Network-based Social Media Analytics

Analysing Twitter data in the context of the COVID-19 outbreak

Leonore Papaloizos
2264897v

15th March, 2020
Code: **github.com/leonore/twitter-parser**
Sample data: **github.com/leonore/twitter-parser/data**

Web Science (Level H) at the University of Glasgow

1) Introduction

For the purpose of analysing Twitter data, software with the following components was developed:

- A simple, streaming crawler that uses the Twitter Streaming API[1] (Section 2a)
- A hybrid crawler that combines the Twitter Streaming API and the Twitter REST API methods[2] (Section 2b)
- Functions to insert Twitter data into a MongoDB collection
- A suite of analytics functions that take the data stored in a MongoDB to analyse Twitter content
    - Extracting topics from Tweets with K-means and TF-IDF vectorisation (Section 3a)
    - Extracting sentiment from Tweets with TextBlob's sentiment analysis tool (Section 3b)
    - Building networks of user and hashtags from different types of Tweets (Section 4)
    - Analysing the networks built in terms of size and connections (Section 5)

The following table describes the process of collecting data:

Crawler	Date	Duration	# of tweets
Streaming	10-03-2020 12:45	60 minutes	52,334
Streaming + filter	10-03-2020 13:45	60 minutes	138,019
Hybrid	07-03-2020 14:40	60 minutes	541,954

The hybrid crawler ran in the early afternoon to attempt and get a global view and content from different time zones.

2) Data Crawl

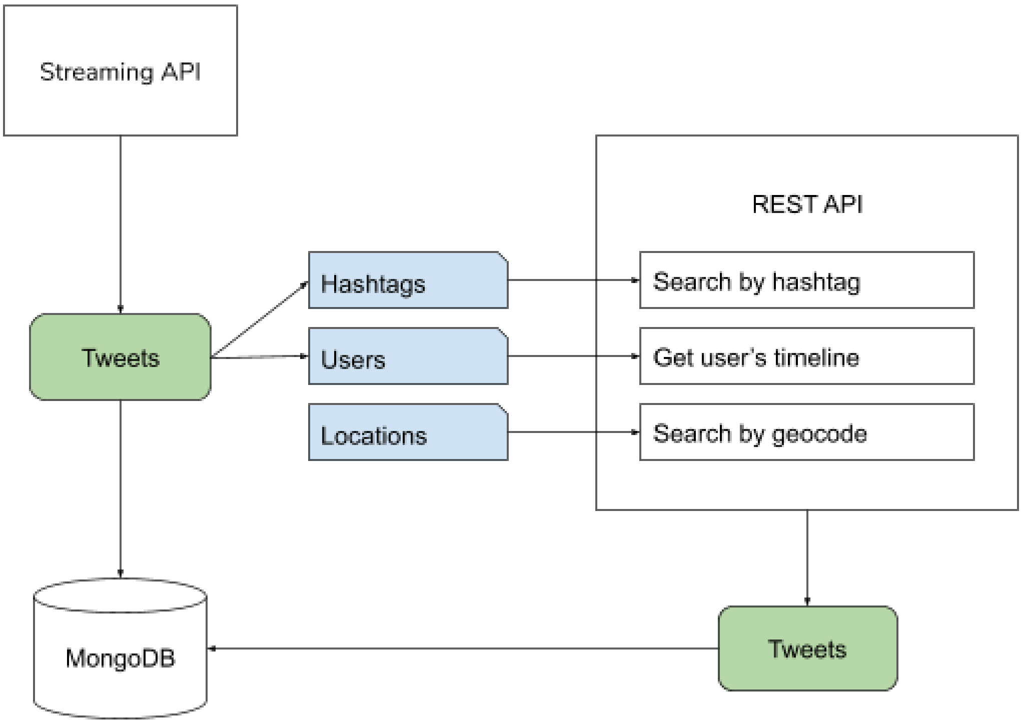
a) Collecting 1% of Twitter data by using the Streaming API

First, Twitter's Streaming API was used in a very straightforward manner. Only English language tweets were streamed, but no filtering keywords were provided. The streamed JSON's date attribute was modified for better handling by MongoDB. This initial simple approach gives a benchmark of performance for later tweaks.

