Analysis of hashtag co-occurence network in tweets related to Covid-19 during the very early stages of the Covid-19 outbreak

Matt Stirling
2208599
University of Oulu
Oulu, Finland
mstirlin18@student.oulu.fi
https://github.com/MattThePerson/SNA_Project_2024

***Abstract*—Discourse on Twitter can be analysed as a network by looking at co-occurrences of hashtags within the tweets. This approach may yield insights about the nature of discourse surrounding a particular topic. In this report, I analyse a network constructed by adding links between tweets based on co-occurrence of hashtags for tweets related to the Covid-19 pandemic published between December 2019 and February 2020. The analysis showed that this network does not follow a power law distribution, contrary to expectation, and that the network has a high transitivity which evolves over time.**

***Keywords*—social network analysis, Twitter, COVID-19, hashtags**

I. Introduction

This report documents my work on the final project for my Social Network Analysis class. The general idea of the project was to use a dataset of tweets to construct a network based on co-occurrence of hashtags and then analyse aspects of the network as well as how some features evolved over time (see next section for problem description). The tweets would be specifically about Covid-19 and were published in the very early stages of the outbreak.

I was able to isolate the components of the network and do some analysis of shortest paths, and I concluded that the distribution of degree centralities of the nodes did not follow a power-law. The transitivity of the network was found too be very high and to evolve in a manner that was peculiar and may require further thought to explain. The evolution of tweet semantics can be partially understood by appealing to the declaration by the WHO of a global public health emergency in late January 2020. I also suggest an alternative approach to analysing similar types of networks which abstracts the network into its complete subgraphs, which are easy to identify in this case. Due to time constraints, I was unable to complete the last two network analysis tasks concerning structural balance and virus propagation model.

A paper that deals with a similar problem description by Türker and Sulak (2018) explores a multiplayer network with the first layer being made by converting co-occurrences of hashtags into links and the second layer by converting semantic relations into links [1]. This paper found that power-law distribution of degree was evident in the multilayer network. This finding of course differs from the result in this report concerning power-law distribution of node degree.

All the code used to perform the network analysis is available in a public code repository [2].

II. Problem Description

The problem description as I see it was to analyse a graph that was to be constructed from a subset of the TweetsCOV19 dataset [3]. The dataset would be limited to tweets between December 2019 and February 2020, and the graph would be constructed by representing individual tweets as nodes and adding edges between any two nodes where the tweets that they represent share at least one hashtag. The graph would be saved in a manner that allows for it to be loaded into a graphing library (in my case an edge list) and the adjacency matrix of the graph would be created.

Then, the following analysis would be done on the network:

- Identify the three largest connected components and determine their average path lengths.
- Plot the degree centrality of the graph and investigate whether it is power law distributed.
- Plot the evolution of *cumulative* average positive and negative sentiment scores of the tweets over the selected timespan and comment on the discrepancy between the two graphs.
- Analyse the evolution of transitivity of the network by looking at the cumulative graph up to each point in time.
- Analyse the evolution of structural balance of the network by looking at the proportion of balanced and unbalanced triangles.
- Investigate the occurrence of unbalanced triangles as a virus propagation case.

After this analysis, I am supposed to comment on the limitations of the overall reasoning pipeline and suggest literature to support the findings.

III. Dataset Description

The dataset from which the network was constructed was Part 1 of the TweetsCOV19 dataset, which is a semantically annotated corpus of tweets about the COVID-19 pandemic. It contains over 8 million tweets between October 2019 and April 2020 which were selected from the TweetsKB dataset, a much larger corpus of nearly 3 billion tweets spanning over 9 years. The basis for selection from this larger dataset was if the tweet text contained keywords related to the pandemic; a seed list of 268 keywords was used. The text of the tweets

themselves were not made available in the TweetsCOV19 dataset and the user IDs were also encrypted for anonymity.

Each tweet instance contains 12 features related to the tweet.

