Imperial College London

The Spread of Information about COVID-19 on Twitter

Author: Ioana Duta

CID: 01506519

Supervisor: Dr Prasun K Ray

Submitted in partial fulfilment of the requirements
for the degree of MSci in Mathematics

Department of Mathematics
Imperial College London
June 2022

This is my own work except where otherwise stated

Signed: Ioana Duta *Date:* 14 June 2022

Contents

List of Figures

List of Tables

Abstract

Social media acts as a forum for the exchange of ideas on a level of accessibility and equalisation unprecedented in human history. Twitter in particular has become a mainstay of socio-political communication. Thus, conversations on Twitter are a microcosm of the spread of ideas through the population as a whole, both as a mirror to information spread via other means, and as a medium of spread in and of itself. The results from analysis of tweet reply networks as a conversation medium reveals that it is possible to use measures of the size, depth and virality of reply cascades to predict classify conversations as to whether they are or are not about COVID, as well as to differentiate between accounts according to their purpose and influence. Investigations of user interaction networks revealed how both verified public figures and ordinary people have the power to influence conversations and become central in conversational communities.

Acknowledgements

Of course none of this would have been possible without the advice, expertise and support of my supervisor, Dr Prasun Ray, who put up with any and all questions at any and all hours. His generous advice and expertise was invaluable and I hope to have done it some degree of justice.

Thanks to my family for immense support throughout. My having been born while my parents were writing theses of their own does not seem to have given me an advantage, but thanks for trying.

Thanks to Nick for his endless patience in the face of my incessant complaining, and for his timely and perceptive coding tips.

I was lucky to have many friends who were there for me during the arduous writing process: special shoutouts to my coursemates for the solidarity, my housemates for having to see me at the end of a day of coding, and Pablo, who was surprisingly understanding for a dog.

1 Introduction

In the modern world, no event goes untouched or uninfluenced by social media. In particular, the influence of social media on politics has risen to prominence over the past few years [3]. Events such as the 2016 Brexit vote and US presidential elections [4], and the SARS-CoV-19 (hereafter COVID or COVID-19) pandemic [5] have played out on the battlegrounds of Facebook, Twitter and Instagram, among others. Politicians, experts, conspiracy theorists and ordinary people alike share and discuss a wide variety of topics, both influencing and being influenced. Both organic and sponsored content reaches a much wider audience than it otherwise would, including users who might not actually be subscribed to the original content creator. Content quality and veracity varies wildly, running the full gamut from reliable, honest and truthful content, to unreliable, untruthful, misinformed and misleading content, [6]. The coverage of the spectrum is not necessarily uniform, and volume and distribution bias towards misleading content could have dire consequences.

1.1 Twitter as a mirror to the world

Twitter dominates its particular niche in the social media landscape. It is a platform for brief multimedia text-focused communication, with multiple mechanisms by which users can reply and endorse content, as well as a verification procedure for public figures, agencies or groups. Prominent figures such as scientists and politicians are all but expected to have an active Twitter presence [7], and their interactions on that platform have a relatively high degree of legitimacy. This is in part due to the platform's reputation, but also to its verification procedure. Twitter operates as a news source for many users, who are able both to interact with such public figures and with the accounts of news outlets, institutions and more [8].

A few features also make Twitter a platform prone to combative content and uniquely lent to controversy. Firstly, anyone can create a Twitter account, and while many prominent figures are verified, still more enjoy anonymity. In addition to this, account verification only establishes that the account belongs to the claimed entity, but ascertains nothing about veracity of the content distributed; in order to become verified, one needs only to 'represent or otherwise be associated with a prominently recognized individual or brand' [9]. This important distinction is all-too-often overlooked – a verified account is not necessarily a reliable source of information, but conflating the two concepts leaves one vulnerable to misinformation by those who would exploit the misunderstanding.

Secondly, tweets are limited to 280 characters [10], and much can get lost in the brevity of a single tweet; if continued across a series of replies (known as a 'thread'), this might help to provide context and clarifications, but only if intentionally accessed and properly read, which might not be the case for a user interacting with the content in bad faith. Thirdly, the culture of the platform has become one of misunderstanding (wilful or otherwise) [11] [12]. The standard user's feed is likely to include respected and trusted public figures arguing with anonymous users who have few followers, and the topics are likely to include a constant stream of news – be it real, fake or satirical (distinguishing between these often being quite hard), or irreverent

memes[1] [13].

Twitter constitutes both an source of a broad range of content and a forum for discussion between an enormous variety of people, providing an exciting digital parallel to the "real" world. Discussions and content circulation on Twitter could therefore be considered as a proxy for discussions and content circulation in the "real" world. In [14], Bruns espouses the use of online networks to provide insight into offline and online social connections and interactions due to their being very rich in data. Rogers [15] goes further, recommending 'following the medium': exploring the online environment first in order to develop new research questions.

Issues related to global and public health are heavily discussed on Twitter [5], which functions heavily as a source of news and information [8]. This was particularly the case during the COVID-19 pandemic, when a combination of rising mistrust in official figures and agencies, and increased boredom induced by prolonged quarantine, turned more people than ever to Twitter [16]. Twitter played an important role in shaping the public opinion on issues surrounding damage limitation measures and vaccines. This, taken together with the fact that Twitter could be considered as a digital parallel to the "real" world, suggests that the study of Twitter content and circulation could provide invaluable societal insight: it may aid in explaining individual and group behaviour, which in turn would inform prediction models for behaviour in any future similar circumstances.

1.2 A brief description of Twitter

Twitter is a micro-blogging and social networking platform primarily focused around sharing posts, or 'tweets', of a maximum of 280 characters (formerly 140 characters until 17/11/2017) [17]. A tweet can also include media content: one each of up to 4 images, one video or GIF, or a poll of at most 4 options [10]. A tweet can contain text only, media only, or both.

Each Twitter user has a **User Page** which contains the following sections:

1. **User Profile** - account photo, cover photo, display name, username, user-submitted bio, date of joining, and optionally the user's location and a custom link), the number of their followers and the people they follow (hereafter 'friends' as in [18])
2. **Tweets tab** - lists the user's originally authored tweets and the user's retweets; this tab does not contain user's replies
3. **Tweets and replies tab** - lists the same content as the tweets tab and also the replies the user has created to their own or others' tweets
4. **Media tab** - lists the user's original tweets or replies that contain images, video or GIFs; this tab does not contain user's retweets
5. **Likes tab** - lists the user's liked tweets.

[1] A contemporaneous example is the Depp v. Heard defamation trial, which took place during the writing of this thesis. Despite the seriousness and complexity of the case, millions of onlookers seemed to be getting their (mis)information exclusively from social media with little regard as to its veracity or the reliability of its source, and memes about it were inescapable.

Figure 1 gives an example of a User Page that displays all these sections.

Figure 1: Screenshot of the User Page of current (May 2022) Twitter CEO Parag Agrawal (@paraga) [1].

A Twitter user account can be in one of two states [10]:

1. **Protected account** – only the accounts of approved user can view the User Page of a private account; the protected account tweets cannot be retweeted
2. **Public account** - all user accounts can view the User Page of a public account; public account tweets can be retweeted

In addition, a Twitter user account can be **verified**, a status which indicates that Twitter has deemed it necessary to identify the account to genuinely belong to the purported user, amongst possible impostors. Politicians and their campaigns, celebrities, and corporations are examples of accounts that are typically verified. This status could apply to either a public or a protected account, though the need to be verified often coincides with having the sort of public presence to warrant a public account.

Users may also be suspended by Twitter. This means that the account is locked against use by the owner, and usually occurs because the account has been hacked, detected as spam, or has been found to violate the Twitter terms of use regarding, say, bullying [10]. In the case of an account being suspended, its tweets are no longer visible to other users.

A Twitter user can interact with the Twitter interface and users who have access to their account in a variety of ways [10]:

1. **Post their own tweets** (known as 'tweeting') – tweets are listed in the **tweets tab** of the User Page
2. **Like** another users' tweet – the liked tweet is listed in the **Likes tab**, which is visible to any other user who has access the user page
3. **Retweet** another user's tweet – re-posts the original tweet; re-tweeted content appears in the user's tweets tab alongside their own
4. **Quote tweeting** – a mechanism by which a user writes a comment regarding another user's tweet such that the result is a new tweet, the comment appearing with the tweet being commented on below. These also appear in the tweets tab unless formulated as a reply
5. **Reply** to a tweet – replies and reply chains form a parallel to a real-life conversation or a forum thread; replies are listed in the **tweets and replies** tab on a user's page.

Retweeting functions as a relatively unambiguous endorsement of the retweeted tweet, whereas the intent behind liking a tweet is slightly more unclear (one might, for example, like a tweet to read later).

Each individual tweet shows the number of times users have interacted with it in each of these ways, as shown in Figure 2.

Figure 2: Screenshot of the first ever tweet [2], by Twitter founder and former CEO Jack Dorsey (@jack). Note the key features (like, retweet, and quote tweet counts), as well as the buttons to (respectively) reply to, retweet, like and share a tweet. Features such as source and date/time postd can also be seen. Reply counts are actually not listed except when a tweet is viewed in the timeline.

There are also several ways in which users can interact with each other:

- Users may **follow** one another:

  - If user A follows user B, B's tweets appear on A's timeline (the main feed user A sees). A is referred to as a follower of B, and this thesis will refer to B as a 'friend' of A.
  - If A and B follow each other, their following is **mutual** (the two users are often colloquially referred to as 'mutuals').

- Users can **mention** one another in a post by preceding an account name with the @ character. This causes the mentioned user to be notified of the mention, and the `@username` string functions as a link to the mentioned user's page.
- Users can **block** or **mute** one another:
  - If user A has muted user B, they will not see any of B's tweets on their timeline. This can occur without A having to unfollow or block B, and B will not be made aware of the muting
  - If user A has blocked user B, B is unable to follow A, see their tweets or in any way interact with them

In addition to following other users, users may also follow **topics**, follow **key words**, or follow **hashtags**. A hashtag is a word phrase (with no punctuation or spaces) preceded by the # character; if included in a tweet, the hashtag becomes a link to a search for all other tweets containing that hashtag.

A user's **timeline** (the main feed of content they see) is primarily made up of tweets and retweets from their friends and any topics, hashtags etc. they follow. Less frequently they will also be shown tweets their friends liked, tweets similar to those the user has recently liked, tweets from users similar to the user's friends, or tweets from topics and key words popular with the user's network [19]. Interspersed throughout is sponsored content and advertising. The content in the timeline is by default not ordered chronologically, but instead according to what Twitter 'thinks' the user will care about the most based on (among other things) who and what they most commonly engage with [19].

Thus, a Twitter user is regularly exposed to content they did not explicitly sign up to see, and has no guarantee of seeing everything they *did* due to the non-chronological ordering of content.

1.3 Current project

This project collects and analyses cascades of replies to tweets from accounts of national health agencies of a series of English-speaking countries, as well as the related user interaction networks. A cascade is a directed tree mapping the branching paths that a single piece of content takes as it spreads through a network, starting from a single original node. The analysis aims to investigate the differences between how replies proliferate from tweets made my different accounts in different geographic regions, the differences in conversations about COVID and those not about COVID, and the properties of the emergent user interaction network, particularly community formation and identification of significant users.

Data collection is made using the python package Tweepy to interact with the Twitter API.

The data collection for a cascade of tweets involves fetching every tweet in a conversation, tracing who replied to whom and then using these retrieved tweets to form cascades – diffusion trees of replies proliferating from a root tweet. The analysis of the retrieved cascades focuses on extracting the properties of the cascades of replies, and the emergent user interaction networks formed by analysing which users reply to each other's tweets. This project focuses on investigating such properties of the networks as depth, size and virality, investigating how these properties vary between account, region, and topic. The project will also investigate community detection and user importance in the user interaction network.

2 Literature Review

Recent previous work has analysed information cascades on Twitter. What follows is a review of some relevant recent papers.

[18] and [6] both analyse information cascades on Twitter, contrasting between different types of original tweet. 'Cascade type' refers to a cascade formed from an original tweet of a certain type. For example, a 'false news cascade' refers to a cascade originating from a tweet espousing false news; an 'image cascade' refers to a cascade originating from a tweet containing an image. The papers are also discussed in [20], which attempts to verify whether the results found in the two papers persist while controlling for cascade type, or if they are in fact explained by the difference in average cascade size between the cascade types.

[21] and [14] look at user interaction networks emerging from replies, studying temporal dynamics, differences in community formation between various topics, and centralities of users in the networks.

2.1 The Structural Virality of Online Diffusion (2016)

[18] contrasts cascades of different types (images, videos, news stories and petitions). The authors distinguish between two types of diffusion: broadcast (large bursts of adoptions from a single 'parent' node) and viral (multi-generational branching, where any given node spreads to only a few others). In particular, they found that online diffusion is characterised by diversity: popular events grow both ways, and in every possible combination of the two.

The paper provides a rigorous definition of a continuous measure of structural virality – this avoids the necessity of binning cascades into one or the other definition, and allows for a more open-ended and subtle approach to relating cascade size to structure. It aims to characterise the structure of the observable patterns arising from an unobserved diffusion process. It increases with the structure's branching factor, and with cascade depth. Problems arise, however, with just considering depth (or indeed average depth): a long chain of single adoptions has great depth, but would not intuitively be considered structurally interesting; a long chain with a sudden large broadcast has high average depth, but since this is due to a single broadcaster, this can't be considered viral either. The authors tackle this problem by instead considering

the cascade's *Wiener index*,

$$W(T) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}$$

the sum of the lengths of the shortest paths between all pairs of nodes. Structural virality $\nu(T)$ is thus defined as the average distance between all pairs of nodes:

$$\nu(T) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}$$

where d_{ij} is the length of the shortest path between nodes i and j, and the diffusion tree T has n nodes. (The authors also present a simple and scalable method for calculating this). The authors note that while this definition is satisfying, it is possible to construct hypothetical examples that have a high structural virality but nonetheless cannot intuitively be considered structurally viral. However, on the whole the authors found that structural virality is a good match for how one might instinctively rank cascades by virality.

The authors built their cascades by first identifying a piece of content; then, for each observation of that content, they then identified:

- The user, X, who posted the observation
- The time X posted the content
- The users followed by X (X's 'friends')

Then, the friends are checked to see if they have also previously posted the content. If they have not, X is labelled a 'root' node of the resulting cascade. If they have, then the friend most likely to have spread the content to X is labelled X's 'parent' in the cascade. (Such a user was usually explicitly credited by X, avoiding ambiguity in most cases). The authors estimate 95% accuracy when identifying how a user learned the information, because there are three ways in which a user can learn from their friends, the first two of which account for 75% of all cases:

- Official retweet: the user directly retweets from a user on their timeline. It's almost certain that they learned the information from the user they retweeted from, and this accounted for 65% of instances.
- Accredited repost: the credit may still be present in the form of a mentioned user, e.g. the tweet starts "RT @username". This has become much less common since the quote tweet was introduced in April 2015 (the year before the publication of this article, and after the end of the data collection window). The user almost certainly learned from the quoted user if they are a friend, or from a friend who follows the quoted user. If no such relationship existed, the user was considered a parent node. This accounted for 10% of cases.
- Uncredited repost: in the absence of any other information about how the user learned the information, the parent is taken to be the friend who most recently tweeted about the content. If no such friend exists, we again label the focal user a root. This has an estimated accuracy of 79%.

The dataset analysed included all tweets from from July 2011 to June 2012 (12 months) containing URLs directing to one of several popular websites. There were approximately 622 million unique pieces of content and approximately 1.2 billion individual posts by users. The authors noted that images and videos are far more numerous than news stories, and petitions are by far the least numerous type of content.

Investigating the relationship between popularity (cascade size) and structural virality, the authors found that for petitions, images and videos structural virality was surprisingly low and largely invariant with respect to size. Even for news, the relationship between structural virality and size was still low, though higher. The authors concluded that on average, even very large events are dominated by broadcasts, and knowing only a cascade's size implies very little about its structure. The authors also note that only the size of the largest broadcast has any meaningful relationship with popularity.

The authors compared empirical observations to predictions from a series of simple generative models, first with Erdős-Rényi (ER) graphs, then with scale-free random networks. The ER models did not correspond to the empirical observations as well as the scale-free models did, despite experimentation with the parameters (for example, taking the infection probability β to be both a constant and drawn from a power-law distribution $\beta^{-\alpha}$). Sweeping over β and α, the authors found certain values of α and r (the so-called 'basic reproduction number') for which mean structural virality, the correlation between size and structural virality, and the probability of a given item becoming popular all reflected the empirical data.

The authors acknowledge a few possible systematic biases that might be affecting their methods. First, the same content may spread not only via Twitter but also via other channels (other social media, email, in person etc. – this is referred to as 'off-channel diffusion'). Thus, two individual introductions of a piece of content by two separate users may in fact be linked, so a single diffusion tree could be mistaken for two independent events. Secondly, there is potential bias due to the preference of reposting over retweeting. The similarity between a user and their friends/followers ('homophily') may mean that two users who are connected in this way may independently post about the same content close together in time. The authors were wary of mistaking this type of similarity for influence.

