\_url. This database obtained contains complete information about the users associated with the tweets and their complete profile containing status, friends count, image etc.

**Input(TAGS) Output(Google Sheet)**



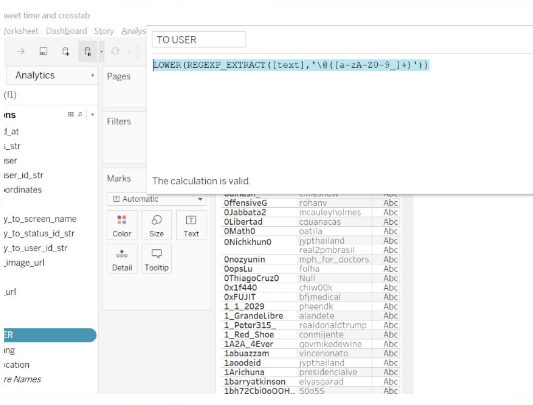
* Pre processing -Extraction of to\_user from the Sheet generated through Tableau (To be done) The Google sheet generated after fetching tweets contains information about “from\_user”. Hence, in order to get the details of “to\_user”, @text is extracted from tweet text attribute. This is operated on Twitter Platform.

Steps followed for the extraction of “to\_user” from tweet text is as follow:

1. The Google Sheet is imported in Tableau platform.
2. From tweet text, the text associated with @ is extracted by using the following code:

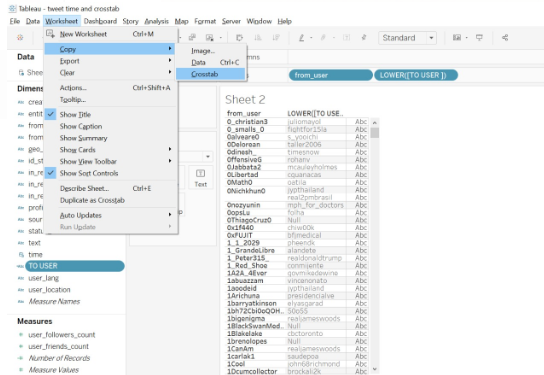
TO\_user =LOWER(REGEXP\_EXTRACT([text],'\@([a-zA-Z0-9\_]+)'))

Extraction of text associated with @ from tweet text



3. After the extraction is done, Sheet 2 is generated containing to\_user and from\_user.

The steps followed are shown below:



* A new dataset- Tweet txt file is created, providing information of “to\_user” and “from\_user”.

**Influencer Analysis:**

Installing package

pip install networkx

Importing Library

import networkx as nx

Loading File (containing “to\_user” and “from\_user”)

INF = nx.read\_edgelist("Tweet.txt",create\_using = nx.Graph(),nodetype=str)

Using betcent to create branching (networking) within the users

betcent = nx.betweenness\_centrality(INF)

Sorting and selecting the top 10 influencer

sorted(betcent, key=betcent.get, reverse= True)[:10]

**Output and Analysis:**

The top 10 influencer obtained are:

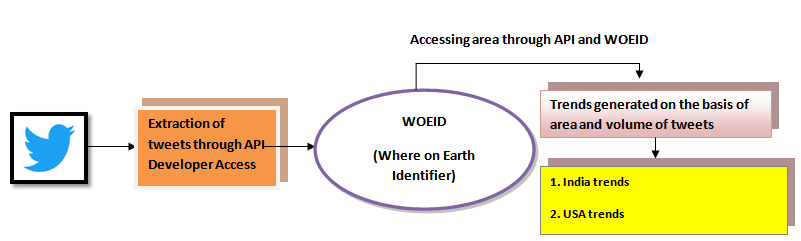
|  |  |
| --- | --- |
| 1. realdonaldtrump | 6. sherrisantosmsn |
| 2. cnn | 7. erinisaway |
| 3. realjameswoods | 8. barackobama |
| 4. NattuVarthakal | 9. alvarorivas8 |
| 5. theprincedylan | 10. AlongiTinna |

The Donald trump has the most influential content of covid-19 over twitter followed up by cnn and realjameswood. People are also influenced by Nattu Varthakal, theprincedylan, barackobama, alvarorovas8. This reflects that their content and tweets regarding Covid-19 are most interactive in nature followed up by retweets, follower ratio and interactor ratio. Depending upon the maximum number of branches and their networking power, these top 10 influencers account can be used as promoter in specific area (say, Covid) of industry.