TABLE I. TWEET INSTANCE FEATURES

#	*Feature*	*Description*
1	Twitter ID	Used internally by Twitter to uniquely identify twitter objects, including tweets, users, direct messages, etc. [4].
2	Username	Encrypted
3	Timestamp	in the format "EEE MMM dd HH:mm:ss Z yyyy"
4	#Followers	Number of followers the user has
5	#Friends	Number of friends the user has
6	#Retweets	Number of times the tweet was retweeted by others
7	#Favourites	Number of times the tweet was favourited by others
8	Entities	
9	Sentiment	Both a positive and negative sentiment score was calculated for each tweet (it is unclear how this was done) and these values are listed as a string represented by a whitespace char
10	Mentions	Usernames (which were unencrypted in this case) that were mentioned in the text of the tweet
11	Hashtags	Hashtags used in the text of the tweet
12	URLs	URLs used in the text of the tweet

Some features – such as sentiment or hashtags – need to be parsed as strings to extract the desired data they contain. The dataset is available for download in both Notation3 and TSV (Tab Separated Value) formats.

IV. GENERAL METHODOLOGY

After limiting the tweets to the desired timeframe (December 2019 – February 2020), the next step was to produce a list of edges from the method outlined in section II. Then, the edge list could be loaded into a python graphing library such as *NetworkX* or *iGraph*, which could be used to conduct the desired network analysis. However, the amount of edges produced made this very challenging as the time- and space-complexity of loading the edges into the library and then analysing the graph in many cases exceeded the time I had and the hardware available to me. To understand why there are so many edges, it is worth considering the nature of the graph that will be created via the described method.

Each hashtag that is associated with more than one tweet will produce a complete graph where the number of edges is:

$$|E(G)| = nC2$$

where n is the number of tweets that use the given hashtag. Each node in this subgraph will therefore have a *minimum* degree centrality of nC2 and the minimum number of complete subgraphs of the whole network will equal the number of hashtags associated with more than one tweet. As a result, our network will be relatively dense, and given the large number of nodes there will also be a large number of edges.

As I already mentioned, I found it challenging to follow the recommended (and perhaps expected) approach of using one of the many python libraries for the network analysis. For instance, simply loading all the edges into an iGraph instance would take over 5 hours on my PC, and the analysis from there could take days. As a result, I have tried to use alternative methods whenever possible, which I will now outline.

Perhaps the easiest point of analysis was the degree centrality, as this information can be extracted from the edge list in a straight-forward manner. (Note that the degree centrality here is the absolute number of edges of each node, as opposed to the normalized values you would get from finding degree centralities using NetworkX). Finding the connected components was also possible without relying on a library by implementing the union find algorithm and then isolating the various components by grouping each nodes based on a shared *parent*.

For some tasks, namely evaluation transitivity, I have modified the way I answered the question. This is because For the analysis of transitivity, the problem description was to load the entire cumulative network up to a given timestamp for 100 equally spaced timestamps. I found that answering the question in this way simply took far too long, and as such I have modified the way I approached it. Instead of using the cumulative network up to each timestamp, I instead looked at a sliding window of *maximum* of 10 time bins.

I would also like to suggest an entirely different approach, one that is enabled by the unique nature of the network we are analysing. Let us construct a different graph which will act as an abstracted version of the full network, where we represent all the unique hashtags as nodes, and add edges between the nodes whenever there is a tweet that shares the hashtag of both nodes. This abstracted graph is of course much smaller, but by analysing this graph we can gain insight or even discover features we want to know about the full graph. This is because each subgraph associated with a hashtag is complete, we know that each node in this subgraph is connected, and also that they are separated by a distance of exactly one. This means that connected components are pretty much a one-to-one correspondence between the two networks. One would simply need to find a list of hashtags in each component, and then find all the nodes that contain at least one hashtag of a given hashtag component. We can also use this approach to estimate the average shortest path distance of the full network, although that takes a bit of mathematical reasoning.

Unfortunately, there were some tasks I was unable to get to, namely finding balanced or unbalanced triangles and investigating the occurrence of unbalanced triangles as a virus propagation case. For the first of these tasks I have suggested a script which should accomplish the task if given enough time and memory.

V. DETAILED METHOGOLOY

As already mentioned, I have conducted all the network analysis in *python* written by myself. The libraries I used include *pandas* for loading the dataset into a data frame, *desk* to load the edge list into a partitioned data frame, *NetworkX* and *iGraph* for some network analysis functionality and

MatPlotLib for plotting data as well as for displaying some of the smaller networks.