To investigate the effects of 'off-channel diffusion', the authors investigated the diffusion of hashtags specific to Twitter. These are less likely to be affected by the bias, because a) hashtags are much less likely to have originated outside of Twitter, and b) they are much less likely to migrate away from Twitter. Analysis of hashtag diffusion yielded qualitatively similar results to the authors' primary analysis, leading them to conclude that off-channel diffusion was probably not significantly affecting their results.

2.2 The spread of true and false news online (2018)

[6] investigates a corpus of verified true and false news retweet cascades, taking news to mean essentially any event or 'rumour' that was tweeted about and contrasting between true and false. The authors used fact checking websites to verify the truth of the news, acknowledging that this may have introduced selection bias into the process, as only certain types of news

attract the attention of checking sites. This bias was addressed by a robustness check, in which they studied a second sample of cascades based on news whose truth or falsehood was not verified by a fact checking website, but by three undergraduates. This sample yielded similar results.

The dataset comprised of about 126,000 rumour cascades, spread by approximately 3 million people more than 4.5 million times (i.e. the total number of nodes across all cascades). More of the false rumours produced fewer than 1000 cascades, whereas true rumours resulted in over 1000 cascades (as percentages of the total). There were peaks of false rumours at the ends of 2013, 2015 and 2016, corresponding to U.S. presidential elections; similarly, during the 2012 and 2016 presidential elections there were sharp increases in the total number of false political rumours, and during the Russian annexation of Crimea in 2014 there was a peak in rumours that were partially true and partially false. The largest category of the rumours was politics (around 45,000 cascades); then urban legends, business, terrorism, science, entertainment, and natural disasters.

Data retrieved directly from Twitter builds a cascade where all retweets point directly to the original tweet. However, this is an inaccurate picture of how retweets actually spread – one can retweet another user's retweet, for example. The authors built the cascade using a method known as time-inferred diffusion, and knowledge of users' follower networks; this approach is based on research in [22]. For example, say a tweet by user X was retweeted by users Y and Z. The information from Twitter would result in a cascade that shows nodes Y and Z both linking directly to node X. However, say that we also know that Z follows Y, and Y follows X, but Z does not follow X: we can infer that Z probably retweeted the tweet from Y, not the original tweeter X. Thus, a more likely cascade shows only links between Y and the root node X, and between Y and Z. Information on follower relationships is inferred at the time of the retweet. Though the authors analyzed replies in other ways (see later), they excluded quote tweets and replies from the analysis of propagation dynamics. Generally speaking, retweets do not contain additional information, so can be taken to indicate that the retweeters endorse the tweet (which is the kind of relationship the paper is interested in). Replies and quote tweets, on the other hand, do not imply endorsement – they could indicate direct disagreement.

The authors quantified depth, size, maximum breadth, and structural virality (using the results of [18]), finding that false news cascades significantly farther, faster, deeper, and more broadly than true news, regardless of the type of news under discussion, and reached far more people.

In addition to analysis similar to that of [18], the authors also investigated the emotional content and 'novelty' of the tweets and replies, as well as data about the accounts who spread and replied to each original tweet, finding that (contrary to their predictions) spreaders of false news had significantly fewer followers, were less active and followed fewer people than spreaders of true news. Emotional content was investigated in an attempt to explain the differences in diffusion, since user characteristics could not, as novel information is perceived as more valuable and therefore more likely to spread. Novelty was assessed by finding the formation distance between a tweet and all prior tweets which users were exposed to before retweeting said tweet. The authors found that false rumours were significantly more novel than true information. The authors then assessed user perception of the news by analysing the emotional associations of

the words used in the replies, finding that replies to false news inspired expressed surprise (as would be expected from the novelty hypothesis) and disgust, while the truth inspired sadness, anticipation, joy and trust.

The authors considered the potential impact of bots when drawing conclusions about human judgement, using a bot-detection algorithm to identify and remove bots prior to analysis. They note, however, than none of the main conclusions were affected when bot traffic was reintroduced into the data. They also found that the spread of both types of news were affected equally by bots, concluding that the greater spread of false news was due to human-driven spread.

2.3 Comparing information diffusion mechanisms by matching on cascade size (2021)

[20] aimed to compare and assess [18] and [6]. The authors noted that neither of the two papers controlled for cascade size, and investigated whether the results found could in fact be explained by the fact that the sizes of the cascades differed according to their type. For example, in [6], false news cascades were much larger (on average) than the true news cascades. The authors used sampling with replacement from both datasets in order to create size-controlled cascades, and carried out the analysis again. They concluded that the findings of [6] are actually accounted for by the differences in size of the two different types of cascade, while the findings of [18] persist when controlling for size. The authors propose that the 'deeper, broader and faster' spreading of false news is due to the higher 'infectiousness' of the information.

2.4 Characterization of the Twitter @replies network: are user ties social or topical? (2010)

[21] looks at the user interaction networks emerging from replies, studying the differences in community formation between various topics. The authors collected used 612,556 Portuguese-language tweets (of which 73,506 were replies) from 49,303 users, made between between March and May 2010, and identified user reply networks to study whether user replies are socially or topically motivated on three topics: religion, politics and sport.

Based on this body of tweets, the authors built an activity network: an implicit network based on interactions (as opposed to the explicit network of user connections via follower-friend relationships). They built the network as a directed, weighted graph where an edge (u_i, u_j) is such that u_i replied to u_j and is weighted by w_{ij}, the number of times such a reply was made.

They identify u_i's ego-centric network, $G'(u_i)$, as all the people who u_i has sent at least one reply to. Then, given topics T_X and T_Y, one can extract from $G'(u_i)$ the restricted networks $G'_{T_X}(u_i)$ and $G'_{T_Y}(u_i)$. These represent the user's replies to other individuals on T_X and T_Y respectively. Then U_X and U_Y are the set of users to whom the user is 'connected' in $G'_{T_X}(u_i)$ and $G'_{T_Y}(u_i)$, respectively. The authors are interested in the relationship between U_X and U_Y – whether they are distinct, or whether they have a degree of overlap – even a complete overlap, or whether one contains the other completely. Denoting by $p(U_X)$ and $p(U_Y)$ the marginal probabilities of u_i interacting with other users on T_X and T_Y respectively, the probability of interacting with the same subset of people on both topics is denoted $p(U_X, U_Y)$. If the user

follows no selection criteria when replying, then probabilities of replying in either group will be independent and $p(U_X, U_Y) = p(U_X)p(U_Y)$. However, users do not choose who to reply to at random. They might be motivated primarily by social ties, or by topical interest:

- If the user's interactions are topically motivated, we will have a lower actual probability of overlap:

$$p(U_X, U_Y) < p(U_X)p(U_Y)$$

- If motivation to reply is social, we have the converse:

$$p(U_X, U_Y) > p(U_X)p(U_Y)$$

The authors use Normalized Pointwise Mutual Information (NPMI) to quantify this overlap. NPMI relates two events' co-occurrence probability to their individual probabilities. In this context each 'event' is a user's interactions with other users on each topic. Thus, given topics TX and TY associated to the sets of users UX and UY, the NPMI is

$$\begin{aligned} NPMI(U_X, U_Y) &= \frac{\ln \frac{p(U_X, U_Y)}{p(U_X)p(U_Y)}}{-\ln p(U_X, U_Y)} \\ &= \frac{\ln \frac{a}{(a+b)(a+c)}}{-\ln a} \end{aligned}$$

Where a is the fraction of users from $G'(u_i)$ who user u_i replied to on both T_X and T_Y, and b and c are the fraction of users from $G'(u_i)$ who user u_i replied to on only T_X or T_Y, respectively. Under the independence assumption we clearly have $NPMI = 0$. If overlap is higher than expected we have $NPMI \in (0, 1]$, and if overlap is lower than expected we have $NPMI \in [-1, 0)$. At the extremes, if U_X and U_Y have no overlap, $NPMI = -1$; if they overlap completely $NPMI = 1$.

The authors used simple keyword-based classifiers to collect messages from three distinct topics: sports (58 keywords), religion (56 keywords) and politics (55 keywords). The topics were chosen as the authors suspected they would have low correlation, being sufficiently different as to actually motivate a user to use selection. The topics were also chosen so as to motivate frequent and constant tweeting, given events in Portugal at the time. Time distributions confirm this. They found that many users interact with more than one topic, with some replies to falling into more than one topic (with three tweets falling into all three topics). However, far fewer replies exist in each possible intersection of topics than the total number of messages in each topic.

The procedure for analysis was as follows. First, for any user and their topic sub-graphs, quantify the subgraph overlap. Then, if one is present, compute the corresponding NPMI values. The authors found that those users that have smaller ego-centric networks have a heavier social influence on their replying habits, and that when a user has more connections, they begin to display a slight tendency to disjoint their network by topic. The authors also concluded that NPMI could be a useful feature for user classification, and could support user profiling processes.

2.5 Mapping dynamic conversation networks on Twitter (2012)

[14] also investigates user reply interaction networks, studying temporal dynamics, differences in community formation between various topics, and users' centralities. The author focused on the reply networks emerging from studying the proliferation of hashtags, and their temporal dynamics. The author builds upon previous work [23], [24], which examines the conversational aspects of reply exchanges, visualising cascades between multiple participants, finding 'at least [...] moderate' responsiveness. The new work aims to consider network properties in more depth than the previous.

The reasoning behind hashtag use is as follows. Other works, for example [25], used a heuristic filtering system, selecting all tweets from the timezone of the event in question during the relevant time window and then filtering by characteristic keywords. Keywords are limited in practicality because different combinations and permutations of keywords may describe the same effect and researchers may struggle to cover all bases. Hashtags, meanwhile, are far more predetermined and a much smaller set characterises a given event or topic, and are less subject to change.

However, the author notes that there are limitations in considering these exchanges as conversations since some responses remain one-offs without sparking an extended conversation. Because only hashtags are tracked, the data collection will miss follow-on tweets that didn't also use the hashtag, so the research presented will often 'capture the beginning more than the conclusion' of Twitter conversations.

The author also notes a few limitations of the methods. At the time of publication, the `in_reply_to_user_id` field was unreliable and researchers often to had to trawl for @mentions. this caught @retweets (see earlier), which he nonetheless considers to 'serve a conversational purpose' as they allowed users to comment. It also meant that a single tweet would be a reply to all others mentioned – be they further up in the chain, or just relevant to the discussion.

Bruns particularly notes the use of visualisation as an approach to identifying key patterns. Though espousing network visualisation specifically as a way of reducing complexity and identifying key clusters and users, he also warns against regarding it as a black box. Static visualisations in particular should not be an end in and of themselves, but should work to find where further study might yield interesting results. Bruns stresses the need for directed edges, so we are able to distinguish prolific repliers, attracters of attention, and those who are both. He notices in an example (concerned with tweets about an Australian election) that mainstream news sources were the emergent members of the latter camp.

In the collection, Bruns includes a timestamp with each tweet (which can for ease of interpretability be converted to time since first tweet). It is then possible to graph total volume of tweets within a timeline at varying resolutions, and perhaps filter for keywords before graphing their 'rise and fall'.

The paper asks the question: how long does a tweet 'last'? It offers several answers, each affecting analysis. Firstly, we may consider that once a tweet is made, it persists. We would see the resulting network grow denser in time, with more edges appearing and existing edges thickening as replies are made.

Conversely, we can say replies last a certain length of time (prolonged by being replied to). Bruns notes the 'ephemerality of online spaces': replies are meaningful only cumulatively and if the connection is upheld through subsequent continuations, so a single reply disappears practically immediately. Such a graph shows rapidly shifting centres of activity.

We can also shift between the two extremes (Bruns makes a comparison to the half-life of radioactive particles). We might consider the time-decay of a tweet, i.e. how much time passes before we no longer consider it part of the network.

The following considerations are made:

- the total volume of tweets currently occurring in network (affects decay time – more tweets, shorter time)
- the volume of replies received by a given user – more in a time, buried by subsequent replies
- that well-connected users (who follow many people, particularly) receive more replies so any given reply is less visible to them
- that if two people reply and are known to/follow one another, this might be a more meaningful reply

The procedure followed is to select a global reply decay time – very large, nearly 0, or variably based on contextual factors. Whichever of these options is chosen, there are several ways to dynamically visualise the network across the time scales. One can graph the whole timeframe, such that all nodes stay visible, while edges appear/disappear. Alternatively, one can visualise only part of the timeframe: only some nodes are relevant, the others floating off to the side.

In closing, Bruns notes that this particular method misses relevant interactions that happen outside the hashtag, or under alternative hashtags. He recommends future work check follow-ons, regardless of whether the hashtag is contained.

3 Methods

All data collection, processing and analysis was performed using own purpose-written code, much of which interacts directly with the Twitter API. There are references to specific data collection functions throughout this section. All functions referred to can be found in the Appendix.

3.0.1 The Twitter API

Twitter offers a comprehensive Application Programming Interface (Twitter API) that provides programmatic access to tweets and public information about accounts. The Twitter API enables the retrieval of large quantities of high-quality content and circulation data that can be used to build interaction models on a large scale.

In order to handle the high volume of requests made to the Twitter API, there are limits placed on the number of requests that can be made in certain windows of time, known as a rate limit [26]. For example, a project or user can make a maximum of 900 tweet look-ups in any 15 minute window, with violations incurring an error code. Some endpoints also have a tweet consumption cap limiting the number of tweets that a project can receive from them in any given month [27]. For example, full-archive search can make at most 2 million requests per month.

3.1 Data collection

Tweets and replies were collected using Tweepy; first, all of an account's original tweet IDs were collected, and then all replies were collected using these as conversation IDs. Data was collected on the 01/01/2022-03/03/2022 period using original tweets and retweets (i.e. no replies) from 9 accounts belonging to public health organisations of 4 English-speaking countries. The following discussion makes reference to purpose-written code, which can be found in the Appendix.

The main functionality that was made use of is as follows. To identify and extract the tweets in a conversation, a root original tweet is first identified, after which all its replies are extracted. For each tweet (both replies and root), the ID, date & time, number of retweets and favorites, and account ID of the tweet author are retrieved. Collection and identification of root tweets was done using the API's **Cursor** method on a particular account. The extraction of replies used the Conversation ID and Full-Archive Search. Follow-up information on the verified status of accounts in a retweet chain, or their follower and following counts, is extracted using User Lookup; the number of retweets and favourites received by tweets in a conversation is given by Tweet Lookup.

3.1.1 Accounts included in data collection

In total 9 accounts were included in the data collection, all official accounts of government health agencies from 4 English-speaking countries. Accounts of individual persons, such as government medical officers, were deliberately excluded for two main reasons: one, because the occupant (and thus the nature) of the office is subject to change; two, because the personal

nature of such an account encourages replies to have a similarly personal slant. English speaking countries were chosen to avoid complicating analysis with the need to obtain, and rely on the accuracy of, translations.

The accounts investigated were:

- United Kingdom
  - UK Health Security Agency @UKHSA
  - Department of Health and Social Care @DHSCgovuk
  - The National Health Service (NHS) @NHSuk
  - NHS England and NHS Improvement @NHSEngland
- United States
  - The Center for Disease Control @CDCgov
- Canada
  - Health Canada and Public Health Agency Canada @GovCanHealth
  - Canadian Institutes of Health Research @CIHR_IRSC
- New Zealand
  - Unite against COVID-19 New Zealand @covid19nz
  - Ministry of Health - Manatū Hauora @minhealthnz

It is worth noting that some tweets were included in more than one account's collection, due to the fact that retweets are included as roots as well. For example, two accounts may retweet the same tweet, or each other (particularly within the same country). This is further discussed in Section 3.2.2.

3.1.2 Collection methods

The first step was to collect all tweets made by an account in a certain period, as well as certain attributes of each tweet. The function `user_tl()` uses Tweepy's Cursor method to collect a specified set of attributes for a specified number of tweets from a specified account. The attributes collected were:

- `full_text`: full text of the tweet
- `id`: id of the tweet
- `favorite_count, retweet_count`: like and retweet counts, respectively, at time of data collection
- `created_at`: date of tweet creation
- `in_reply_to_screen_name, in_reply_to_status_id`: username of the author of the tweet replied to, and id of the tweet replied to, respectively (if applicable: None otherwise)

- `retweeted`: whether tweet was a retweet (NaN otherwise)
- `source`: app or device type from which the tweet originated (if specified, None otherwise)
- `is_quote_status`: whether tweet was a quote tweet (NaN otherwise)
- `rt_from`: username of tweet author (if tweet is a retweet: NaN otherwise)

This was all collected as a dictionary, converted to to a dataframe, and saved as a .csv which could then be used for cascade extraction.

The dataset of recent tweets was filtered to exclude replies. The data was processed to include the original ID of retweeted tweets: Twitter gives retweeted tweets a 'retweet ID' unique to each instance of retweeting, and the ID collected is by default this new ID.

The Conversation IDs were found and looped through to get the conversations, resulting in a dataset containing all the non-reply tweets made or retweeted by the account in the time period and all replies to these tweets, with a number of attributes also recorded (see Section 3.1.3).