The Streaming API approach was also used with filtering keywords. The keywords used are the same ones that were used in the hybrid crawler to be described in Section 2b. This was to give a better comparison of Tweet counts and usage of Twitter's APIs

[1] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/
[2] https://developer.twitter.com/en/docs/api-reference-index

b) Collecting Twitter data through hybrid Streaming & REST API architecture



The figure above gives a system overview of the hybrid crawler's structure. The idea was to parse entities collected from the Streaming API tweets in order to perform adjacent searches with the REST API. For every tweet that came in, the JSON was parsed for its hashtags and user mentions. Each of these were added to their own queue. Separate threads were running to retrieve tweets from user timelines and perform search by hashtag. Moreover, there was a location thread which randomly picked a location from a pre-set list and performed a tweet search by geolocalisation.

Requesting a user's timeline had separate limit rates as search, hence it was added as a probe to get an additional number of tweets. In order to get tweets from only from users that seem that they might not be spam, a user's tweets were only processed if they had more than 10 followers. Moreover, if a user had more than 15,000 thousand followers, their friend network was added to the user queue to process, as they seem to have importance to have this many followers, so they might follow other power users too.

The hashtags were added in a search query because of the reasoning that it might be part of a larger trend. The locations used for search were not parsed from Tweets, but were chosen based on locations that might have interesting activity, e.g. important governmental organisations being headquartered there.

As this crawler was being developed at a time when the coronavirus outbreak was developing and spreading outside of China, it was decided to add coronavirus-related keywords to the Streaming API's filter parameter. The topic of the virus seemed to be appearing more and more often in Tweets, and choosing to track this would allow us to analyse the Tweets in regards to an event that is affecting us. Following this, the pre-set list of locations was changed to Washington DC, London, Geneva (WHO headquarters), and Wuhan, in an attempt to get the latest tweets from governmental organisations as well as the latest news from the topic.

3) Tweet Grouping

a) Topic extraction

A first method was attempted using K-means with a TF-IDF vectoriser. Although the quality of clusters might not be the best, this can be applied to an unfiltered Twitter dataset where a specific topic was not tracked in order to obtain groupings of Tweets. This method was applied to our 1% streamed dataset. K-means performance was analysed for 0 to 100 clusters with a step of size 2. The sum of the square distances (SSE) between each centroid and members of the clusters was lowest for K=76. K-means was then run on the TF-IDF vector data with K=76. 76 clusters were obtained and the top 10 words for each of the clusters were retrieved. Some notable examples are reported below:

Cluster	Top 10 words	Estimated topic
Cluster 1	cancer, moves, refinement, forte, effect, le, aquarius, aries, deny, today	Astrology
Cluster 7	heart, class, pissed, showed, cat, ok, like, face, joy, tears	Funny tweets
Cluster 14	mean, make, back, think, hi, trump, polls, lose, people, say	Politics
Cluster 19	fans, shows, shut, truly, completely, sold, republic, concerts, czech, inform	Music
Cluster 25	eurocentric, transwomen, black, white, happy, beauty, men, day, international, women	International Women's Week[3]
Cluster 42	turning, crap, pigeons, virus, spread, italy, people, outbreak, covid, coronavirus	Coronavirus

Some other clusters were less informative, e.g. a cluster's top words were "focused, foh, fold, folded, folder, focusing, zzz, works, art, artists".

b) Sentiment analysis

For the study of COVID-19 data, the tweets were instead grouped based on sentiment analysis. As the topic of conversation is already provided here, sentiment analysis was an alternative path to take to obtain groupings of tweets. The analysis of the distribution of sentiments could also yield interesting results, as COVID-19 related tweets might contain more negative than neutral or positive tweets compared to regular tweets.

[3] Tweets were retrieved during International Women's Week

For sentiment analysis, the Python TextBlob module was used[4]. A first attempt was made at using Afinn[5] but TextBlob seemed to get the overall sentiment of the tweet better, and also had a subjectivity score if this software were to be extended.

In order to extract sentiment from a tweet, it was first cleaned up and tokenized. The cleanup process was as follows:

1. Turn text into lowercase
2. Remove URLs with regex
3. Transform emojis into text with the Python emoji package[6] to convey further emotion, for example: 😧 → :anguished_face: → anguished
4. All non-ASCII characters and special characters are removed
5. Whitespace is stripped from either sides
6. All Tweets are turned into a list of words with nltk and stop words and removed
7. The words are joined back together in a string

Once the text was cleaned and tokenized, its overall sentiment score was obtained with TextBlob. TextBlob gives a score in the range [-1.0, 1.0]. After trying a few sentences out, it was decided that a score between 0 and 0.2 would correspond to a neutral sentiment, a score below 0 would be a negative sentiment, and a score above 0.2 would be a positive sentiment.