For generating the edges, an approach had to be taken that would not have too large of a time-complexity. The naïve approach would be to doubly iterate over each tweet and then create edges whenever at least one hashtag is found the hashtag lists of both tweets. This would have a time complexity of roughly:

$$O(n) = n^2$$

and would surely have taken many days just to generate the edge list. In contrast, the below figure shows the pseudocode for generating the edge list in a matter of minutes.

```
// Get list of tweets for each hashtag

set TWEETS_WITH_HASHTAG to map (string -> array)
for tweet in tweets
  for hashtag in tweet.hashtags
    add tweet.id to TWEETS_WITH_HASHTAG[hashtag]

// Create edges

set EDGES_CREATED to "empty list"
for (hashtag, tweet_ids) in TWEETS_WITH_HASHTAG
  set amount_of_ids to (length of tweet_ids)
  for i in list(0, 1, …, amount_of_id-1)
    for j in list(i+1, i+2, …, amount_of_ids)
      set new_edge to (tweet_ids[i], tweet_ids[j])
      add new_edge to EDGES_CREATED
```

Fig. 1. Pseudocode for generating list of edges from co-occurances of hashtags between tweets. Note that the tweets here are represented with their Twitter IDs. In actual implementation the Twitter IDs should be mapped to a list of indeces starting from 0, as the Twitter IDs are 19 character long strings and cannot be converted to *int32* for optimization purposes.

As mentioned in the previous section, for finding connected components I implemented the Union Find algorithm. Below is the pseudocode which accomplishes this task.

```
// Helper functions

function find(x, parents)
  if parents[x] ≠ x
    set parents[x] to find(parents[x])
  return parents[x]

function union(a, b, parents, ranks)
  set ap to find(a)
  set bp to find(b)
  if ranks[ap] < ranks[bp]
    set parents[ab] to bp
  else if ranks[ap] > ranks[bp]
    set parents[bp] to ap
  else
    set parents[ab] to bp
    increase ranks[bp] by 1

// Initialize union find

set PARENTS to (0, 1, 2, …, n-1, n)
set RANKS to (0, 0, 0, …, 0, 0)

for (source, target) in EDGES
  union(source, target, PARENTS, RANKS)

// produce list of nodes in each component

set COMPONENTS to empty container
set NODES to {x : (x,x) in EDGE LIST }
for node in nodes
  p = find(node, PARENTS)
  add node to COMPONENTS[p]
```

Fig. 2. Pseudocode of Union Find algorithm for finding list of nodes in each connected component given an edgelist of the graph.

VI. Results and Discussions

A. *Nodes and Edges*

The total number of unique hashtags used by at least one tweet was 292 264. However, the number of hashtags that were used by more than one tweet was 81 931 (28%), the other 78% of all unique hashtags used will not contribute edges to the network. Figure 3 shows the frequency of use of the 10 most used hashtags, and figure 4 shows the number of edges in the complete graph created by connecting all tweets that use that hashtag for these same hashtags. We can see that the *coronavirus* hashtag was most used and it will contribute far more edges than any other hashtag.

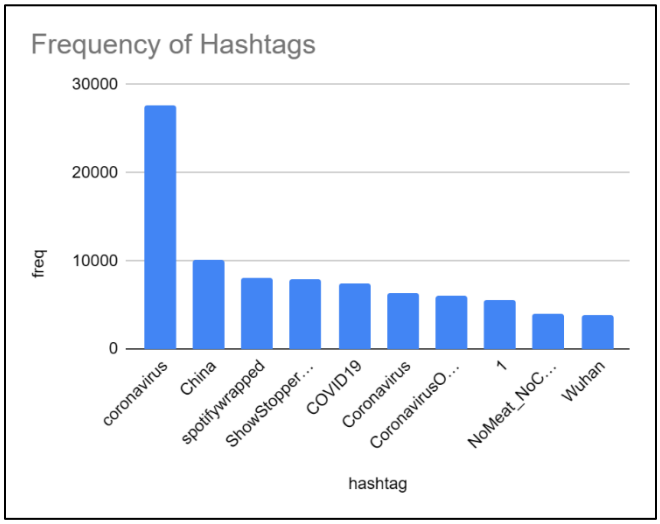

Fig. 3. Frequency of use of top 10 hashtags.