Twitter recently introduced the Conversation ID as an attribute of tweets [28]. This is the tweet ID of the original/root tweet in a conversation. However, at the time of data collection Tweepy had not yet included Conversation ID in the attributes of a tweet status object, so it was necessary to devise a separate mechanism for both a) extracting a tweet's Conversation ID and b) searching all tweets by a given Conversation ID. The following functions performed these tasks:

- `get_CID(id)` takes a tweet ID and finds the conversation ID of that tweet [29].
- `get_Conv(conversation_id)` given a conversation ID (the ID of the conversation's originating tweet), returns a list in which each element each element is a dictionary containing the conversation ID, ID, and text of a reply to the original tweet. The first element of the list is the original tweet of the conversation [29].[2]
- The function `add_params()` takes as input a) a list such as the one returned by `get_C` and b) a list of desired parameters; for each dictionary in the first list, it adds the attributes in the second list for the tweet in question. Crucially, it will always add the values of `'in_reply_to_status_id'` and `'in_reply_to_user_id'`, which are required to make the cascades and interaction networks.

However, not all replies were picked up by this method. This process interacted directly with the Twitter API, and since this was a large-scale search which took long enough to perform that it would have been impractical to supervise, it was difficult to deal perfectly with the possibility of incurring rate limit violations (see Section 3.1.2). The code was to take a shorter pause after every request and a longer pause after a predetermined number of requests in order to avoid this as much as possible, but these pause had to avoid being so long as to impractically extend the runtime. This meant that some rate limit violations were unavoidable, so for example some tweet lookups failed when searching for replies. This was the cause of some of the missing tweets:

[2]This and the previous function were written with reference to [29]

a discussion of the reconstruction of cascades to pick up tweets missed by data collection, and the treatment of unrecoverable tweets, is found in Section 3.2.1.

3.1.3 Structure of the dataset

The following attributes were collected for each tweet:

- `conversation_id`: the ID of the root tweet of the conversation of which the tweet is a part
- `id`: the tweet's own ID
- `text`: the tweet text
- `in_reply_to_status_id` and `in_reply_to_user_id`: ID of the tweet being replied to, and of its author (both 'Root' if no tweet was being replied to)
- `user_name` and `user_id`: user ID and username of tweet author
- `created_at`: date of tweet creation
- `verified`: whether the user is verified (at the time of data collection)
- `followers_count` and `friends_count`: the number of users who a) follow the tweet author and b) are followed by respectively, the tweet author (at the time of data collection)
- `cascade`: which account was being swept when this tweet was picked up

The numbers of tweets, users and conversations in the raw dataset are in Table 1.

Country	Username	Tweets	Cascades	Users
UK	UKHSA	8804	235	3626
	DHSCgovuk	14126	273	6304
	NHSuk	1608	76	1010
	NHSEngland	3759	139	1888
US	CDCGov	9241	149	3620
Can	GovCanHealth	7526	253	2739
	CIHR_IRSC	1087	109	451
NZ	covid19nz	1015	58	238
	minhealthnz	1981	67	805
All	–	42883	1200	17466

Table 1: Number of tweets, cascades and users in the raw dataset, split by account. Values in the final row are not the sums of the corresponding values in the other rows because certain conversations, tweets and users were included in more than one account's data.

3.1.4 Reply networks

The mechanism of replying in Twitter is that which has the clearest parallel to real life communication. The functionality is simple – one can make one tweet in reply to a single other tweet, that can in turn have multiple replies to it. Thus it is possible to construct a 'cascade' or 'reply tree' of replies to a single original 'root' tweet in the form of a directed graph in which one node, which represents a single tweet, has one outgoing edge linking it to the single tweet to which it is a reply (unless that tweet is the root, in which case it has no outgoing edges), but can have multiple incoming edges from nodes representing tweets that replied to it. The properties of a reply cascade are illustrated in Figure 3.

In the Twitter interface, a reply may list several users as being 'replied to' – these will be not only the author of the tweet to which the tweet in question is a direct reply, but potentially also any users mentioned in said tweet, and any authors further up this 'branch' of the cascade. This project only makes a link to the tweet being directly replied to.

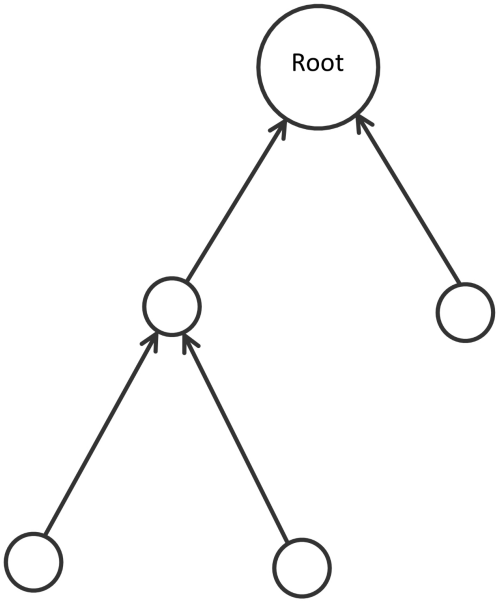

Figure 3: Diagram of a simple cascade. The large circle is the root tweet, and the smaller circles its replies. A tweet 'points to' the tweet it is replying to. Note how some tweets have two replies, but a tweet can only 'point' (reply) to one other.

3.2 Processing

3.2.1 Reconstruction and missing tweets

Occasionally, cascades may run into the issue of 'missing' tweets. As is illustrated in Figure 4, this occurs when a tweet in the cascade – the missing tweet – was not able to be recovered, but tweets replying to it were. We know the missing tweet exists and we know its ID from the tweets replying to it, but we do not know to which tweet it replied. This commonly happens because the missing tweet was deleted by its author, or because the author's account was suspended or made private. During data collection, due to the high volume of requests being made (and in part due to the dependency on occasionally inconsistent WiFi), some tweets were not picked up due to a rate limit violation during their lookup (see Section 1.1). That is, the tweets existed and were recoverable, but the data collection nonetheless failed to pick them up. Their existence was known due to their tweet ID's presence in the `in_reply_to_status_id` list, but this was absent in the `id` list. Therefore the dataset could, to an extent, be reconstructed: by checking the for ID numbers present in the former list and not the latter, looking up the corresponding tweet, filling in its information in the relevant columns, and appending both lists accordingly. If the tweet could not be looked up, its ID was nonetheless appended to the `id` column to record the attempt, and every other piece of information was recorded as the string `'error'`. 2539 lookups were performed in total, of which 144 resulted in errors; 2395 tweets were successfully recovered.

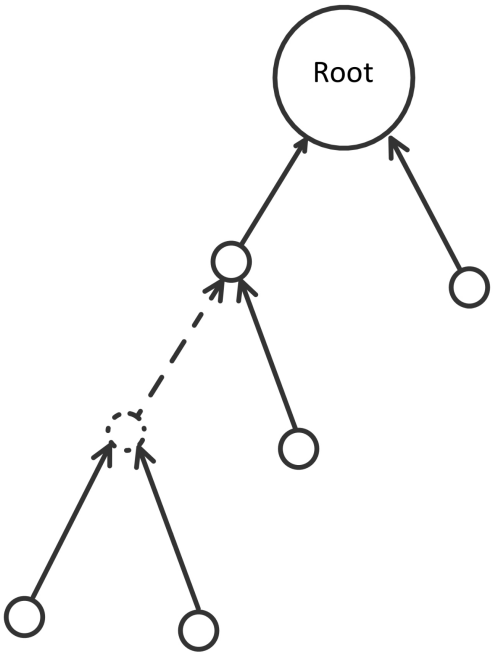

Figure 4: Illustration of what it means for a tweet to be 'missing'. This is cascade as in Figure 3, but the reply represented by a dashed circle is 'missing', and the link represented by the dashed arrow is unrecoverable. We know the ID of the missing tweet and about the links represented by the arrows pointing to it from the replies to it, which we do have.

If a tweet could not be recovered at all, it is not possible to ascertain where in the cascade it actually belongs. The missing tweet in Figure 4 could, for example, have linked back to the root node, resulting in a cascade with a different depth to the truth. It could have linked to another missing tweet, resulting in a different size from the true value. In lieu of making any false assumptions about the cascade properties there are two options:

1. Remove all conversations in which there is at least one missing tweet
2. Remove all missing tweets, their replies, replies to those replies etc. but keep the rest of the conversations in which those tweets are found

Either way, valuable information is lost. In the end, the second option was chosen. Missing tweets disproportionately affected larger cascades, perhaps because of their propensity to attract controversy and cause users to tweet content that would get them suspended, so to remove all conversations with a missing tweet would have removed 22260 tweets in total. However, removing all missing tweets and their replies removed 441 tweets, which is 1.98% of the tweets in the affected cascades, so the loss of information is comparatively small.

3.2.2 Duplicate tweets and outliers

In some cases, the same conversation was included in more than one data collection. This happened when either one account retweeted another of the ones included in collection, or more than one account retweeted the same tweet from an account that was not included in collection. When considering tweets aggregated from more than one account, for example when

constructing the interaction network of a particular nation, a reduced dataset, with duplicates removed, was used. However, a duplicated conversation is relevant separately to each collection in which it appears. For example, @minhealthnz retweeted @covid19nz on more than one occasion. The resulting cascade is relevant to each of the two accounts separately, but should only be counted once when considering the whole dataset or the New Zealand dataset.

The dataset was also cleaned of outliers, taken to be those conversations containing more than 650 tweets. This removed the largest 3 conversations, comprising 2192 tweets, from the set. The composition of the cleaned and thresholded set is given in Table 2.

Country	Username	Tweets	Cascades	Users
UK	UKHSA	8007	234	3343
	DHSCgovuk	13250	272	6008
	NHSuk	1603	76	1009
	NHSEngland	3078	138	1713
US	CDCGov	8346	148	3615
Can	GovCanHealth	7465	253	2738
	CIHR_IRSC	1085	109	450
NZ	covid19nz	1011	58	237
	minhealthnz	1960	67	805
All	–	40047	1197	16745

Table 2: Number of tweets, cascades and users in the processed dataset, split by account

4 Analysis

This section will discuss: the formation of cascades, the methodology of obtaining their properties, and how the distributions of these properties were compared; the splitting of the dataset by discussion topic, and the fitting and assessment of classifiers to sort unseen tweets into topics; and the formation of user interaction networks and calculation of certain measures of the network structure.

4.1 Reply cascades

Reply cascades were formed by taking the lists of tweet IDs and the IDs each tweet was replying to, and looping over them to create a list of directed graph edge tuples. Conversation IDs were also looped over to label each cascade by its conversation. Following the convention in [21], the edge (t_i, t_j) represents tweet t_i replying to t_j.

4.1.1 Cascade analysis: The Kolmogorov-Smirnov test

The analysis focused on the following particular key properties of the cascades, using the python package NetworkX's implementations:

- Cascade size: the number of tweets within it
- Cascade depth: the length of the longest path between any two replies in the cascade
- Virality: a graph's virality is its Wiener index, the sum of the shortest-path distances between each pair of reachable nodes, normalised by graph size.
  - This is only computable for strongly connected undirected graphs. This value was evaluated on a cascade formed with undirected edges

The **Kolmogorov-Smirnov** (K-S) test is a statistical test of the equality of one-dimensional probability distributions [30]. It can be used to determine the likelihood that two samples were drawn from the same (albeit unknown) distribution – this is known as a two-sample test. The null hypothesis that the two samples are drawn from the same distribution is tested by quantifying the distance between the two samples' empirical distributions.

The empirical cumulative distribution function F_n for n independent and identically distributed observations X_i is defined:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(X_i < x)$$

where $\mathbb{I}$ is the indicator function. That is, the sum of elements in the sample less than x, normalised by the sample size. Then the K-S test statistic is

$$D_{n,m} = \sup_x |F_n(x) - F_m(x)|$$

where F_m, F_n are the distributions being compared. That is, we take the largest absolute difference between the two distributions across all values of x. The p-value corresponding to this test statistic is drawn from the Kolmogorov distribution.

If the test statistic is small or the p-value is high, then there is insufficient evidence reject the null hypothesis (which is that the two distributions are identical [31]). The threshold for what was considered 'high' was 0.05: a comparison was considered significant if the p-value was less that this threshold and non-significant otherwise.

In order to investigate the underlying distributions of the collected data on size, depth and virality, K-S tests were performed on every possible pair among the 9 accounts. This is a total of 108 comparisons: 9 accounts $\times$ 36 possible pairs $\times$ 3 measures (size, depth and virality). Only those pairs of accounts for which either all comparisons were significant or all insignificant are reported.

4.2 Analysis of tweet topics

The accounts used, being primarily agencies concerned with multiple aspects of a whole country's healthcare (with the exception of the specialised `@covid19nz`), did not only tweet about COVID, and not all their conversations concerned COVID. This raises the question of how the cascade properties differed between COVID and non-COVID discussions. Tweets were classified based on the root tweet; it would hardly be unlikely for a reply to a tweet that was not about COVID to itself mention COVID and even spark further replies about COVID, but the conversation as a whole (and the other replies) was not intended to be about COVID. Separating these two types of conversations out was key to performing this analysis: however, since manually checking every single root tweet was infeasible, this separation was done by keyword checking.

Two lists of keywords were used to separate the root tweets in the dataset:

- List 1: tweets that included at least one of these were considered definitely about COVID: ['omicron', 'covid', 'corona', 'coronavirus']
- List 2: tweets that included at none of these were considered likely not about COVID: ['omicron', 'covid', 'corona', 'vaccine', 'jab', 'pfizer', 'moderna', 'astrazeneca']

This split the dataset into 3 groups. The first group comprised COVID conversations: those whose root tweet contained at least one of the words in the shorter list. The second comprised non-COVID conversations: those whose root tweet contained none of the words in the longer list. Of course, this left the third group of tweets: their root tweets contained at least one word in the second list but none of those in the first. These conversations may or may not have been about COVID – we can take them to be ambiguous, and they were excluded from subsequent analysis. For example, several tweets from the UKHSA concerned influenza, and contained such words as 'vaccine' and 'virus' without being about COVID. These would have fallen into the ambiguous group.It is also possible that there were some conversations wrongly designated non-COVID, that simply used different words; the second list was designed to avoid such cases. Lastly, the first list was designed such that there was a vanishing probability that a tweet would contain one of those words and not be about COVID. All tweets were made lowercase before searching to avoid case sensitivity errors.

One question that naturally arises is whether one can predict, based on certain attributes,

whether a conversation is about COVID or not – that is, are the topics characterised by these attributes? Both supervised and unsupervised classifiers might be suited for this purpose.

The attributes used in classification were virality, cascade size, cascade maximum depth, and the number of replies to a tweet made within 10 hours of the root tweet being posted. These features were normalised before the classifiers were applied.

4.2.1 Unsupervised learning: k-means clustering

k-means clustering is an unsupervised learning algorithm which assigns each of n observations (each of which has p features) into one of a predetermined number k of clusters. The assignment is such that each observation belongs to the cluster with the nearest centroid [32]. The algorithm works iteratively; starting with k randomly placed centroids in p-dimensional space, at each iteration each point is assigned to the cluster of its nearest centroid, after which the new centroid of each cluster is recomputed. The algorithm runs until convergence, i.e. until the centroids move a distance less than a predetermined tolerance between iterations. The tolerance used was 10^{-10}.

For the analysis of tweet topics, the clustering should split the data into two groups, corresponding to COVID and non-COVID conversations. The prediction is that the splitting into two clusters is optimal, but this must be verified. This is done by analysing which number of centroids results in the greatest decrease in the loss we wish to minimise, which for k-means is the average distance between each point and its assigned cluster's centroid. Increasing the number of clusters always decreases the loss (as we go from 0 clusters to as many clusters as there are points, points necessarily get closer to the centroids), so we look for 'elbows' in the plot of loss against number of clusters, i.e. where there are particularly sharp decreases.

Measures for the effectiveness of the k-means method include accuracy and confusion matrices [33]. These rely on the true positive (TP) and true negative (TN) values (indicating the correct identification of, in this case, a COVID or non-COVID cascade, respectively), and false positive (FP) and false negative (FN) (respectively the incorrect identification of a non-COVID cascade as COVID, and vice versa). The confusion matrix is so-called because the values are usually represented in a table as in Table 4.2.1.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Clustering accuracy is defined as the proportion of correct identifications, as given by

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

4.2.2 Supervised learning: random forests

A classification decision tree is a decision-making tool which makes a prediction based on several variables, by successively partitioning the variable space and making a prediction of what class each final partition is most likely to belong to [34]. This partitioning can be visualised as a binary tree-like structure. The optimal decision tree is such that, after all splits have been performed on a training set, most of the samples in each region are in the same class [35].

While decision trees have the advantage of being easy to interpret, they are very sensitive to changes in the data; even a small change in the data can induce a large change in optimal tree structure. Decision trees also frequently have a relatively low accuracy. Deeper trees also tend to overfit to training data, whereas shallower trees have lower accuracy [35].

These shortcomings can be remedied by the fitting of a random forest to the data instead. A random forest is a classification (or regression) method that fits multiple decision trees to a training set [36]. The training algorithm for random forests utilises bagging (bootstrap aggregating). Given a training set X with responses Y, the bagging procedure involves sampling randomly with replacement B from the training set and training responses, and fits trees to the sample. The output of the random forest is the class selected by the most trees.

As this is a supervised learning method, it is necessary to check its performance on unseen data, which was done using cross validation.

