University of Sheffield

Visualising COVID-19 misinformation networks

Lucian Murdin

Supervisor: Carolina Scarton

A report submitted in partial fulfilment of the requirements
for the degree of Bachelors in Computer Science

in the

Department of Computer Science

May 23, 2022

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: Lucian Murdin

Signature: Lucian Murdin

Date: 19/05/2021

Abstract

Misinformation on social media causes mistrust in public authorities and the media, apprehension for public health programs, and other negative societal effects, with far reaching consequences beyond its social network domain. COVID-19 has highlighted this, and study of misinformation spread through social networks is important. This project created Twitter diffusion networks on two COVID-19 Twitter datastreams split into misinformation on COVID-19 and general COVID-19 conversation from March to October 2020. It visualises these on Gephi, an open source visualisation tool. The visualisations group nodes into communities, and novel network structures are revealed. Classifiers were ran on the tweets to delineate claim topics and debunks vs. misinfo, exposing conclusions on the homogeneity of twitter communities. Network properties are calculated on these networks to quantify their characteristics. This project finds visualising these networks is possible, informative and novel, and opens questions for further COVID-19 network analysis.

Acknowledgements

I would like to thank my supervisor Carolina Scarton for her continual guidance and support throughout this project, helping me develop my own ideas and allowing me the freedom to direct the project in ways I found interesting. I would also like to thank Jacob Crawley for his collaboration, helping me implement his system into my own project.

Finally I would like to thank my housemates and friends for being a great support throughout my whole time at university, as well as my parents for being there for me at every step of the way through my education and life.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Misinformation surrounding COVID-19 on social media has been prevalent since the beginning of the pandemic. It is damaging in many aspects, not only spreading mistrust for health professionals and public health projects like vaccine distribution, but also diffusing dangerous rumours: 700 people died from drinking denatured alcohol to combat the coronavirus, spurred by misinformation online [12]. Information shared on social media is often unverified, yet is still shared easily. This has led to what is being called an 'Infodemic', defined as 'too much information including false or misleading information in digital and physical environments during a disease outbreak' by the WHO [11]. The requirement for identification and analysis of misinformation, as well as differentiation from reliable information, has never been more important than over the course of the pandemic.

Misinformation often spreads through unverified sources on social media websites such as Twitter, Facebook, or Reddit: as mentioned above, this has serious real world effects, and more needs to be done to find how this spreads through these networks, and where it comes from. Network analysis concerns the analysis of metadata of information shared on social media, using this information to build an image of a network with nodes and edges on a graph. On Twitter, metadata might include how many followers the author has, how many retweets the post might receive, and how the claim might spread within certain communities on social media, tracked through retweets [10].

1.2 Aims and objectives

This project aims to combat the diffusal of misinformation by visualising tweet networks, using Twitter datastreams containing COVID-19 misinformation claims and general COVID-19 claims. This project will use network analysis techniques to process Twitter data that has already been collected into a repository (sourced by The University of Sheffield [24]) It will investigate trends within two COVID-19 datastreams. Topology analysis will be conducted, using quote tweets to build graphs representing diffusion networks that form within such data. Analysis on these networks will include measuring network properties and metrics. Network analysis techniques have in fact previously been applied to Twitter datasets [31] to analyse the influence of social media on election results : principal 'superspreaders' of misinformation were identified. Similar techniques will also be applied in this project to identify well connected nodes in diffusion networks relating to COVID-19.

These aims can be split into primary and secondary sections. The primary aim is to display these twitter networks with currently available visualisation tools, such as Gephi. This will produce visualisations of both datastreams, informing on differences between them, and giving an idea on the key network structure behind the twitter dataset. The secondary aim is to enrich the tweets with further data, from classifiers that would identify the types of COVID topic, or whether the tweet is misinformation or a debunk of that misinformation. It also includes calculating network properties on the graphs produced.

Constraints in this project include the handling of personal information, as the project will be handling potentially identifiable data such as Twitter usernames. These concerns will be addressed by an ethics application, and data gathered from public domains like Twitter will be suitably anonymised if it will be displayed in any way. Other constraints might include the vastness of the Twitter datasets that will be handled, as larger datasets, potentially reaching millions of tweets, might be computationally unmanageable. The results will need to be selected carefully to retain a broad picture, but not lose out on relevance during analysis.

In summary, this project will collect COVID-19 claims from a Twitter repository in order to visualise Twitter networks in graphing software. The project will also analyse networks created on these data with network properties. The project further aims to identify novel differences between the two datastreams when creating the networks. These differences can then be used to show how the two types of networks behave through network visualisation, with the secondary aim of enriching the nodes of the network with extra attributes. This project will display these data both statistically/numerically, and with network visualisation tools.

The rest of this report will consist of a literature review, which will detail some previous work done on gathering misinformation, including COVID data, as well as its analysis and classification in various techniques: this will provide the relevant background information for a starting point into this project. It will also describe multiple network analysis and visualisation techniques. A section on requirements and analysis will follow, detailing exactly what this project will deliver and how it will do so, with a description of tools to be used in development. Then follows the design section, which will propose a plan for the application to build the Twitter network and introduce other technologies such as classifiers that will be used. The implementation section will recount the progress of the project, including unseen challenges and limitations that affected development. Then the report will discuss and display the results, with the visualisations on show and the statistical analyses accompanying them, as well as any novel conclusions from these results.

Aside: Relationship to my degree

This project demands experience with several aspects of Computer Science and various skills, some of which have been covered in my degree, and some of which are new to me. Topics which I have experience in include text processing for dealing with the tweet data, working with Python and RESTful APIs for the main application, aswell as JSON object handling.

Topics in this dissertation which are new to me involve working with big data: pulling large datasets from online sources and processing them, using Python to do this. Network analysis is also new to me: learning about new ways of analysing graphical data and the properties of network graphs were particular challenges. Broader skills of mine were also developed: report writing was new to me, so I was challenged to write a scientific report, building a system and gathering results to discuss.

Chapter 2

Literature Survey

This section of the dissertation will outline the background for the project: firstly, it will describe studies on misinformation on social media. Then it will show some examples of datasets produced on social networks for analysis. Network analysis techniques and visualisation tools will then be described, so as to provide context for the objectives of this project.

2.1 Misinformation in social media

2.1.1 Overview

Misinformation is defined in many ways : in this project, it will be defined as “false or inaccurate information that is spread on social media”. It differs from ‘disinformation’ in that it is not deliberately created to deceive [37].

Misinformation is prevalent on social media : the ease of communication and spread afford wide reaching exposure across many platforms. Even before the pandemic, studies on misinformation relating to health were completed. Wang et al. reviewed the “nature and drivers of health-related misinformation” [36]. It was concluded that previously studied topics from 2012 to 2018 mostly investigated health-related misinformation such as that surrounding vaccines or cancer, and that many studies employed social network analysis. It also proposed future studies should examine the susceptibility of different social groups to spread misinformation: why different types of people might share claims or not. This study was published before the outbreak of the pandemic, so its review on health misinformation does not include studies on COVID-19: the pandemic has brought misinformation research to larger public attention.

OSNs (Online social networks) have announced they are attempting to limit the spread of misinformation, with Facebook claiming to have removed over 3000 accounts spreading COVID-19 and vaccine misinformation [28]. COVID-19 has generated wide response over misinformation: previously, some networks stopped short of banning misinformation-linked accounts, arguing harm on free speech. Social network providers have now detailed ways in which they combat misinformation, from attaching warning labels to banning accounts and removing content [26]. However, there is no standardised response from all companies, which is important given the context of what this project aims to address: a way to more effectively analyse misinformation.

2.1.2 Misinformation and COVID-19

Since the COVID-19 pandemic began in early 2020, scientific factsheets and studies have been produced to categorise COVID-related misinformation to varying degrees of detail. COVID-related social media claims have been shown to be diverse, spanning a wide range of topics and platforms [16, 17]. The following studies investigate where this COVID-19 specific data comes from in comparison to previously studied misinformation, how it is spread, who it is spread by, and how its level of falsity correlates with how fast the claims might spread within a network: understanding these points is key to further analysis of this data.

The Reuters Institute produced a study concerning COVID-19 misinformation on social media during the start of the pandemic from January to March 2020 [16]. The factsheet concluded that there were many different types of misinformation, spread with many different intentions; more influential actors also had higher levels of engagement. The study showed how follower count is related to how influential a user might be within a social network: if they have a large base of followers to share content with, their retweets/tweets can reach a larger audience and diffuse a claim more widely. The study also categorised COVID claims into groups, e.g. “Public authority action” and “Community Spread”, correlating to claims concerning actions or policies public authorities are taking on the virus and how the virus spreads within communities respectively. This categorisation showed the diversity of COVID-19 misinformation. The difference between bottom-up and top-down misinformation was also specified, where misinformation comes from members of the broader public and more influential users like politicians and celebrities respectively. Top-down misinformation was shown to account for 69% of engagements within social media content, whilst only making up a 20% share of content produced/spread overall, implying that more prominent users on social media drive misinformation sharing: this shows how further investigation of these diffusion networks with focus on influential users is necessary in combating misinformation spread.

Representation of the way that misinformation spreads through a network has been investigated in “The COVID-19 Infodemic” [17]. The study described the previously mentioned ‘Infodemic’, defining the explosion of information, true and false, surrounding COVID-19, so as to formally investigate whether this was truly occurring in an epidemic-like way on social networks. It made use of already existing epidemiological models to fit information spreading about the pandemic, which involved calculation of the basic reproduction number R_0. The number represents the factor by which misinformation multiplies for each new ‘exposure’, and the factor was calculated for multiple social networks, including Twitter, where it was calculated to be between 4.0 and 5.1, indicating misinformation was highly ‘viral’ and would be reproduced many times over. This study also compared both questionable and reliable information sources, and found that on Twitter both types are amplified equally. This has interesting implications for this dissertation, as it will similarly investigate two datastreams: do they show similar properties to questionable and reliable information (equal amplification) through network analysis?

Another exploratory study into propagation of misinformation was conducted on Twitter data. The study analysed tweets connected with fact-checked COVID-19 claims, collecting both false and partially false claims. It found how verified twitter accounts were involved in either retweeting or tweeting new misinformation. It also investigated the speed of propagation of misinformation through the Twitter network [34]. This could indicate that the verification status of an account might have an effect on its tendency to diffuse misinformation, which could be investigated with this project's twitter data.

2.2 Online Social Network (OSN) Datasets

2.2.1 Overview

Social networks contain large amounts of data that can inform on various trends. The sheer scale of information poses a serious challenge for any data analysis tasks. For example, on Twitter there are 500 million tweets posted each day [33]. Twitter datasets that this project will analyse need to be manageable in size and specific to COVID-19. Since the beginning of the pandemic there have been datsets produced that meet these criteria, detailed below, often making use of the COVID-19 Twitter stream [13].

2.2.2 COVID-19 misinformation datasets

TweetsCOV19 is a dataset formed of 20 million tweets spanning October 2019 to April 2020 [18]. It was taken as a subset from Gesis' (Leibniz Institute for the Social Sciences) TweetsKB, a dataset of more than 2 billion tweets produced for data consumers analysing Twitter data [19]. TweetsCOV19 was developed using 268 keywords (facemask, covidiot, pandemic) on TweetsKB, and contains various metadata including: #Retweets, #Followers, Tweet ID. It also analyses sentiment of the tweet body. The paper described potential use cases for this dataset, like investigating the societal impact of the pandemic, and for analysing effectiveness of information campaigns on public opinion. This dataset is diverse and provides a good base for data analysis, and has the advantage of including sentiment score with each tweet. However, compared to other datasets it is quite small.

The COVID-19 Twitter stream is a datastream produced by Twitter themselves: the stream returns tweets based on Twitter's internal tweet annotation [13]. It delivers real time full data on tweets related to COVID-19, determined via hashtags such as #covid2019, as well as strings in the tweet text. This filtering by hashtags is useful to narrow down large streams of data. However, the datastream is relatively unorganised compared to dedicated COVID datasets, as the stream purely provides tweets and their metadata, with no extra information. It is a useful tool for researchers to then build their own more advanced datasets.

Researchers at The University of Sheffield collected COVID19 twitter data in both English (41.5M tweets collected so far) and Portuguese (13.6M collected tweets). This datastream was gathered as part of the paper "Categorising Fine-to-Coarse Grained Misinformation" [24], which formed a manually annotated dataset from these tweets. The tweets were annotated as primarily debunk or misinformation, with other categories question, comment, relevant / irrelevant making up the ambiguous data. This paper provides an important resource for examples of debunks and misinformation, and the two datastreams it gathered will be used in this project for sampling.

2.3 Network Analysis

As shown in the introduction, the aim of this project is to carry out network analysis techniques on Twitter data. Network analysis is a broad term, and it should be specified how it applies to social networks like Twitter, and what the takeaways from this can be for this research. This section will also show how some users on Twitter are more influential than others, and how identifying these users within diffusion networks is important in network analysis.

Before the pandemic began, there were already studies being carried out on misinformation and disinformation within social media. Italian disinformation surrounding the EU elections was studied by Pierri et al. [31]. This research built Twitter diffusion networks by corresponding original tweets, where the user had not retweeted or replied to other content but had written it themselves, to nodes in the network. Then, directional edges were built when the tweet received replies or retweets, and when a user was mentioned or quoted by another. These networks were where the researchers identified highly debated topics of misinformation, as well as classifying communities within Twitter sharing similar content and 'superspreaders' within those communities. A community in the context of network topology is a dense subgraph within a network, which is separated from other communities: the implication within the network described above is that original content is shared mostly within contained groups. The report is limited by its scope, given that it is focussed on a specific country (Italy) in a specific topic (EU elections), and uses content from a limited set of 'disinformation outlets' determined by the researchers themselves. However, the publication does usefully describe metrics used on the diffusion networks, notably k-core, in-strength/out-strength, and centrality values, all of which can be applied to networks built using COVID-19 misinformation. It should be noted that this report classifies disinformation, which again differs in definition from misinformation in that it is said to be actively spread by malicious actors, whereas misinformation are rumours spread with no underlying motive.

So far these studies have shown analysis on single network topologies, but one of my project aims is to compare different topologies of multiple networks. A 2020 report from

Pierri et al. [32], following on from their previous investigation into disinformation referenced above, compared diffusion networks of both misleading and mainstream sources. The diffusion networks were built in the same manner as in [31], but an important caveat was noted. A true diffusion network cannot be constructed with Twitter, because a retweet of a retweet will point back to the originally authored tweet, hence the full cascade of interactions cannot be represented. The study calculated a range of network properties, including k-core measures, and computation of how connected each component (node) of the graph is, giving rise to strongly connected components and weakly connected ones. The relevance of these properties is then shown: communities sharing misleading information are found to be more well connected and clustered together than mainstream networks, which have a larger global audience and more weakly connected components, showing that reliable information is not so bound to tight Twitter communities. This report shows how a number of analyses can differentiate between two networks, showing novel differences in their structure and properties. Since comparison has not been carried out in a similar way on COVID data, this gives a good base to apply to COVID-19 misinformation and debunks.

Further analysis of influential users was explored in [23], where a Weighted Correlated Influence (WCI) measure was calculated to represent features of a user's impact online. The measure aimed to combine both useful statistics on a user's profile (total number of followers, following etc.) and underlying network topology features (betweenness of node, in-degree/out-degree etc.). This combination of both user specific statistics and network properties gives a useful way of combining both sets of information, supplying a more meaningful metric to measure influence of particular nodes in a network. This metric could be calculated on nodes with linked users within diffusion networks in this project to identify particularly influential Twitter users.

Another report which used centrality metrics to extract useful information from Twitter was conducted on COVID-19 data [30]. The work characterised 'information leaders', users on Twitter with high popularity, by producing a Twitter diffusion network and calculating centrality metrics based on users' follower counts, retweet counts, and 'following' number. However, it was found that these values alone do not predict high influence within a network (compared via centrality metrics). This further reinforces the need for network analysis on COVID data, as it is only when this is combined with obvious measures of popularity like follower count that useful conclusions can be made about which users diffuse misinformation on social networks.

2.4 Network Visualisation

Basic network visualisation represents nodes as points on a graph, and edges between these nodes as lines connecting them: a set of entities linked by relations [25]. When these data are represented graphically, valuable visual insights are gained by users who may be unfamiliar with the intricacies of the data: concepts such as well connected nodes can be displayed numerically, and will make it immediately apparent which nodes are important in the dataset.

When representing small graphs, with few nodes and edges, these can be visualised simply and clearly. However, when dealing with large amounts of data, such as those handled on social media potentially in the 1000s of nodes, visualisation becomes complicated and may produce illegible graphs. Innovative ways of representing networks and useful network properties are required: network visualisation is not just about showing data in a human readable format, it is about telling a story with the data and highlighting its most interesting features.

