Cardiff University
CM3203 Final Year Project

Resurfacing of External Sources of Misinformation within Social Media

Author:
Joseph Harris

Supervisor:
Professor Alun Preece

Resurfacing of External Sources of Misinformation within Social Media

1 Abstract

Twitter has proved instrumental in the spread of misinformation, with fake news posts propagating faster on the platform than factual posts, and no policies within Twitter for flagging or removing fake news tweets. Twitter allows users to supplement their posts with references to external websites, often used to support a claim they may be sharing. This project delves into how individuals spreading misinformation make use of this feature within Twitter, and more specifically how these referenced external websites may resurface after their initial usage, being cited and augmented in future forged posts.

Resurfaced content is of particular interest, as it represents misinformation that can stand the test of time, potentially capable of supporting many false narratives, therefore finding this content is important both in the direct fight against fake news, but also in understanding what it is that allows a source of misinformation to be of continuous malevolent use.

This project directly looks to find examples of external websites used to support tweets promoting misinformation, observing explicitly those that reappear as supporting material for later fake news tweets, thus finding a subset of external websites that demonstrate this resurfacing phenomenon. Results obtained will give insight into how sources of misinformation may be refactored and reused for new fake news claims, potentially giving different interpretations of the same supporting material.

2 Introduction

Misinformation is a term used to describe information shared that is unintentionally false - typically due to lost context [Hil]. Individuals are able to harness this misinformation and spread it with purpose or to supplement a forced agenda, in an attempt to convince an audience that the falsities presented are in fact truths [CH20]. In no other place is this misinformation more prevalent than within social media - in the run up to the 2016 U.S. presidential election there were more than 30 million shares of posts relating to Donald Trump that were classified as containing misinformation [AG17], and it is this sharing of 'fake news' that heavily influenced the results of this election [BM19].

In masking fake news as genuine on Twitter, individuals put great efforts into ensuring what they put forward looks credible, including building a large follower base to give credence to information put forward, and citing external sources to support claims made within posts [CMP11].

In a study conducted by the University of Southern California [Shi+17], it was discovered that over 70 percent of rumour tweets [1] contain a hyperlink to external websites. These rumours being spread may flare up multiple times, even months after the original tweet post [Fri+14]. Often times, the way in which these external websites are used to support rumour tweets may be altered to suit a certain agenda, by adding new information or entirely changing how the external website should be perceived [Im+11]. Repeated misinformation gives rise to the illusory truth effect [SEL17], whereby a reader will be more likely to believe a story if continually presented it, and thus misinformation resurfacing multiple times in social media proves a genuine threat to the integrity of the platform in which it is shared.

Research into social media posts that utilise external sources to supplement their claims is so far inadequate [Shi+17], and this is the problem area investigated in this paper.

The project aims to explore the resurfacing of external websites within Twitter posts that may be sharing misinformation, analysing how the websites may be used as supporting evidence for potentially erroneous and malicious claims.

The project will be taking a very data-centric, research heavy approach, with relevant software systems only being produced as research progresses. The main tasks required for reducing the data to show potential resurfaced external websites are:

1. Producing a subset of the data that will contain only potential rumour tweets
2. Refining these rumour tweets to only the most provocative (those that have been 'called out' most by users as spreading misinformation)
3. Find examples of rumour tweet pairs, where each rumour tweet contains the same external website embedded within it, with a given length of time between tweets

[1] tweets that are sources of misinformation

3 Background

This project delves into researching how tweets flagged as being potential sources of misinformation use these external websites to push 'fake news', and how the external websites referenced within these tweets may reappear at a later date within other fake news tweets.

The project has been executed in conjunction with Cardiff University's Crime and Security Research Institute (CSRI), who have provided the tweet data used throughout the research done. This data supplied by the institute has been created by inputting several search criterion [table of criterion] into their social media enrichment platform Sentinel [Pre+17], producing a series of tweets for given dates wherein the tweet contains one or more of the search criterion texts.