4.2.3 Cross-validation

Cross-validation is a method used to ensure that a model is optimally trained, such that it does not overfit a training set. It involves splitting the data into k roughly equal subsets; at each of k iterations, one is taken as the test set and the rest of the data is pooled into a training set. The model is fit to the training set and evaluated on both the training and test sets. At the end, the average of all testing and training accuracies is reported. In this case the absence of an explicit training set necessitated the use of cross-validation to verify model efficacy.

4.3 User networks

User interaction networks were constructed by looping over the lists of user IDs and the IDs of the users who were being replied to in order to form a list of directed graph edge tuples. Following once again the convention in [21], the edge (u_i, u_j) represents user u_i replying to u_j. This edge is also associated with the weight w_{ij}, proportional to the number of times u_i replied to u_j.

Occasionally users reply to themselves, in order to write more than can be contained within a single tweet. Such self-replies were not included in the graphs, as they serve no conversational purpose but can cloud useful results by creating a heavily weighted self-loop which falsely skews the distribution of degrees. Cleaning of duplicates, as described in Section 3.2.2, was vital here so that the same link did not appear falsely more than once within a country's graph and incorrectly increase edge weights. There were 15 tweets for which user lookup failed. These were also excluded, as these do not have the requisite user information needed to make an edge.

Visualisation of these networks, as well as community detection and computation of centrality, was carried out using the programme Gephi [37].

4.3.1 The PageRank Algorithm

PageRank is an algorithm which measures the centrality or importance of a node by counting the number and quality of links to it, based on the principle that more important nodes are likely to be linked to more times by others [38]. PageRank is also used by Twitter to determine who a given user might want to follow [39]. The PageRank of a node u in a network is denoted $PR(u)$ and must satisfy the property that

$$PR(u) = \sum_{v:(v,u)\in E} \frac{PR(v)}{d_{out}(v)}$$

Where E is the network's edgelist, and $d_{out}(v)$ is the out-degree of the node v, or its number of outgoing edges. That is, the rank of a node is the sum of the ranks of the nodes that link to it, divided by the out-degree of those nodes [40].

It is calculated by solving the following [40]:

$$PR(u) = \alpha\frac{1}{n}(I - (1-\alpha)A_{out}D_{out}^{-1})^{-1}\mathbf{1}$$

where A_{out}, D_{out} are the outward adjacency matrix and the diagonal matrix of node out-degrees, respectively; α is a given probability value, usually set to 0.15 or something similarly low; and I, $\mathbf{1}$ are the identity matrix and 1-vector of appropriate size, respectively. Mathematically, the algorithm is equivalent to considering a weighted random walk in which the random walker follows the transition matrix of the graph $A_{out}D_{out}^{-1}$ with probability α, transitioning to any node with probability $(1-\alpha)$.

4.3.2 Community detection: modularity and the Louvain method

Modularity is a measure of the structure of networks or graphs which quantifies the strength of division of a network into modules [41]. It is the difference between the fraction of the edges that fall within the given groups and the expected fraction if edges were distributed at random. Commonly, this randomization of the edges is done so as to preserve the degree of each vertex. For directed graphs modularity is calculated as follows [42]:

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i^{in}k_j^{out}}{2m}\right]\delta(c_i, c_j)$$

where A_{ij} is the edge weight between nodes i, j; k_i^{in}, k_j^{out} are the sum of the incoming and outgoing weights of the edges attached to nodes i, j respectively; m is the sum of all edge weights in the graph; c_i, c_j are the communities in which the nodes i, j respectively are; and δ is the Kronecker delta function (1 if the nodes are in the same community and 0 otherwise).

Modularity is often used in community detection, by choosing the optimal partition into communities as that which maximises modularity. The higher the modularity, the better the

community structure. The Louvain community detection algorithm [43] utilises two passes to efficiently maximise modularity:

1. Each node in the network is assigned to a community of its own. Then each node i is removed from its own community and joined with the community of each of its neighbours j. The resulting change in modularity is calculated:

$$\Delta Q = \left[\frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

   Where Σ_{in} and Σ_{tot} are, respectively, the sum of all the weights of the edges *inside* the community i is moving into, and the sum of all the weights of the edges *to* nodes in that community. k_i is the weighted degree of node i, $k_{i,in}$ is the sum of the weights of the edges between i and other nodes in the community that it is moving into, and m is as before.

   After this process has been completed for all the neighbouring communities of i, i is moved into a neighbouring community such as to maximise ΔQ.

2. Secondly, all nodes within the same community are grouped to form a new network where each node is a community from the previous network. Within-community edges are represented by self-loops. Links to a node in a different community from multiple nodes in a community are represented by a weighed edge. The first step is then applied on the new network, and algorithm repeats.

5 Results

5.1 Cascades

5.1.1 Properties of the data

Tables 9 (see Appendix) and 3 give the statistics for the distributions of cascade size, depth and virality for all the accounts, both separately and aggregated, for the raw and processed datasets, respectively.

Country	Username	Size		Virality		Depth	
		mean	std	mean	std	mean	std
UK	UKHSA	34.22	59.70	1.631	0.7251	4.910	4.100
	DHSCgovuk	48.71	92.11	1.306	0.9362	4.592	5.569
	NHSuk	21.09	62.30	1.510	0.4578	3.343	2.413
	NHSEngland	22.30	61.51	1.534	0.6565	3.710	4.111
US	CDCGov	56.39	103.0	1.532	0.4591	4.770	4.225
Can	GovCanHealth	29.50	60.28	1.573	0.6675	4.379	5.209
	CIHR_IRSC	9.954	37.11	1.652	0.4545	2.954	1.960
NZ	covid19nz	17.43	7.819	2.709	0.6638	10.78	3.868
	minhealthnz	29.25	71.63	1.860	0.5496	5.612	3.259
All	–	33.99	76.17	1.675	0.747	4.739	5.135

Table 3: Cascade statistics by account after processing. Values reported to 4sf

Figure 5 gives the distribution of values for the size, depth and virality measures for the whole processed dataset. The distributions of values for these measures for the individual accounts are given in Figures 15,16 and 17, which are given in the Appendix.

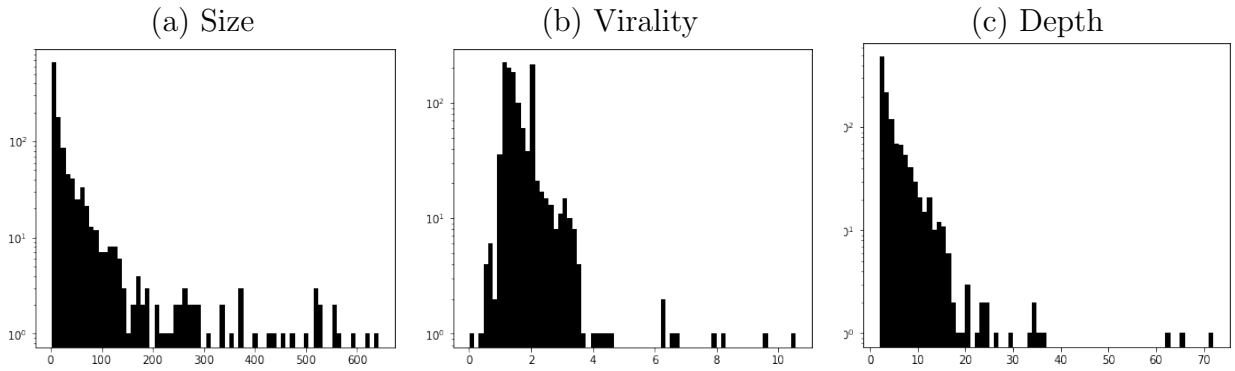

Figure 5: Semilog histogram plots for (a) cascade size, (b) virality, and (c) maximum cascade depth for whole dataset

Figure 6 gives a histogram of the number of replies made at discrete time intervals after the posting of the conversation's root tweet.

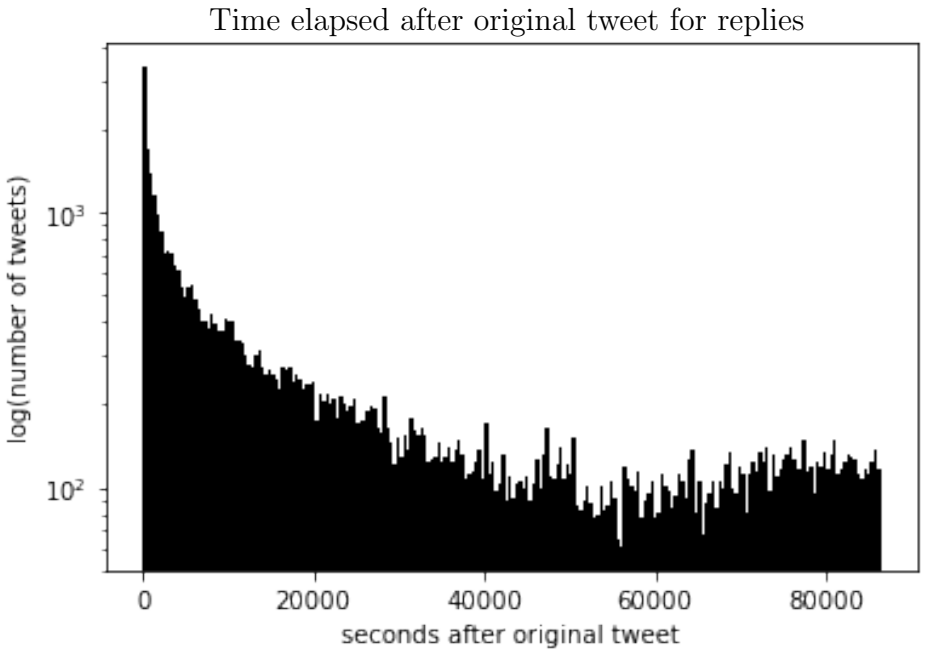

Figure 6: Histogram of the distribution of time elapsed after original tweet for each reply (logarithmic scale)

Table 4 gives the number of root tweets in each account's collection which was tweeted by that account (i.e. how many roots were original, not retweeted).

Country	Username	Tweets	Original tweets	Percentage original
UK	UKHSA	234	109	46%
	DHSCgovuk	272	163	59%
	NHSuk	76	53	69%
	NHSEngland	138	109	78%
US	CDCGov	148	137	92%
Can	GovCanHealth	253	182	72%
	CIHR_IRSC	109	22	20%
NZ	covid19nz	58	58	100%
	minhealthnz	67	54	81

Table 4: Percentage of tweets in each collection that were original

5.1.2 Distribution comparisons

The results of the K-S tests are given in Figure 7. Pairs of accounts that yielded a significant p-value (i.e. $p < 0.05$ – see Section 4.1.1) in all three comparisons – on size, depth and virality measure distributions – are marked S. Pairs whose comparisons yielded 3 non-significant p-values are marked N. Pairs which yielded both significant and insignificant comparisons are left blank.

	UKHSA								
UKHSA		DHSCGovUK							
DHSCGovUK			NHSUK						
NHSUK				NHSEngland					
NHSEngland			N		CDCGov				
CDCGov	N					GovCanHealth			
GovCanHealth	N						CIHR_IRSC		
CIHR_IRSC	S	S			S	S		covid19nz	
covid19nz	S	S	S	S	S	S	S		minhealthnz
minhealthnz		S	S	S	S		S	S	

Figure 7: Table showing which pairs of accounts yielded significant (S) and insignificant (N) K-S test comparisons on all 3 attributes

Figure 8 gives the distribution of p-values from the significant and non-significant comparisons.

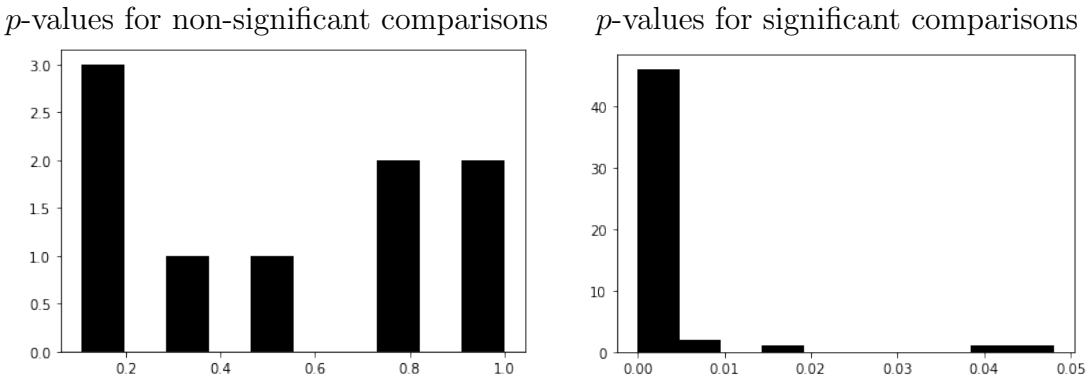

Figure 8: Distributions of p-values

5.2 Topics

5.2.1 Statistics

In total, there were 26265 tweets from 503 conversations in the COVID dataset, and 15507 tweets from 646 conversations in the non-COVID dataset. Table 5 gives the summary table of dataset structure.

Topic	Tweets	Cascades	Users
COVID	26265	503	10505
Not COVID	15507	646	8290

Table 5: Number of tweets, cascades and users in the COVID and non-COVID dataset

Figure 9 gives plots of the distributions of size, depth and virality for the two datasets, and scattergraphs comparing each pair of variables.

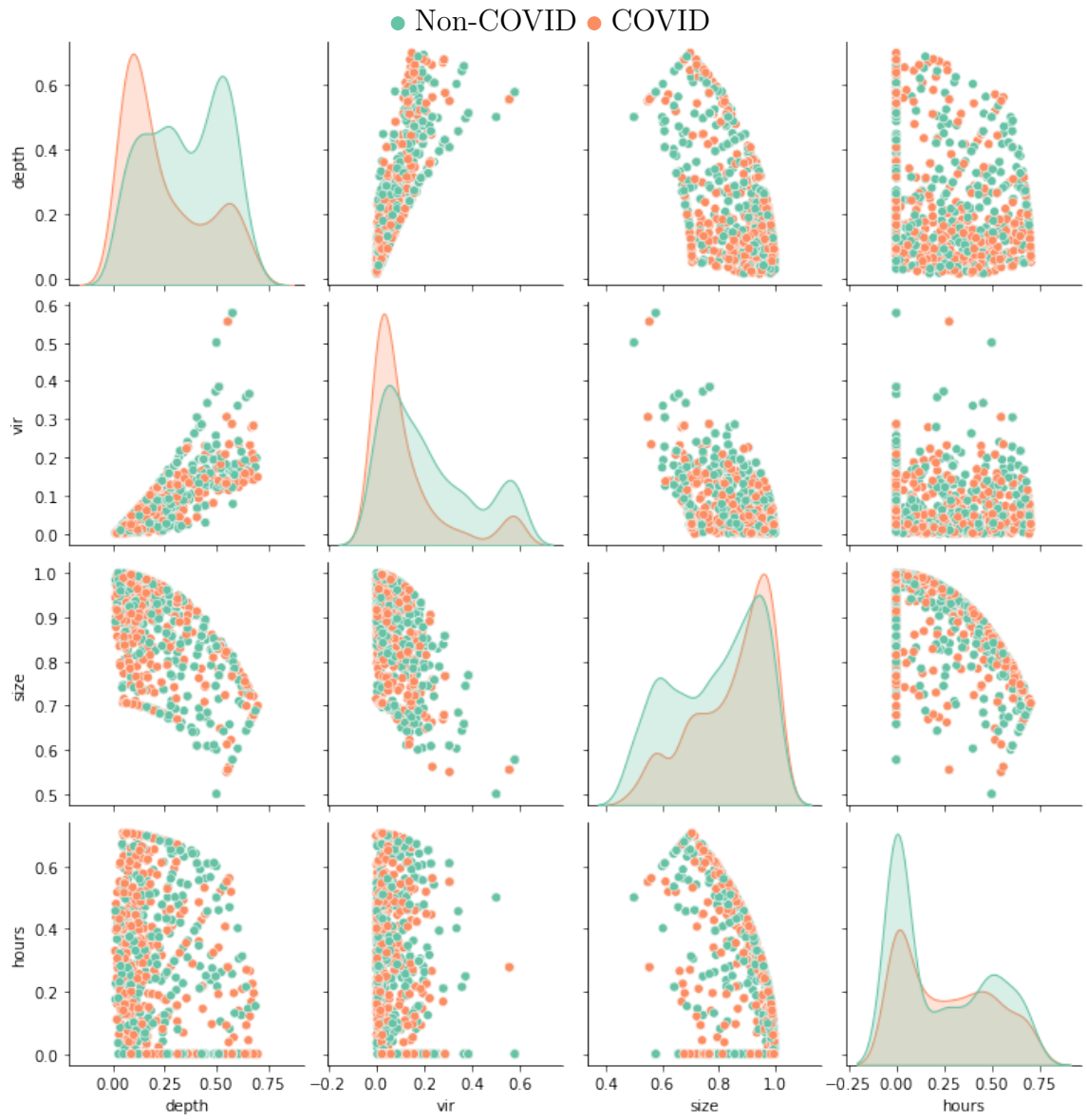

Figure 9: Pairplot comparing the distributions of the attributes between topics. The diagonal cells show marginal distributions of those attributes, and off diagonals scatter plots of each pair of attributes

K-S test p-values for comparisons of the distributions of size, depth and virality between the two sets are given in Table 6.