Studies have proposed ways of visualising large datasets: published in 2009, NodeXL is a toolkit for visualisation of network features on such datasets [35]. It was designed to work on the Excel spreadsheet software, an obvious limitation, but it still provides a basis for many visualisation techniques to be broadly applied. The software aimed to compute overall network metrics, like total number of edges and nodes, as well as advanced statistics like node rankings, with betweenness and centrality taken into account. The paper describes the unrefined data visualisation, which shows as an unorganised and illegible graph. It then proposes displaying nodes with their node-size as proportional to their in-degree, usefully highlighting well-connected nodes. NodeXL is still being developed today from this original paper. The techniques explored are used further in future visualisation tools.

Building on NodeXL, Gephi is another visualisation tool that processes large graphs using a 3D render engine, making use of graphics processors [15]. It was originally proposed in 2009, but has had constant development through the years, with the last stable version 0.9.2 released in 2017 [21]. The application is well optimised, and can network more than 100,000 nodes. It calculates network statistics and metrics similar to NodeXL, and has various customisation options: gephi is more suited to dealing with much larger amounts of data. It can show networks changing over time, taking in real time data : there are also capabilities to filter the data according to queries on edge weight, degree range, and other network properties. Gephi uses layout algorithms to determine the best shape for the graph, providing readability and showing potential communities [5].

Chapter 3

Requirements and Analysis

3.1 Aims, Objectives and Requirements

The purpose of this project is to construct networks, and conduct network analysis techniques on the University of Sheffield COVID Twitter repository. These results will then be visualised. The visualisations will be explored to find novel trends. The gathered Twitter data will consist of two datastreams from the tweet repository, with tweets gathered by filtering hashtags as both misinformation-related, and general COVID-related. These will produce multiple visualisations of twitter data.

In order to achieve this, individual tweets in the datastream will be represented as nodes, with edges representing the link between a quote and its original tweet, so as to show the way in which tweets propagate within the bounds of the datastream. This network can then be visualised. Tweets will also be enriched with other potential attributes that could inform on trends within the datastream. Examples of these could include: user attributes (follower count, location etc.); entities lifted from tweet text (hashtags, URLs); or labels from classifiers.

The visualisation application Gephi provides utility to compute network properties on the networks fed into it. This feature will be used to gain statistical insight into the types of networks produced. Examples of these properties include: K-core measures, connectedness measures, centrality measures, betweenness measures etc. Since Gephi calculates these within its software, they should be easy to extract for multiple datasets.

The aims consist of a primary and secondary objective. The primary objective is to visualise the two datastreams of *covid19misinfo* and *covid19all*, by first building diffusion networks with nodes as tweets and edges as relationships between tweets. This will use Twitter data ranging from March to October 2020, the period over which the Sheffield tweet repository was gathered. Gephi will be used for visualisation. This objective will produce visualisations to be included in the final report, which will potentially provide useful insights into this dataset.

The secondary objective is to identify novel trends within the dataset, in order to inform on possible ways of identifying COVID-19 misinformation. This objective includes both calculating network properties on the datastream as mentioned above, and enriching the data with classifiers, for example the one produced by the WeVerify Annotation team @ EUVsVirus Hackathon [29], which categorises COVID-19 claims in types as identified by the Reuters Institute study [16]. Another classifier to be used is one being worked on by fellow dissertation student Jacob Crawley, which classifies the tweets into both debunks and misinformation, based on the textual content of the tweet and other features. In addition, attributes such as number of retweets can be added to nodes to adjust node size, and show which nodes have large engagement over a specified period of time.

Enumerated requirements from these objectives are set out as follows:

Primary Objective

1. Acquire section of Elasticsearch datastream.
2. Build diffusion network of tweets through Twitter.
3. Visualise network in useful configurations.

Secondary Objective

1. Calculate and compare network properties on Gephi.
2. Enrich nodes with attributes from classifiers.

3.2 Elasticsearch dataset

The Elasticsearch Twitter dataset developed at the University of Sheffield consists of more than 50m tweets. This size is too large and unwieldy for a project of this scale, so a section of the dataset will be produced for data visualisation. The data was collected using COVID-19 hashtags similarly to previously described papers: the data was split into streams, using hashtags that might indicate misinformation e.g. #Plandemic, #CCPVirus ; and hashtags that were more general e.g. #COVID19. The figures below show the hashtag make-up of each datastream, *misinfo* and *all* (fig 3.1 and fig 3.2 respectively).

3.3 Evaluation techniques

Evaluation of the success of this project poses a challenge in that the primary objective's success could be subjective. Creation of the visualisation itself will be most of the fulfilment of the primary objective: if the visualisation properly captures all aspects of the tweet networks

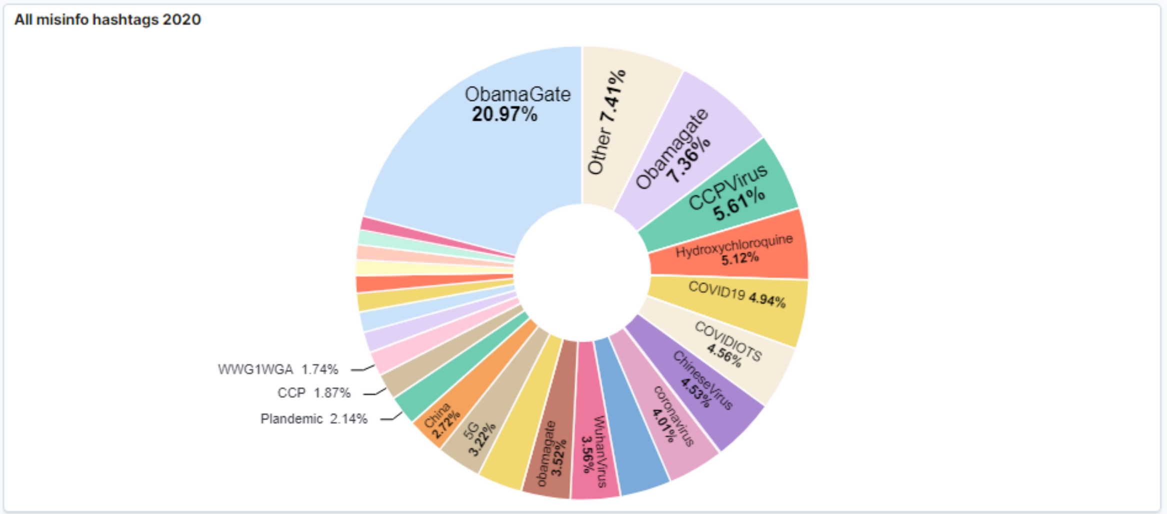


Figure 3.1: *COVID19MISINFO datastream hashtag make-up over 2020*

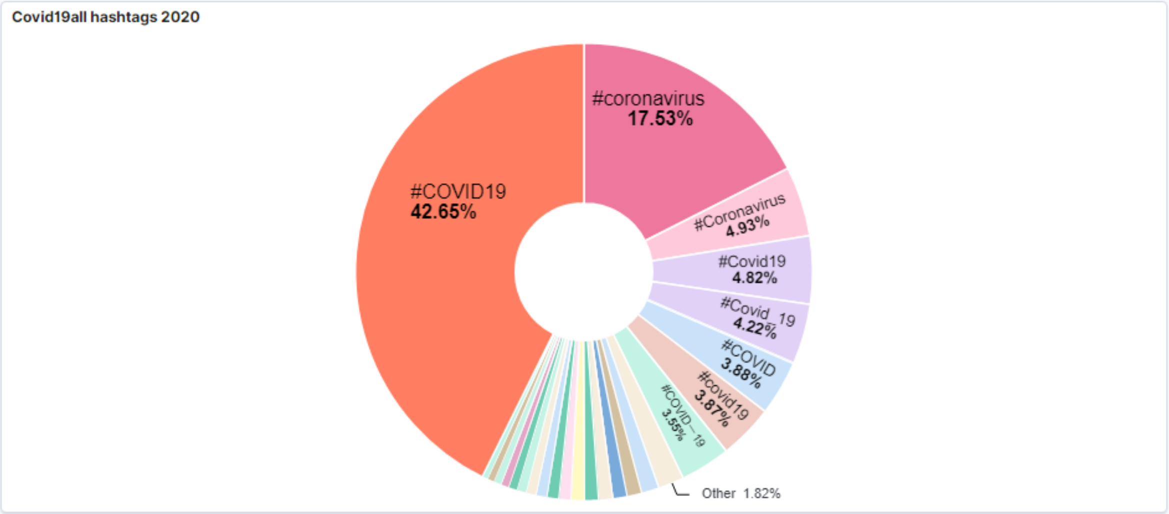


Figure 3.2: *COVID19ALL datastream hashtag make-up over 2020*

such as paths between nodes and collecting all quote tweets present, it could be called a success. However, judgement of a visualisation as ”useful” is subjective : there is not an easy way to measure its usefulness objectively as that depends on a number of factors. A visualisation could be said to tell a good story if it is legible, and relevant trends are apparent and highlighted within the structure of the nodes. This could also mean overlaying other data, such as user profile statistics, to highlight the relationships between these statistics and the produced network graphs.

For the second objective, the classifier would be successful first if they were implemented and worked well on the test data. Further to this, the results that come from the classifiers should enrich the nodes and provide some useful insight into their nature, that of the nodes surrounding them, and overall trends in the data. Calculation of network properties is straightforward: evaluation of success would come from the difference between the two datastreams in the network properties. The aim is that they show key numerical differences.

3.4 Tools for development

3.4.1 Elasticsearch

Elasticsearch is a search engine useful for handling large amounts of data. It indexes these data into an inverted index and stores them as JSON documents, correlating keys with their values (booleans, strings, array, and other types of data) [3]. It differs from traditional relational databases in that it uses RESTful GET and POST requests to manage data, immediately making it more compatible with web-related data management, and using JSON/CSV files. It also provides an advanced querying system that has the power to retrieve summaries of data, and will retrieve the most relevant results in order. These queries are advanced, and make use of fuzzy searching and other search techniques for flexibility in query language.

3.4.2 Gephi

Through the research done on currently available network visualisation tools, it is the best choice to use Gephi to visualise the Sheffield datastreams. Since the application is open-source, it will not require fees or licensing to use within this project. Gephi provides a rich number of tools for visualisation like node filtering and grouping. It also automatically calculates network metrics such as betweenness and centrality. Gephi is well used not only in social network analysis, but also for social science and biology, so a rich ecosystem of tutorials and forums surround it: it should be easy to find support when required.

3.4.3 Python

Python will be used to construct the application which will build networks on these datasets. Python is a high level readable language, meaning any code produced should be reproducible and understandable. Python also provides many useful data analysis libraries, and is supported by packages like NetworkX and Elasticsearch which will prove useful. It is well-supported in academia and online, meaning there will be ample opportunities for help if problems are encountered during development. It is well suited for data manipulation and task repetition.

3.5 Analysis of results

After networks have been built on the dataset, and network properties have been calculated, the statistics of the two datastreams, misinfo and all, will be compared. Interesting nodes in these networks (ones which might be well connected) will be evaluated. The project should conclude by summarising novel trends in each class, showing the visualisation results and results of classification/enriching nodes with attributes.

Results will be displayed in a tabular format and graphically: results for visualisation will show each datastream as a graph on Gephi, where nodes and edges are nicely organised and give meaningful insight into the data. Network properties will also be shown in tables for each datastream, and properties for influential and interesting nodes will be highlighted.

3.6 Associated risks

Risks that were considered in this project might involve a number of considerations. One might be the fact that the dataset section taken from Elasticsearch might not be accurate or large enough to display trends found in COVID misinformation. Steps taken to mitigate this include reducing the size of individual tweet documents so as to save processing time, and allowing for a larger dataset. Another risk could be that no trends are found in the data : however, this is an interesting conclusion in of itself. It might suggest that again the dataset is too small, but it may also show how the two datastreams were not significantly different in their networks.

Chapter 4

Design

This chapter will introduce the plan which will achieve the aims outlined above: it will cover the structure of twitter data, elasticsearch queries, as well as the methods for building networks out of this data : limitations of elasticsearch and twitter data will be covered, and how this factored in to the proposed design. Much of this design was produced dynamically as experience was gained with the way that Elasticsearch, Python, and other parts of the project interacted. This means that this chapter represents an overview of the implementation chapter in its final state, so some techniques and challenges encountered during development, as well as detail on the way the design developed from first concepts, will be explained further in the next chapter.

4.1 Elasticsearch Twitter data structure

This project makes use of the elasticsearch dataset provided by The University of Sheffield, gathered for a COVID annotation study [24]. The dataset is a datastream of tweets authored from April 2020 to November 2020. It used the Twitter API to gather two streams: one, *covid19misinfo*, gathered tweets which included hashtags typical of misinformation, e.g. #Plandemic, #CCPVirus. The second, *covid19all*, gathered tweets hashtagged with general COVID-19 terms, such as #COVID19, #Coronavirus etc.

4.1.1 Tweet structure

The tweet objects contained in the database follow the object structure of tweet data gathered from the Twitter API [9]. They follow a hierarchical attribute structure, where child sets of fields are contained within parent level fields, which can themselves be contained in higher parent level fields. This structure is useful when accessing the original tweet from a quote tweet referencing it, as the quote tweet will contain a field with the body of the original tweet.

The highest parent entities include amongst others the author of the tweet (user), tweet text, creation date, tweet id, and quoted_status. Further information is contained under the attribute ”entities”, or ”extended_entities”. These fields contain information parsed from the tweet text as individual entities, such as hashtags and urls. The attribute quoted_status is of particular interest, as it will provide a method of building the network as follows. A tweet also contains a unique ID, which will be necessary to use as a representation of a node on a graph.

4.1.2 Quote Tweets

As mentioned by Pierri et. al [32], the caveat for Twitter data was that a true diffusion network could not be built upon a dataset using retweets, as every retweet points back to the originally authored tweet, not the retweet a user might have seen to cause them to click ’Retweet’. Due to this limitation, it was decided to analyse ’Quote’ tweets, tweets which are type of retweet: they contain the tweet body which they are quoting, similar to ’Reply’ tweets, but in addition to the author’s own thoughts/comments.

The tweet body stored in Elasticsearch sometimes contains a field ’quoted_status’, which will only exist if the tweet is a quote tweet. This field contains another tweet body, of the original tweet. The inclusion of this field allows representation of both tweets as nodes, and a directed edge between the original and quote tweet. This is not mirrored by retweets, giving quote tweets the advantage of building true diffusal networks. These properties effectively allow a network to be formed of all of the quoted tweets in the elasticsearch dataset, by iterating through all tweets with ’quoted_status’ present, and recording their connections to original tweets that are also present in the elasticsearch dataset.

4.2 Building a Twitter network

As part of the primary aim, an application will need to be built to query the Elasticsearch server at The University of Sheffield, and represent that result in a format that can be visualised with Gephi.

Python was previously justified for use in this project, as it provides a range of relevant packages like networkx and elasticsearch: networkx is able to represent nodes as unique IDs with edges between them and assign attributes to each node, whilst being able to export this data as a GEXF file which Gephi can read. The elasticsearch package is Elastic’s own python package, designed to make connecting to and querying and elasticsearch server easy, as it provides methods for authentication and building elasticsearch queries [14].

The Python application will pull a subsection of tweets using the elasticsearch API, and represent those tweets as nodes and edges with NetworkX. NetworkX provides several data structures for representing graphs [6]. One of thise structures, the DiGraph, can represent directional edges: a directional edge will be used to represent the original tweet transferring information to the quote tweet, allowing paths to be tracked through the network. NetworkX can also translate this data into a format for Gephi to visualise the results. Since the elastic-search query DSL is able to filter the results of the query by timestamp created, this can be used to pull any subsection of the datastream, and convert it into a readable graph format.

4.3 Attaching attributes

The secondary aim is to attach attributes to the tweet objects, with the goal of further identifying trends. There are many potential attributes that could be included, with some examples following:

1. Number of retweets per tweet
2. Hashtags of tweet
3. URL of tweet
4. Tweet text of tweet
5. User follower count
6. Classified by topic of misinformation
7. Classified as debunk or misinformation

Not all of these were applied in the final implementation, but offer an extension to give further features to each tweet. Some of these attributes, such as URL, hashtag, and tweet text, are easily lifted from the tweet object. However, calculating retweets and implementing classifiers require more detail: below is a description of the initial plan for attaching each attribute.