**3. Trend Analysis**

**Analysis Process:** The approach used for understanding trend analysis of tweets is WOEID (Where on earth identifier). The area is accessed through API and WOEID, which in turn helps in accessing data country-wise and analyze them on the basis of country and volume of tweets.

In the study below, the identification of trends w.r.t India and world is done and their volume of tweets per trending tweet is identified.



**Data source and Extraction:**

Installing packages

pip install twitter

Importing libraries

import twitter

Data extraction through API access

CONSUMER\_KEY="AgnlboLoCE8D8mGD306NV52HP"

CONSUMER\_SECRET="zTGF55AeviuW7qiqHD8RWdNXkekknhKiJXTtnlfpLlRtX9qF2O"

OAUTH\_TOKEN="1266975421851107328-juQcFThlBlyra6qd3A4AMqqTrXDibH"

OAUTH\_TOKEN\_SECRET="LhRKi4Cna87r4ttrzwCEjhWB1sFLMKGA2th9UwA8gwnHo"

#Replace with your accesssecret

Defining twitter\_API

auth = twitter.oauth.OAuth(OAUTH\_TOKEN, OAUTH\_TOKEN\_SECRET,

CONSUMER\_KEY, CONSUMER\_SECRET)

twitter\_api = twitter.Twitter(auth=auth)

print(twitter\_api)

**Trend Analysis:**

**WOEID (Where on Earth Identifier Place**) is a unique 32-bit reference identifier, which is used with API accessing Twitter trends. This is used to identify countries and the trends associated. Thus, we choosed India and world for analyzing their giving WOE\_ID for both.

Defining Woeid\_ID

WORLD\_WOE\_ID = 1

INDIA\_WOE\_ID = 23424848

Identification of India trends w.r.t volume of tweets: The Woeid is accessed using api.trends.place () where the trends on the basis of volume of tweets are analyzed.

india\_trends = twitter\_api.trends.place(\_id=INDIA\_WOE\_ID)

for trending in india\_trends[0]['trends']:

print(trending['name'], '-----' , trending['tweet\_volume'])

Identification of World trends w.r.t volume of tweets

world\_trends = twitter\_api.trends.place(\_id=WORLD\_WOE\_ID)

for trending in world\_trends[0]['trends']:

print(trending['name'], '-----' , trending['tweet\_volume'])

**Output and Analysis:**

The trends w.r.t to the volume of tweets are generated for India and U.S.A

|  |  |
| --- | --- |
| **India\_Trends** | **Volume of tweets** |
| BLACK\_LIVES\_MATTERS | 494996 |
| Antifa | 2523457 |
| Jadon Sancho | 76306 |
| 1 Billion | 163436 |
| baghdad | 87488 |

|  |  |
| --- | --- |
| **World\_Trends** | **Volume of tweets** |
| SarkaruVaariPaata | 1447379 |
| GeorgeFloydProtests | 290568 |
| AllHeartsWithSidharth | 507276 |
| Anonymous | 1148428 |
| SindhRejectsBogusDomiciles | 19216 |

***Overall conclusion:***

* Digging into the multi-dimensional sentiment analysis, we found that the sentiment “Assertive” went up, and “Fearful” went down with the Covid scenario. Besides, the Sentiment Density indicates that the public turned out to be less loaded with emotions. At last, the topics behind the sentiments unfolded more details.
* To fight the coronavirus not only needs the guidance from the government but also a positive attitude from the public. Our analysis provides a potential approach to reveal the public’s sentiment status and help institutions respond timely to it.
* The government can use these to 10 influencers for maximum outreach within the network, in case of any step and concern related to Covid-19. This way, people will respond and interact in the most responsive manner.