Measure	p-value
size	1.167×10^{-28}
depth	1.241×10^{-7}
virality	6.926×10^{-7}

Table 6: p-values for the K-S tests between the distributions of the measures of the COVID and non-COVID datasets

5.2.2 k-means clustering

Accuracy for the clustering was 59.05% and the confusion matrix is given in Table 7.

		Predicted class	
		COVID	non-COVID
Actual class	COVID	288	349
	non-COVID	115	381

Table 7: Confusion matrix for k-means clustering

Figure 10 gives the variation of loss (average distance from centroid) against number of clusters in the k-means algorithm. This was plotted both with and without including the measure of tweets within 10 hours, in order to investigate the effect of this measure on the performance of the model.

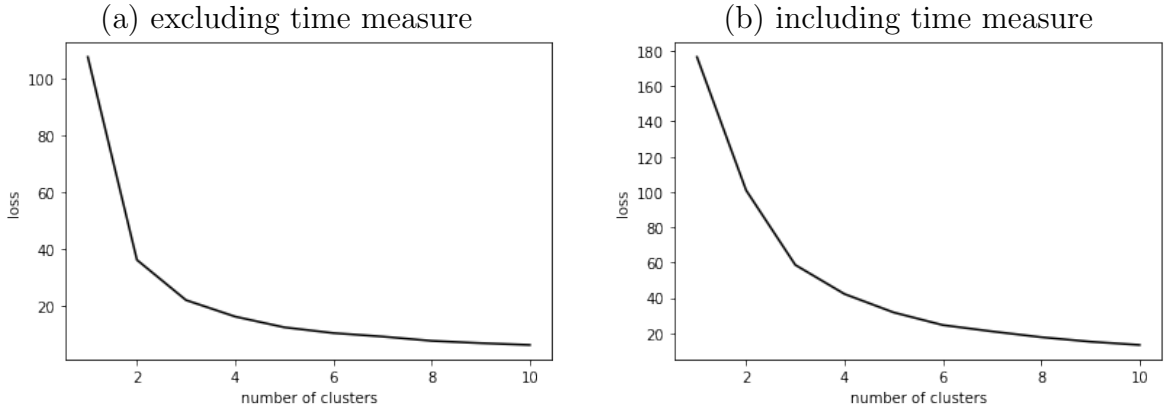

Figure 10: 'Elbow' plot showing how loss decreases with number of clusters for the data: (a) excluding temporal measure, and (b) including temporal measure

5.2.3 Random forests

Running the Random Forest algorithm on 10-fold cross-validation yielded a mean training accuracy of 90.03% and a test accuracy of 63.30%.

5.3 User interactions

Figure 11 gives a reduced version of the user interaction network for the whole processed dataset. There were 1746 nodes and 25867 edges in the original dataset, so the network diagram in its original form appears cluttered. First, modularity was computed via the Louvain algorithm and the PageRank for each node were calculated. Then the node sizes were scaled proportional to PageRank and the nodes coloured according to their modular class. Finally, any node with a degree of 3 or less was removed from the visualisation. The users with the highest PageRank are labelled with their username.

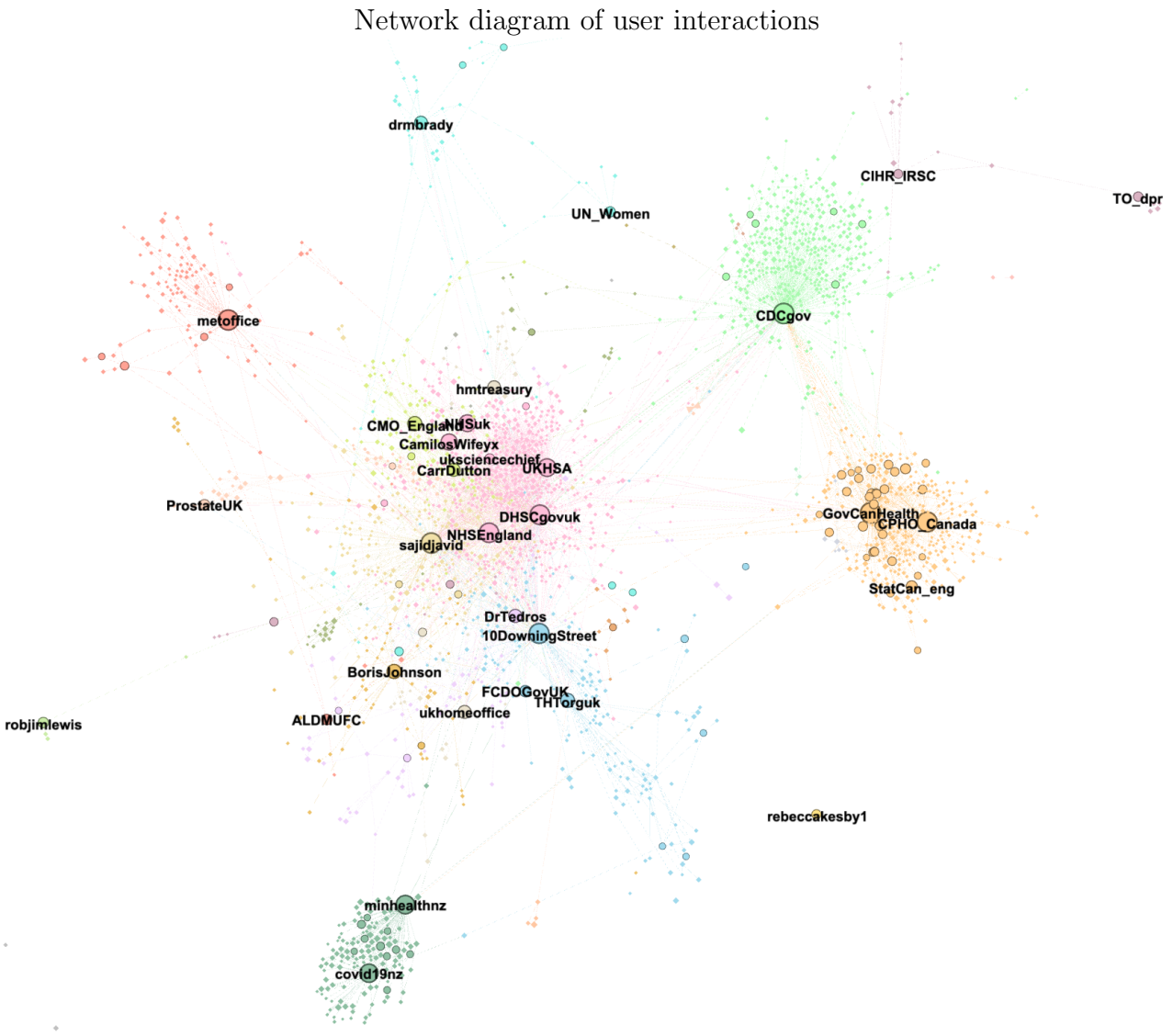

Figure 11: Network diagram with modularity for community detection applied. Node colour corresponds to modular class, and node size to PageRank. Some nodes have been omitted for clarity.

Figures 18, 19, 20 and 21 give similar user interaction network diagrams for the processed datasets limited to each country. Nodes with degrees below 3 were removed after calculation of PageRank, degree and modularity in order to make the diagram clearer. These diagrams are in the Appendix.

The number of nodes and edges in each graph are given in Table 8. Note that the number of nodes in the networks is not the same as the sums of the users in each account's dataset (see Table 3.1.3) because users may appear in more than one account's dataset and must not be counted twice in the country dataset.

Country	Nodes	Edges
UK	10268	15139
US	3620	4972
Can	3138	4318
NZ	954	1440
All	17466	25867

Table 8: The number of nodes and edges in user interaction network for each country

Once again, node size corresponds to PageRank and colour to modular class, and the users with the highest PageRank are labelled with their username. The distribution of PageRank values for each account is given in Figure 12.

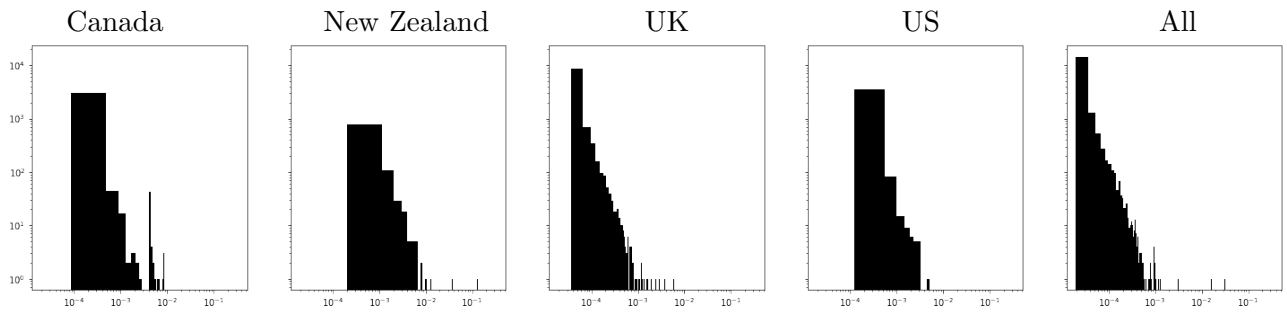

Figure 12: Distribution of PageRank values for the user networks of each country, log-log scale

Visualisation algorithms were Force Atlas with no overlapping enforced as much as possible, and Yifan Hu Proportional [44].

5.3.1 Most central accounts

In order to evaluate popularity and importance in the user interaction networks, for each geographic region the top-k users (those with the highest PageRank) in each of the n largest modularity classes were identified. Then each of the top-k users was marked according to whether they were verified, and their follower count was taken.

For all countries, k was taken to be 3. As can be seen in Figure 13, the distribution of class size was somewhat variable between countries.

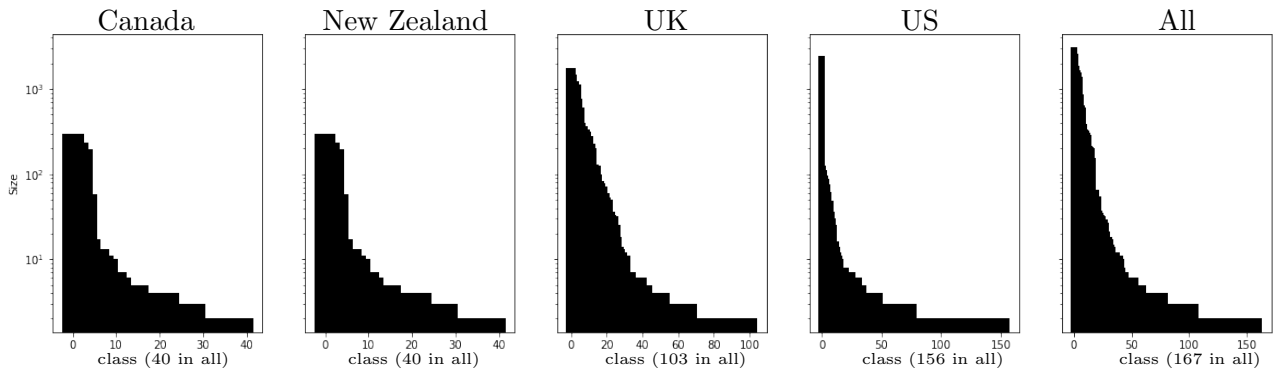

Figure 13: Distribution of the modularity class sizes

n was taken to be 4 as this number of classes contained most of the users for all graphs. For all countries but the UK, the 4 largest classes contained between 70 and 80% of the users; for the UK the 4th largest classes only formed 54% of the users. The 4 largest classes in the graph for all countries contained 85% of the users. The results are given in Figure 14. The Louvain algorithm detected different communities in the graph of all users than in the one limited to individual countries, even though each country's network is a subgraph of the all-user network. For this reason, the rightmost column (which concerns the all-user network) displays proportions independent to those of the other four columns.

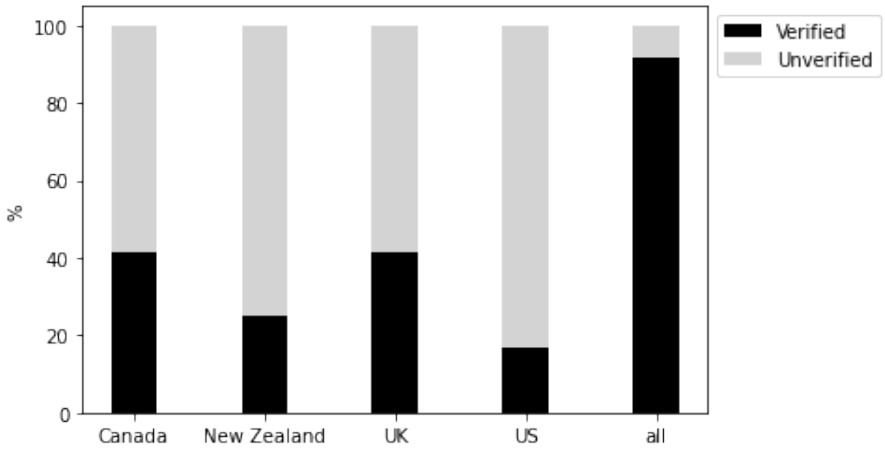

Figure 14: Stacked bar chart showing the proportion of the 3 most central users in the 4 largest modularity classes of each network that were verified. Classes in the all-user network (rightmost column) were calculated separately from those in the country networks. They, and their composition, are therefore independent from those of the other networks

In order to look more deeply into who was most influential in each network, the most central users in each of the largest modularity classes in each graph are presented in Table 10, with their verified status, their follower count and notes about their identity – whether they are a public figure, and if so, who they are. Results are reported in descending order of modularity class size. The 6 largest classes are reported for the whole dataset due to its size. The table can be found in the Appendix and also lists the sizes of the largest classes.

6 Discussion

6.1 Cascades

Table 2 shows that the accounts whose cascades had the largest number of tweets were @UKHSA (UK Health and Safety Authority), @DHSCgovuk (UK Department of Health and Social Care) and @CDCGov (US Center for Disease Control) (in descending order). The distributions of values for cascade sizes for @DHSCgovuk and @CDCGov have large spreads and the largest means. Additionally, the outlier conversations which were removed during thresholding belonged to these accounts. This could be explained by the fact that these accounts belong to primary government executive and advisory bodies for health legislation in their respective countries. This may be why these accounts tweet more and generate larger conversations.

Table 3 and Figure 5 show that across all the accounts, the distributions of values for the size and depth of cascades are skewed very small. In practical terms, many more tweets garnered the interest of a few people than attracted mass attention. The skew of the cascade depth distributions indicates that users are more likely to respond to the original tweet or a low-level reply rather than replying to or continuing a lengthy conversation.

The same can be seen in the distributions of size and depth when separated by individual accounts, shown in Figures 15 and 16. Many of the size distributions show a large degree of skew, as demonstrated by the standard deviation being larger than the mean. Similarly, many of the standard deviations of the depth distributions have a standard deviation comparable to or larger than the mean, and show skew (see Figure 5 again). A notable exception are the New Zealand accounts, which have a more central and evenly spread distribution of depths – users in this dataset were more likely to maintain a consistent level of interest in continuing or contributing to a lengthier conversation.

Figures 5 and 17 show that the distribution of virality has a less extreme skew than the other two measures, both across the whole dataset and for individual accounts. The distributions for individual accounts all show a similar mean and similar degree of spread, with the obvious exception being @DHSCgovuk, which appears to display a two-humped distribution corresponding to a slew of low-virality tweets. @DHSCgovuk is one of the most prolific accounts in the data, and it seems that a large part of this generous output was tweets that failed to spark interest.

Interestingly, cascade sizes did not seem tied to country size. The New Zealand accounts, originating in a country of 5 million, had similar - indeed larger – mean sizes than the Canadian, coming from a country of 38 million. Similarly, the population of the UK is 67 million, nearly 5 times less than the US at 331 million, but the distributions of size values did not reflect this.

As for the distribution of tweets in time after posting of the original tweet (Figure 6), the plot shows an initial flurry of activity, then sharp decrease until around 17 hours. After this point, there was neither a flat plateau nor a continuing decline, but in fact a small increase in activity.

6.1.1 Account comparisons

Figure 7 shows that the New Zealand accounts and @CIHR_IRSC (Canadian Institutes of Health Research) displayed significant p-values in most of their comparisons, with @covid19nz (Unite against COVID-19 New Zealand) having only significant comparisons and the other two accounts only having two comparisons that were not significant on all 3 measures. A significant p-value is evidence to reject the null hypothesis, that the underlying distributions are the same. The distribution of the p-values for these comparisons (see Figure 8) shows that almost all are very small indeed, mostly under 0.02 and only two greater than 0.04.