4.3.1 Calculating retweets

The number of retweets of a specific tweet can be accessed as a field in the tweet object, however since each tweet object is fed into the datastream the moment it is created due to the hashtag filtering, this field will always be zero. It is not possible for the tweet at its point of inception to have been retweeted.

Therefore a method is required to calculate the number of retweets a tweet has at the end of a certain subsection. This will be calculated by use of the retweeted_status field in the tweet object. This field is similar to quoted_status, in that it only surfaces when a tweet is a retweet of an original. The field contains the tweet body of the original along with its ID. This structure inspires an algorithm that will retrieve all of the retweets posted in the same time period as the subset of quote tweets, and record the amount of occurrences of a certain tweet ID, indicating total retweets of a specific tweet at the end of that period.

4.3.2 Classifiers

The two classifiers that will be ran on the data include one from WeVerify EUvsVirus, and one from collaboration with Jacob Crawley, a fellow dissertation student.

WeVerify Annotation team

The first classifier, named COVID-19 claim categoriser, classifies COVID-19 claims into one of ten categories. These categories are taken from the study conducted by Reuters Institute on the COVID-19 infodemic [16]. The classifier was produced in the #EUVsVirus hackathon in early 2020, organised by the European commission, with the aim to "develop innovative solutions for coronavirus related challenges" [7]. The team co-ordinated by TUOS (The University of Sheffield), called "WeVerify Annotation", trained deep learning models to automatically classify covid claim text. The classifier is available as a "Text-Analytics as a Service" provided by TUOS. This service can be accessed through a REST API, and can handle large batches of documents. This classifier can be run on a batch of tweets, or one at a time, and the results will be fed back in as an attribute of the node.

Misinfo/Debunk Classifier

The second classifier was created by fellow dissertation student Jacob Crawley as part of his dissertation: his body of work involved the use of BERT classifiers for classification of COVID claims as primarily misinformation or debunk. The classes were set out in "Categorising Fine-to-Coarse Grained Misinformation" [24], with additional categories for comments and relevant/irrelevant claims. The classifier code collaborated on is written in python, so is compatible with this project's application. It takes tweet text, and labels it with MISINFO, DEBUNK etc. These labels will be attached as attributes to their respective nodes.

The classifier produces and uses extra features based on the tweet object, such as calculating subjectivity, positive and negative words, followers and verified status. The confusion matrix including the five categories and best-performing classifier scores for Debunk and Misinfo classification are shown in figure 4.1 and table 4.1. The confusion matrix shows the classifier predicted many MISINFO labels correctly, and the high F1 scores on MISINFO and RECALL show that although the experimental classifier returned an overall accuracy of 54% for all classes, it was effective in differentiating those main classes. The classifier also compressed the classes "Misinformation" and "Related Misinformation" [24] into one "MISINFO" class, and similarly again with "Debunk" and "Related Debunk".

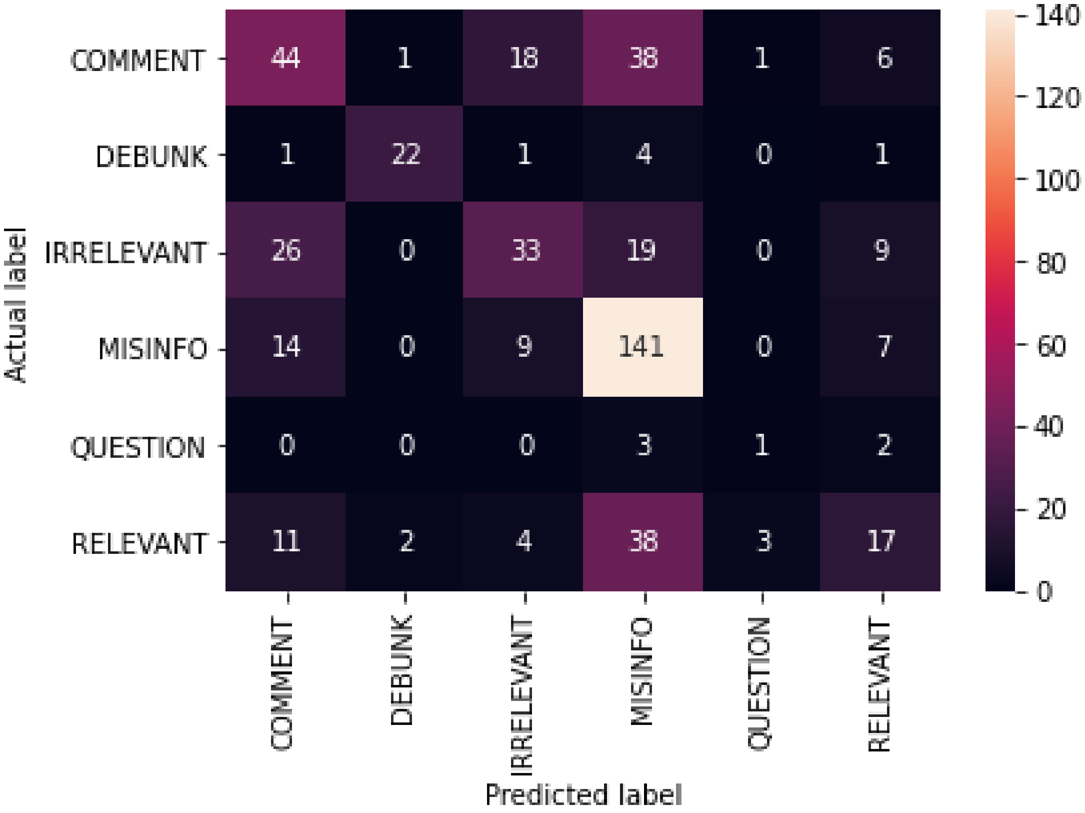


Figure 4.1: *Confusion matrix*

Metric	**Score**
Accuracy	0.54
Misinfo Precision	0.58
Misinfo Recall	0.82
Misinfo F1	0.76
Debunk Precision	0.88
Debunk Recall	0.76
Debunk F1	0.81
Misinfo/Debunk F1	0.75
Macro F1	0.47

Table 4.1: *Classifier Scores*

4.4 Translating to Gephi

NetworkX provides a write to GEXF (the Gephi data format) function: once the nodes and edges between them are built into a NetworkX Graph object, these can be converted to columns and rows of tabular data, ready to be visualised in Gephi.

Gephi is a rich open source system for visualising graphs. It provides a number of visualisation algorithms built in such as ForceAtlas2 [21], an algorithm designed for Gephi which is based on the simple principle of nodes repulsing each other and edges attracting their nodes. Other algorithms are provided such as Yifan Hu [20], which use similar principles. The comparison of these is described further below.

Gephi also comes with built in network property calculation. It automatically returns the number of nodes and edges within a loaded network: other network properties can be calculated on demand, such as average node degree or path length. Graphs of metrics like modularity are also provided for common tasks like community detection. These properties will be further explored in the results section, when the graphs are built.

Chapter 5

Implementation

This section will describe the implementation of the design ideas proposed in the previous chapter. It will recount the development process, with specific hurdles encountered along the way. It will describe each part of the finished implementation in detail.

5.1 Initial challenges

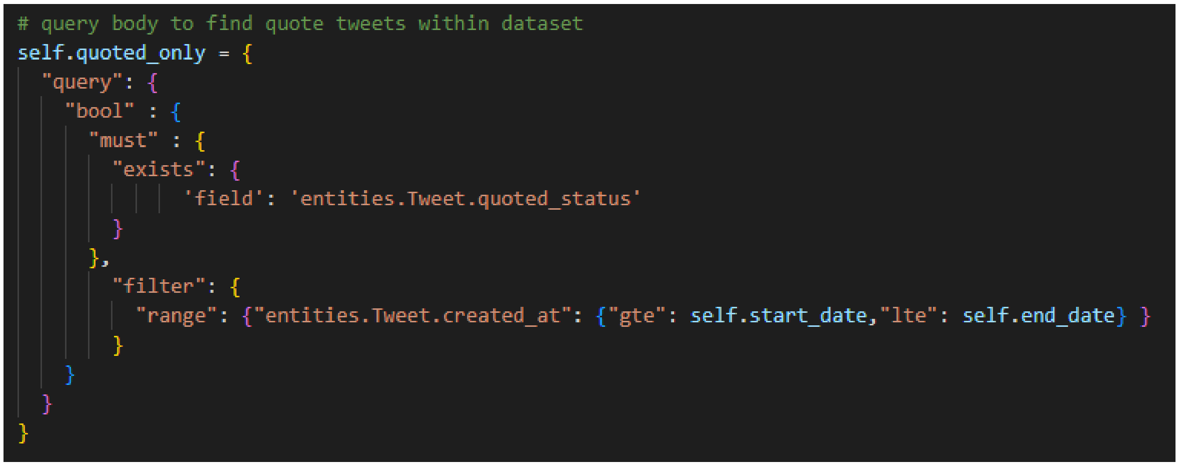
Initial challenges included gaining experience with Elasticsearch, figuring out twitter object datastructures, and making use of REST APIs.

5.2 Accessing the Elasticsearch cluster

The Elasticsearch cluster is hosted by The University of Sheffield, and had been gathered for use in a previous paper, and for further research purposes [24]. The cluster required authentication details which were provided by Carolina Scarton: the Elasticsearch cluster uses SSL encryption for security, and authentication into HTTP headers is sufficient for access. The cluster hosted by the university uses Elasticsearch version 7.13.4 : it was important to consider this as the newest available release is version 8.2.0 (as of 11/05/2022), so some features present in later docs were not available, and some syntax had changed. Since the cluster is hosted internally at the university, a VPN into the university network was also required for access.

5.2.1 Elasticsearch structure and API

The Elasticsearch architecture consists of clusters of nodes, where nodes are servers containing indexed data. The cluster contains nodes which store data, and nodes which serve other



```
# query body to find quote tweets within dataset
self.quoted_only = {
  "query": {
    "bool" : {
      "must" : {
        "exists": {
              'field': 'entities.Tweet.quoted_status'
        }
      },
        "filter": {
          "range": {"entities.Tweet.created_at": {"gte": self.start_date,"lte": self.end_date} }
        }
    }
  }
}
```

Figure 5.1: *Elasticsearch Query DSL to match quote tweets with range filter*

purposes, like the master node for controlling other nodes in the cluster [38]. This architecture is abstracted down for access by Elasticsearch's REST APIs. These allow use of typical RESTful queries like GET and POST to access or write to the Elasticsearch cluster.

Important to note is the indexing techniques used in Elasticsearch: the Elasticsearch cluster can contain many indexes, for storing different loads of data. When querying the cluster the index must be specified, as otherwise the intended target of the search may not be found. The university cluster contains multiple indices for each month of data collected : one for each datastream, COVID19ALL and COVID19MISINFO. With the data ranging from March to September, this gives 14 indices in total. The individual records in the indices are known as documents, and the categories like tweet_id as fields.

5.2.2 Elasticsearch queries

Elasticsearch has a rich querying DSL (Domain Specific Language) to construct complex and comprehensive queries to be passed into the cluster. The DSL can be thought of like an abstract syntax tree, where queries can be made up of sub queries, and can contain multiple types of conditions and filters [1]. Leaf queries will find specific terms in fields according to conditions like ”match” or ”range” for timestamps and integers, whilst compound queries build leaf queries together: conditions like ”bool” give logical boolean operators over multiple queries. Elasticsearch is capable of handling more expensive queries: fuzzy queries allow slight term differences (fox matches box), regexp queries allow use of regular expressions, and wildcards can also be implemented into queries.

Queries primarily used in this application include boolean compound queries, where two matching terms were required for a match: the query shown in fig 5.1 matches documents where the field quoted_status exists with the leaf query ”exists”, but also makes use of the filter context to retrieve only tweets that match the field created_at in a specified range. The ”bool” context enables both of these conditions to apply to the same query.

5.2.3 Using Kibana

The Elasticsearch stack is the environment of analytical tools provided by elastic.co for use with Elasticsearch clusters. It consists of: Elasticsearch, the search engine itself; Logstash, a server-side tool for collecting data and processing; and Kibana, a data analytics tool for visualising large indices, pulling statistics, and creating graphs [2].

Kibana is a useful tool for investigating the documents stored on the Elasticsearch cluster. Since this datastream was not initially familiar, Kibana provided useful opportunities: it displays all of the available fields in the datastream. The ability to view all of these fields proved useful when building queries, and investigating structure, as fields are displayed as hierarchical structures : "Tweet.quoted_status.created_at" shows exactly how to access the created_at field, since the JSON returned into python queries uses dictionaries which follow this hierarchy. These fields can be used as axes on a graph: two fields could be mapped against each other, such as created_at as time on the x axis, and count of records on the y axis (see 6.1 for example). Pie charts are also useful for showing make-up of textual data: see fig 3.1, which was built with Kibana.

5.3 Python Implementation and Testing

The development process began with an initial application to test Elasticsearch querying. This then evolved to become the main project, which consists of a main class containing the methods that pull and process the tweets, and a run file to specify query options like date range and query size. The version of python used was 3.9.1.

5.3.1 Packages

The main packages used were the Elasticsearch python package, and the NetworkX package. Also used was a native python package "os", which was required to access environment variables storing server authentication details. The versions used were elasticsearch 7.14.0, and networkx 2.6.3. The elasticsearch python version, as mentioned above, must match the version of Elasticsearch on the cluster: newer versions of the python package are available, but cannot be used.

5.3.2 Elasticsearch Queries implemented in Python

The elasticsearch package contains a catch-all .search function under the search API, which takes an index and query body as arguments: size of the result can also be changed as an optional argument. The parameter query body can be defined beforehand, which is useful for code readability. This function will return a nested dictionary, "hits", which will contain all of the documents found that match the query as JSON objects.

Limitations

Whilst experimenting with queries, the COVID19ALL stream was tested: due to the largeness of this dataset, an error was encountered. The return size limit of queries is 10,000, and often periods were selected from the datastream that matched more than 10,000 tweets. This issue was a worry to begin with, as it could potentially affect completeness of results: what if some important quote tweets were missed out of the network when being built, and insightful structures were lost? There are other APIs to retrieve more than 10,000 results with Elasticsearch, such as the scroll API, which paginates results indefinitely. This method could have been used, however processing 10,000 results with attributes attached was already computationally intensive, so it was decided to limit the scope with this size limit.

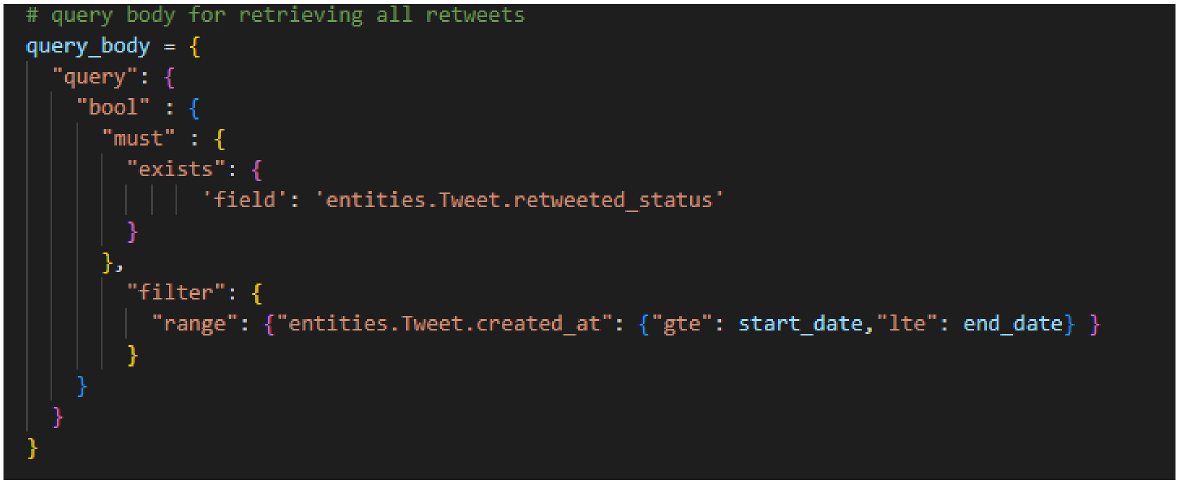
5.4 Building a network in Python with NetworkX

The main python class defines a StoreRetweets object to handle all processing on the tweets. The class attributes contain a NetworkX DiGraph (directional graph), to store the graphical representation of the datastreams. This DiGraph object has directional edges, so can represent the relationship between an original and a quote tweet. The attributes also contain the start and end timestamp for the section being pulled, as passed in when a StoreRetweets object is constructed.