`Search criterion for creation of dataset`

`Fake news`	`Disinformation`
`Propaganda`	`Rumours`
`Active measures`	`Lies`
`Subversion`	`Interference`
`Influence`	`Misinformation`
`Mainstream media`	`Hoax`
`Conspiracy`	`Untrustworthy`
`Deep state`	`Useful idiots`
`Fact Checking`	`Fabrication`
`Manipulate`	`False`
`Fraud`	`Deceive`
`Populism`	`Troll`
`Unreliable`	`Made-up`
`Bogus`	`Inaccurate`
`Doctored`	

The tweets span 3 time periods - Summer 2019 (26th April 2019 - 30th June 2019), January 2020 (1st January 2020 - 15th January 2020), and March 2020 (4th March 2020 - 25th March 2020), with each day of each period having their own JSON file. This allows for a snapshot of the Twittersphere at these times, giving an easy way of enacting the constraint of rumour tweets being some period of time apart in order for repeated website usage to be classified as 'resunrfacing', as this will require just finding websites used in rumour tweets in more than one of these time periods.

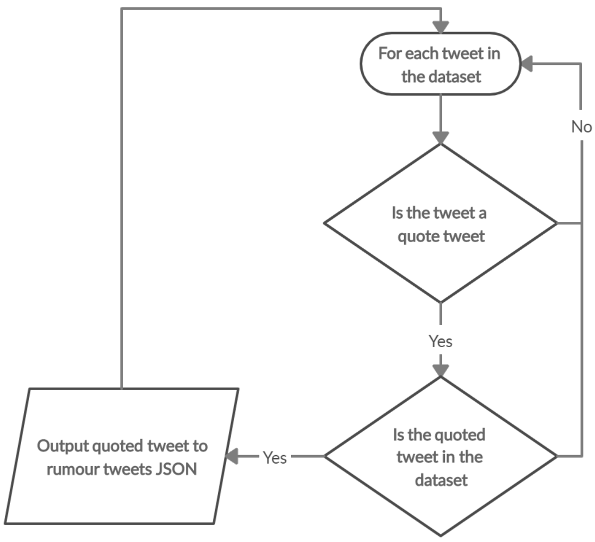
It is assumed for this project that tweets that are trying to convey or share misinformation will *not* contain any of the search criterion. This assumption is due to research completed regarding the language used in fake news, wherein it was discovered that positive abstract generalities such as 'truth' and 'freedom' are commonly used words within misinformation [Ras+17], but their antonyms are not (and it is these antonyms that form the search criterion used to create the dataset). As all tweets within the dataset provided by CSRI contain these criterion, the focus is on tweets that reference tweets **not** within the dataset.

This is possible by observing quote tweets, which are posts made in response to another tweet. Each tweet in the dataset contains full information about itself [Twi], with quote tweets having the addition of containing full information about the tweet it is responding to.

The basic theory adopted to obtain rumour tweets is to look at these quote tweets within the data, and see if the tweet referenced (quoted tweet) is within the supplied dataset. If so, then this means they contain one of the search criterion, which goes against the assumption of rumour tweets not containing the search criterion. If they are *not* within the dataset, then it can be assumed that they are attempting to pose as genuine tweets, and as other Twitter users have 'called them out' (which can be observed from the original quote tweet), they can then form the rumour tweets subset.

This assumption can be made in part due to research done in regards to user's responses when confronted with misinformation on Twitter [WZ18], showing that anywhere between 5-17% of individuals observing fake news will either ask for confirmation on the legitimacy of a story, or directly cast doubt in the form of a reply, and this is the foundation on which the creation of these rumour tweets is built.

`Flowchart showing process of obtaining rumour tweets`



The method of producing rumour tweets makes use of only quote tweets in the provided dataset. This will only be a fraction of the overall tweets within the dataset, however other forms of tweet (original, retweet) go against the assumption that rumour tweets will not contain search criterion words, and so are of no use at this stage in the creation of these rumour tweets.

The current stage of research into the resurfacing of external sources of misinformation supporting fake news posts in social media seemingly extends only to the research done in [Shi+17], and the focal point of the research was more in relation to the most cited websites for a specific rumour, as oppose to a specific website resurfacing for rumours over an extended period. Many systems used in helping individuals uncover misinformation within a platform have difficulties in adapting their approach to deal with non-linguistic content such as embedded external websites [Rog+], with some systems [FM16] choosing to remove all instances of URLs completely.