@covid19nz was the only account significantly different to all the others; its distributions of depth and size values had higher means than the others, particularly for depth, hinting at greater engagement in the form of larger conversations which were more likely to be continued by users. @covid19nz was the only account specifically designed to report about COVID, while the other accounts were less targeted and frequently posted about other health-related topics. Table 4 shows that @covid19nz was also by far the most original account, with no retweets. @CIHR_IRSC was the only account run by a research institute, while the others were mainly ministerial departments. @CIHR_IRSC was also by far the least original account, with 80% of its tweets being retweeted from other sources. These differing purposes and properties may explain why these accounts differed so heavily from the others.

@minhealthnz (Ministry of Health New Zealand) has less distinguishing characteristics: it is a ministerial account, like several of the others; though small, it is of similar size to several others in terms of number of conversations and tweets; nor is it remarkably original (at 81% original tweets it is between @NHSEngland with 78% and @CDCGov with 91%). The accounts with which it did not have exclusively significant comparisons were @UKHSA and @GovCanHealth (Health Canada), with which it @minhealthnz has key similarities: the accounts are all health agencies with similar purposes. As will be discussed, the comparison between those @UKHSA and @GovCanHealth was highly non-significant.

The pairs of accounts that yielded exclusively non-significant comparisons were @UKHSA and @CDCGov, @UKHSA and @GovCanHealth, and @NHSEngland and @NHSuk. This is evidence that the null hypothesis should not be rejected, and suggests that the distributions are likely the same for these accounts.

A possible reason for the similarity between the distributions of @NHSEngland and @NHSuk might be that they are both run by the NHS (UK National Health Service) for the purpose of providing general health advice to UK citizens. @NHSuk had around half the number of tweets and cascades than @NHSEngland. This is surprising: England is a constituent of (and thus smaller than) the UK and @NHSuk is an older account with more followers. Two possible explanations may be that users are more likely to engage with a more local authority, or that the more high-level @NHSuk limits itself to more relevant tweets, but either way the volume of replies is proportional to the volume of tweets between the two.

For the comparisons between @UKHSA and @CDCGov and between @UKHSA and @GovCanHealth, the reason for the similarity might be due to the fact that they are run by organisations with similar purposes – executive government agencies.

6.2 Topics

Table 5 shows that while there were more conversations in the non-COVID dataset, there were more tweets in the COVID dataset. It seems that while more conversations are being started about topics other than COVID, they fail to generate lasting and involved discussion. This could be because users are more concerned with discussing COVID, or that root tweets about COVID are more relevant. The ratio of tweets to cascades for COVID conversations is 53.2, while the ratio for non-COVID conversations is less than half this, at 24.0. However, there are only 2215 more users in the COVID set, which is 21.1% of the total COVID users. This suggests that a conversation being about COVID was enough to attract significantly more replies, but not to attract proportionally as many new people.

As for the K-S tests of the distributions of measures, the p-values (see Table 6) are all extremely low. This indicates there is good evidence to reject null hypothesis and conclude that the data are likely to be drawn from different distributions. Since the ratio of tweets to cascades is much lower for non-COVID conversations it is unsurprising that the distributions of cascade size differ significantly between the two topics. The depth and virality p-values also indicate that the distributions of these measures also differ significantly between accounts.

Visual inspection of Figure 9 reveals that the datapoints may not be linearly separable by category. In those regions containing more points, there is often a seemingly random distribution of both classed of conversation. There is also overlap in the distributions of the measures of tweets within 10 hours and virality for the two topics. The histograms on the diagonal also demonstrate the differences between the depth and virality of the topics. Though they attracted fewer replies, it seems non-COVID conversations are better at inspiring deeper conversation, while COVID conversations are large but shallow, as more people reply to the root or low-level replies. While the distributions of virality have the same two-humped shape, the non-COVID conversations have a higher hump for higher virality and a lower hump for lower. Also, within 10 hours of the root being posted, non-COVID conversations were typically smaller than COVID conversations.

6.2.1 k-means Clustering

k-means clustering resulted in an accuracy score that was better than chance, though the score warrants investigation into where inaccuracies occurred.

Looking to the confusion matrix in Table 7, there was a great deal of mis-classification, particularly with the positives: there was a tendency to false negatives, and only 45% of the positives were correctly classified. However, most negatives (76%) were properly classified. This may because while there are many regions of the variable space in which in which there were both many COVID and non-COVID conversations, there are more and larger regions of only non-COVID conversations than only COVID conversations.

The elbow plot generated without the time measure (Figure 10a) shows a very clear elbow at 2 clusters. This is less clear when including the time measure (Figure 10b) – then there is a shallower curve and equal change in loss at 3 clusters, indicating that time measures induces more granularity. In particular may indicate some subclustering, which means it may

be worth investigating further the causes and correlations with time taken to reply. Perhaps it might be because a non-COVID conversation featured a reply about COVID that derailed the conversation. This may also explain the significant tendency towards false negatives.

6.2.2 Random forest

The random forest performed much better on training set but didn't generalise as well, though maintaining an accuracy much better than chance.

The improvement could be because the successive partitioning of the variable space deals much better with how the datapoints from both classes are spread; k-means can only assign to one of two distinct splits of the space, but the random forest is able to deal much better with the fact that many areas of the variable space contain points from more than one class. The model might be further improved with more data, allowing more robust model fitting.

It is also possible that the model would have performed better if trained on a different set of variables. To investigate this, leave-one-out cross-validation could be performed to investigate which variables have the most impact on the fit, and potentially additional attributes could be introduced and checked in the same way.

6.3 User interactions

From Figure 13 it can be seen that the US data has a clear single largest modularity class, and the Canada data has two clearly largest classes. Canada' largest two classes have size 10 times that of the next largest, and the US's largest class is 20 times larger than the next largest. This is probably because data collection only focused on one account in the US (@CDCGov) and two in Canada (@CIHR_IRSC and @GovCanHealth) but nonetheless, the other two graphs do not show this kind of property: for example, the New Zealand collection focused on 2 accounts, but there are not 2 clear largest modularity classes. This is likely linked to the fact that the @CDCGov was a highly original account, with only 9% retweets; it did not allow for the introduction of outside communities in the same way that less original networks did. The Canadian accounts are less original, but it may simply be that their retweeted tweets were less influential or relevant and did not generate discussion. These properties can also be seen in the network diagrams (Figures 18, 19, 20, 21 and 11).

Figure 12 shows that some networks (Canada, US) have a much sharper dropoff than others (New Zealand, UK) between the highest and next-highest PageRank values. This also reflects the distribution of modularity class size as discussed above; membership of a large community allows for users to become more influential compared to those in a smaller community.

Figures 18, 19, 20, 21 and 11 are not results in and of themselves (as in [14]), but serve to illustrate the other results. For example, it can be seen how higher PageRank users have more connections to other users, or that modularity classes (designated by colour) often have more connections within them. The sizes of these classes are clearly visible, and it becomes clear that some communities are more insular than others – these are usually based on popular retweets.

For example, there are lots of very large and insular communities in the UK network in par-

ticular. This was because the accounts involved in this collection frequently retweeted each other, as well as other governmental accounts. These retweets often concerned topics which were unique within the dataset. For instance, a distinct community formed around @metoffice tweets retweeted around the onset and aftermath of Storm Eunice in February; the storm posed a number of injury risks to the public.

Table 10 shows that the most central users in each class were generally verified public figures with followings mainly in at least the tens of thousands, especially in the larger classes. This is expected, these were all very famous, influential outside of Twitter with a clear political link to public health.

However, in smaller classes, particularly when splitting by country, it was possible for ordinary people with smaller followings to dominate a community. For example one Canadian class's most central user had 4 followers but beat even the Canadian government department of Heritage's account in terms of PageRank.

In Figure 14 it is shown that the top 3 of the largest 4 classes were mainly unverified personal accounts. People getting stuck in have a lot of power and can have a lot of influence. However, the proportions of verified accounts in the UK and Canada's pools of most influential users were around twice those of the US and New Zealand. This reflects the UK network's tendency to insular networks focused around a retweet from an official account, something the Canadian network also displays to a lesser degree but which is a much less prominent phenomenon in the other two networks. The most influential users in the largest communities detected in the all-user network are mainly verified; when taking the most general view possible, it is expected that their influence might be more prominent.

7 Conclusion

This project set out to investigate how conversations on about issues related to Twitter differ between national healthcare accounts, as well as how they differ based on whether they are about COVID or not. It also aimed to investigate community formation and the driving force behind user popularity within the emergent user interaction networks.

The distributions of values of cascade size suggest that a lot of people get involved a little, and few get involved a lot; many more tweets garnered the interest of a few people than attracted mass attention. The distribution of values of cascade depth indicates that users are more likely to respond to the original tweet or a low-level reply rather than replying to or continuing a lengthy conversation, except for users in New Zealand dataset, who were more likely to maintain a consistent level of interest in continuing or contributing to a lengthier conversation. The distribution of values of cascade depths for @DHSCgovuk suggests that while this is one of the most prolific accounts in the data, a large part of this output was low-engagement tweets, a high proportion of which are retweets, that did not inspire large-scale user interaction.

Analysis of the distribution of replies in time after the posting of the original tweet shows a pickup in replies after 17 hours, following an initial burst of activity and a subsequent sharp decline. This may correspond to a next-morning rush to continue conversations from the

previous evening – further investigation into when in the day root tweets were posted might shed light on this.

Comparisons of the distributions of cascade size, depth and virality distributions between pairs of accounts reveals that only account that was significantly different from all the others is @covid19nz. This was the only targeted COVID account and also the only account putting out 100% original tweets. It was also one of the accounts with highest mean virality and one of the few accounts whose depth did not skew close to 1. Clearly, engagement is different for focused original accounts, to the point of being more involved and impactful, and more able to generate greater discussion. This may be because of the trustworthiness and reliability associated with focused accounts that are dedicated to their own content.

The results support the idea that measures of size, depth and virality are a good reflection of this difference in perception, suggesting that these measures may be used to predict perception and engagement in contexts other than healthcare, as it might in fact be possible to tell the difference in account types entirely from these properties. In order to fully investigate this, further work could look into comparing the whole corpus of tweets from focused accounts with tweets on that topic from a general account: for example, all of @covid19nz's tweets with only those @DHSCgovuk tweets that focused on COVID. Further work could also look into whether this reflection persists with the inclusion of other network properties, such as breadth and branching, as measures. From the findings it seems that accounts that are able to generate the most impact are those which are focused and mainly post original content, especially when it comes to reacting to a crisis such as the COVID pandemic.

The other very different account in terms of mainly significant comparisons and differing account type is @CIHR_IRSC. It was the least original account by some way, and the only one belonging to a research group. Its depth and size distributions had relatively small mean compared to those accounts with which it was significantly different. Based on the findings with @covid19nz, it seems that a switch to original content might drive higher engagement with research-based accounts.

The differences between the distributions of the values of size, depth and virality between the two topics were significant, which supports the idea that these properties can be used to differentiate between accounts. The classification results are also promising. To drive up the accuracies of the clustering and random forests, leave-one-out cross-validation might be used to investigate the impact each measure has on the model performance, and to investigate the potential introduction of other measures, such as breadth or branching. Investigation of the attention garnered by a tweet, such as its retweet and favorite counts or those of its quote tweets, might also be useful measures of a tweet's popularity.

The subclustering result in k-means is also interesting. Further work could look more closely at the temporal properties of these cascades and how time-evolution of discussions varies between topics, as well as when in the day and week different types of conversations are started and continued. This might reveal the underlying data structure driving the 3rd cluster 'detected' by the inclusion of the temporal measure.

Future work might also link investigation of topics with the differences in accounts, splitting

each topic's dataset by account to see whether the differences in topic focus are driving the differences in the distributions of account properties. It could also look at the effectiveness of different sets of keywords in topic splitting, or bolster the splitting process with manual checking.

As expected, verified public figures and organisations drive much of the conversation in the networks. Particularly in the largest communities, such accounts emerge as the most central users. However, smaller communities and second, third etc. most central users are much less prone to this: personal accounts with a low following have the power to become very central and drive ongoing conversation, and are simply less likely to be the single most central user. Further work could investigate other centrality measures, such as weighted degree and eigenvector centrality, to complement investigation of PageRank, as well as looking in more depth at the number of followers an account has and the ratio between followers and friends. This could help form a more complete profile of users within a network in terms of their popularity and influence both within and outside the network in question. Further work could also investigate to what extent highly central users become involved in deeper conversations. Time-dynamic versions of the user interaction network structures could provide more insight into the evolution of discussions that would complement the discussion of structure carried out in this project.

This project was limited in scope by the amount of data it was possible to collect within the project time frame and using a single Twitter Developer Account. Expanding the focus to more countries and more accounts from within those countries may allow for more robust results, or even a comparison with accounts that tend to spread misinformation. The decision to exclude personal accounts (e.g. health ministers) from collection and analysis may not have to be made by a project which was able to account and mitigate for the different ways personal and organisational accounts are perceived by users.

To conclude, his project has shown that it is possible to use measures of cascade size, depth and virality to predict whether a conversation is about COVID or not, and to tease out differences in the purpose and reach of an account. Investigations of user interaction networks also revealed the power private individuals have to become central in a sizeable community within all interactions discussing public health on a national or global scale.

8 Bibliography

References

[1] Agrawal P;. Twitter userpage [accessed 23.05.2022]. Available from: https://twitter.com/paraga.

[2] Dorsey J;. [Tweet] available from jack @jack, [accessed: 23.05.2022]. Available from: https://twitter.com/Jack/status/20.

[3] Enli GS, Skogerbø E. Personalized campaigns in party-centred politics: Twitter and Facebook as arenas for political communication. Information, communication & society. 2013;16(5):757-74.

[4] Gorodnichenko Y, Pham T, Talavera O. Social media, sentiment and public opinions: Evidence from# Brexit and# USElection. European Economic Review. 2021;136:103772.

[5] Rufai SR, Bunce C. World leaders' usage of Twitter in response to the COVID-19 pandemic: a content analysis. Journal of public health. 2020;42(3):510-6.

[6] Vosoughi S, Roy D, Aral S. The spread of true and false news online. Science. 2018;359(6380):1146-51.

[7] Ekman M, Widholm A. Twitter and the celebritisation of politics. Celebrity studies. 2014;5(4):518-20.

[8] Kwak H, Lee C, Park H, Moon S. What is Twitter, a social network or a news media? In: Proceedings of the 19th international conference on World Wide Web; 2010. p. 591-600.

[9] Twitter. About Verified Accounts;. [accessed 25.05.2022]. Twitter Help Center. Available from: https://help.twitter.com/en/managing-your-account/about-twitter-verified-accounts.

[10] Twitter. Twitter Glossary;. [accessed 23.05.2022]. Available from: https://help.twitter.com/en/resources/glossary.

[11] Saveski M, Roy B, Roy D. The structure of toxic conversations on Twitter. In: Proceedings of the Web Conference 2021; 2021. p. 1086-97.

[12] Ott BL. The age of Twitter: Donald J. Trump and the politics of debasement. Critical studies in media communication. 2017;34(1):59-68.

[13] Zubiaga A, Spina D, Martínez R, Fresno V. Real-time classification of twitter trends. Journal of the Association for Information Science and Technology. 2015;66(3):462-73.

[14] Bruns A. How long is a tweet? Mapping dynamic conversation networks on Twitter using Gawk and Gephi. Information, Communication & Society. 2012;15(9):1323-51.

[15] Rogers R. The end of the virtual: Digital methods. vol. 339. Amsterdam University Press; 2009.

[16] At 199 million, Twitter logs 20% user growth as pandemic posts surge; 2021. [accessed 24.05.2022]. Business Standard News. Available from: https://www.business-standard.com/article/technology/at-199-million-twitter-logs-20-user-growth-as-pandemic-posts-surge-121043000235_1.html.

[17] Rosen A, Ihara I. Giving you more characters to express yourself; 2017. [accessed 25.05.2022]. Twitter Blogs. Available from: https://blog.twitter.com/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.

[18] Goel S, Anderson A, Hofman J, Watts DJ. The structural virality of online diffusion. Management Science. 2016;62(1):180-96.

[19] Twitter. Twitter Timeline;. [accessed: 23.05.2022]. Available from: https://help.twitter.com/en/using-twitter/twitter-timeline.

[20] Juul JL, Ugander J. Comparing information diffusion mechanisms by matching on cascade size. Proceedings of the National Academy of Sciences. 2021;118(46).

[21] Sousa D, Sarmento L, Mendes Rodrigues E. Characterization of the Twitter @replies network: are user ties social or topical? In: Proceedings of the 2nd international workshop on Search and mining user-generated contents; 2010. p. 63-70.

[22] Goel S, Watts DJ, Goldstein DG. The structure of online diffusion networks. In: Proceedings of the 13th ACM conference on electronic commerce; 2012. p. 623-38.

[23] Honeycutt C, Herring SC. Beyond microblogging: Conversation and collaboration via twitter. In: Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS; 2009. .

[24] Boyd D, Golder S, Lotan G. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In: 2010 43rd Hawaii international conference on system sciences. IEEE; 2010. p. 1-10.

[25] Mendoza M, Poblete B, Castillo C. Twitter under crisis: Can we trust what we RT? In: Proceedings of the first workshop on social media analytics; 2010. p. 71-9.

[26] Twitter. Rate Limits;. [accessed 25.05.2022]. Docs — Twitter Developer Platform. Available from: https://developer.twitter.com/en/docs/twitter-api/rate-limits.