The main method uses an Elasticsearch DSL query to match all tweets with the quoted_status field present, filtering only the Tweets that fall within the timestamp range previously specified. When this query is returned, all of the quote tweets are iterated through. For each quote tweet, fields in the tweet body are lifted to fill out the network: firstly, the quote id and quote tweet body are lifted, then the original id and original body, which can be accessed within the quote tweet body. Then additional attributes are captured: tweet text, hashtag, and calculated retweets. Classifiers use the tweet text to return further attributes, as explained further below. These attributes are combined when NetworkX DiGraph's add_node method is called, which takes the first id as the new node, a second id as the node to build the edge towards, and the attributes to be added to the new node. Some nodes do not contain all of the attributes specified, so these are replaced with null values when adding new nodes. Further still, any node which is added through other routines which does not fall within the timestamps specified is labelled with "Outside DB", to limit the amount of data processed. Finally, the DiGraph is written as a GEXF file for use in Gephi.

5.4.1 Calculated Retweets

Retweets are calculated separately: a different function defines a new Elasticsearch DSL query (fig. 5.2), with the "exists" leaf changed to "retweeted_status". The function collects all retweets in this manner, and iterates through them. It finds the tweet id of each originally retweeted tweet, and records how many times this id appears, effectively counting the retweets



```
# query body for retrieving all retweets
query_body = {
  "query": {
    "bool" : {
      "must" : {
        "exists": {
              'field': 'entities.Tweet.retweeted_status'
        }
      },
        "filter": {
          "range": {"entities.Tweet.created_at": {"gte": start_date,"lte": end_date} }
        }
    }
  }
}
```

Figure 5.2: *Elasticsearch Query DSL to match retweets with range filter*

for each unique id. Since this query uses the same date range, it finds the total retweets for each id at the end of that date range, as it pulls all retweets. This function works well enough to fill out attributes, but is limited by the way it only counts retweets that have also been collected by the datastream.

5.4.2 WeVerify Classifier

The WeVerify classifier is provided by the GATE Platform from TUOS [4]. This platform is a ”Text-Analytics-as-a-Service” site, where many useful classifiers can be used on demand through APIs. A REST API is provided for the WeVerify COVID-19 Claim Categoriser, where an API key is provided for authentication. The API simply takes a text string and returns a class.

Implementation was straightforward, as the text for each tweet had already been gathered. The sample text was fed by POST into the API, and the resulting label was attached to each node.

5.4.3 Collaborative classifier

The classifier provided by Jacob Crawley proved more of a challenge to implement. The classifier is built in python, so can be stored in the same directory as the other files. It is designed to work on batches of CSV twitter data, with tweet bodies provided in a large list: then the classifier will assign a tweet label with the class determined to each CSV row.

This architecture is initially incompatible with this project’s methodology where quote tweets are iterated through one at a time, not processed all at once. Modifications were made to the main loop to record the text of each tweet at processing into a pandas dataframe

[8], and the classifier was slightly adjusted to accept this new datatype. When all tweets were classified, the IDs and labels were stored corresponding together in a python dictionary, which NetworkX can accept and add attributes to those existing nodes.

5.4.4 Testing and Development Techniques

Since the project produces no ”correct” output, test strategies evolved dynamically throughout the project, and the main application has been thoroughly tested and improved throughout its lifetime.

Development of the project used Visual Studio Code, an open source IDE. The IDE provides highlighting features for Python, and auto code completion. It has a rich ecosystem of extensions that provide useful additions such as bracket highlighting, and source control for Git repositories.

The main testing tools used during development were the Visual Studio Code debug console for variable watch, and breakpoints to step through. Particular challenges encountered during development included the referencing of fields like ”tweet_text” within elasticsearch queries : kibana was used extensively to target appropriately sized sections of the tweet repository for analysis, and to cross reference field names. In addition, the in-IDE console included within Visual Studio Code was used to monitor network-building processes, as networks often took between 15 minutes to a few hours to build: it was useful to track progress of network-building.

5.5 Gephi Implementation and Testing

5.5.1 Use with NetworkX

The completed NetworkX DiGraph contains all the information needed by gephi to visualise the graph, but requires a certain format GEXF to read it in. NetworkX provides a function to convert the DiGraph into a GEXF file, and this worked well.

5.5.2 Visualisation algorithms

When the network graphs are first visualised by Gephi, the initial result is an unorganised square of nodes and edges (fig 5.3). Gephi provides visualisation algorithms that re-organise the nodes and edges to group similar nodes, and show communities.

Here are a few examples of some of the visualisation algorithms that were used with Gephi to create graphs. Initial experimentation was done with a few of the algorithms, Yifan Hu (fig. 5.4a) and OpenOrd [27] (fig. 5.4b) for example: Yifan Hu proved unsuccessful in extracting

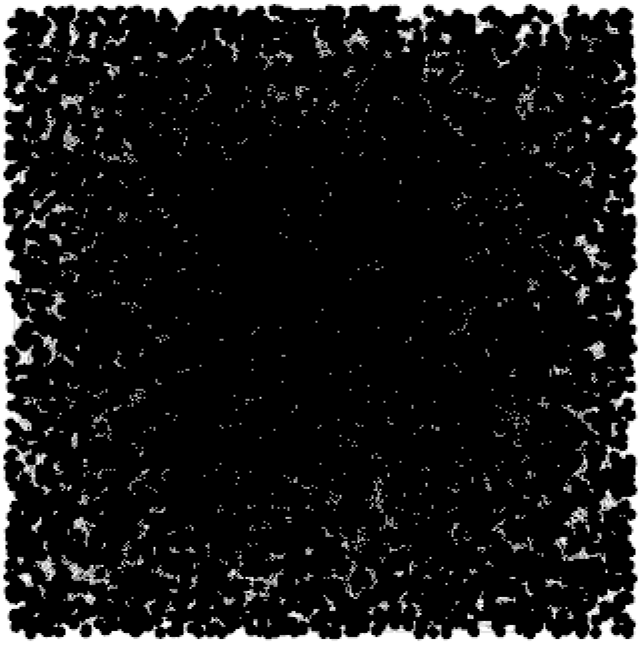
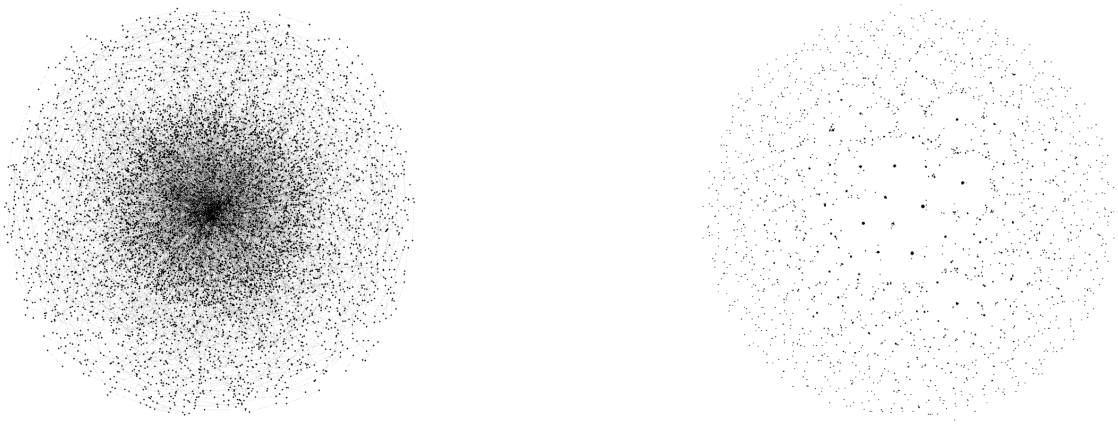


Figure 5.3: *Unstructured nodes as first loaded*



(a) *Yifan Hu applied on unstructured*

(b) *OpenOrd applied on unstructured*

Figure 5.4: *Yifan Hu and OpenOrd*

any community structure, but OpenOrd fared better, producing distinct groups. However, OpenOrd took considerable time to complete: ForceAtlas2 produced the same structure more quickly.

ForceAtlas2

ForceAtlas2 [22] (fig 5.5) proved to be the most useful for visualisation, as it organised the nodes into communities quickly and discretely. Groups of nodes in the results tended to be grouped in separate communities, with no linking edges. This property allows ForceAtlas2 to separate the communities immediately by pulling nodes together, with edges repelling in a spring-like fashion. The force-directed algorithm simulates a physical system, using two principles: "repulsion formulas of electrically charged particles" and "attraction formulas of springs". It also takes into account the degree of each node in the repulsion force, bringing weakly connected nodes to larger connected ones: this separates distinct groups efficiently. The parameters of the system (repulsion and gravity) could be adjusted, and optimal values of 1.5 and 1.5 were found to keep graphs from expanding too far outwards. However, after this algorithm had organised the nodes, they were not visible as individual points.

Noverlap

Noverlap is an anti collision algorithm that uses a repulsion force to separate nodes until none are overlapping (fig. 5.6). When applied to the result of the ForceAtlas2 algorithm, it produces readable graphs with distinct nodes. These graphs produced with both ForceAtlas2 and Noverlap proved to be the most readable and were used for the results displayed below.

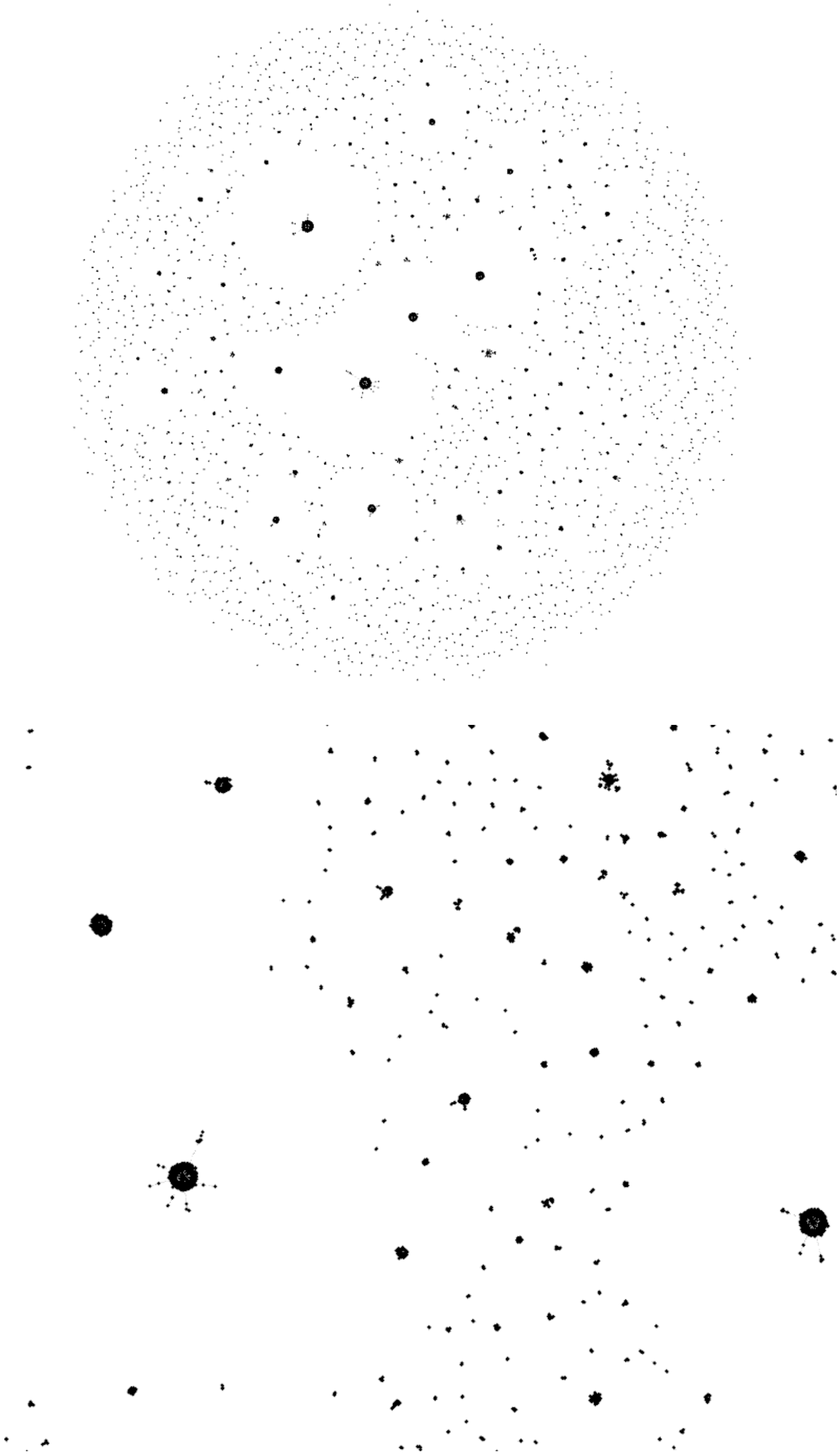


Figure 5.5: *ForceAtlas2 applied on unstructured data with zoomed subsection*

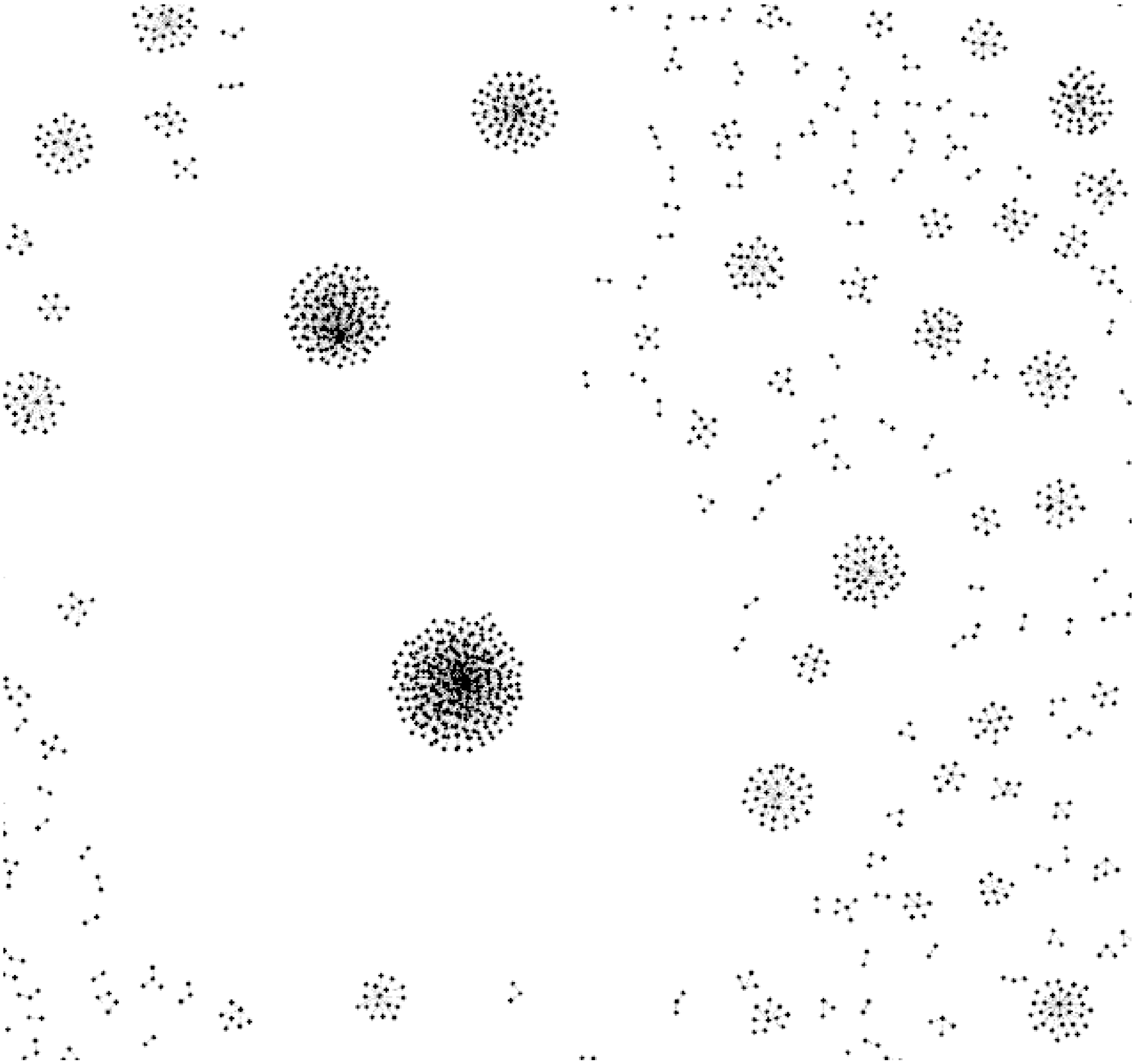


Figure 5.6: *Noverlap pass over ForceAtlas2 result*

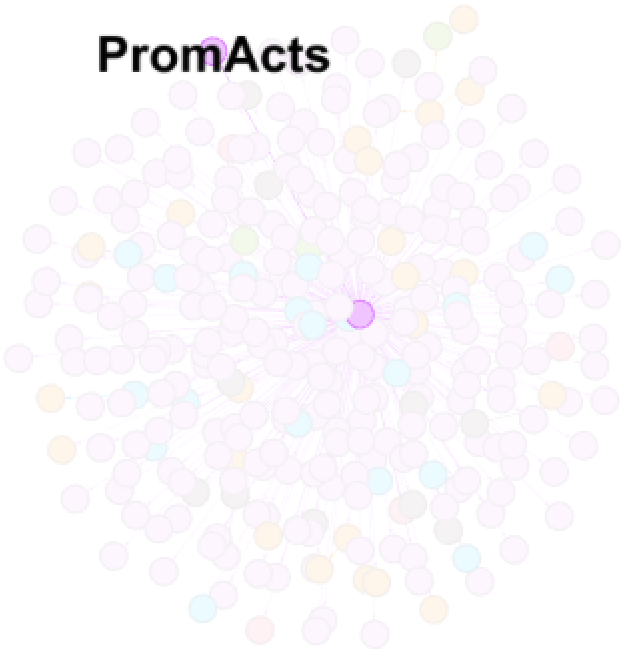


Figure 5.7: *Tweet label overlaid onto Gephi*

5.5.3 Filtering, Seeing labels

Gephi provides a feature to see any label overlaid onto node data (fig 5.7). This feature enables us to view the tweet text, hashtag, or any other attribute of a node in the Gephi viewer. It is interesting to see how a tweet is classified (misinfo, debunk etc.) in the network based on its content. Gephi can also filter out nodes which don't meet certain conditions, such as having a degree less than or equal to one.