The following results for sentiment grouping were obtained:

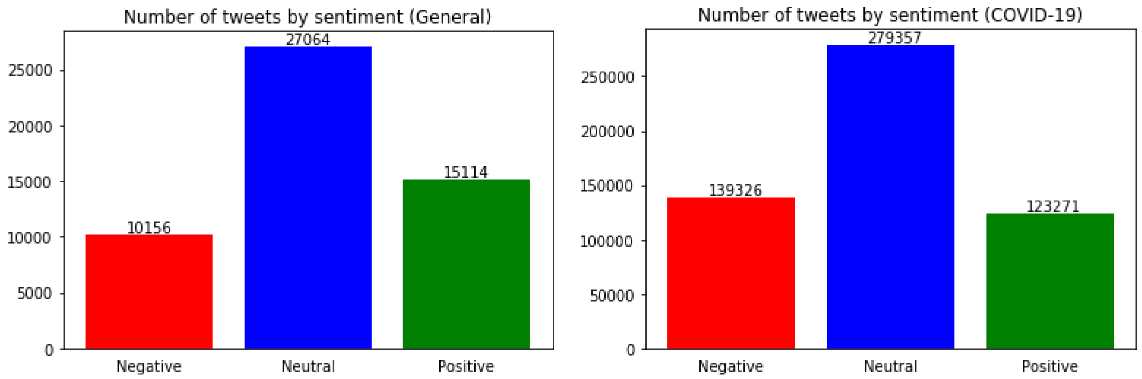


Figure A (left): *Number of tweets by sentiment detected for 1% unfiltered sample data.*
Figure B (right)*: Number of tweets by sentiment detected for COVID-19 related crawled data.*

Figure A shows the sentiment distribution of the regular streamed 1% Twitter data without coronavirus word tracking filters. Figure B shows distribution on the COVID-19 larger dataset. Although the proportion of neutral tweets stays similar, we can see that there are more negative tweets in the COVID-19 Tweets than the general tweets. The overall proportion of sentiment in COVID-19 related tweets is 26% negative, 51% neutral, and 23% positive, compared to 19%, 52%, and 29% in the unfiltered Tweets, respectively.

[4] https://textblob.readthedocs.io
[5] https://pypi.org/project/afinn/
[6] https://pypi.org/project/emoji/

Here are the texts of some example Tweets from each category:

- Negative: “If I show up to an arena and there ain't no fans in there, I ain't playing." LeBron on possibly playing NBA games without any fans in attendance due to coronavirus outbreak.
- Neutral: @nytimes Handwashing, I agree. How about trade in and consumption of wild animals + climate change + long-running cuts to health care systems?
- Positive: did y’all know that there are 96,950 cases of COVID-19 right now? Did y’all know 3,308 people have died from it? BUT did y’all know that 53,981 of these cases have already recovered? Stop letting the media control your brain. Scare tactics are real tactics. Do your research.

c) General statistics

The following table reports the count of tweets, retweets, quotes and replies in the collection. Note that there might be some overlap. For example, a retweet might also be a quote status. However, quote statuses and replies have been differentiated.

Dataset	Total	Retweets	Quotes	Replies	Average Tweet length
Full	541,954	413,716	128,779	33,262	160
Negative	139,326	115,776	35,774	5,609	185
Neutral	279,357	204,810	63,975	19,854	145
Positive	123,271	93,130	29,031	7,799	168

From this we can see that most activity Twitter seems to be getting comes from retweets. Moreover, there seems to be a lot more exchange of positive replies than negative replies.

d) Username, hashtag, entity extraction

The next step was to look at recurrent usernames, hashtags and concepts that might appear often in the data overall, as well as in groups. This section describes the methods for extracting important entities as well as reports findings.

In order to retrieve the most important usernames from the group of Tweets, both retweet counts and mention counts were considered.

- For each tweet, user entities were retrieved from the Tweet JSON. The correct entity field depends on whether or not the Tweet had been truncated. Once correctly retrieved, each hashtag and user mention is tallied up in a dictionary. Hashtags were put into lowercase before being saved. If either is mentioned more than once its tally is increased.