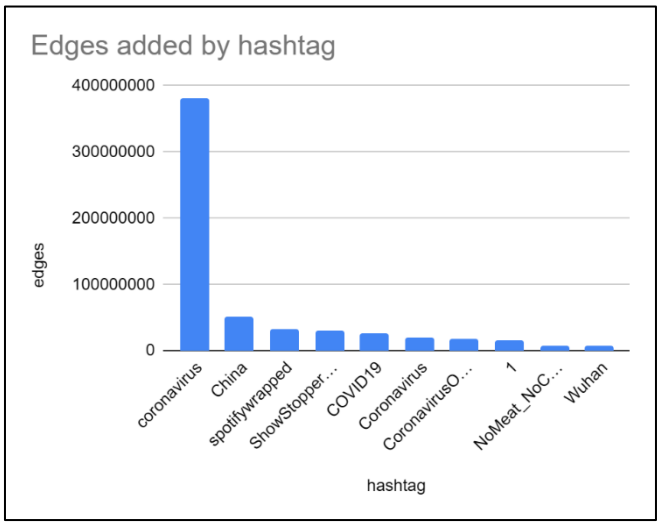

Fig. 4. Amount of edges in subgraph added to the network by the hastag. This is the minimum amount of edges added by that hashtag.

The complete graphs for each hashtag will be connected to other complete graphs by tweets that used both hashtags. Most tweets didn't use any hashtags, but for those that did, it would appear that the number of hashtags used per tweet follows a power law (I haven't tested this for goodness-of-fit). The histogram of this distribution is shown in figure 5 below. Some tweets used upwards of 30 hashtags, which means that they were very highly connective to other complete subgraphs. If a tweet only used one hashtag (which was used by another tweet as well) it's transitivity would be 1, because it is only connected to tweets in one of these hashtag-representing subgraphs, and that subgraph is complete. As the number of hashtags used goes up, the transitivity would go down, although I would still expect the overall transitivity to be quite high (close to 1). We will return to transitivity later on.

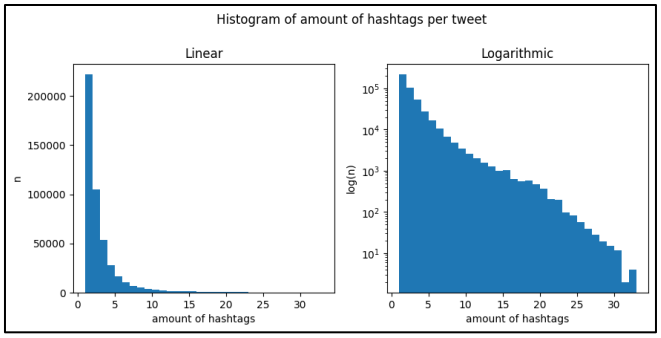

Fig. 5. Histogram of amount of hashtags per tweet, ignoring tweets that contain no hashtags (which are not added to the network, although they are included in sentiment analysis later on).

In the dataset there are 1 848 756 tweets within our desired timeframe of between December 2019 and February 2020 (3 month period). Of these, 462 901 tweets (25%) have one or more hashtags, and when we generate the edge list we find that 410 885 tweets (89% of previous value) are connected to another tweet. That leaves 52 016 tweets that have hashtags, but those hashtags are not used by any other tweet so they aren't connected. From now on, the set of tweets not connected to any other tweet will not be considered as part of the network, although they will be considered during the sentiment analysis which doesn't require a network as such.

The number of edges in the network is 684 732 453. The average degree of the network is therefore 1666.5, which supports our earlier hypothesis that the network would be very dense. Despite the density, if we consider the adjacency matrix representation of the graph only around 0.4% of the elements contain non-zero values (the only non zero values in this network is one). Interestingly, the amount of edges in the largest complete subgraph – which is associated with the hashtag *coronavirus* – is around 380 million. This accounts for around 55% of all the edges in the network.

B. Connected Components

Via my implementation of the Union Find algorithm, I was able to find 7,746 connected components in the graph with more than one node. As we can see from the table below, the vast majority of nodes (94%) are in the largest of these components. All components smaller than the largest have fewer than 100 nodes, and their average size is only 3.3 nodes. Since the smallest component size is 2 (I am ignoring isolated nodes for the purpose of analysis) this suggests that a great many components are only 2 or 3 nodes in size. The average component size with the largest component included is 53.