Chapter 6

Results

This chapter will set out some examples of visual results gained from the networks built upon tweets. It will also show network properties as calculated on those networks. It will show how these visualisations can inform on important trends in these datastreams, and the results of using two classifiers on these data.

6.1 Justifications

6.1.1 Data limit considerations

Due to the sheer size of the tweet repository, some prior research was required to determine on which time periods to gather results. For example, the COVID19MISINFO datastream in April 2020 gathered $\tilde{1}$.8M tweets in total from the 7th of April to the 30th. Per day, the count of tweets gathered ranged from 146k to 48k gathered. In addition, the number of tweets coming into the datastream varies over the day, shown in figure 6.1.

However, not all tweets are considered by the python application, as it is only those that are a quote tweet of others that are pulled down from elasticsearch for analysis. Kibana shows that on April 15th, the unique count of quote tweets is 3,596. This is under the maximum pull size for a single Elasticsearch query, which is 10,000 documents.

With this limitation, it makes sense to select the periods at which the tweets per unit time are the least, as they give the most time for a tweet to be replied to before reaching the 10k limit: this is with the aim in mind of producing the deep networks, with the idea they show more insight. With COVID19MISINFO, there are many full days in which less then 10,000 quote tweets are produced, so building a full network over the course of a day is feasible (see fig 6.2). There are a few exceptions to this, such as mid-May, which shows an explosion of quote tweets published.

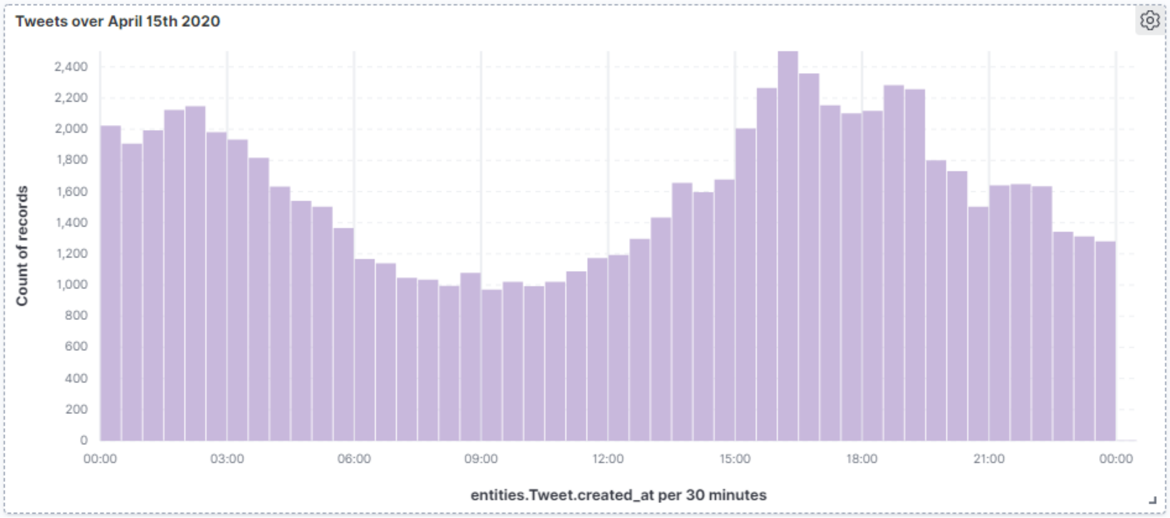


Figure 6.1: *MISINFO datastream count of Tweets created over April 15th*

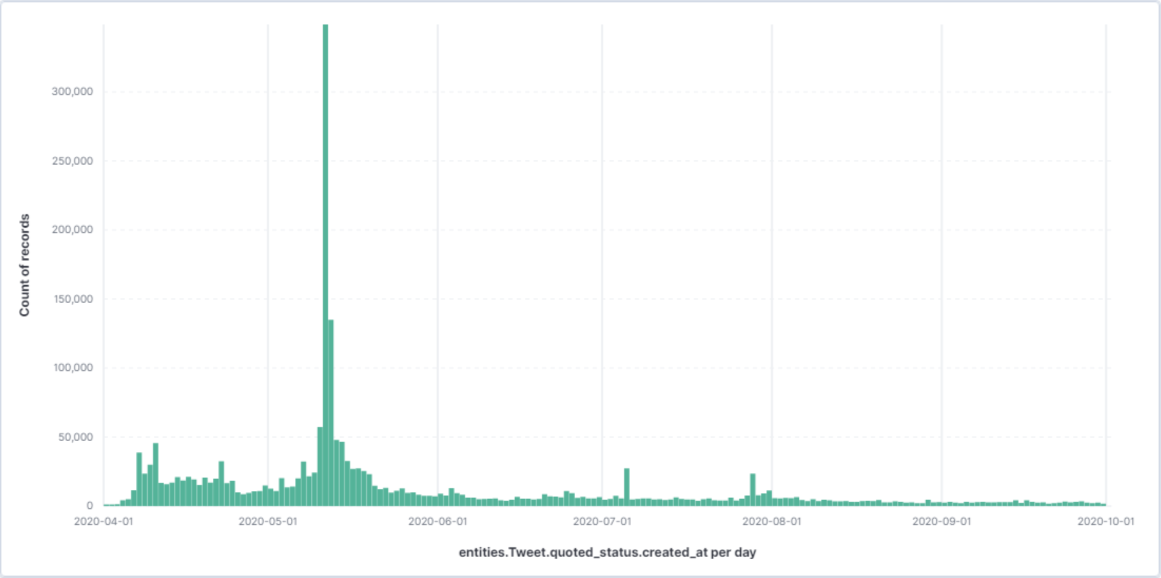


Figure 6.2: *MISINFO datastream tweets count over time*

However, COVID19ALL is a much larger datastream, with a range of 100k to 900k quote tweets posted per day (see figure 6.3). This volume of data causes issues in that the time window for gathering tweets needs to be carefully selected to pull total documents under 10k.

Aside: Elasticsearch Relevance scoring

The reasoning behind limiting the Elasticsearch query time window is to keep a complete view of the data. Elasticsearch queries rank documents by relevance score, so in this situation

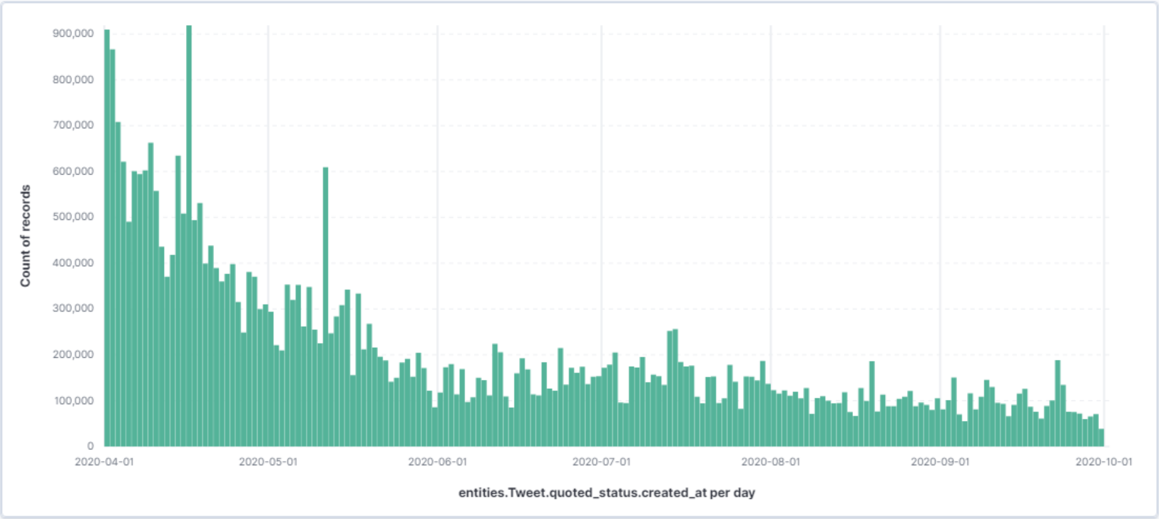


Figure 6.3: *ALL datastream quote tweets count over time*

where more than 10,000 documents are matched by an elasticsearch query, the documents returned will be a result of their relevance score, hence returned with a constant bias.

Figure 6.4 also displays the daily trend of tweets that peaks in the evening and is most reduced in the morning. For this reason, it is justifiable to select time periods for analysing COVID19ALL that fall in the morning. This aims to retain completeness of relevant data.

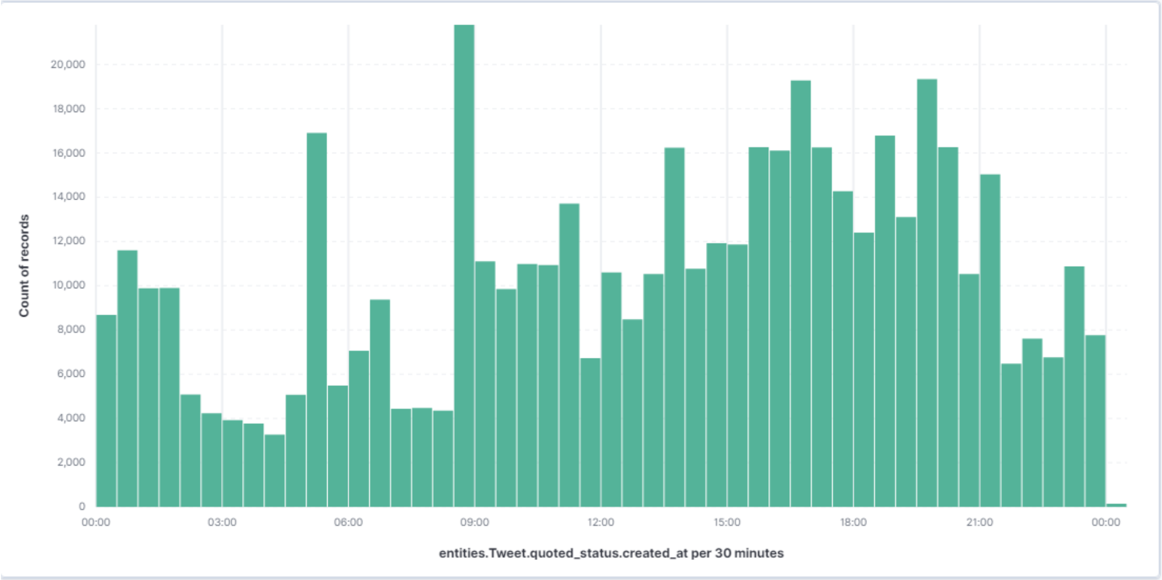


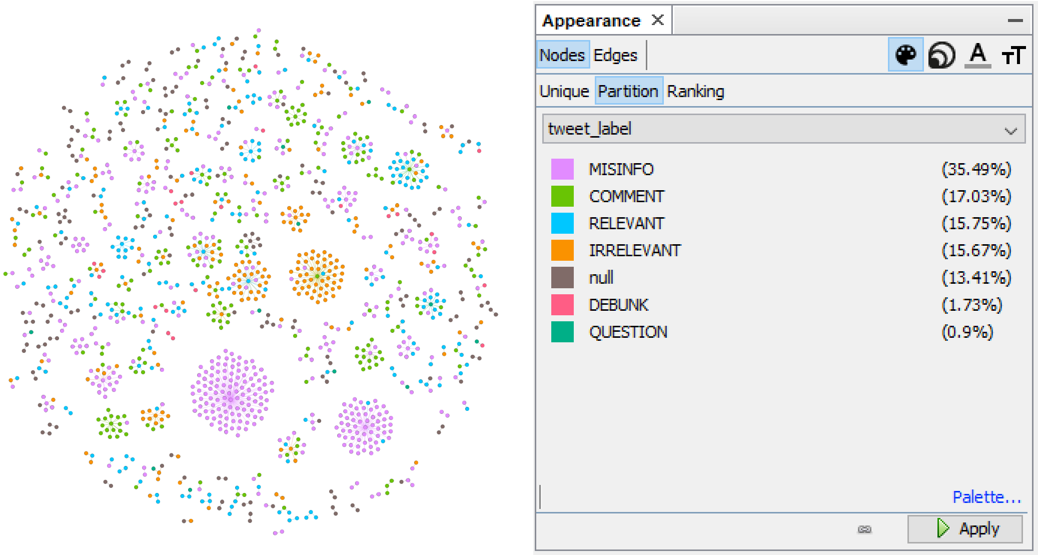
Figure 6.4: *ALL datastream quote tweets count over Apr 15 2020*

6.2 Gephi Examples

To demonstrate the visualisation results, here follows some examples of graphs built from the Twitter datastream from sample dates over the time period it was gathering tweets. These graphs are the result of building the network in Python from Elasticsearch Twitter data, using NetworkX to define nodes and edges, then translating to Gephi format to visualise within the software.

6.2.1 COVID19 MISINFO Examples

Visualisation of the MISINFO datastream was successful in showing networks with many attributes and connections. The following figure 6.5 shows an example of a subsection of the datastream gathered on Apr 15 2020 from 1600 to 1900. It consists of 1327 nodes with 1031 edges, with the largest out degree of 113. In this example, the dataset was visualised using the ForceAtlas2 [22] algorithm, then with a pass of Noverlap, an algorithm to space out leaf nodes from central ones. This subset of the datastream is selected as an initial example as it is what was used primarily for testing and development, and contains a small enough number of nodes to be exported as a PNG with visible detail.



tweet_label

🟪	MISINFO	(35.49%)
🟩	COMMENT	(17.03%)
🟦	RELEVANT	(15.75%)
🟧	IRRELEVANT	(15.67%)
🟫	null	(13.41%)
🟥	DEBUNK	(1.73%)
🟩	QUESTION	(0.9%)

Figure 6.5: *MISINFO datastream visualised in Gephi on Apr 15 2020 from 1600 to 1900*

This graph and others that follow are produced from the Gephi export feature, which takes a snapshot of the network coloured according to user preference. In this example, the nodes are coloured according to tweet_label, an attribute gained from the misinfo classifier provided by Jacob Crawley.

The graph contains many groups of nodes: the largest groups are often a single original tweet, with many quote tweets surrounding it. In this example, most of the nodes are in small star topologies with only 2-5 tweets, each with a central original tweet: 92.6% of nodes have an out-degree less than or equal to 2. The average degree is 0.777, showing most nodes have less than 1 outwards edge, ending the cascade of tweets as 'leaf nodes'. The average directed path length is 1.084, showing how most paths through the this network remain at near one, perhaps representative of how many sub graphs are original tweets with one layer of quote tweets one directed edge away. The network diameter is 3. This trend is particularly evident in this graph: following graphs show some deeper networks. There are a few 'giant' components, tweets which have stimulated a particularly large amount of quote tweets within the network. Since the application only gathers tweets which are a quote tweet, there are no single isolated nodes. This example is coloured according to the misinfo/debunk classifier, and these large groups of nodes often appear homogeneous. This initial example introduces a few key trends to be further explored - limited depths of networks, short path lengths, and large isolated groups with similar characteristics. The network property subsection will explore these themes of community structure.