Research pertaining to the resurfacing of external sources of misinformation is seemingly not actively pursued within the field of situational understanding, and so this project aims to give an insight into this specific area, and evaluate the effectiveness of this approach to situational understanding of misinformation.

The project itself will be conducted using the Python language, with initial programs employing a bottom-up approach to find as many potential rumour tweets as possible, with further programs working in a top-down manner, pushing these rumour tweets through a series of constraints, ultimately obtaining a list of resurfaced external websites, and all tweets that contain them within the dataset.

Initially the program will be required to find quote tweets from the data, which can be achieved by simply looking for the *quoted_status* field within each tweet - if a tweet contains this field then they are a quote tweet, and vice versa. In order to process this great quantity of JSON files, it is necessary to make use of the JSON python library [Pyta], primarily for loading data into programs for analysis, and for dumping output from programs to more refined JSON files. As the data is split up initially into 3 epochs, then per day within these epochs, the *os* python library [Pytb], will be used to allow creation of an array of all files within a folder (in this case all dates within an epoch), permitting the processing of an epoch in full, rather than manually processing each individual date within an epoch sequentially.

4 Specification and Design

The project will consist of several programs, each applying constraints to the main dataset in order to refine it down to a single JSON file containing the best examples of resurfaced external websites.

5 Classifying Rumour Tweets

Posts that tweets quote, that are not within the dataset themselves (i.e. do not contain search criterion terms) are classified as rumour tweets. The tweets within the dataset that quote these rumour tweets are classified as 'call out tweets', and it's required the program keeps a count of the number of these call out tweets per rumour tweet. This will allow later constraints upon these rumour tweets to only using those that have been called out more than a certain threshold, reducing the chance of false positive rumour tweets.

Initially the data will need to be condensed considerably, as each tweet in the original datasets contain around 300 lines of JSON information, and the dataset contains 21138306 tweets, so due to the limited processing power and storage available this will be infeasible to process and store. Therefore, the first program to be built will immediately search through each data file for only rumour tweets, and then of these rumour tweets it will only store: Twitter ID (unique identifier for the tweet), username, text, URLs, quoted users, hashtags, favourite count, retweet count, and date of tweet creation. This information will be stored in a simple dictionary, which will go into the 're-duced data' folder for that epoch, as a JSON file.

```
Example of reduced JSON entry of a tweet:

{
   "tweet_id": 1121360345157779457,
   "username": "inquirerdotnet",
   "text": "The Manila Times clarified on
   Thursday that one of its editors did not
   resign but was asked to do so, and stood
   by its matrix piece published in the
   paper. https://t.co/BLTb7Upi5W",
   "urls":[
         "url":"\https"://t.co/BLTb7Upi5W",
   ],
   "quoted_users":[
         "screen_name":"khallareINQ"
   ],
   "hashtags":[
   ],
   "created_at": "Wed May 04 11:15:43 +0000 2019"
   "favorite_count": 4,
   "quote_count": 1,
   "retweet_count": 5
}
```

In order to find and store specifically 'rumour tweets', a method must first be outlined for finding (1) tweets quoting other tweets, and (2) if these tweets being quoted are within the dataset (i.e. containing search criterion text). To find quoting tweets, it's necessary to search for the 'quoted_status' field within tweets:

```
quoted_at_tweets = []
if "quoted_status" in tweet:
    quoted_at_tweets.append(tweet)
```

Then to find if the tweets that are being 'quoted at' are within the dataset, it is necessary to make use of the tweet ID each unique tweet possess, and cross reference these 'quoted at' tweet IDs against the list of original unique tweet IDs within a dataset. To make the list of original unique tweet IDs within the dataset, the program simply runs through each tweet in the dataset, and writes their tweet ID to the file 'uniqueids.txt'. Then when cross referencing the 'quoted at' tweet IDs to see if they are within the original dataset, the program appends all of the IDs in uniqueids.txt to an array, and checks if the 'quoted at' ID is within this array.

```
uniqueids = []
with open('uniqudids.txt', 'r') as unqdoc:
    for id in unqdoc:
        uniqueids.append(id)
for tweet in quoted_at_tweets:
    if tweet['tweet_id'] not in uniqueids:
        # this is a rumour tweet
```