TABLE II. TOP 10 LARGEST COMPONENTS

nth largest component	**Amount of nodes**
1	385,468
2	95
3	80
4	78
5	71
6	68
7	67
8	59
9	57
10	52

Thanks to the small size of these other components, it is possible to visualize them in a meaningful way. Figures 6-8 show the 2nd, 3rd and 6th largest components respectively. Many of the components I checked were complete (such as the component in figure 3) and had perhaps one or two hashtags associated with each node. Components 2 and 6 had a larger set of hashtags, but were still obscure enough to not be connected to any other tweets.

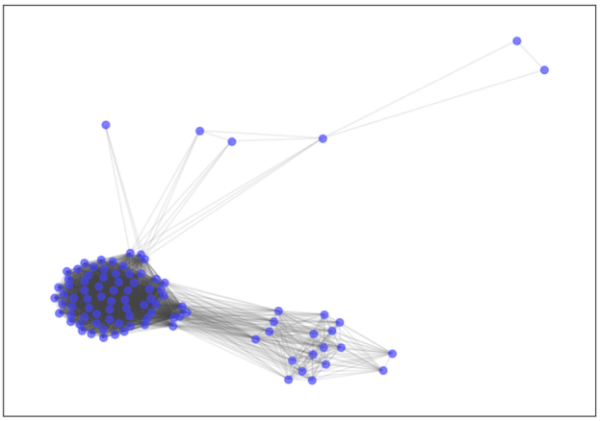

Fig. 6. 2nd largest connected component visualized with NetworkX and MatPlotLib

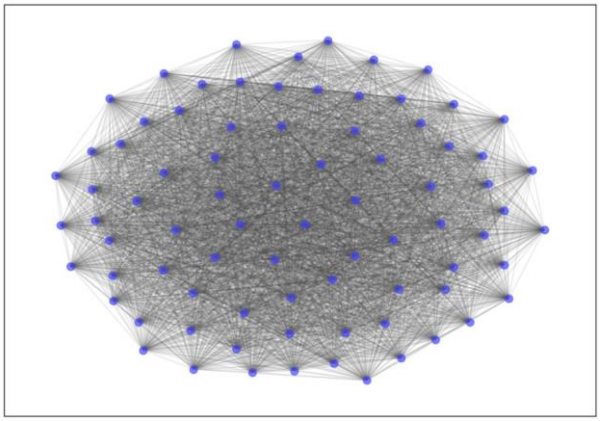

Fig. 7. 3rd largest connected component

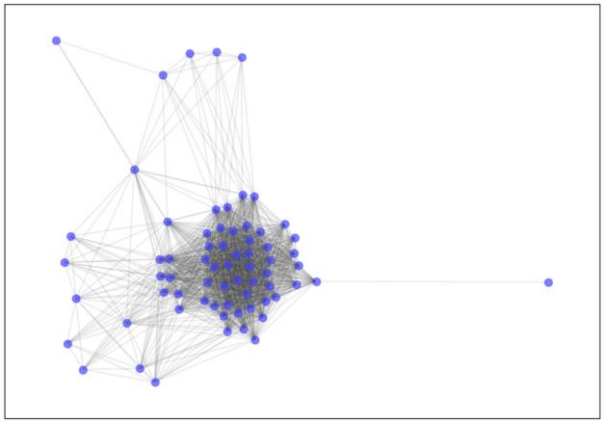

Fig. 8. 6th largest connected component

These isolated communities can actually be seen quite well by visualizing the abstracted version of the network, where nodes represent hashtags. Figure 9 shows a small section of this abstracted graph which was visualized using *cosmograph.app*, an online tool for visualizing large graphs. The whole network is far too large for this tool to handle, but it was able to visualize the hashtag graph, which contained 81 931 nodes and over 815 000 edges. The algorithm for determining the layout of nodes caused the smaller connected components to clump together. (Note that the components shown are abstractions as they show the hashtags, so a component with 2 nodes could represent a component in the full graph with 50 nodes, but with 2 hashtags used within the component.)

Fig. 9. Section of the abstracted hashtag graph, which was visualized with https://cosmograph.app.

Due to shortage of time and difficulty of computation, I was unable to get an exact figure for the average shortest path length of the largest component in the network. Below are the average shortest path lengths for the other 10 largest components.