6.2.2 COVID19 ALL Examples

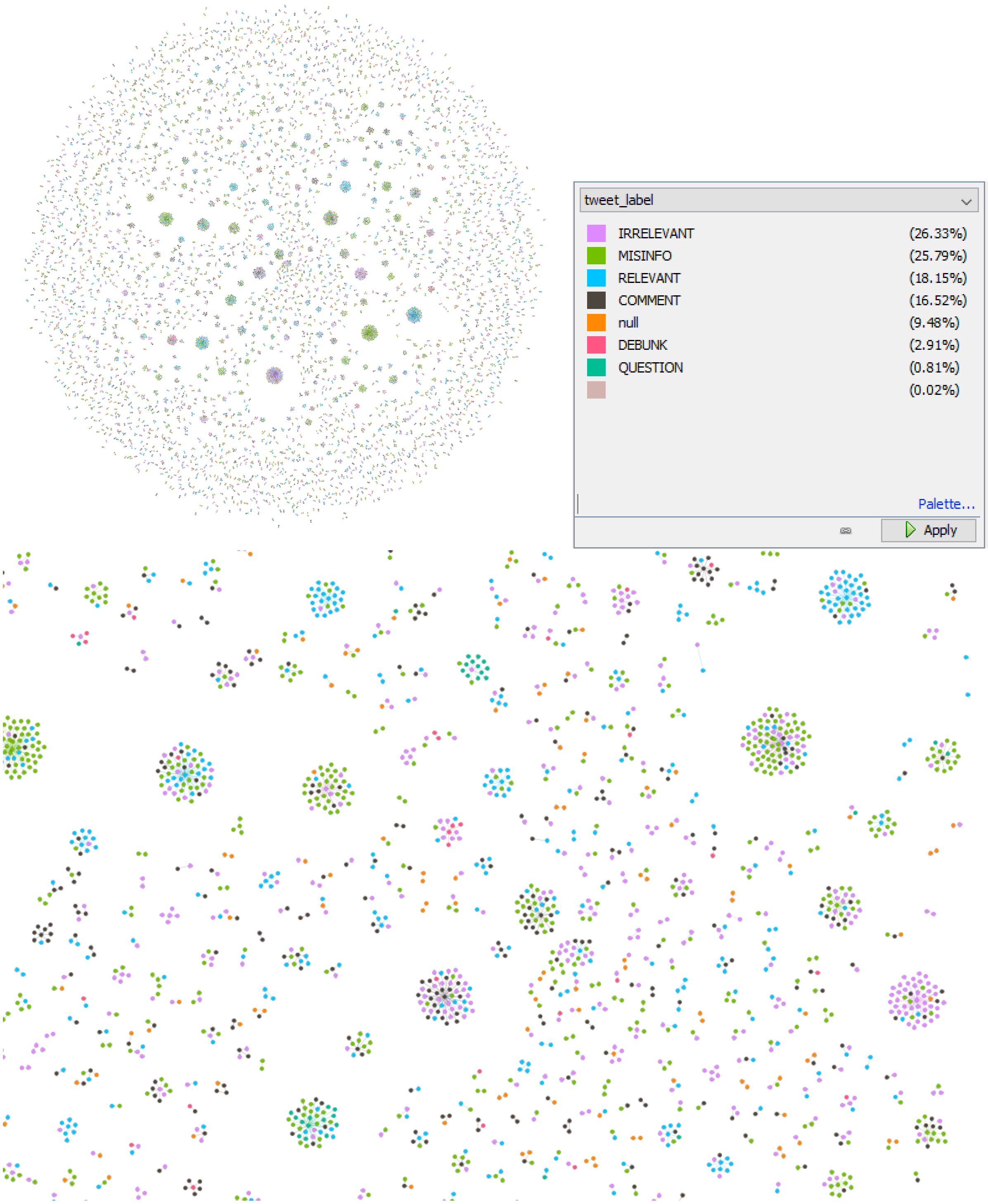
The ALL datastream is much larger than the MISINFO datastream, so the time range for gathering data had to be cut down to fall within the 10,000 query size requirement. The following is an example gathered on April 15th 2020 from 0600-0800 GMT. The network consists of 10789 nodes and 7232 edges. A subsection is provided here to show more detail, as the overview is too large to see individual clusters (fig 6.6).

This example exhibits some differences between the MISINFO dataset. A significant difference is the depth of the network: the network diameter computed by Gephi is 7, indicating that the largest path between two nodes is 7 edges. This particular path is shown in fig 6.7, where the network has built a long tweet chain, indicative of more conversational properties than a single original tweet with surrounding one level quote tweets. The property eccentricity, the maximum distance from one node to another, is shown to be high at the end of this path.

Another graph built on the same ALL datastream used the time range 1600-1900 showed further interesting node patterns. This graph contains 11098 nodes and 7545 edges. A pattern exhibited in fig 6.8 showed a large component with an outlying sub - group, indicating that these giant components are not always simple.

6.2.3 Drawbacks of visualisations

These visualisation results provide a large amount of data to look at: Gephi allows investigation of individual communities by their labels as described in Chapter 5. However, these visualisations reach up to 10k tweet nodes: individual investigation of groups of nodes can be time consuming. In addition, these visualisations contain many groups of nodes which



🟪	IRRELEVANT	(26.33%)
🟩	MISINFO	(25.79%)
🟦	RELEVANT	(18.15%)
⬛	COMMENT	(16.52%)
🟧	null	(9.48%)
🟥	DEBUNK	(2.91%)
🟩	QUESTION	(0.81%)
🟫		(0.02%)

Figure 6.6: *ALL datastream visualised in Gephi on Apr 15 2020 from 0600 to 0800, with subsection for detail*

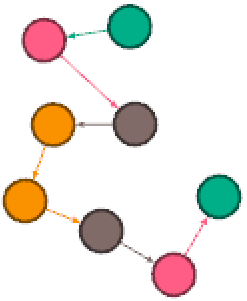


Figure 6.7: *Group of nodes coloured by eccentricity in COVID19ALL*

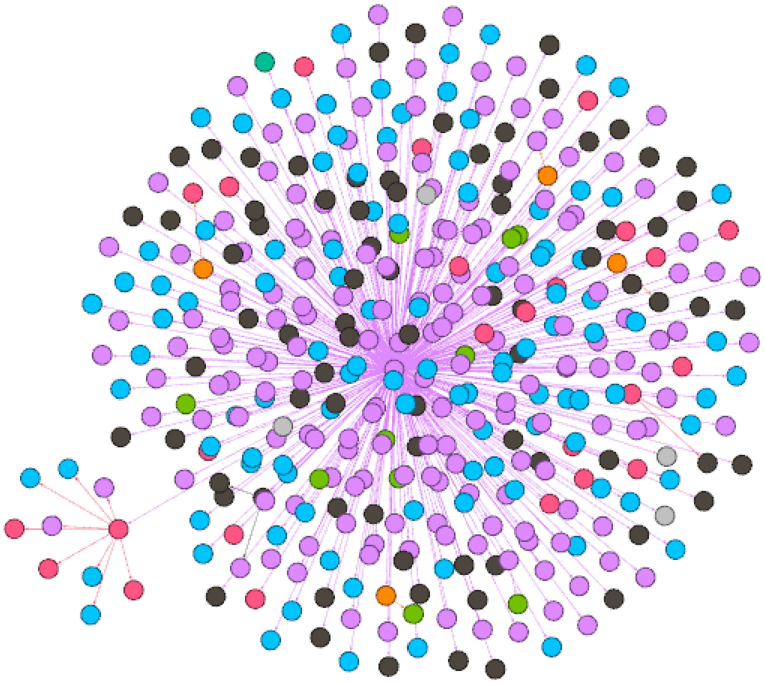


Figure 6.8: *Group of nodes with outlying sub group in COVID19ALL*

contain only a few nodes: these groups fill up space on the visualisation, where more filled space means less focus on larger groups of nodes from a purely visual standpoint.

6.3 Graph studies and Network Properties

The network metrics that can be calculated with Gephi are useful to analyze these large graphs without manually searching for interesting groups of nodes. Indeed, it is difficult to manually search each network for differences / trends, so these metrics prove useful in extracting those differences. A number of networks from different points in the datastreams were gathered, in order to calculate network properties and identify potential trends/differences between the datastreams.

6.3.1 Network property overview

It is useful to recap some of the network properties available on Gephi, in order to assess their meaningfulness in our context. It is important to note again that the graph is directed, and also that there are no graph cycles in the network. This is because of the logical way in which the network is built: there could not be a quote tweet pointing back towards an original node, because that node was created before it, and directional edges represent original pointing to quoted tweet.

This table represents the results of the collection of 12 networks built over both the MISINFO and ALL datastreams. The MISINFO networks are built on a full day from the months of April to September, whilst the ALL datasets are built from 06-00 to 08-00 and 12-00 to 14-00 on the months of April, May, and June.

Degree represents the number of edges connected to a node. It can be specified as In-degree or Out-degree, where graphs are directed, and edges coming in to a node will be counted as in-degree, and vice versa. In the calculation in Gephi, the average degree takes into account out-directed-edges, so is based on the average number of edges leading out of a node.

Paths are sequences of edges through nodes : with these directional networks, long paths represent a chain of quote retweets. The average path metric takes all directional paths in the network and find the average length.

Diameter of the network represents the shortest path between the two most distant pair of nodes. In this directional context, where most nodes are grouped together in communities, and no part of the graph is cyclic, it shows the longest edge path between two nodes. If this group (fig. 6.7) was considered on its own, the network diameter would be 7.

	Subsection: all							
Time	Date	Nodes	Edges	Avg degree	Avg Path	Diameter	Communities	Modularity
0600-0800	Apr-15	10789	7232	0.67	1.075	7	3557	0.998
1200-1400	Apr-15	11297	7564	0.67	1.075	3	3733	0.998
0600-0800	May-15	9980	7201	0.722	1.078	4	2779	0.993
1200-1400	May-15	10725	7304	0.681	1.086	5	3421	0.993
0600-0800	Jun-15	9023	7776	0.862	1.039	3	1247	0.812
1200-1400	Jun-15	9335	7333	0.786	1.056	4	2002	0.951
Average		**10191**	**7402**	**0.732**	**1.068**	**4.3**	**2790**	**0.958**

	Subsection: misinfo							
0000-2359	Apr-15	3248	2877	0.886	1.103	3	371	0.961
0000-2359	May-15	7025	5318	0.757	1.099	3	1707	0.99
0000-2359	Jun-15	4375	3380	0.773	1.104	4	998	0.975
0000-2359	Jul-15	3655	2628	0.719	1.112	3	1027	0.982
0000-2359	Aug-15	2552	1930	0.756	1.131	4	622	0.967
0000-2359	Sep-15	1856	1342	0.723	1.151	3	514	0.956
Average		**3785**	**2913**	**0.769**	**1.117**	**3**	**873**	**0.972**

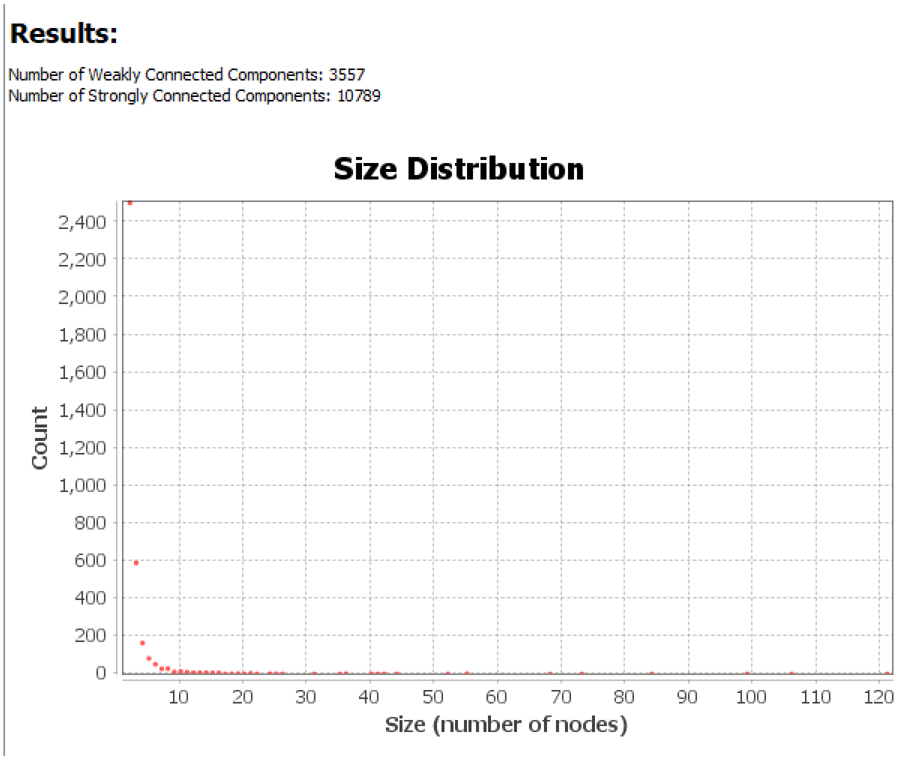
Table 6.1: *Network properties for both COVID19ALL and COVID19MISINFO graphs*

6.3.2 Communities and clustering

Strongly and Weakly connected components

Strongly connected components are portions of network graphs where each vertex has a path to another vertex. Weakly connected components are directed sections of the network graphs which are unreachable from other nodes of the graph. Gephi can calculate the number of both strongly and weakly connected components, but when it was ran on these graphs, the number of strongly connected components were calculated as the same as number of nodes, and weakly connected components usually the same as number of communities.

The example figure 6.9 shows the graph output from Gephi when running the 'Connected components' metric. The graph shows the count of weakly connected components at a certain node size : the graph exhibits a logarithmic pattern, with the many weakly connected components containing a small number of nodes. This corresponds visually to the many small degree components surrounding a few giant components found in the examples above. When this routine is run across other examples, similar graphs are produced.



Results:

Number of Weakly Connected Components: 3557
Number of Strongly Connected Components: 10789

Figure 6.9: *Running 'Connected components' on ALL graph, Apr15 0600-0800*

Communities

Communities are an important concept in analysing these graphical results. They are densely connected groups of nodes, which are weakly connected to the other groups surrounding them. This definition is not as strict as weakly connected components, however the two become analogous as mentioned because of the way the graph is built: there are very few instances where two similarly sized groups of nodes are connected by an edge.

The Gephi algorithm partitions the set of nodes into communities, and returns an output of the total communities across the current dataset. On average, the ALL datasets had 2790 communities, whilst the MISINFO datasets had 873. This is roughly the same proportionally to the amount of nodes in the dataset: for ALL, average nodes / average communities = 0.273, whilst for misinfo the proportion of nodes to communities is 1: 0.231.

6.4 Enrichment of tweets with extra attributes

Gephi provides multiple ways of visualising attributes on nodes. The primary method is through colouring of the nodes: they are partitioned with a colour palette, according to their class type: classes which appear the most infrequently to the point of insignificance are often put together as a single grey class. This feature also returns the percentage of nodes as each attribute across the whole network. An example that would work well would be one of the classifier's attributes: various colours are assigned to debunk/misinfo/irrelevant, for example. Another Gephi feature is to size the node according to an integer attribute. This works well for the calculated retweets, as they are a pure integer value.

6.4.1 Calculated Retweets

The retweets count attribute is visualised here by sizing the node respective to the count value, between a range. Below in fig 6.10 is an example of a graph that has had its nodes sized according to retweet count. As expected, the size of a node is typically larger the more quote tweets that are connected to it, indicating twitter engagement is much higher on nodes that have more conversations surrounding them.

6.4.2 WeVerify Annotation

The coronavirus topics classifier can be overlaid onto the network graphs with the colouring feature in Gephi. This node colouring feature also provides the percentage of nodes in each class. The tables below represent the percentage of nodes classified as each class from the application.