The final stage here is to ensure the rumour tweet contains at least one URL, as this is the key element of the rumour tweet being observed.

```
if tweet['urls'] != []:
```

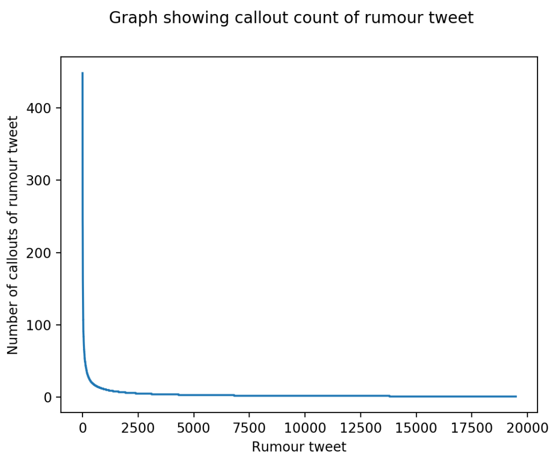
With these rumour tweets now collated, the next stage is to find the most provocative among them.

6 Finding Most Provocative Tweets

Finding the most provocative rumour tweets is necessary as of the assumption that the more people 'call out a tweet' (replying to a tweet using search criterion terms), the more likely it is the tweet is promoting misinformation

When creating the rumour tweets dataset, any instance of a rumour tweet (a tweet called out by a quote tweet in the original dataset) was placed into the rumour tweets dataset. This meant that if a rumour tweet had been called out by many tweets in the original dataset, there would be that many versions of the rumour tweet in the rumour tweets dataset. This is important, as it allows for counting the number of times a specific tweet has been called out, and for reducing this rumour tweet set down to containing only unique instances of rumour tweets. Within this set each tweet will also contain a variable 'count', which pertains to the number of times they have been called out.

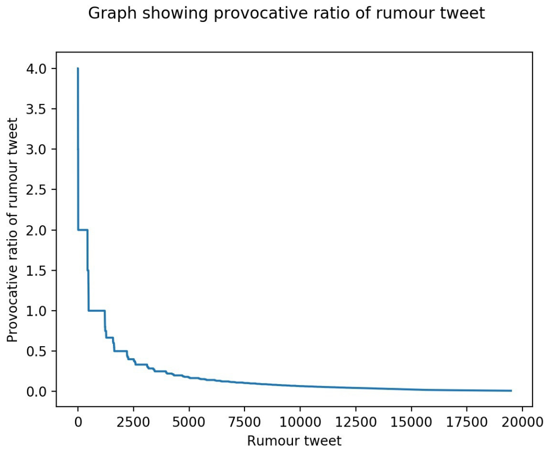
The initial approach to classify if a tweet was 'provocative' or not was to have a threshold of 'number of times a tweet has been called out', wherein if a tweet exceeded this threshold with its 'count' value, it could be classified as provocative. The issue with this approach is that this led to only twitter accounts with a large enough audience being classified as provocative due to naturally having more responses, but having a high number of followers is not indicative of the account having an increased chance in spreading misinformation - in fact quite the opposite [Gur+16]. Graphed below is how the spread of 'provocative' rumour tweets looked given this approach.



The approach instead adopted was to create a 'call-out ratio', which measured the number of times a rumour tweet had been called out, versus the number of times people had quoted the rumour tweet *without* calling it out. This made use of the 'quote_count' field in the JSON data for the tweet, so producing the call-out ratio required:

```
callout_ratio =
tweet['count'] / tweet['quote_count']
```

for each tweet. This ratio allowed better accommodation for all rumour tweets within the rumour tweet dataset, mindless of follower count, as provided the ratio of people calling the tweet out versus not calling the tweet out was high enough, it would be classed as 'provocative'. The graph showing the spread of provocative rumour tweets is shown below.