TABLE III. AVERAGE SHORTEST PATH LENGTHS

nth largest	Average shortest path length
2	1.47122
3	1.0
4	1.0
5	1.0
6	1.42757
7	1.0
8	1.0
9	1.0
10	1.0

C. Degree Centrality

The histogram of the degree centralities of all then nodes in the graph can be seen in figure 10. Also, the maximum, minimum and mean degree centralities can be seen in table III. Looking at the histogram, I notice immediately that there is a strange spike which I would visually estimate to occur around degree 27 000. At first, I thought this spike was either an error, or indicative or data corruption, but I believe that this spike can be explained by the largest hashtag-representing subgraph.

Recall that the most used hashtag was *coronavirus* and that it had approximately 27 000 tweets. In a complete graph the degree centrality of each node is simply the number of nodes minus one. In a complete subgraph in which the nodes can be connected to outside nodes, the degree centrality of each node can only be higher than that. In figure 11 I have first converted the y-axis to be logarithmic to better see the patterns in the histogram, and second added a bar to represent

this minimum degree centrality histogram of the subgraph associated with the hashtag *coronavirus*. We can see that it lines up pretty well with the spike, and I believe it is the reason for this anomaly (the bar is slightly offset from the spike in the graph because it doesn't perfectly align with the bins used in the histogram).

TABLE IV. DEGREE CENTRALITY METRICS

Metrics	
Max	33 748
Min	1
Mean	1 666

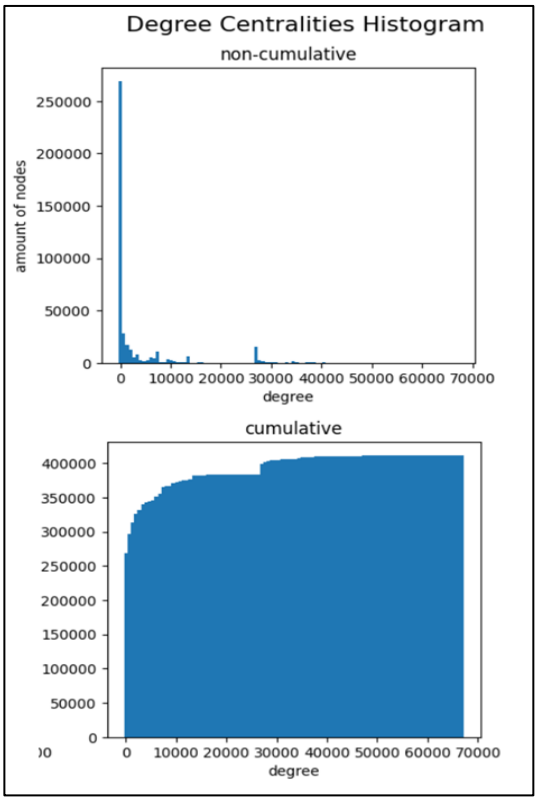

Fig. 10. Non-cumulative and cumulative histograms of degree centralities of all the nodes in the network.

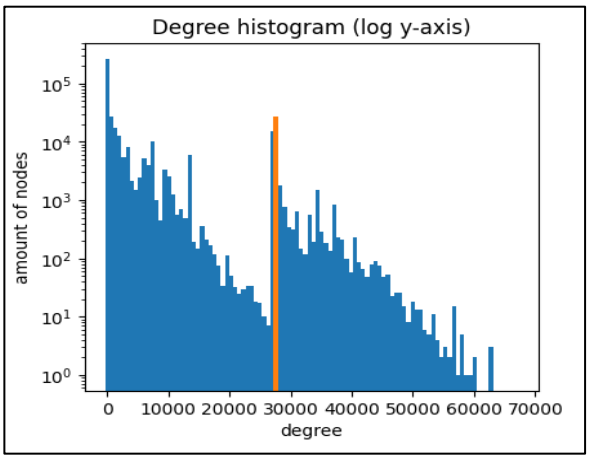

Fig. 11. Log plot of degree centralities. The orange bar represents the subgraph representing the *#coronavirus* hashtag (most used hashtag), where the x-placement represents the *minimum degree centrality* of each node in the subgraph and the height represents the amount of nodes in the subgraph.

Then is the question of whether the degree centralities are power law distributed. As seen from figure 12, the degree centralities of the nodes do not seem to follow a linear distribution in a log-log plot. Instead, the plot starts out very straight, but then enters an arching pattern which spreads out the further right you go. I believe the spreading out is due to the exponentially larger bins used for the x-axis.