- Retweets are a special case. Each Tweet classified as a “retweeted_status” was retrieved and the original tweet’s retweet count was tallied up next to the creator of the tweet. Note that retweet counts coming from the retweeter could not be tallied up as the Tweet coming in were often just posted, and it is static content.

For entity extraction, each Tweet’s text body was retrieved, tokenized with the method described in Section 3, and appended to a corpus of text. Once all tweets were parsed, the corpus was analysed by nltk’s Frequency Distribution tool, which counts how many times each word appears.

The following table reports the top 10 most mentioned elements obtained in each list with this method. For the most frequent words, general words not bearing much meaning such as “know”, “like”, were not considered.

Dataset	Top 10 retweeted users	Top 10 mentioned users	Top 10 hashtags	Top 10 words (concepts)
Ungrouped	BTS_twt, LizSpecht, tedlieu, BarackObama, realDonaldTrump, Harry_Styles, _SJPeace_, FactTank, Louis_Tomlinson , AirlinesDotOrg	realDonaldTrump, POTUS, tedlieu, LizSpecht, elonmusk, DrDenaGrayson, ayoair, Whitehouse, OnlyAtarii, cschans61	covid19, coronavirus, covid_19, covid-19, coronavirusoutbreak, china, covid, iran, coronavirususa, breaking	“coronavirus”, “covid”, “people”, “face”, “trump”, “cases”, “need”, “health”, “hands”, “test”
Negative	LizSpecht, tedlieu, WhySharksMatter, _SJPeace_, realDonaldTrump, BTS_twt, elonmusk, kobebryant, charliekirk11, BrianKarem	realDonaldTrump, elonmusk, tedlieu, POTUS, LizSpecht, ayoair, _alicejay, watchidid, SJimons, BernieSanders	covid19, coronavirus, covid_19, economicterrorism, medicalterrorism, covid-19, china, coronavirusoutbreak, iran, coronavirususa	“coronavirus”, “covid”, “people”, “trump”, “due”, “risk”, “panic”, “hands”, “stop”, “sick”
Neutral	BTS_twt, BarackObama, FactTank, Harry_Styles, realDonaldTrump, carterjwm, DrDenaGrayso	realDonaldTrump, nimmserk, JHUSystems, FactTank, EmericanJohsnon, DrDenaGrayson, SarabiElly,	covid19, coronavirus, covid-19, coronavirusoutbreak, china, covid_19, covid, iran, coronavirususa, bbb20	“coronavirus”, “covid”, “trump”, “people”, “cases”, “health”, “test”, “public”,

	n, bts_bighit, _SJPeace_, NiallOfficial	goodmiad, SANJAYGHARMALKA, metalmaugly		“years”, “global”, “time”
Positive	AirlinesDotOrg, Harry_Styles, realDonaldTrump, Louis_Tomlinson, BarackObama, MrBeastYT, BTS_twt, JKCorden, NiallOfficial, youngttsunami	realDonaldTrump, OnlyAtarii, VP, Potus, WhiteHouse, Mike_Pence, AirlinesDotOrg, DrDenaGrayson, brianbeutler, anikauwu	covid19, coronavirus, covid-19, coronavirusoutbreak, china, covid, coronavirusinsa, covid_19, coronaviruschallenge, coronavirususa	“coronavirus”, “covid”, “people”, “joy”, “tears”, “cases”, “trump”, “good”, “right”, “light”

4) Capturing and Organising User and Hashtag Information

This section describes how user and hashtag information was captured. Subsequent analysis is done in Section 5.

a) Capturing User information

The next step was to look at how users were interacting across tweets. The process was similar to the one used in Section 3b for tallying up entity counts. The idea was to create three graphs based on retweets, user quotes and replies, and global user mentions in Tweets. The initial structure for the graphs was a dictionary, but they were then also built into a graph data structure with the help of the networkx Python package[7]. The interactions between users are directed on the basis of who is mentioning whom. For example, if user A is retweeting user B, that is a directed edge from user A to user B. User B does not necessarily have to acknowledge user A back. We can also count the number of times user A mentions user B in order to see where links are stronger and interaction is recurrent.

