

Analysing the Political and Topical Makeup of r/conspiracy

Luca Antonio Passariello

A dissertation presented for the degree of
Master of Science



Advanced Computer Science MSc
Supervised by Professor Alun Preece
School of Computer Science and Informatics
Cardiff University

2022

Contents

1	Introduction	5
2	Background	7
2.1	Technologies	7
2.1.1	Reddit	7
2.1.2	Pushshift.io	8
2.2	Techniques	8
2.2.1	Data Collection	8
2.2.2	Topic Modelling	8
2.2.3	Supervised Machine Learning	10
2.2.3.1	Support Vector Machine Classification (SVC)	10
2.2.3.2	Multinomial Naive Bayes Classification (MNB)	12
2.3	Libraries	13
2.3.1	Built-In	13
2.3.2	Third-Party	13
3	Data Collection	15
3.1	Reddit Post and Comment Structure	15
3.2	Gathering Data	17
4	Data Exploration	19
4.1	Data Composition	19
4.2	Early NLP Analysis	19
4.2.1	Top URLs	19
4.2.2	Top Mentioned Subreddits	21
4.2.3	User Account Creation Dates	24
4.2.4	Named Entity Recognition (NER)	26
4.2.4.1	Top People	27
4.2.4.2	Top Organisations	27
4.2.4.3	Top Locations	28
4.2.4.4	Top Groups	29
4.3	Topic Modelling with BERTopic	30

4.3.1	Model Creation	30
4.3.2	Model Application	32
4.3.3	Model Evaluation	34
4.3.4	Model Findings	37
4.3.4.1	Topic Analysis Over Time	37
4.3.4.2	Single-Issue Users	40
4.3.5	Next Steps	41
5	Political Classification	42
5.1	Data Gathering	42
5.2	Pre-Processing	43
5.3	Text Vectorisation	44
5.4	Training	44
5.5	Model Application	46
5.6	Findings	47
5.6.1	Overall Distribution	47
5.6.2	Partisanship Across Topics	48
5.6.3	Partisanship Over Time	49
6	Conclusion	51
6.1	Summary of Findings	51
6.2	Discussion of Aims	51
6.3	Future Work	52
6.3.1	BERTopic Topic Modelling	52
6.3.2	Political Classification	52
	Appendices	55
A	plot_without Listing	55
B	Separated Partisan Topic Graphs	57
C	JSON Dictionary to DataFrame Function	58

List of Figures

1	Diagram of the workings of BERTopic	9
2	Example of a 2D feature-space	10
3	Example of a 2D feature-space with hyper-plane	11
4	Example post from r/conspiracy	15
5	Example comments from r/conspiracy	16
6	Graph showing the counts of monthly collected comments	18
7	Brigading prevention measure on r/conservative	20
8	Pie chart showing top URLs in r/conspiracy comments	21
9	Pie chart showing top subreddits in r/conspiracy comments	22
10	Sentiment of different subs in r/conspiracy	23
11	Users by Account Creation Month	24
12	Users by Account Creation Day (December 2020 to March 2021)	25
13	Price of GameStop Stock from January 15th 2021 to February 15th 2021 [1]	25
14	Top 10 PERSON Entities	27
15	Top 10 ORG Entities	28
16	Top 10 GPE Entities	29
17	Top 10 NORP Entities	29
18	Screenshot of the labelling web application	30
19	Pie chart of topic distribution with and without 'other'	33
20	Normalised frequency of the n-gram 'george floyd' across topics	34
21	Top Eight words for each topic in the model	36
22	Evolution of all topics over time	37
23	Topics over time without 'other'	38
24	Topics over time without 'other' and 'us_politics'	39
25	Top Topics for Single-Issue Users	40
26	r/conspiracy comment partisanship distribution	47
27	Partisanship across topics	48
28	Partisanship Over Time (Daily)	49
29	Partisanship Over Time (Daily) - Subsection	50
30	Right-Wing Topic Breakdown	57
31	Left-Wing Topic Breakdown	57

32	Neutral Topic Breakdown	58
----	-----------------------------------	----

List of Tables

1	Table showing possible outcomes	11
2	Classifier Training Corpus Sizes	43
3	F1 scores for different classification configurations	45

1 Introduction

The aim of this project is to analyse the topical and political makeup of the **r/conspiracy** subreddit. This involves gathering comments from the subreddit, modelling comments into coherent topics, and classifying the political leaning of comments across topics, and over time.

Reddit is an online social network, where users join groups called 'subreddits', which focus on a specific topic, for instance one might be for a television show such as **r/gameofthrones** or for a sport such as **r/formula1**. Users make posts in these 'subs' and can then partake in a nested comment section. All content can then be voted up (upvoted) or voted down (downvoted) by other users. Reddit uses a complex algorithm to determine which posts to show users, including some recommendations based on past activity, post/comment scores, and relative post 'freshness' [2][3].

One particularly notable subreddit is called **r/conspiracy**, which describes itself as a 'thinking ground', and a 'forum for free thinking'. It is a community of over 1.8 million users who discuss conspiracy related content, such as flat earth theories, moon landing conspiracies, and others, but also have a keen interest in U.S. politics.