[27] Twitter. Tweet Caps;. [accessed 25.05.2022]. Docs — Twitter Developer Platform. Available from: https://developer.twitter.com/en/docs/twitter-api/tweet-caps.

[28] Twitter. Conversation ID;. [accessed 25.05.2022]. Docs — Twitter Developer Platform. Available from: https://developer.twitter.com/en/docs/twitter-api/conversation-id.

[29] Robinson L. Retrieving specific conversations using Tweepy;. [accessed 10.12.2021]. Stack Exchange. Available from: https://math.stackexchange.com/q/65398427.

[30] Massey Jr FJ. The Kolmogorov-Smirnov test for goodness of fit. Journal of the American statistical Association. 1951;46(253):68-78.

[31] SciPy. SciPy 2-sample K-S test, scipy.stats.ks_2samp;. SciPy v1.8.1 Manual. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html.

[32] Hartigan JA, Wong MA. Algorithm AS 136: A k-means clustering algorithm. Journal of the royal statistical society series c (applied statistics). 1979;28(1):100-8.

[33] Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006;27(8):861-74. ROC Analysis in Pattern Recognition. Available from: https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[34] Quinlan JR. Simplifying decision trees. International journal of man-machine studies. 1987;27(3):221-34.

[35] Hastie T, Tibshirani R, Friedman JH, Friedman JH. The elements of statistical learning: data mining, inference, and prediction. vol. 2. Springer; 2009.

[36] Breiman L. Random forests. Machine learning. 2001;45(1):5-32.

[37] Bastian M, Heymann S, Jacomy M. Gephi: an open source software for exploring and manipulating networks. In: Proceedings of the international AAAI conference on web and social media. vol. 3; 2009. p. 361-2.

[38] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab; 1999.

[39] Gupta P, Goel A, Lin J, Sharma A, Wang D, Zadeh R. Wtf: The who to follow service at twitter. In: Proceedings of the 22nd international conference on World Wide Web; 2013. p. 505-14.

[40] Spielman DA. PageRank and Random Walks on Directed Graphs. Yale University Department of Computer Science; 2010. Lecture Notes, Yale University Department of Computer Science. Available from: http://www.cs.yale.edu/homes/spielman/462/2010/lect16-10.pdf.

[41] Barabási AL. Network Science. Cambridge University Press; 2015.

[42] Leicht EA, Newman ME. Community structure in directed networks. Physical review letters. 2008;100(11):118703.

[43] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008 oct;2008(10):P10008. Available from: https://doi.org/10.1088/1742-5468/2008/10/p10008.

[44] Hu Y. Efficient, high-quality force-directed graph drawing. Mathematica journal. 2005;10(1):37-71.

9 Appendices

9.1 Summary of raw data

Country	Username	Size		Virality		Depth	
		mean	std	mean	std	mean	std
UK	UKHSA	37.46	74.26	1.620	0.7373	3.532	3.721
	DHSCgovuk	51.74	102.7	1.316	0.9789	3.388	3.832
	NHSuk	21.16	62.45	1.506	0.4623	3.026	2.032
	NHSEngland	27.04	82.22	1.592	0.9710	3.187	3.187
US	CDCGov	62.02	120.9	1.905	4.667	5.932	30.98
Can	GovCanHealth	29.75	61.35	1.564	0.6548	3.439	2.532
	CIHR_IRSC	9.972	37.13	1.651	0.4571	2.872	1.977
NZ	covid19nz	17.50	7.925	2.697	0.669	10.14	4.696
	minhealthnz	29.57	72.94	1.843	0.558	4.463	3.036
All	–	40.96	90.76	1.517	1.840	3.895	11.43

Table 9: Cascade statistics by account from the raw dataset. Values reported to 4sf

9.2 Distribution histograms by account

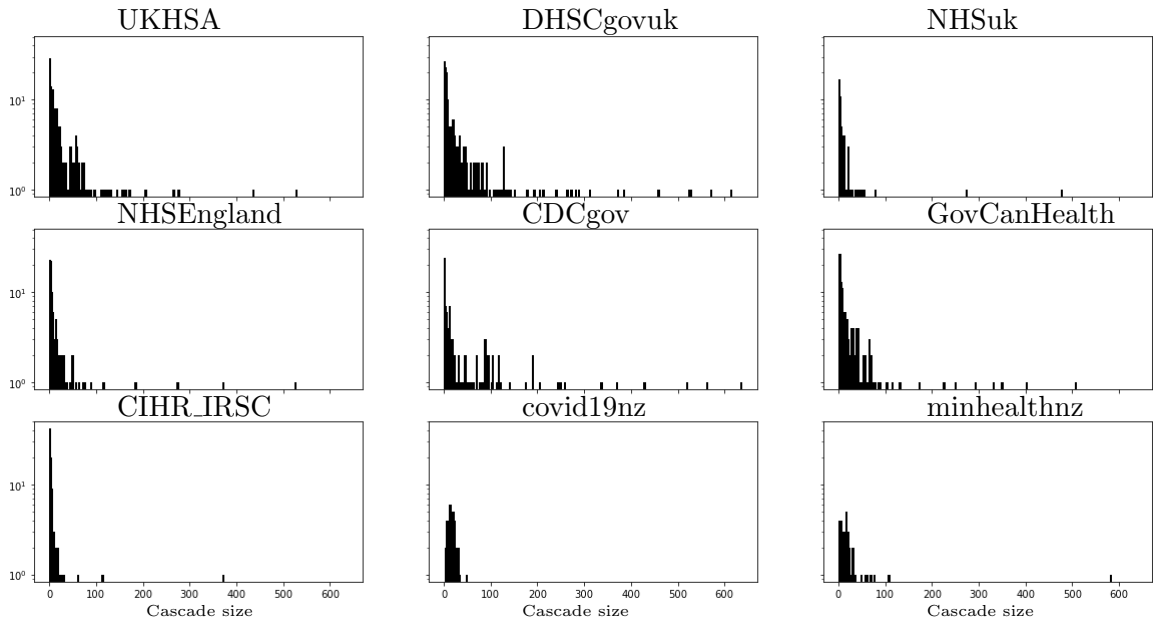

Figure 15: Histograms of the distribution of the sizes of the cascades, by account

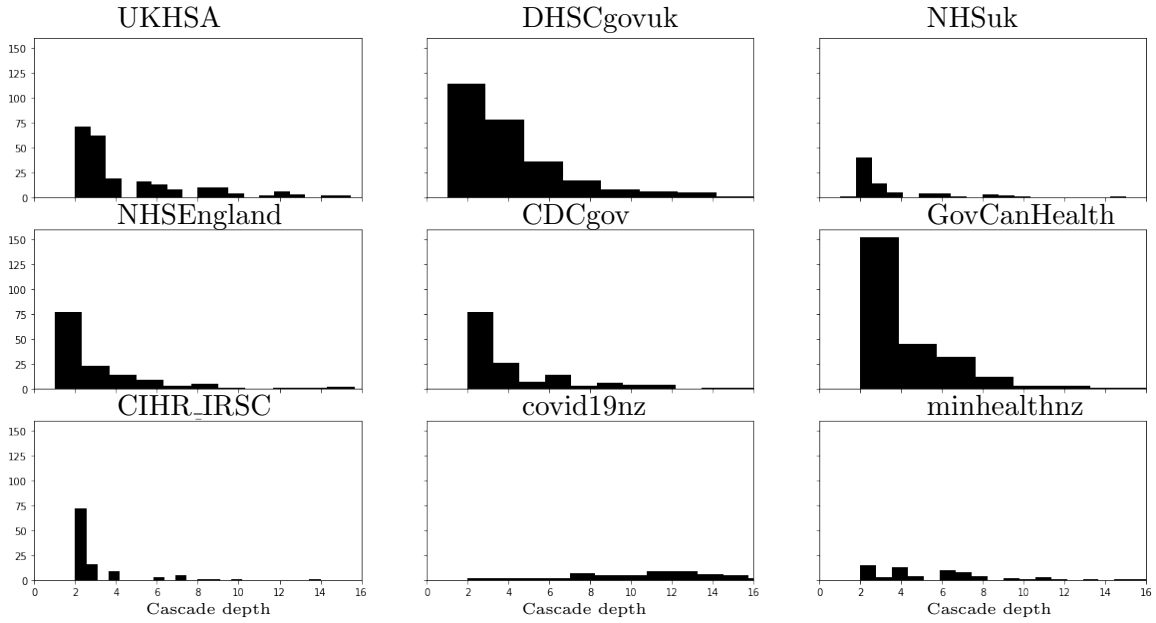

Figure 16: Histograms of the distribution of the depths (longest directed path) of the cascades, by account

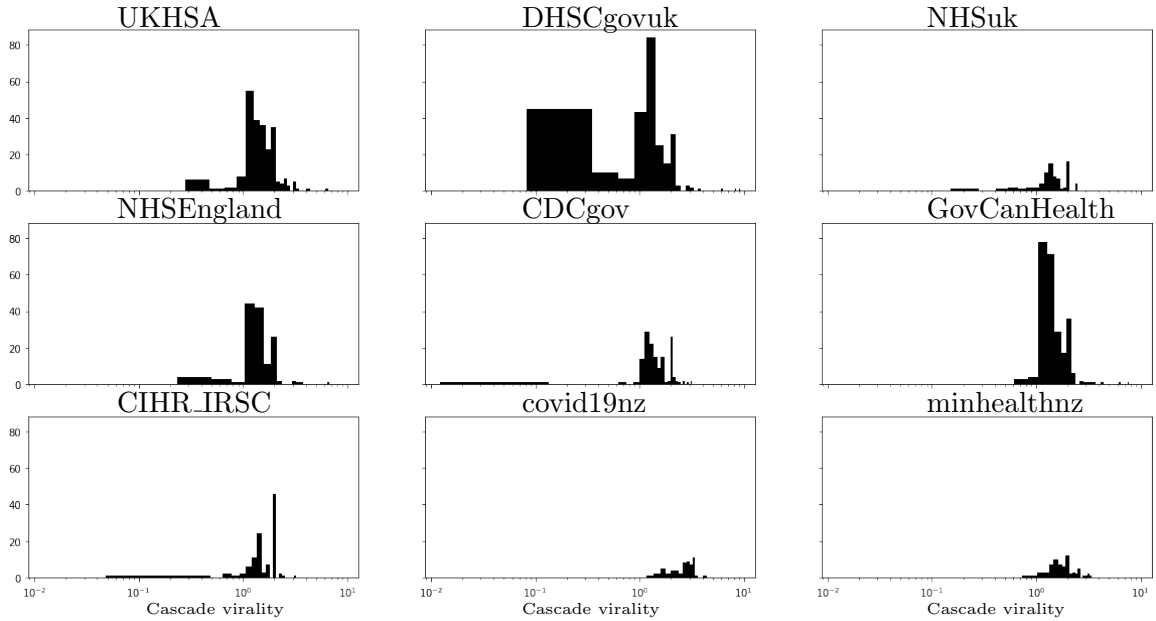

Figure 17: Histograms of the distribution of the viralities of the cascades, by account

9.3 User interaction graphs by country

Network diagram of the Canada dataset

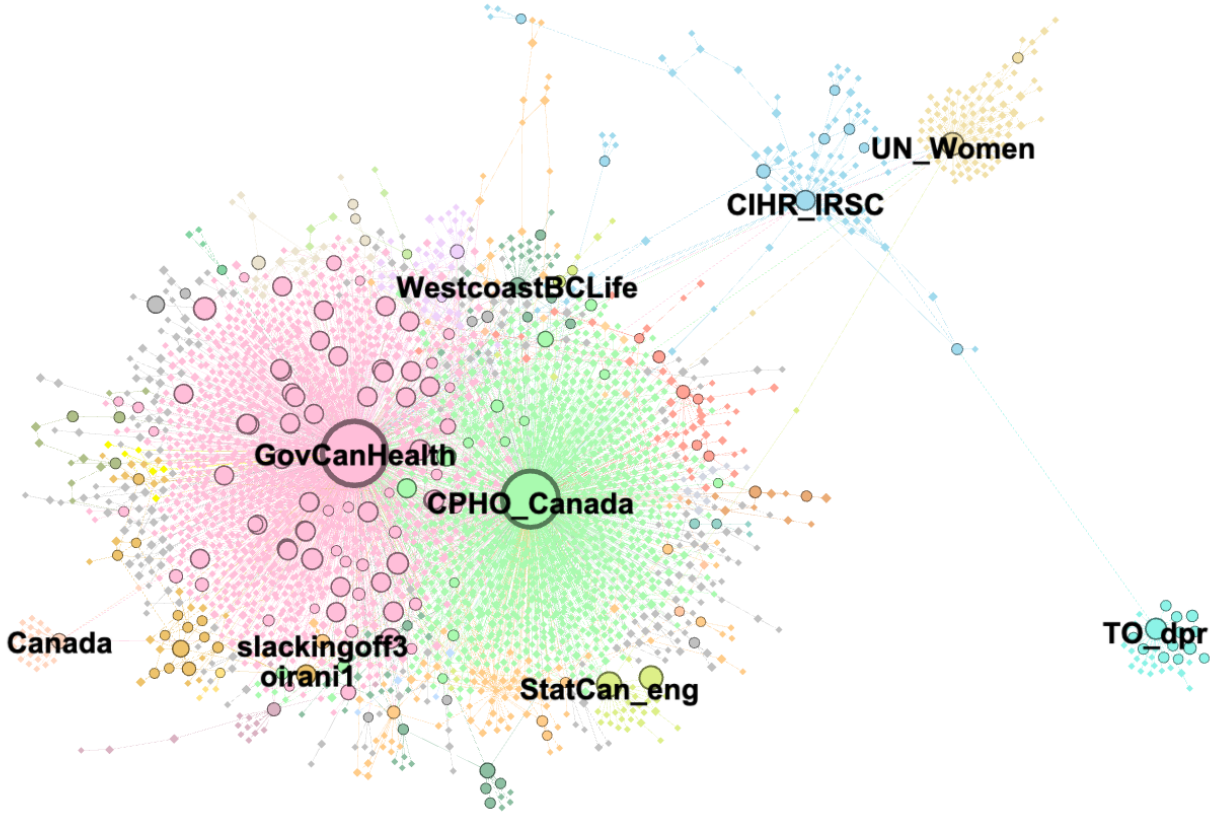

Figure 18: Network diagram of the Canada dataset with modularity for community detection applied. Node colour corresponds to modular class, and node size to PageRank. Some nodes have been omitted for clarity.

Network diagram of the New Zealand dataset

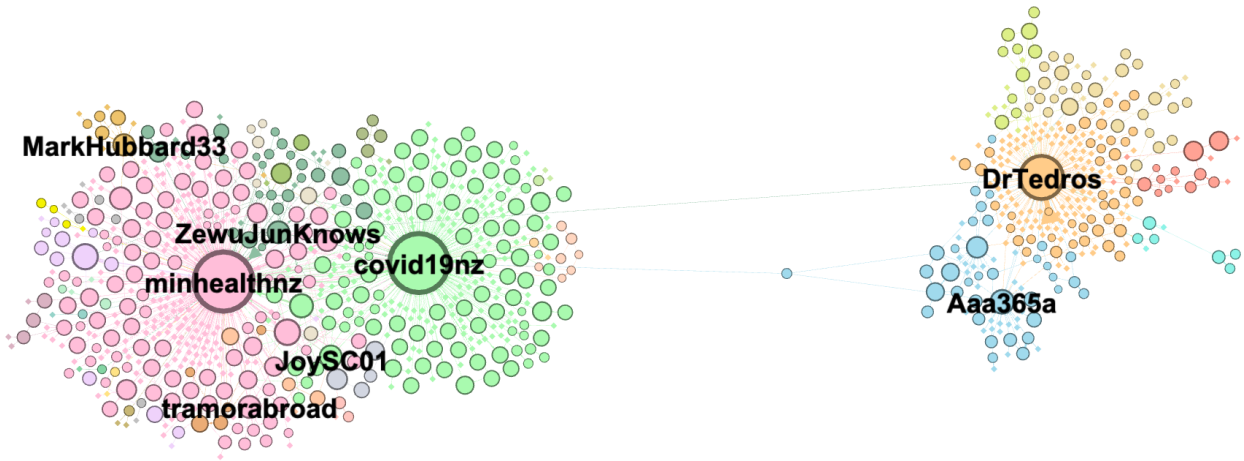

Figure 19: Network diagram of the New Zealand dataset with modularity for community detection applied. Node colour corresponds to modular class, and node size to PageRank. Some nodes have been omitted for clarity.

Network diagram of the UK dataset

Figure 20: Network diagram of the UK dataset with modularity for community detection applied. Node colour corresponds to modular class, and node size to PageRank. Some nodes have been omitted for clarity.