- In the case of retweets, an edge is created from the retweeter to the author of the original tweet. If the retweeter were to retweet another one of that author’s tweet, the weight of the edge between the two would be increased.
- Quotes and replies were tallied up in the same graph, in the same way as retweets. An edge is created from the quoter/replier to the quoted/replied to. The tally is increased for subsequent mentions.
- Finally, overall mentions in the body of the tweet are retrieved from the JSON’s entity file. Edges are created from the Tweeter to each user mentioned. This creates overlap with quotes and replies but gives an overview of user interaction.

[7] http://networkx.github.io

b) Capturing Hashtag information

The process for gathering hashtags co-occurrence information was different: hashtags do not 'mention' each other as users do, so sorted list of hashtags were initially inserted in nested lists. They were then added to an undirected graph. Each unique hashtag in the list would be a node, and if it was in the same list as another hashtag, an edge would be created between them.

c) Network output

The interactions were first tallied up in a dictionary, and then built into a graph with the help of the networkx package. Networkx is however quite slow for computations, so although it was used to gather some general metrics, some other defined methods are used for network analysis in terms of connections such as ties and triads in Section 5.

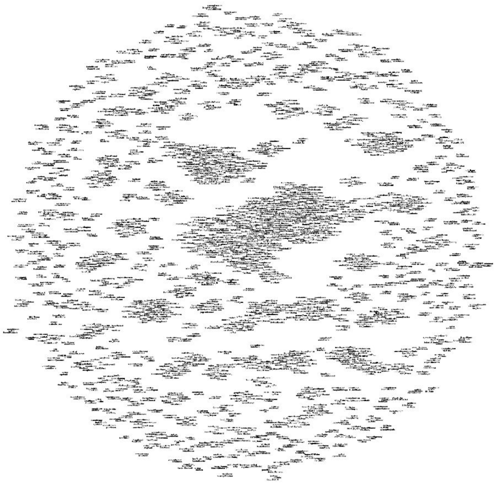
The following table provides some graph information obtained through Networkx.

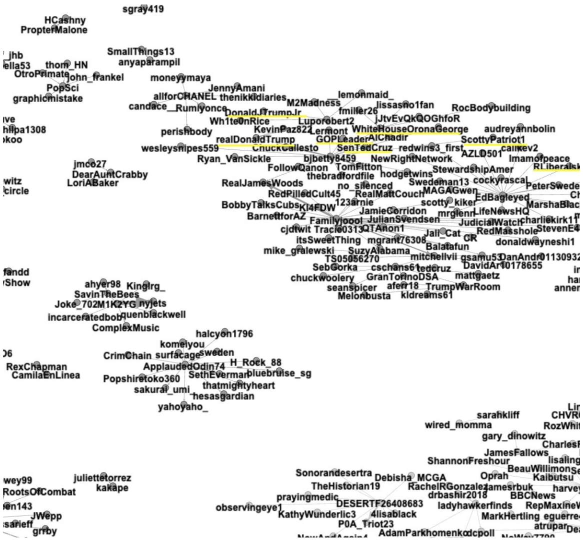
Dataset	Type of network	# subgraphs	# nodes	# edges	Average sub-network size
Ungrouped	Retweet	6,816	224,080	295,851	32 users
	Quote	5,516	73,360	81,112	13 users
	General	3,271	94,674	149,895	28 users
	Hashtag	10,227	35,794	110,196	3 hashtags
Negative	Retweet	2,421	83,832	98,345	34 users
	Quote	2,231	21,919	21,361	9 users
	General	1,225	28,576	40,344	23 users
	Hashtag	3,016	9,442	21,601	3 hashtags
Neutral	Retweet	4,725	122,831	149,487	25 users
	Quote	3,648	42,485	44,677	11 users
	General	2,131	55,164	74,804	25 users
	Hashtag	7,076	23,529	67,047	3 hashtags
Positive	Retweet	3,087	65,924	74,068	21 users
	Quote	2,461	21,562	20,630	8 nodes
	General	1,480	32,721	50,874	22 users
	Hashtag	4,179	13,614	33,854	3 hashtags

The data from the table seems to suggest that retweet networks grow a lot further than quote/reply networks do.

d) Drawn graphs

Out of interest, a random subgraph of the network built from neutral retweets was plotted using Gephi, as networkx's matplotlib drawing tool is too slow for the scale of the networks. This was to illustrate how a graph might look once visualised. The following figures highlight how the network looks at different scales.





As we can see the network is built of larger networks and smaller ones. The small network highlighted here shows a network clustered around similar users (US Politics).