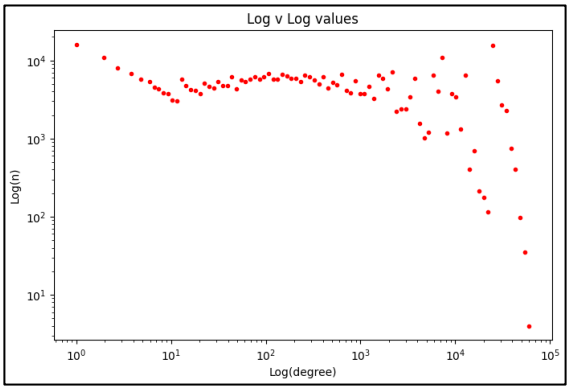

Fig. 12. Log-log plot of degree centralities

Without performing any analysis for goodness-of-fit, I feel confident concluding that based on a visual analysis of the log-log plot of degree centralities that they are not power law distributed. However, data points below degree of 100 might be power law distributed.

D. Evolution of Sentiment

In the following two subsections the evolution of the network/dataset was analysed by dividing the three month period that we are looking at into 100 equally spaced time bins and then analysing the cumulative network/dataset up to each time bin. In the current subsection we are looking at how the positive and negative sentiment scores evolved over this time. This analysis doesn't require analysis of the network per se, and was therefore pretty straightforward. (Also, note that we are including *all* 1.8 million tweets from this time period in this analysis).

In figure 13 we see the plot of average positive (green) and negative (red) sentiment scores for the cumulative dataset. Both plots have a general trend downwards. Also, both have a bit of variance or noise at the start, which I believe is due to the smaller number of tweets used for the data points at the start which results in random fluctuations affecting the data more.

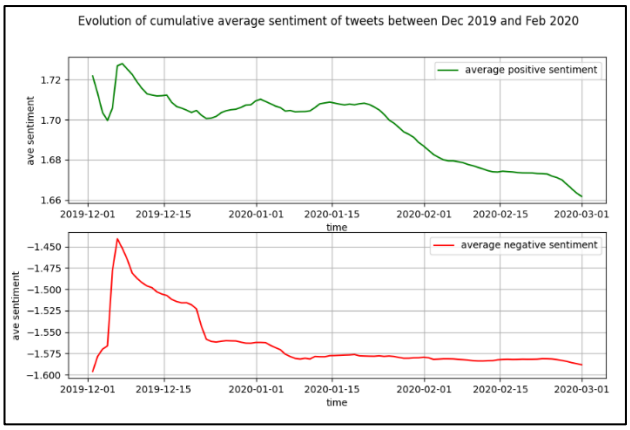

Fig. 13. Evolution of average positive and negative sentiment scores of the cumulative network up across the timespan.

As for the differences, the negative sentiment plummets early and then stays quite low, while the positive sentiment stays quite high until it drops off around late January. I am not sure why there is such a discrepancy in when the plots plummet, however. I do think that the trend in the positive sentiment can be explained in the following way: cases of novel coronavirus were first detected in China in December 2019 and the WHO declared a Public Health Emergency of International Concern (PHEIC) on January 30th 2020 [5]. This was only characterized as a pandemic in March 11th (which falls outside the time period I am analysing). I believe that this drop off in positive sentiment can be explained by the increased negative reporting on the pandemic around late January as well as the declaration of a PHEIC by the WHO. Why the negative sentiment was unaffected by this I cannot say for certain. Perhaps people who made negative tweets were pessimistic from the start.

E. Evolution of Transitivity

For analysing the evolution of transitivity I utilized the same 100 time bins as before. The project description required calculating the transitivity for the cumulative network up to each time bin, but due to lack of time I found it necessary to adopt a slightly different approach, where instead I looked at a sliding window with a maximum size of 10 time bins and analysed the network consisting of all tweets within that window. Figure 14 shows the result of this analysis.