These broad data shown in tables 6.2 and 6.3 are not necessarily related to the graphical

	percent classified, Datastream:all					
Time	**Date**	**PromActs**	**PubAuthAction**	**CommSpread**	**Consp**	**GenMedAdv**
0600-0800	Apr-15	26.3	19.8	17.5	13.0	5.74
1200-1400	Apr-15	25.6	18.9	18.4	14.1	6.19
0600-0800	May-15	29.7	16.5	17.8	14.0	7.08
1200-1400	May-15	26.7	18.1	16.3	13.0	6.35
0600-0800	Jun-15	48.3	9.8	23.4	6.04	3.80
1200-1400	Jun-15	37.5	14.7	18.1	10.2	5.82
Average		**32.3**	**16.3**	**18.6**	**11.7**	**5.83**
	Datastream: misinfo					
0000-2359	Apr-15	29.9	14.8	7.27	17.2	20.9
0000-2359	May-15	47.1	8.73	3.40	12.6	2.11
0000-2359	Jun-15	37.4	8.55	6.74	17.3	4.41
0000-2359	Jul-15	29.3	10.5	8.56	17.6	4.30
0000-2359	Aug-15	37.7	7.13	5.88	18.9	4.35
0000-2359	Sep-15	25.8	9.16	5.98	24.5	3.12
Average		**34.5**	**9.80**	**6.31**	**18.0**	**6.52**

Table 6.2: *WeVerify topic by percentage classified on dataset, pt. 1*

	Datastream: all						
Time	**Date**	**PubPrep**	**Vacc**	**VirTrans**	**VirOrgn**	**Outside DB**	**None**
0600-0800	Apr-15	3.15	1.33	0.73	0.48	3.22	6.68
1200-1400	Apr-15	2.99	1.21	0.71	0.59	3.03	6.65
0600-0800	May-15	3.10	1.90	0.54	0.33	2.76	4.77
1200-1400	May-15	3.25	2.21	0.77	0.40	3.55	7.02
0600-0800	Jun-15	1.51	0.85	0.71	0.59	1.30	2.66
1200-1400	Jun-15	3.48	1.41	0.82	0.24	1.95	4.36
Average		**2.92**	**1.49**	**0.71**	**0.44**	**2.64**	**5.36**
	Datastream : misinfo						
0000-2359	Apr-15	0.34	1.63	0.31	0.58	3.39	0.58
0000-2359	May-15	0.63	1.34	0.23	0.14	7.76	7.66
0000-2359	Jun-15	0.66	2.29	0.27	0.34	8.66	5.05
0000-2359	Jul-15	2.65	2.93	0.57	0.27	9.9	5.69
0000-2359	Aug-15	1.37	2.00	0.43	0.12	8.78	5.45
0000-2359	Sep-15	0.92	2.81	0.43	0.43	11.5	5.71
Average		**1.10**	**2.20**	**0.38**	**0.31**	**8.33**	**5.02**

Table 6.3: *WeVerify topic by percentage classified on dataset, pt. 2*

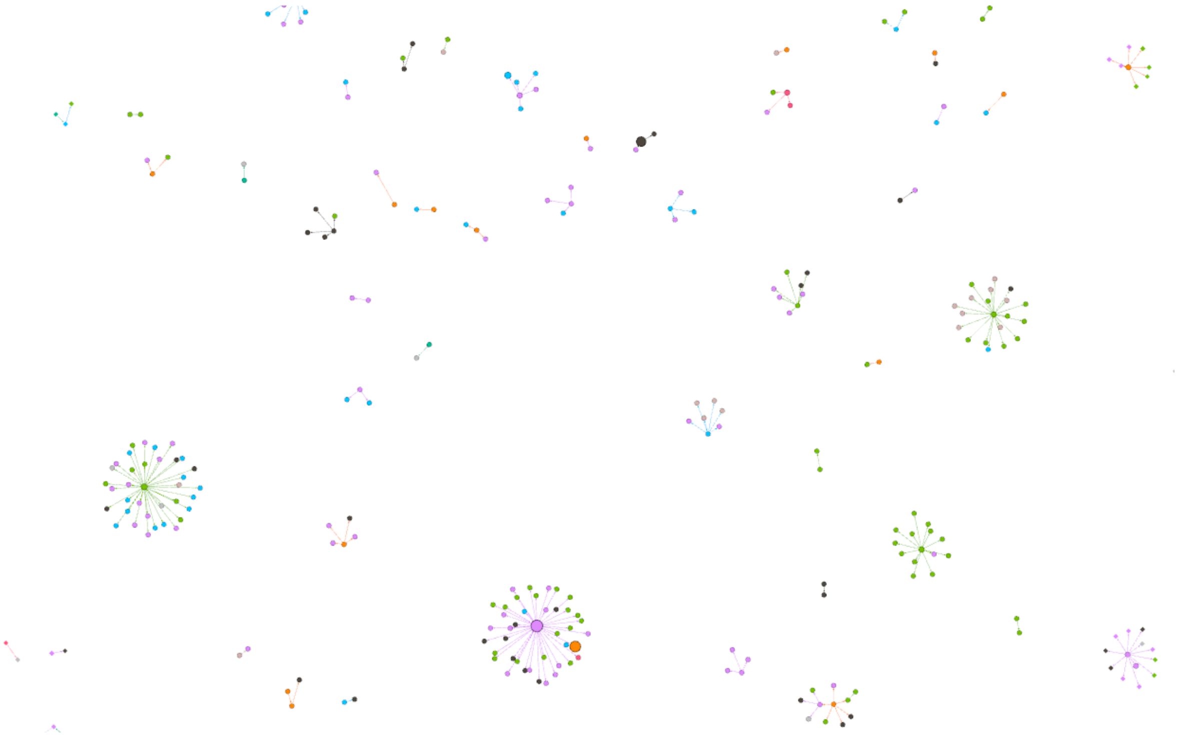


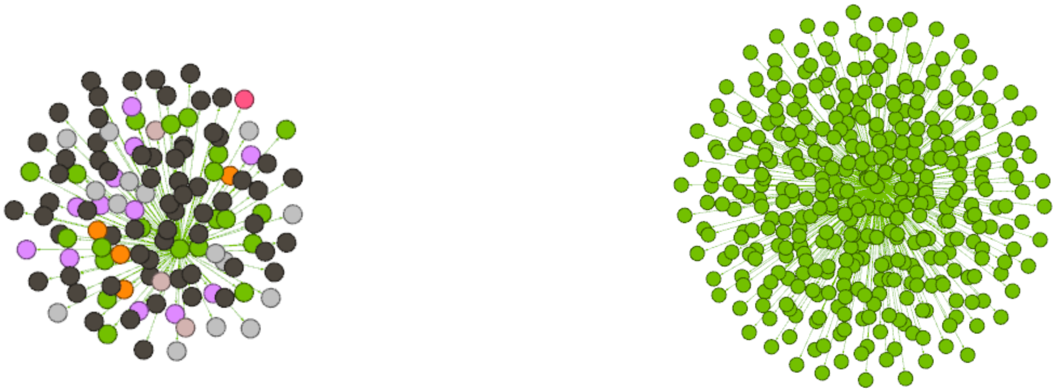
Figure 6.10: *Running size by calculated retweet on ALL graph, Apr15 0600-0800*

properties of each dataset , but it is useful to show an initial overview of the types of claims that are most prevalent in these datastreams. The sample shown in table 6.4 is broadly supported by these data, with some differences. This sample is gathered from misinformation articles, not Twitter posts, so this difference could be shown in comparison here. The largest classified topic was PromActs, at an average of 32.3% and 34.5% for both ALL and MISINFO. In the Reuters study this equated to 23% of their sample, a large proportion, but not PubAuthAction, which had 39% of the sample. The rest of the topics, such as VirTrans and VirOrgn, made up a very small section of the classified data: these topics were also at the bottom for Reuters' study, but not at 0.7% of the sample like the classifier. These results could imply that Twitter hosts more conversations on Prominent Actors, which include companies or famous people (celebrities, politicians): this would make sense given Twitter is a social media platform where these entities receive the most engagement, and would naturally generate the most conversations.

Making a comparison between the all and misinfo datasets is also interesting: the topic ”Consp”, a category classifying conspiracy theories, saw an increase from 11% to 18% percent from ALL to MISINFO. This confirms an immediate assumption about the type of claims present in the misinformation datastream, that it would be more likely to contain conspiracy theory claims than the all datastream.

Claim Topic	Percentage of Sample
Public Authority Action	39
Community Spread	24
General Medical	24
Prominent Actors	23
Conspiracy Theories	17
How virus transmits	16
Virus Origins	12
Public Preparedness	6
Vaccine Development	5

Table 6.4: *Percentage topics of sample containing types of claim: Reuters [16]*



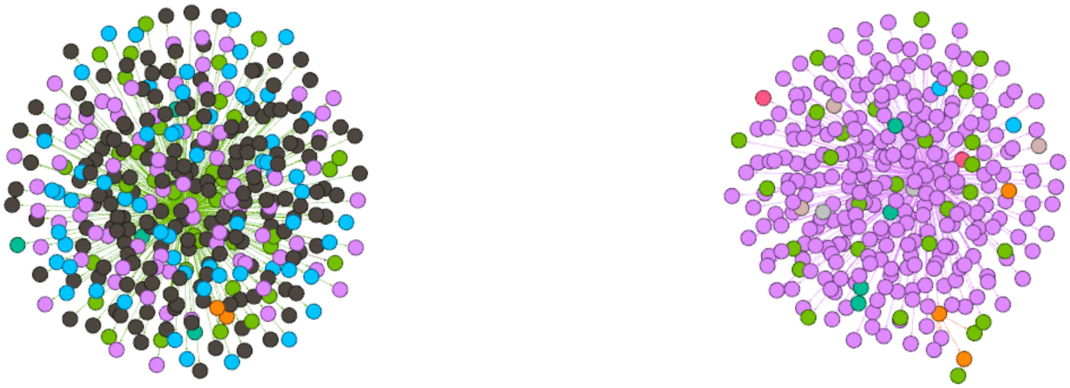
(a) *Apr 15 ALL giant component*

(b) *Apr 15 MISINFO giant component*

Figure 6.11: *Apr 15 giant component ALL vs MISINFO comparison, coloured with WeVerify*

Homogeneity in communities

This classifier takes tweet text from nodes and assigns the class as an attribute, so it would be valuable to investigate whether the topics it classifies are consistent within communities. Comparing the giant component (component with the largest amount of nodes) across a few datastreams, the classifier results in that component are often homogeneous but not completely, with a majority of one class making up a component. In both datastreams, the giant components are often well over 100 nodes large: in the example shown in 6.11a, the component is made up of mostly Conspiracy classified nodes (black), but there are also PromActs (Pink) and PubAuthAction (Green) featuring substantially. Conversely, the misinfo component in 6.11b, is completely homogeneous : the topic is GenMedAdv. This trend is continued in fig 6.12, where the ALL component is varied, and the misinfo component is much more homogeneous.



(a) *May 15 ALL giant component*

(b) *May 15 MISINFO giant component*

Figure 6.12: *May 15 giant component ALL vs MISINFO comparison, coloured with WeVerify*

6.4.3 Misinfo/Debunk Classifier

In a similar fashion to the WeVerify classifier, the misinfo/debunk classifier can be overlaid onto the data with the same node colouring feature, and the percentage of each class can be found.

The proportion of each class in each datastream, shown in table 6.5 informs some immediate conclusions. The misinfo datastream produced an average 32.8% MISINFO classifications, whilst the all datastream produced 25.4%. This shows that the misinfo datastream, at least according to this classifier, does indeed contain more misinformation than the main COVID-19 conversation on Twitter. Other differences include the amount of comment classified nodes, where datastreams all and misinfo had 32.1% and 23.3% classified respectively. Assuming the all datastream is a more general conversation on COVID, this informs a difference where the misinfo datastream spreads more misinformation, as opposed to merely a comment on a certain topic: misinformation more likely to cause more misinformation to propagate. Also to note is the high proportion of null values in the misinfo datastreams : when the application detects no text field available (if the tweet is outside the range of the query, or does not contain the correct field), it leaves null as the category. Investigating further, many of the null values are found in small node pairs outside of large communities: this could be indicative of an issue with the program, or of the scope of the queries used on elasticsearch. Fortunately these null values do not tend to appear in more significant components of the graph.

	Percentages as classified by misinfo classifier (3 s.f.) Subsection: all							
Time	Date	MISINFO	DEBUNK	COMMENT	QUESTION	REL	IRREL	null
0600-0800	Apr-15	25.8	2.91	16.6	0.81	18.2	26.3	9.48
1200-1400	Apr-15	22.7	1.95	27.2	1.17	12.4	25.7	8.94
0600-0800	May-15	26.5	2.14	34.8	1.03	11.2	17.4	6.87
1200-1400	May-15	31	0.98	18.4	1.66	22.4	16.7	8.85
0600-0800	Jun-15	14.3	1.64	58.2	0.270	9.79	12.7	3.16
1200-1400	Jun-15	31.8	0.470	37.2	0.300	12.2	13.2	4.83
Average		**25.4**	**1.68**	**32.1**	**0.873**	**14.4**	**18.7**	**7.02**
	Subsection: misinfo							
0000-2359	Apr-15	21.5	0.250	62.4	0.620	5.73	3.91	5.63
0000-2359	May-15	32.8	0.160	19.4	0.940	16.3	8.21	22.3
0000-2359	Jun-15	32.11	1.23	15.9	0.940	18	11.5	20.3
0000-2359	Jul-15	37.4	1.42	13.7	0.850	11	14	21.7
0000-2359	Aug-15	41.7	0.590	13.1	1.65	12.7	10.5	19.8
0000-2359	Sep-15	31.5	0.920	14.9	1.08	13.2	13.7	24.7
Average		**32.9**	**0.762**	**23.2**	**1.01**	**12.8**	**10.3**	**19.1**

Table 6.5: *Table representing misinfo/debunk classification percentages*

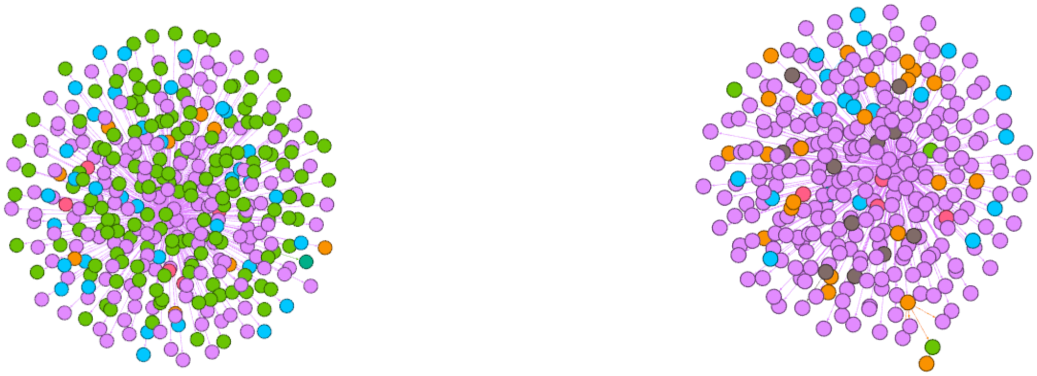
Homogeneity in communities

Similarly to the WeVerify classifier, here we examine the misinfo/debunk classifier make-up of communities of node groups. These large communities exhibit different levels of homogeneity, with figure 6.14a displaying a largely homogeneous community with the class MISINFO. 6.13a, 6.13b, and 6.14b all show a larger diversity in the node class present : both 6.13a and 6.14b indicate no overall class majority.

6.5 Implications and discussion of results

6.5.1 Implications of network property results

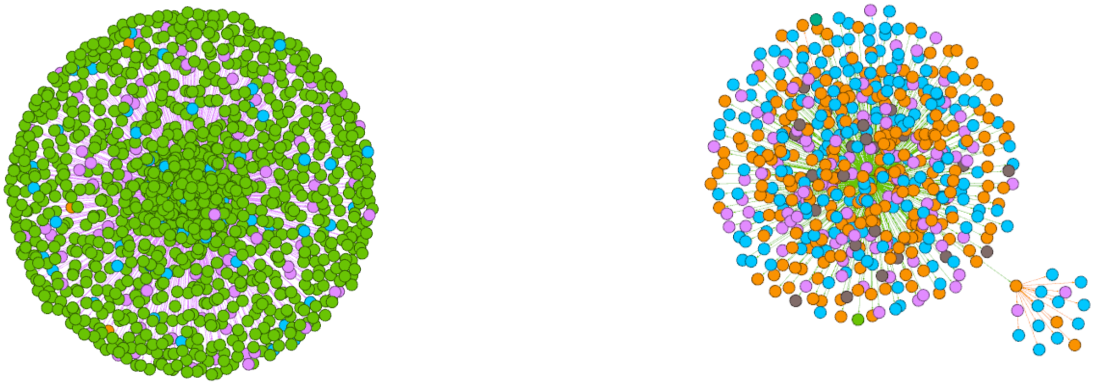
These network properties inform much about the nature of each dataset gathered, and at a quick glance give an overview of the size and make-up of any of them. The more informative metrics involved path length: both Avg Path Length and diameter. On average, the misinfo dataset had a higher average path length, but a lower network diameter. The higher average path could indicate that misinfo spreads more deeply into the twitter network - more quote nodes are shared again, creating longer paths to expand the network deeper. The lower network diameter could indicate fewer long chains of nodes in misinfo, which seems to refute the previous claim: however, since the all datastream samples are much larger than the misinfo samples, perhaps they are more likely to produce long node chains because of their size.