5) Network Analysis

For terminology, a tie is created when user A mentions user B. A loop is created when user B mentions user A back. There are many different types of triads[8], but for easier and reproducible calculations across hashtag and user networks, a triad is defined as an interaction between three nodes A, B, C, regardless of the directions and the number of edges between those nodes.

a) User networks

In the case of users, we can methodically look for ties, loops and triads in the following way:

- For each user A, go through each user B it has mentioned. The tally for links increases for all the user A has mentioned.
- Then, get user B's mentioned list from the dictionary. If user A is in user B, a loop is formed.

[8] http://vlado.fmf.uni-lj.si/pub/networks/doc/triads/triads.pdf

- If the user A has more than 2 mentions the appropriate number of triads is formed, i.e. all (non-ordered, non-repetitive) combinations of 2 in the list. Each triad will be of the form A-(B-C) for each pair of users B, C from the user's mentions.
- If user B has other mentions in its list, transitive links are formed, as well as triads. There are x extra triads for the number of people in B's mentions list, i.e. a new triad for A, B, and each other user in B's mentions.

The following table reports on the number of computed ties, loops, triads and transitive links in the datasets.

Dataset	Type of network	Ties	Loops	Transitive links	Triads
Ungrouped	Retweet	295,581	786	37,980	58,662,157
	Quote	81,112	1,434	23,076	5,464,443
	General	149,895	901	44,064	19,843,152
Negative	Retweet	98,345	197	4,692	5,341,466
	Quote	21,361	422	4,012	324,270
	General	40,344	160	6,703	1,413,241
Neutral	Retweet	149,487	462	17,179	21,675,305
	Quote	44,677	795	14,674	2,511,086
	General	74,804	551	27,144	8,945,505
Positive	Retweet	74,068	201	6,739	5,751,713
	Quote	20,630	417	6,077	613,423
	General	50,874	252	10,472	3,001,062

Positive tweets seem to create longer paths (connections) between users than negative tweets do, considering the groupings contain a similar amount of Tweets.

Both groups and ungrouped data follow the same structure: retweets get the biggest networks, before quotes. A lot more connections are made in ungrouped data as evidently not all communications shared between users will all be of the same sentiment. So if a negative tweet made by user A mentioned user B but user B had not tweeted something negative, less ties, loops, transitive links and triads would be made.

b) Hashtag networks

We can look for ties and triads in our hashtag networks by looking through the obtained nested list structure.

- Lists of size less than 1 can be discarded, as we cannot link that hashtag to any other one.
- Lists of size 2 are ties and potentially triads if any hashtag in that list is contained in another list and connected to other hashtags.
- Lists of size 3 and above contain triads.

Visited list of hashtags are stored in an adjacent data structure so as to not go through the same list of hashtags twice. For each list of hashtags, all combinations (non-ordered, non-repetitive) of ties and triads possible are summed up and added to a tally. Then, for each element in the hashtag list, we look through the hashtags that have already been visited. If a hashtag from the list is found in those previously visited lists, triadic connections are created between those visited elements and the other hashtags in the original list. To illustrate, if we look through the list [1, 2, 3], and 3 is found in another visited list [3, 4, 6], triadic connections are counted for the following groups: [1, 4, 6] and [2, 4, 6].

The following table reports on the number of ties and triads in the hashtag network for each dataset.

Dataset	Ties	Triads
Full	169,099	357,947,986
Negative	29,411	16,122,192
Neutral	99,465	116,952,725
Positive	44,327	25,406,212

The numbers have to be put into perspective with the proportion of each sentiment. There are a lot more links and triads tallied up for hashtags in neutral tweets but there are also twice as many neutral tweets in the dataset. However, we could infer that fact-related tweets (neutral tweets) hold more hashtags as organisations use hashtags as a means of communicating information through channels. Moreover, there is a close share of negative and positive tweets in the dataset, but there is a lot more ties between hashtags in positive tweets.
The full dataset contains more triads as hashtags used across types of tweets are then combined together.

6) Conclusions

This report gave information on possible analysis methods to use for Twitter data analysis. Twitter provides limited access to its data through free APIs. A large amount of data can be gathered through the use of efficient REST probes.

This report focused in part on analysing network statistics of groups of COVID-19 related Tweets in the midst of the outbreak of the disease around the world, but the methods described here can be applied to any set of Tweets. The code to carry out this analysis is available on Github.