The threshold used was based upon reading through the tweets outputted by this program, attempting to find a value that allowed as many rumour tweets to pass through, while reducing false positive as much as is possible. This part of the development was heavily based upon qualitative research rather than further technical development, and resulted in the value of 0.1 as the threshold to which a tweet's provocative ratio was heavily indicative of its chance of spreading misinformation.

7 Finding Resurfaced External Websites

One of the issues faced when beginning analysis of URLs used in rumour tweets was that many of the URLs had been shortened. It was important that fully expanded URL were being analysed, as otherwise this may lead to comparisons of two different versions of the URL for the same website, giving false negatives.

The way in which this issue was resolved was through use of the **requests** module in Python, which is a HTTP library that allows for simple HTTP requests to be made [Rei].

```
def resolve_url(url):
    try:
        r = requests.get(url)
    except requests.exceptions.
            RequestException:
        return (url)
    if r.status_code != 200:
        longurl = url
    else:
        longurl = r.url
    return (longurl)
```

This function allows input of a URL as an argument, then using the requests module to obtain connection to the URL via a HTTP request. If connection is made, a 200 status code will be issued, at which point the URL from the loaded HTTP page is returned, which will be its expanded version.

With URLs now standardised across all provocative rumour tweets containing them, it's possible now to begin searching for resurfaced URLs within provocative rumour tweets, across periods of time. As defined earlier, resurfaced websites are websites used in rumour tweets within multiple time periods, which are defined as Summer 2019, January 2020, and March 2020.

In order to detect these resurfacing, a dictionary is created for each of the three timeframes, with each tweet in the provocative rumour tweets subset being a single entry.

```
summer_tweets = []
with open('summer.json', 'r') as summer:
    for tw in summer:
        tweet = json.loads(tw)
        summer_tweets.append(tweet)
```

For each URL within each tweet, a dictionary entry was made wherein the key is the URL created using the resolve URL function, and the value is the full tweet entry from the JSON file.

```
summer_urls = {}
for tweet in summer_tweets:
    urls = tweet['urls']
    for url in urls:
        summer_urls[url] = tweet
```

This now allows effortless comparisons of entries within dictionaries between timeframes, by simply iterating through the keys in each timeframe's dictionary, comparing them to keys in the others' dictionaries, and making a note of matches between the timeframes. Results are stored in the 'final' dictionary, which will take the resurfaced URL as key, storing all tweets containing this URL within an array, which is stored as the value of that entry.

```
final = {}
for sum_url in summer_urls.keys():
    for jan_url in jan_urls.keys():
        if sum_url == jan_url:
            if jan_url not in final.keys():
                final[jan_url] = []
            sum = summer_urls[sum_url]
            jan = jan_urls[jan_url]
            final[jan_url].append(sum)
            final[jan_url].append(jan)
```

With this completed, all URLs that resurface between timeframes are now stored, alongside all the tweets containing these URLs. The final task is in presenting these resurfaced URLs and tweets in a way that is useful for understanding exactly how the URLs are resurfacing among rumour tweets.

8 Interface to Present Resurfaced Content

With the now finalised list of resurfaced content containing tweets, the challenge was in presenting this information in a way that intuitively showed how, for each resurfaced website, tweets were written using the website as supporting material. This required showing how external websites resurfaced within rumour tweets over time, and so the first challenge was in organising the tweets to show this resurfacing in chronological order.

The date a tweet was posted is within the 'created_at' field of the tweet, however the formatting of the date value at this stage [Thu May 02 10:19:13 +0000 2019] will not allow for easy ordering of the tweets. In order to rectify this, it's possible to slice the created_at value for each tweet, to ascertain the year, month, and date values.

```
tweet_year = tweet['created_at'][-2:]
tweet_month = tweet['created_at'][4:7]
tweet_date = tweet['created_at'][8:10]
tweet_full_date = str(tweet_date) + "-" +
str(tweet_month) + "-" + str(tweet_year)
```

Given a list of these tweet_full_date's, it is now possible to organise the list chronologically using the datetime module within Python.

```
date_list.sort(key=lambda
date: datetime.strptime(date, "%d-%b-%y"))
```

With the methods of organising dates in place, the website dictionary must now be iterated through (where keys are equal to the unique resurfaced website, and values correspond to an array of rumour tweets containing this website). For each of these iterations, a dictionary is created, wherein the key is the date of the tweet's posting, and value is the tweet in full. With this in place, it's now necessary to iterate through the dates from date_list in order, find the tweet value this corresponds to in the tweet_date dictionary made, and then append this to the new chronologically ordered list. This will then be appended to a new JSON file, 'finalresults.json'.