Network diagram of the US dataset

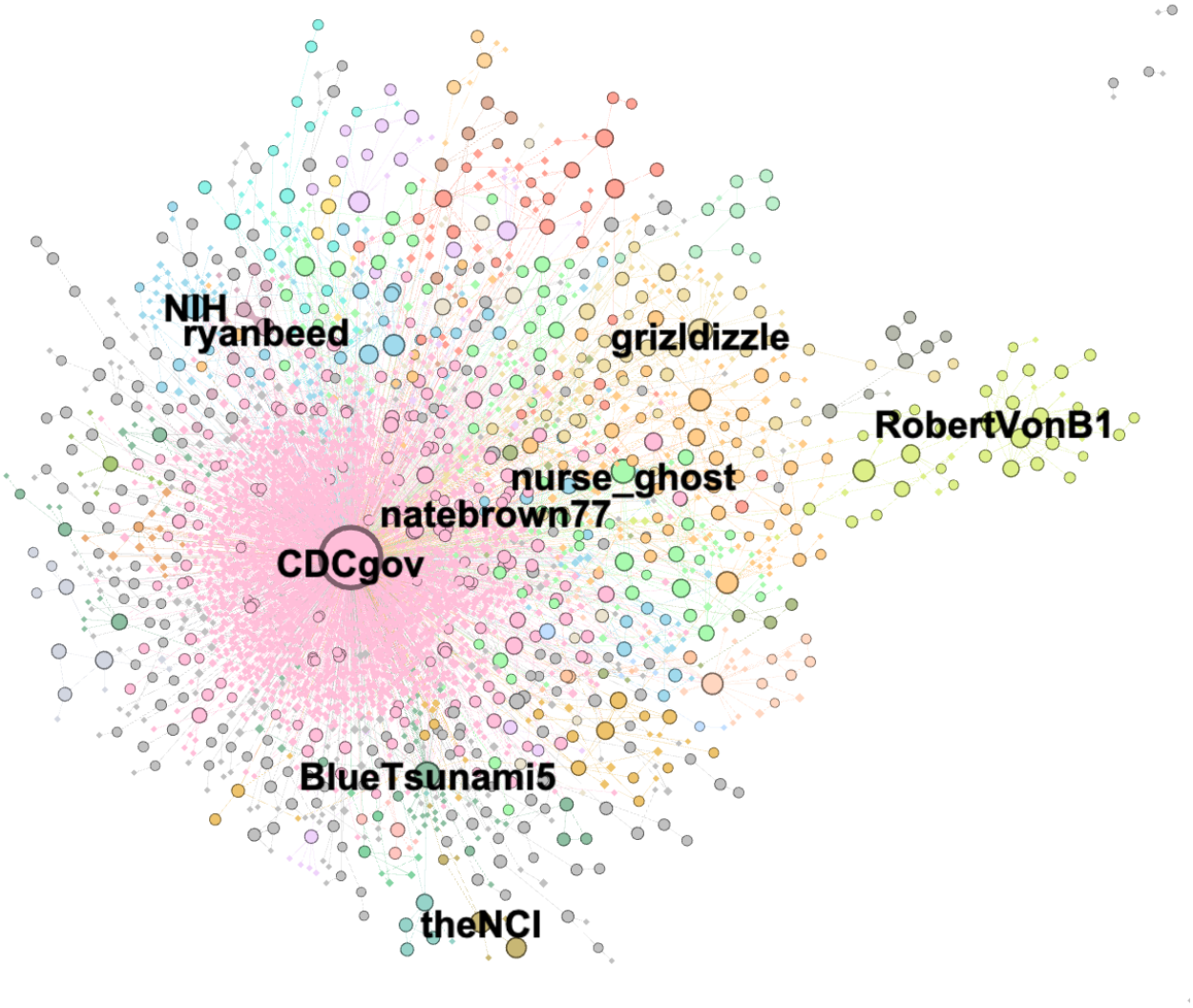

Figure 21: Network diagram of the US dataset with modularity for community detection applied. Node colour corresponds to modular class, and node size to PageRank. Some nodes have been omitted for clarity.

9.4 Most central users in each geographic network

	Username	Verified	Followers	Class size	Notes
All	sajidjavid	True	287k	6159	UK Secretary of State for Health & Social Care
	CDCGov	True	5.04m	3417	US Center for Disease Control
	GovCanHealth	True	433k	2735	Health Canada
	10DowningStreet	True	5.9m	2448	UK Prime Minister's Office
	metoffice	True	878k	656	UK Weather Service
	minhealthnz	True	49k	637	New Zealand Ministry of Health
Ca	GovCanHealth	True	433k	1109	Health Canada
	CPHO_Canada	True	302k	1045	Chief Public Health Officer of Canada
	CIHR_IRSC	True	66k	97	Canadian Institute of Health Research
	slackingoff3	False	4	97	Personal account
NZ	minhealthnz	True	49k	278	New Zealand Ministry of Health
	covid19nz	True	45k	238	New Zealand Unite against COVID-19
	DrTedros	True	1.75m	159	Director-General of the WHO[3]
	Aaa365a	False	80	61	Personal account
UK	sajidjavid	True	287k	1766	UK Secretary of State for Health & Social Care
	10DowningStreet	True	5.9m	1473	UK Prime Minister's Office
	DHSCgovuk	True	707k	1219	UK Department of Health & Social Care
	NHSEngland	True	509k	1110	National Health Service
US	CDCGov	True	5.04m	2428	US Center for Disease Control
	nurse_ghost	False	1.2k	122	Personal Account
	KathleenConfor3	False	5	116	Personal Account
	NIH	True	1.6m	108	US National Institute of Health

Table 10: Most central users in largest modularity classes

9.5 Data Collection Code

9.5.1 Setup

```
auth = tweepy.OAuthHandler(api_key, api_key_secret)
auth.set_access_token(access_token_new, access_token_secret_new)
api = tweepy.API(auth)
```

9.5.2 Collect user's timeline

```
def user_tl(n, user, attr, term=''):
    """
    collects all tweets made by a user (tl = timeline)
    Parameters
    ----------
    n : number of tweets to fetch
    user : user whose tweets to fetch
    attr : attributes to record
    term : TYPE, optional
        if searching for a keyword. The default is ''

    Returns
    -------
    df : data frame containing attribute values
    """

    # initialise output
    output = {}
    for a in attr:
        output[a] = []
    if 'retweeted' in attr:
        output['rt_from'] = []


    # get data
    user_list = tweepy.Cursor(api.user_timeline, screen_name=user,
            tweet_mode='extended').items(n)

    sleep(60)

    # process data & add to dict
    for tweet in user_list:
        if term in tweet.full_text:
            for a in attr:
                b = getattr(tweet,a)
```

```
                if a == 'created_at':
                    b = pd.to_datetime(b)
                output[a].append(b)

        sleep(10)


    #  finally, make dataframe
    df = pd.DataFrame.from_dict(output)
    return df
```

9.5.3 Get conversations

```
def get_bearer_header():
    """
    returns bearer header of elevated account
    """
    uri_token_endpoint = 'https://api.twitter.com/oauth2/token'
    key_secret = f"{api_key}:{api_key_secret}".encode('ascii')
    b64_encoded_key = base64.b64encode(key_secret)
    b64_encoded_key = b64_encoded_key.decode('ascii')

    auth_headers = {
       'Authorization': 'Basic {}'.format(b64_encoded_key),
       'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'
       }

    auth_data = {
       'grant_type': 'client_credentials'
       }

    auth_resp = requests.post(uri_token_endpoint, headers=auth_headers,
                data=auth_data)

    bearer_header = {
       'Accept-Encoding': 'gzip',
       'Authorization': 'Bearer {}'.format(bearer_token),
       'oauth_consumer_key': api_key
    }
    return bearer_header

def get_CID(ID):
```

```
    # gets the conversation ID of a tweet
    uri = 'https://api.twitter.com/2/tweets?'

    params = {
       'ids':ID,
       'tweet.fields':'conversation_id'
    }

    bearer_header = get_bearer_header()
    resp = requests.get(uri, headers=bearer_header, params=params)
    return resp.json()['data'][0]['conversation_id']


def get_Conv(conversation_id):
    """
    given a conversation ID (the ID of the orig tweet), returns a list
    each element corresponds to a reply to the orig tweet
    each element is a dictionary containing the conversation_id, id, and text
    first element is original tweet
    """

    uri = 'https://api.twitter.com/2/tweets/search/all'

    params = {'query': f'conversation_id:{conversation_id}',
       'tweet.fields': 'in_reply_to_user_id',
       'tweet.fields':'conversation_id',
              'max_results':500
    }

    bearer_header = get_bearer_header()
    resp = requests.get(uri, headers=bearer_header, params=params)

    assert resp.json() != {'meta': {'result_count': 0}}, "error encountered"

    if resp.json() == {'meta': {'result_count': 0}}:
        print('no tweets found in lookup for', conversation_id)
    orig = {'conversation_id':conversation_id, 'id':conversation_id,
            'text':api.get_status(conversation_id)._json['text'],
                'in_reply_to_status_id': 'Root'}
    whole = [orig] + resp.json()['data']
    return whole
```

9.5.4 Add parameters and original IDs

```
def add_params(lis):
    """
    given a list 'lis' of dictionaries of tweets, adds user ID and screen name,
    user and status being replied to
    if applicable, and time of creation
    """
    for dic in lis:
        twid = int(dic['id'])
        stat = api.get_status(twid)
        dic['in_reply_to_user_id'] = stat.in_reply_to_user_id_str
        dic['in_reply_to_status_id'] = stat.in_reply_to_status_id_str
        if stat.in_reply_to_status_id_str is None:
            dic['in_reply_to_user_id'] = 'Root'
            dic['in_reply_to_status_id'] = 'Root'
        dic['user_name'] = stat._json['user']['screen_name']
        dic['user_id'] = stat._json['user']['id_str']
        dic['created_at'] = pd.to_datetime(stat._json['created_at'])
    return lis

def orig_id_list(df):
    """
    if tweet is a retweet, return ID of original tweet and not
    that of retweeted instance
    """
    idlist = list(df['id'])
    # go through tweet IDs
    for i,ID in enumerate(idlist):
        # lookup tweet
        tweet = api.get_status(ID)
        # is it an RT?
        if 'retweeted_status' in tweet._json.keys():
            ID = tweet._json['retweeted_status']['id']

        # root tweet ID
        idlist[i] = get_CID(ID)
        sleep(15) # to avoid rate limit
    return idlist
```

9.5.5 Retrieve conversations en masse

```
def scrape(ids):
    """
    given list of original tweet ids, retrieves the full conversations
    """
    from time import sleep
    bigdf = pd.DataFrame()
    issues = 0
    for ID in ids:
        try:
            # fetch conversation
            cascade = get_Conv(ID)
            # add attributes
            cascade = add_params(cascade)

            # add onto output
            tempdf = pd.DataFrame(cascade)
            bigdf = bigdf.append(tempdf)
        except:
            # track errors
            print('something went wrong with ' + str(ID))
            issues += 1

        sleep(30) # avoid rate limits

    print('number of tweets that raised issues: ', issues)
    return bigdf
```

9.5.6 Add user info

```
def adduserparams(casc):
    """
    adds a user's follower, friend count and verified status to dataframe
    """
    users = casc['user_id']
    # avoid unnecessarily looking people up twice
    checked = []
    issues = 0

    # initialise output
    userinfo = {'verified':[None]*len(users), 'followers_count':[None]*len(users),
            'friends_count':[None]*len(users)}
```

```
    for i, u in enumerate(users):
        if u in checked:
            # if we've already looked up the user just use previous info
            j = checked.index(u)
            userinfo['verified'][i] = userinfo['verified'][j]
            userinfo['followers_count'][i] = userinfo['followers_count'][j]
            userinfo['friends_count'][i] =  userinfo['friends_count'][j]
        else:
            try:
                # user lookup, value adding
                user = api.get_user(user_id=u)
                userinfo['verified'][i] = user._json['verified']
                userinfo['followers_count'][i] = user._json['followers_count']
                userinfo['friends_count'][i] = user._json['friends_count']
            except:
                # track errors
                print('something went wrong with ' +str(u))
                userinfo['verified'][i] = 'error'
                userinfo['followers_count'][i] = 'error'
                userinfo['friends_count'][i] = 'error'
                issues += 1
        sleep(10)

        checked.append(u)

    userdf = pd.DataFrame(userinfo) # make df

    # add to input df
    casc['verified'] = userdf['verified']
    casc['followers_count'] = userdf['followers_count']
    casc['friends_count'] = userdf['friends_count']
    print('number of users that raised issues: ', issues)

    return casc
```

9.5.7 All an account's conversations

```
def collect(attr, name, earliest, latest, csv1, csv2, n=100000):
    """
    collects all conversations from an account
    attr: attributes to collect information on in initial tweet collection
    name: name of account we collect from
```

```
earliest: earliest date to collect tweets from
csv1: name of the csv to save the inital tweet collection to
csv2: name of the csv to save the cascade to
n: max no of tweets to collect info on in initial collection
"""
# collect tweets
tweetdf = user_tl(n, name, attr, GetRT=False, term='')

# don't go back too far in time
tweetdf = tweetdf[~(tweetdf['created_at'] < earliest)]
tweetdf = tweetdf[~(tweetdf['created_at'] > latest)]

for i,n in enumerate(tweetdf['in_reply_to_status_id']):
    if n is int:
        if n == None or np.isnan(n):
            tweetdf['in_reply_to_status_id'].iloc[i] = 'Root'
            tweetdf['in_reply_to_user_id'].iloc[i] = 'Root'
# store
tweetdf.to_csv(csv1)
print(name, ': save 1 done') #track progress

# only original tweets
tweetdf = tweetdf[pd.isna(tweetdf['in_reply_to_status_id'])]

sleep(180)

# original, not retweet, id
idlist = orig_id_list(tweetdf)

sleep(180)

# get all conversations
casc = scrape(idlist)

# mark root tweets
for i,n in enumerate(casc['in_reply_to_status_id']):
    if n is int:
        if n == None or np.isnan(n):
            casc['in_reply_to_status_id'].iloc[i] = 'Root'
            casc['in_reply_to_user_id'].iloc[i] = 'Root'

# track progress
print(name, ': scrape done')
```

```
    casc = adduserparams(casc)
    print(name, ': added user params')

    # final save
    casc.to_csv(csv2)

    return casc
```

9.5.8 All conversations from a series of accounts

```
def collect_all(attr, namelist, earliest, latest, csvlist1, csvlist2, bigcsv, n=10^5):
    """
    runs collect() on a series of accounts
    attr: attributes to collect information on in initial tweet collection
    namelist: names of account we collect from
    earliest: earliest date to collect tweets from
    csvlist1: names of the csv to save the inital tweet collection to
    csvlist2: names of the csv to save the cascade to
    n: max no of tweets to collect info on in initial collection
    """
    # initialise
    bigdf = pd.DataFrame()

    # loop through all accounts and collect all conversations
    for i in range(len(namelist)):
        casc = collect(attr, namelist[i], earliest, csvlist1[i], csvlist2[i])
        bigdf = bigdf.append(casc)
        sleep(360)
    bigdf.to_csv(bigcsv)
    return bigdf
```

9.5.9 Reconstruction

```
def recons_df(df, attr, savename):
    """
    trawls through the dataframe looking for missing tweets and
    attempting to recover them
    this is lengthy -- every 50 attempted recoveries, reports progress and saves
    """
    err = 0
    look = 0

    ids =  list(df['id'])
```

```
reply = list(df['in_reply_to_status_id'])
conv = list(df['conversation_id'])
reply[0] = ids[0]
origid = ids.copy()

output = {}
for a in attr:
    output[a] = []

for n,i in enumerate(reply):
    # check if we know about this tweet
    if i == 'Root':
        pass
    elif i in ids or int(i) in ids:
        pass
    else:
        if i == '0' or i == 0 or i == "Root":
            break
        try:
            look += 1
            # look up the tweet
            stat = api.get_status(i)
            # record it in list of id's
            ids.append(i)
            # make sure we also know what conversation it's part of
            # (same one as the reply)
            conv.append(conv[n])

            # if this is an original tweet, mark it as such
            if stat.in_reply_to_status_id == None:
                reply.append('Root')

            # if not, get the ID of the tweet it's replying to
            else:
                reply.append(stat.in_reply_to_status_id)
                # reply.append(conv[n])
            sleep(10)

            # finally, add everything to the dictionary
            for a in attr:
                if a == 'conversation_id':
                    b = conv[n]
                else:
                    b = getattr(stat,a)
```

```
                    if a == 'created_at':
                        b = pd.to_datetime(b)
                output[a].append(b)
                print('lookup no errors')


        except:
            # if the above didn't work, usually the tweet was deleted or something
            # this is ONLY called if the tweet lookup didn't work
            # print('status ' + str(i) + ' raised an error')
            # record it anyway
            ids.append(i)
            # make this link back to the root node
            reply.append(conv[n])
            conv.append(conv[n])

            for a in attr:
                if a == 'conversation_id':
                    b = conv[n]
                else:
                    b = 'error'
                output[a].append(b)

            err += 1
            sleep(10)
            print(n,i)

        if look%50==0 and look != 0:
            print('performed '+str(look)+' lookups,
            and gone through '+str(n)+' tweets
            with '+str(err)+' errors')
            print(output)
            print([len(output[a]) for a in output.keys()])
            newdf = pd.DataFrame.from_dict(output)
            newdf.to_csv(savename)

            sleep(60)

print('had to lookup ' + str(look) + ' times')
print('issues with ' +str(err) + ' tweets')

newdf = pd.DataFrame.from_dict(output)

return newdf
```