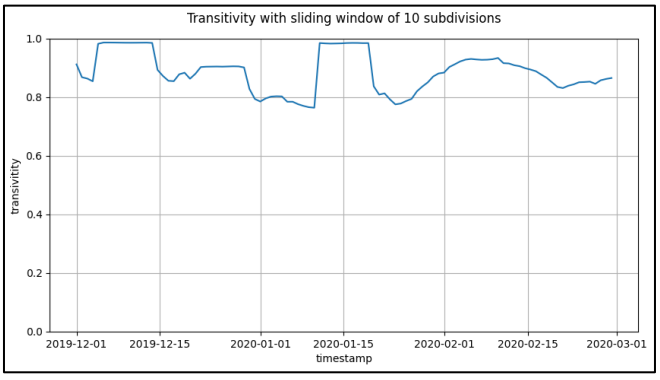

Fig. 14. Evolution of transitivity of a sliding window of max size 10.

The transitivity of the graph is generally quite high, usually above 0.8. At two separate times in the graph the transitivity gets very close to 1 and stays there with very little change. At other times the graph evolves in a more organic way, and sometimes the graph jumps suddenly. My hypothesis as to why the graph has this shape is that a hashtag gets very popular very suddenly, and if there are fewer overall hashtags being used in a time period the transitivity of the graph will be higher and more stable. Also, because of my use of a sliding window, if a group of tweets that use a particular hashtag falls out of the window, the transitivity can have a sudden change which could explain the sudden drops as well.

VII. Conclusion and Prespectives

Unfortunately I did not manage to complete all the tasks in this project, and those that I did complete I did not do as thoroughly as I would have liked. I believe that my methods could have been improved and optimized even further, and with proper planning the tasks could have been completed satisfactorily. That being said, the network I was given to analyse was indeed quite large (particularly because of the amount of edges), and given the constraints of time and hardware - as well as the fact that I was working alone - I am decently satisfied with the work I have produced. What follows is a discussion of further work I believe could be done.

I found the degree centrality distribution to be very interesting, and I think more work could be done to understand its shape. For instance, one could generate random graphs that follow the same basic structure as this network (a network made up of interconnected complete subgraphs with varying sizes) and vary the size of the complete graphs as well as how connected they are together (perhaps starting with no connections between complete subgraphs) and analyse the effect these changes have on degree centrality.

I was also interested in the plot of transitivity over time. I think more thought could be given as to the exact reasons why the plot has the features it does. I would be interested to see the plot of transitivity compared to the number of tweets, number of hashtags, and distribution of hashtag groups so see if any connections could be made.

I believe that the hashtag version of the network could have been of more use here, and also of use in additional analysis. I would have liked to see how average shortest path length could be mathematically inferred (or estimated) from looking at the average shortest path length of the hashtag network, along with its structure. Furthermore, the hashtag network could be used to estimate the structural balance of the whole network by making simplifying the search for balanced and unbalanced triangles within the complete subgraphs. These methods could be useful to analysing very large networks similar to the structure of this one.

Two additional points of analysis that the hashtag network can be useful for (which are beyond the scope of the project description) are the diameter of the network and community detection. Firstly, the diameter (or the longest shortest path)

of the full network can be exactly found by taking the diameter of the hashtag network and simply adding one. For community detection, this approach could be useful as community detection algorithms can be prohibitively computationally expensive, and if we treat all nodes in the same complete subgraph as belonging to the same community, we could get meaningful results from detecting communities in the hashtag network and generalising to the whole network.

VIII. References

[1] İ. Türker and E. E. Sulak, "A multilayer network analysis of hashtags in twitter via co-occurrence and semantic links," *International Journal of Modern Physics B,* vol. 32, no. 4, 2018.

[2] M. M. Stirling, "Project for Social Network Analysis 2024," GitHub, 2024. [Online]. Available: https://github.com/MattThePerson/SNA_Project_2024. [Accessed 2024].

[3] D. D. Erdal Baran, "TweetsCOV19 - A Semantically Annotated Corpus of Tweets About the COVID-19 Pandemic," 4 June 2020. [Online]. Available: https://zenodo.org/records/3871753.

[4] "Twitter IDs," Twitter, [Online]. Available: https://developer.twitter.com/en/docs/twitter-ids. [Accessed May 2024].

[5] "Coronavirus disease (COVID-19) pandemic," World Health Organization, [Online]. Available: https://www.who.int/europe/emergencies/situations/covid-19#:~:text=Cases%20of%20novel%20coronavirus%20(nCoV,pandemic%20on%2011%20March%202020.. [Accessed 13 May 2024].