With all dates chronologically ordered, a final python program is produced that iterates through each line of finalresults.json (with each line representing a resurfaced website), printing the website URL, and all tweets containing them in chronological order.

This is the final product, allowing for visualisation of the resurfacing of external sources of misinformation within Twitter (a snippet of this final output is shown below, showing an example of a resurfaced website, all rumour tweets using this website as supporting material, and the number of users that called out this tweet in the original dataset).

```
Resurfaced website: https://climatism.blog/
2019/03/18/cognitive-bias-climate-change-
alarmists-refuse-to-accept-the-science-
that-proves-extreme-weather-events-are-
not-increasing/

Number of rumour tweets website is in: 2

19-Mar-19
@JWSpry: EVERY SINGLE METRIC of
#ExtremeWeather is *declining* in
frequency and intensity, or no trend as
#CO2 emissions rise.
THE mainstream media hides this news from
you because it wrecks their man-made
#ClimateChange narrative.
FIGHT BACK! :point_right: RT
https://t.co/DfCJphCgUw via @JWSpry
2 tweets ‘called out’ this tweet

10-Sep-19
@JWSpry: @MRobertsQLD ONLY reason a
'majority of Australian's' apparently
believe that the \climate is changing"
is because they are exposed to
innumerable amounts of mainstream
media propaganda &amp; sampling events.
ACTUAL #ExtremeWeather is declining
across most major metrics.
https://t.co/DfCJphCgUw
1 tweets ‘called out’ this tweet
```

With the interface now built and all resurfaced URLs available to view alongside the context in which they are used, it's possible now to analyse the results and evaluate the extent to which this method gives an insight into new sources of misinformation.

9 Results and Evaluation

The interface shows 28 unique websites that have resurfaced as supplementary material for rumour tweets referenced in multiple time period's tweets, 16 being static webpages, and 12 dynamic.

The context in which these websites are used throughout their lifetime seems to be generally consistent, serving to support the same underlying narrative - even after months of not being used within rumour tweets.

There are examples to the contrary within the results however, with one resurfaced external website relating to 'Agenda 21' being used for vastly

different narratives in its lifetime.
In 06/08/2018, a user tweets:

***@danferg0063**: @MichelleObama @WhenWeAllVote ((Obama Covertly Signed the US Over to the UN - https://humansarefree.com/2017/03/what-is-agenda-21-depopulation-of-95-of-the-world-by-2030.html))*
This tweet was called out by 84 different tweets within the dataset.

Then in 05/07/2019, a different user tweeted:

***@ResistanceGrou9**: @AOC ((UN, Democrats amp; Global Elites solution for climate change by 2030.)) ((UN Agenda 21/2030 - https://humansarefree.com/2017/03/what-is-agenda-21-depopulation-of-95-of-the-world-by-2030.html))*
This tweet was called out by 7 different tweets within the dataset.

This shows the same supporting article being used by multiple users, for differing narratives, with user @danferg0063 using the external website to support his claims of Obama's move to make the US become closer with the UN, whereas @ResistanceGrou9 using the same external website to claim 'Agenda 21' is the democratic party's approach to solving climate change.

While not a regular occurrence within the results, having proof of different users using the same piece of misinformation to promote their own different fake news is noteworthy, and looking into this source of misinformation in future work could allow for a better understanding of how static misinformation can lead to dynamically changing fake news.