(a) *May 15 ALL giant component*

(b) *May 15 MISINFO giant component*

Figure 6.13: *May 15 giant component ALL vs MISINFO comparison, coloured with misinfo/debunk*



(a) *Jun 15 ALL giant component*

(b) *Jun 15 MISINFO giant component*

Figure 6.14: *Jun 15 giant component ALL vs MISINFO comparison, , coloured with misinfo/debunk*

Although community based metrics are useful, in this instance both datasets had similar partitioning into communities, hence gave similar metrics. These results, where modularity and communities were similar across datastreams, could be indicative of some of the limitations of visualising Twitter data. When translated to a network diagram in the methods outlined in this report, tweets will produce star topologies, and few links between separate communities are found. Further work could be done to filter out communities such as fig 6.15, and follow cascades/diffusion paths further. A useful tool as mentioned above would be the Twitter API, which would facilitate further investigation into specific interesting tweets, by retrieving all of a single tweet's quote tweets and further filling out the network. In addition, these results could indicate that misinformation spreads in similar ways to general conversation on Twitter, at least within the context of this dataset.

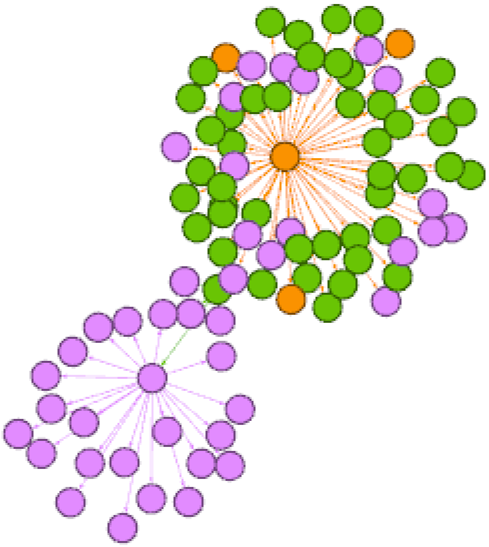


Figure 6.15: *Section of Jun 15 ALL, showing two interlinked communities*

6.5.2 Implications of classification results

Both classifiers produced interesting results on the datastreams. Of particular interest was the way nodes in the same communities were classified. Sometimes a nodes in a community had mostly similar classification topics, but some communities were much more varied. A conclusion from the WeVerify classifier was that misinfo communities tended to be more homogeneous in their classification topics: this could imply that misinformation is less varied than the general conversation on Twitter, with fewer new or challenging ideas. Furthermore, the general statistics showed that the topic Prominent Actors was much more prevalent throughout both datastreams than other topics, informing a logical conclusion that Twitter would be most likely to host conversations about celebrities, famous companies, etc.

The misinfo/debunk classifier showed interesting results: keeping in mind that it is a new untested classifier, it identified the misinfo stream as having more misinformation. It also identified more comments in the all datastream, perhaps a conclusion on the more social behaviour of the all datastream as opposed to pure misinformation on the misinfo datastream. Studies on node communities were less conclusive: many communities were a mix of MISINFO, COMMENT, etc.: there were fewer instances of one class dominating completely. This could be an indication of the new nature of the classifier, or the ambiguity of the test data.

6.6 Fulfilment of requirements

For the first objective, visualising the datastreams, the project has been largely successful. Graphs were created and displayed in Gephi, and interesting subsections were observed. The implementation worked well in transforming the unstructured Twitter data into a directed graph with sets of nodes and edges. The end result was less complex than originally hoped, but gave enough detail for further analysis in the second objective.

For the second objective, calculating network properties and adding extra attributes, it was completed fully, with some conclusive results. Calculating the network properties gave concise ways of representing multiple graphs, and some differences between the datastreams implied novel conclusions. However, the two datastreams were largely similar from a numerical standpoint. This was disappointing as the project aimed to identify key differences between both datastreams through network analysis, as differences between misleading and reliable information networks had been discovered in previous papers [31]. Improvement upon the methods of building the Twitter networks or using different network analysis techniques could potentially show differences: further research is required

Adding in the classifiers proved more successful: the results showed differences between the datastreams and conclusions were interesting. Both of the classifiers were successful in classifying the nodes, and gave a unique aspect on the data. Shortcomings include some un-classified nodes: the tweet text was not always available for each tweet, so some were not classified. Implementation of the Twitter API to retrieve the relevant text given the tweet ID would solve this failure.

6.7 Limitations of system

Some key assumptions were made throughout the project that might limit the accuracy of these results, and the content of their conclusions.

The misinfo datastream has been assumed to be a true representation of misinformation on Twitter. However, the only conditions for tweets to be gathered into this stream were the defined hashtags: #CCPVirus etc. The datastream could also be made up of users using these hashtags to refute/debunk misinformation, not spread it. Therefore the conclusions made on the basis of the results rest on the assumption of the two datastreams being discretely "misinformation" and "not misinformation".

In addition, the results gathered were only from a few select days, with approximately 10k tweets each: the entire repository is over 50 million tweets large. The days that were selected could have been biased, and the time periods of 0600-0800 were small. The data collected is in no way comprehensive, and this limitation could be a reason why the results are lacking in differences.

Another assumption is that quote tweets fully represent the conversation on Twitter. There are multiple ways a Twitter user can interact with a given tweet: they can retweet, reply, or quote tweet. Simply gathering quote tweets does not capture the whole story surrounding a particular tweet, so the networks produced will be limited in their characteristics. When first setting out in the project, it was imagined there would be a large network of tweets with interlinking edges between communities: however, this was not the true nature of the dataset.

In addition, the second classifier used was not fully tested: the model from Jacob Crawley is sound but not a full implementation like the WeVerify classifier. The related analysis could be re-done in the future when classifier reliability improves.

6.8 Potential improvements and further work

Given the amount of possible attributes that could be attached to these tweet objects, the potential for further investigation is large. With the addition of user fields, such as follower count, Weighted Correlated Influence could be calculated and overlaid onto nodes, showing influential users within the network. This and other advanced network metrics could provide further insight. The Twitter API could be used to find out user profile statistics, as well as further work with the dataset. Since many of the nodes are of degree one or less, the API could increase the complexity of the data by querying each node for its quote tweets, extending the network. Identifying potential superspreaders that share more misinformation, the ones at the center of communities, would be a useful contribution from gathering user data with the API.

Organising this data into networks, and attaching attributes to nodes gives a whole host of potential new features for misinformation classification. Degree of a node could be taken into consideration when classifying misinformation automatically in OSNs, and building a network over known misinformation sources could help classify these sources. Detection of misinformation is important with the pandemic, and these new features could increase classification effectiveness. Furthermore, these networks do not have to be built solely on misinformation: they could identify different kinds of community in OSNs, such as ones which diffuse reliable information.

These datastreams come from the dataset gathered in Categorising Fine-to-Coarse Grained Misinformation [24]: this study also annotated a set of twitter claims with DEBUNK, MISINFO as defined in the classifier used above. An extension/different route for the project could be to use the Twitter API to get the quote tweets for each MISINFO or DEBUNK tweet, then compare the networks generated. Although this dataset is only 1800 tweets large, compared to the millions in the datastream, it provides solid definition of misinformation and debunk tweets, and building these networks could provide interesting insight. This would combat the limitation of using the misinfo and all datastreams, which do not discretely define misinformation.

Chapter 7

Conclusion

This report has detailed the progression of a Twitter COVID-19 misinformation network analysis study. It has described previous relevant studies, along with COVID-19 datasets and network analysis techniques. It has defined the aims and objectives of the project, as well as the evaluation techniques. It summarised the application design, and the implementation of that application. The results of the network analysis were shown, with conclusions drawn.

This project began with a problem to tackle: misinformation surrounding the COVID-19 pandemic. This misinformation causes real life harm, and research into its nature, the way it spreads, and what can be done to stop that spread is at the forefront of much research around the world. Much has already been done in the two years since the start of the pandemic to identify misinformation: it is this work that this project is inspired to contribute to.

The datastreams gathered by the university provided a huge repository of tweets to access: the natural structure of Twitter with tweets and retweets lends itself to network building opportunities. The primary aim was to build a twitter diffusion network on these tweets, as had been done in previous papers pre-pandemic with ”fake news”. The secondary aim was to analyse these networks with network analysis techniques, and use classifiers on the twitter data to enrich the network nodes with more information. These aims evolved throughout the project, with the broad overall objective to extract useful differences and trends between the two datastreams.

The design and implementation of the Twitter network evolved dynamically through experimentation with elasticsearch and python, finding things that were possible (querying for quote tweets and building off them) and things that weren't (limitations of size of queries). Some aspects of the project, like finding retweets, proved more of a challenge than was initially expected. The learning process involved much changing of project direction, but the end implementation achieves the aims as laid out. The work with classifiers was a stimulating task which proved to produce novel results.

The results of this project satisfied the main aim: visually interesting graphs were produced out of the twitter datastreams, and the insight was enhanced by the classifiers. The network analysis metrics showed the general characteristics of both datastreams, and conclusions were drawn about their similarity in terms of these results : perhaps the data was not comprehensive enough, or the two datastreams simply did not show differences in their network structure. The classifiers proved to be key in identifying differences between the two datastreams: the WeVerify classifier showed the primary makeup of these tweets, Prominent Actors, and showed the homogeneity in class across communities.

All in all, this project showed that Twitter datastreams can be visualised to produce visually interesting results, and that this can be used to differentiate communities within datasets, as well as draw conclusions about the nature of such data. Further work includes improving the project methods and using different datasets: gathering tweets with the Twitter API to expand the network, increasing the size of these networks, applying the methodology to known misinformation/debunks. In addition, the networks created could be used as a feature in machine learning classifiers, in order to potentially differentiate sets of information present in OSNs, further contributing to identification of false/misleading information to stop harmful spread. Overall, this project's scope proved to be large. Focus could have been improved in the beginning of the project to limit the final results: a target of gathering only large communities in the dataset, or using pre-classified misinfo/debunks would have helped to give more concrete results. However, the project largely achieved its visualisation aims and objectives.

Bibliography

[1] Elasticsearch query dsl, . URL https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html.

[2] The elk stack: From the creators of elasticsearch, . URL https://www.elastic.co/what-is/elk-stack.

[3] What is elasticsearch? elasticsearch developer website, . URL https://www.elastic.co/what-is/elasticsearch. Accessed: 11-05-2022.

[4] Gate cloud: Text analytics-as-a-service. URL https://cloud.gate.ac.uk/.

[5] Features of gephi, gephi developer website. URL https://gephi.org/features/. Accessed: 11-05-2022.

[6] Networkx documentation. URL https://networkx.org/.

[7] Euvsvirus matchathon, . URL https://www.euvsvirus.org/.

[8] Pandas dataframe - pandas 1.4.2 documentation, . URL https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html.

[9] Tweet object — docs — twitter developer platform. URL https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet.

[10] *Tweet metadata timeline — docs — twitter developer platform.* URL https://developer.twitter.com/en/docs/twitter-api/enterprise/data-dictionary/tweet-timeline. Accessed: 11-05-2022.

[11] Who health topics: Infodemic. URL https://www.who.int/health-topics/infodemic. Accessed: 11-05-2022.

[12] 700 dead in iran after drinking toxic alcohol to 'cure coronavirus', Apr 2020. URL https://www.independent.co.uk/news/world/middle-east/coronavirus-iran-deaths-toxic-methanol-alcohol-fake-news-rumours-a9487801.html. Accessed: 11-05-2022.

[13] Twitter covid-19 stream, twitter community, Apr 2020. URL https://twittercommunity.com/t/new-covid-19-stream-endpoint-available-in-twitter-developer-labs135540. 11-05-2022.

[14] Python elasticsearch client, 2022. URL `https://elasticsearch-py.readthedocs.io/en/v8.2.0/`.

[15] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*, 2009.

[16] J Scott Brennen, Felix M Simon, Philip N Howard, and Rasmus Kleis Nielsen. Types, sources, and claims of covid-19 misinformation, 2020.

[17] Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. The covid-19 social media infodemic. *Scientific Reports*, 10(1):1–10, 2020.

[18] Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. Tweetscov19-a knowledge base of semantically annotated tweets about the covid-19 pandemic. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2991–2998, 2020.

[19] Pavlos Fafalios, Vasileios Iosifidis, Eirini Ntoutsi, and Stefan Dietze. Tweetskb: A public and large-scale rdf corpus of annotated tweets. In *European Semantic Web Conference*, pages 177–190. Springer, 2018.

[20] Yifan Hu. Efficient, high-quality force-directed graph drawing. *Mathematica journal*, 10 (1):37–71, 2005.

[21] Mathieu Jacomy. Gephi 0.9.2 : A new csv importer, Sep 2017. URL `https://gephi.wordpress.com/2017/09/26/gephi-0-9-2-a-new-csv-importer/`. Accessed: 11-05-2022.

[22] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014.

[23] Somya Jain and Adwitiya Sinha. Identification of influential users on twitter: A novel weighted correlated influence measure for covid-19. *Chaos, Solitons & Fractals*, 139: 110037, 2020.

[24] Ye Jiang, Xingyi Song, Carolina Scarton, Ahmet Aker, and Kalina Bontcheva. Categorising fine-to-coarse grained misinformation: An empirical study of covid-19 infodemic. *arXiv preprint arXiv:2106.11702*, 2021.

[25] Lothar Krempel. Network visualization. *The SAGE handbook of social network analysis*, pages 558–577, 2011.

[26] Nandita Krishnan, Jiayan Gu, Rebekah Tromble, and Lorien C Abroms. Research note: Examining how various social media platforms have responded to covid-19 misinformation. *Harvard Kennedy School Misinformation Review*, 2(6):1–25, 2021.

[27] Shawn Martin, W Michael Brown, Richard Klavans, and Kevin W Boyack. Openord: an open-source toolbox for large graph layout. In *Visualization and Data Analysis 2011*, volume 7868, page 786806. International Society for Optics and Photonics, 2011.

[28] Bickert Monika. Taking action against vaccine misinformation superspreaders, Aug 2021. URL `https://about.fb.com/news/2021/08/taking-action-against-vaccine-misinformation-superspreaders/`. Accessed: 11-05-2022.

[29] Olga Papadopoulou. Weverify annotation team @euvsvirus hackathon, Oct 2020. URL `https://weverify.eu/news/weverify-annotation-teameuvsvirus-hackathon/`. Accessed: 11-05-2022.

[30] David Pastor-Escuredo. Characterizing information leaders in twitter during covid-19 crisis. *arXiv preprint arXiv:2005.07266*, 2020.

[31] Francesco Pierri, Alessandro Artoni, and Stefano Ceri. Investigating italian disinformation spreading on twitter in the context of 2019 european elections. *PloS one*, 15(1): e0227821, 2020.

[32] Francesco Pierri, Carlo Piccardi, and Stefano Ceri. Topology comparison of twitter diffusion networks effectively reveals misleading information. *Scientific reports*, 10(1): 1–9, 2020.

[33] David Sayce. The number of tweets per day in 2020, Dec 2020. URL `https://www.dsayce.com/social-media/tweets-day/`. Accessed: 11-05-2022.

[34] Gautam Kishore Shahi, Anne Dirkson, and Tim A Majchrzak. An exploratory study of covid-19 misinformation on twitter. *Online social networks and media*, 22:100104, 2021.

[35] Marc A Smith, Ben Shneiderman, Natasa Milic-Frayling, Eduarda Mendes Rodrigues, Vladimir Barash, Cody Dunne, Tony Capone, Adam Perer, and Eric Gleave. Analyzing (social media) networks with nodexl. In *Proceedings of the fourth international conference on Communities and technologies*, pages 255–264, 2009.

[36] Yuxi Wang, Martin McKee, Aleksandra Torbica, and David Stuckler. Systematic literature review on the spread of health-related misinformation on social media. *Social science & medicine*, 240:112552, 2019.

[37] Liang Wu, Fred Morstatter, Kathleen M Carley, and Huan Liu. Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter*, 21(2):80–90, 2019.

[38] Product Marketing Lead Yifat Perry. Elasticsearch architecture: 7 key components, May 2021. URL `https://cloud.netapp.com/blog/cvo-blg-elasticsearch-architecture-7-key-components`.