Other results that showcase resurfaced websites being used to support the same sort of misinformation still provide a great insight into the fake news stories that are repeatedly being shared, the websites that are being used to support false claims even after weeks or months after their initial usage, and the users that are referencing these websites (even when there is more up-to-date, pertinent websites they could be using). A great area of development building off of the work done during this project would be to create an interconnection network of these users citing the same sources of misinformation, as in the example relating to Agenda 21 there is almost a year between the users' posts, and it would be of interest to see if these two users share any other similarities in postings.

One of the issues within the results, that was not predicted, was the recurrence of constantly-changing websites (such as https://www.ctvnews.ca/), of which created 12 of the 28 resurfaced websites. As the project is aiming to look at how a website's contents may be used with altered narratives over time, these sorts of websites are not of any use, as its contents will change continuously. However, this does reveal the website domains that continuously come up in provocative rumour tweets as supporting material, which could be useful to monitor in future for further analysis into external sources of misinformation.

10 Future Work

As this project was done in conjunction with Cardiff's Security and Research Institute, and utilising data from their social media enriching platform SENTINEL, a big step that could be made from the project's current stage would be to begin integrating this functionality into the platform itself, so that the results from this project could be achieved automatically by making use of the data it already obtains on behalf of the institute. This should pose no great issue, as the program itself functions by directly taking the JSON tweet output from SENTINEL, then running it through a series of programs before producing the output. By changing the initial method of obtaining the data (from looking for it in my personal computer's directories to instead looking at where SENTINEL outputs the data) then results could be achieved continuously with relative ease.

This project has clearly shown there is value in looking for resurfaced content in relation to misinformation, and future work would likely follow from this theme. It would be possible to alter the code quickly to instead look for other features than URLs within tweets, such as hashtags. Users will often make use of hashtags to topic-tag, categorising their own tweets into what the content of the post is about [Sha14]. Observing a hashtags lifetime - and specifically the context in which it is used in - could be of great use in situational understanding when looking at rumour tweets, and could provide information into how rumours develop under the remit of the same underlying hashtag.

I feel that at current the project has taken the concept of searching for resurfaced URLs as far as it could usefully go, and moving towards searching for other sorts of resurfaced content - like hashtags - would be of great benefit to systems' situational understanding.

11 Conclusions

This project aimed to use the notion of resurfacing external websites within rumour tweets to find sources of misinformation, and to understand how these resurfaced websites were being used to support the rumour being spread within these tweets. To this extent, the project was hugely successful, the interface created shows (based on the method of finding rumour tweets, and the defined ruling for what classifies as 'resurfaced') all examples of resurfaced websites among rumour tweets, and the contexts in which they are used to support a tweet's claims. The results further show that in some cases, narratives can be altered while still utilising the same underlying supporting material, and research can be furthered in future work to understand what it is about these specific supporting materials that allows them to be open to malleable to supplement different sorts of misinformation.

It is clear that given the 21138306 tweets inputted into the program, the results are quite minimal. However, this was in part due to the change in how a rumour tweet qualified as being provocative, as utilising a tweets 'provocative ratio' led to many rumour tweets being disqualified from being part of this subset.

Overall the project was successful in its task of finding external sources of misinformation by looking for resurfacing URL content among rumour tweets, and this success can be easily seen through the interface produced as part of this project.

12 Reflection

I feel this project has overall been a success, however I am aware that the finished project is not at all like what was initially planned. Originally the project was to take a far more machine learning, neural network approach to situational understanding, however upon talking to members of the Cardiff Security and Research Institute, it came to our attention that resurfaced content was an untapped area of misinformation both for the institute, and in general. As a result, we decided as a collective that this was the route that the project would take, as it would prove to be both useful for the institute's efforts in situational understanding, and also a truly original area to research and develop.

Along the way there were many issues that I had to face in doing this project. For a large part of the project I was unable to utilise university computers to process data (due to COVID-19 lockdown restrictions), and hence the processing, storage, and analysis of data was to be completed on my less capable laptop. This choke-holded my progress immensely, and while efforts were made to facilitate this new mode of working, it definitely impacted the overall effectiveness of the project. What's more, ascertaining data from the institute became a real challenge, the transferring of datasets had to be achieved through online methods (as oppose to USB sticks), and the institute became quickly inundated with work due to the misinformation being spread relating to the pandemic, and so naturally it became a lesser priority to send me data, ultimately meaning I had less time than expected to process and analyse the data.

My initial plan served me well for the initial weeks where mostly bottom-up research was done on the data, and I kept on track during this time. As the project divulged into resurfacing content research however, the plan became quickly irrelevant, though I did still maintain the concepts of solid milestones and deliverables for the institute.

13 References

References

[CMP11] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. *Information Credibility on Twitter.* 2011. DOI: 10.1145/1963405.1963500.

[Im+11] Yung-Ho Im et al. "The emerging mediascape, same old theories? A case study of online news diffusion in Korea". In: *New Media Society* (2011). DOI: 10.1177/1461444810377916.

[Fri+14] Adrien Friggeri et al. "Rumour Cascades". In: *International AAAI Conference on Weblogs and Social Media* (2014), p. 108.

[Sha14] Allison Shapp. "Variation in the Use of Twitter Hashtags". In: (2014).

[FM16] M. Moh F. Yu and T. S. Moh. "Towards extracting drug-effect relation from twitter: A supervised learning approach". In: *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)* (2016). DOI: 10.1109/BigDataSecurity-HPSC-IDS.2016.53.

[Gur+16] Supraja Gurajala et al. *Profile characteristics of fake Twitter accounts.* 2016. DOI: 10.1177/2053951716674236.

[AG17] Hunt Allcott and Matthew Gentzkow. *Social Media and Fake News in the 2016 Election.* 2017.

[Pre+17] Alun Preece et al. *Sentinel: a Co-Designed Platform for Semantic Enrichment of Social Media Streams.* 2017. DOI: 10.1109/TCSS.2017.2763684.

[Ras+17] Hannah Rashkin et al. *Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking.* 2017. DOI: 10.18653/v1/D17-1317.

[Shi+17] Jieun Shin et al. "The diffusion of misinformation on social media: Temporal pattern, message, and source". In: *Computers in Human Behavior* (2017), pp. 891–921. DOI: https://doi.org/10.1016/j.chb.2018.02.008.

[SEL17] Briony Swire, Ullrich K. H. Ecker, and Stephan Lewandowsky. "The Role of Familiarity in Correcting Inaccurate Information". In: *Journal of Experimental Psychology Learning Memory and Cognition 43* (2017). DOI: 10.1037/xlm0000422.

[WZ18] Bairong Wang and Jun Zhuang. "Rumor response, debunking response, and decision makings of misinformed Twitter users during disasters". In: (2018). DOI: 10.1007/s11069-018-3344-6.

[BM19] Alexandre BovetHernán and A. Makse. *Influence of Fake News in Twitter during the 2016 US Presidential Election.* 2019. DOI: 10.1038/s41467-018-07761-2.

[CH20] Marlie Celliers and Marie Jacoba Hattingh. "A Systematic Review on Fake News Themes Reported in Literature". In: *Responsible Design, Implementation and Use of Information and Communication Technology* (2020). DOI: 10.1007/978-3-030-45002-1_19.

[Hil] Kashmir Hill. *Hurricane Sandy, @ComfortablySmug, and the Flood of Social Media Misinformation.* URL: https://www.forbes.com/sites/kashmirhill/2012/10/30/hurricane-sandy-and-the-flood-of-socialmedia-misinformation. (accessed: 11.05.2020).

[Pyta] Python. *json — JSON encoder and decoder.* URL: https://docs.python.org/3/library/json.html. (accessed: 02.05.2020).

[Pytb] Python. *os — Miscellaneous operating system interfaces.* URL: https://docs.python.org/3/library/os.html. (accessed: 02.05.2020).

[Rei] Kenneth Reitz. *Requests: HTTP for Humans^{TM}.* URL: https://requests.readthedocs.io/en/master/. (accessed: 04.05.2020).

[Rog+] David Rogers et al. "Real-time Text Classification of User-Generated Content on Social Media: Systematic Review". In: ().

[Twi] TwitterDeveloper. *Tweet objects.* URL: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json. (accessed: 05.05.2020).