Conspiracy theories have progressively become a significant issue in today's society, exacerbated by the COVID-19 pandemic, and the advent of more divisive and alienating politics.

Disinformation is now seen as one of the most important threats facing European democracy [4]. The Russian government has been accused in recent years of launching widespread, complex disinformation campaigns, from elections in the United States [5], to their invasion of Ukraine [6].

The advent of the coronavirus pandemic, and subsequent public health measures, such as strict 'lockdowns' [7], led to a rapid growth in conspiracy theory related activity [8], especially in online social media, and in some ways brought conspiracy theories to the mainstream. As people were now confined to their homes, the majority of social interaction was performed through social media, with the vast majority of people who were forced to stay home reporting an increase in their social media usage [9]. This then led to more exposure to conspiracy related content, which then inevitably leads to more believers in a given theory.

Conspiracy theories have been a part of culture for much of human history, with one particularly famous theory dating back to the first century AD, when numerous theories spread pertaining to the death of roman emperor Nero.

While these theories are not a new part of society, they are able to spread much easier due to the ubiquity and accessibility of the internet, and other electronic media. Anyone can reach an audience of millions instantly, allowing for the dissemination of potentially harmful and dangerous ideas.

These ideas can result in alarming real-life consequences, from the burning of critical mobile data infrastructure [10], attacks on British Telecom engineers [11], or the spread of disease resulting from the denial of COVID-19 [12].

These theories are sometimes characterised as beliefs which allow believers to feel as part of a collective, who all subscribe to some form of the belief, whereas others can have a very strong link to subscription to a particular political ideology, as shown in a paper from 2022 [13].

The aim of this project is to analyse the **r/conspiracy** subreddit, examining the political and topical makeup of the subreddit, with the aim of corroborating whether the majority of discourse in the subreddit is right-wing, by determining the partisanship of comments, both over time and across topics.

2 Background

The tools to be used and data to be processed are the most important components of any study. This Section discusses the technologies and techniques which will be used throughout this project in order to achieve the aims set out in the Introduction. Initially discussed is Reddit, and why it is useful for textual social media analysis. The second area of discussion of this section evaluates the techniques to be used for data collection, topic modelling, and classification, while the third part discusses the the libraries used in Python to achieve these goals.

Most, if not all data analysis and manipulation will be conducted in Python, this is because the 'pip' ecosystem provides over 350,000 packages which allow Python to do nearly anything. The Python version in use is 3.9.0, which is the latest major release available.

Jupyter notebooks were used for most of the exploratory analysis to find the limits of the data. For more data intensive operations, stand-alone Python files were used to quickly access more memory. This was used for tasks such as processing large data sources or training the classifier.

2.1 Technologies

2.1.1 Reddit

As previously discussed, Reddit is a social media platform where users organise themselves into groups related to specific subjects. It can be thought of as a collection of information and can act as a way of seeing what is going on in the world right now, this is why it is known as the 'front page of the internet'. The meritocratic system rewards interesting content, which promotes content users want to see to other users.

Reddit has long been a platform which results in important research findings: from analysis of news sites [14], to analyses of domestic abuse discourse [15], and more meta concepts about Reddit itself [16]. There are two main benefits for the usage of Reddit in research:

1. The scale of the textual dataset - 366 million posts were created in 2021, with 2.3 billion comments across them, and 43 billion total 'upvotes'. [17]
2. The structure of Reddit - various studies have been conducted on data flow within and between communities, such as a paper from 2021 which studied user migration around Reddit, for instance, following trolls across subreddits [18].

2.1.2 Pushshift.io

Pushshift is a community data repository created by members of the subreddit **r/datasets**, it has two main components - the API, and data dumps. The data dumps are especially useful as they are continuously scraped from the Reddit API and go back to December 2005. This data is collected for various fields, such as comments, submissions, subreddits, moderators, and more.

For my purposes, I am using the comments dataset which contains every Reddit comment, from every subreddit since 2005. As this would be an extraordinary amount of data, I have narrowed my analysis down to the period of January 2020 to June 2022. Every comment in the dataset from this period was about 600GB compressed.

To easily interact with these files, a tool was created by a developer called **PS-REDDIT-TOOL**, which provides a command line interface with the files, allowing for them to be downloaded, and decompressed. This tool also provides facilities for extraction of subreddit-specific and time-specific comments from the dataset.

2.2 Techniques

2.2.1 Data Collection

Data collection was achieved with the previously mentioned tools Pushshift and PS-REDDIT-TOOL. These tools enabled the efficient gathering of a large dataset which was then analysed using NLP techniques. This is discussed further in Section 3.

2.2.2 Topic Modelling

To easily classify posts into their respective conspiracy, I made use of topic modelling, which is an unsupervised method of organising a set of documents into a specific number of groups or 'topics'. This would allow me to easily classify posts and compare NLP analysis across different conspiracy areas.

BERTopic was ultimately chosen, as it has been shown to be far superior to other topic modelling methods such as LDA and NMF, across many datasets, and standard measures of a successful topic modelling algorithm, such as topic coherence, and computation time [19].

BERTopic is quite an advanced algorithm. It first uses UMAP for dimensionality reduction to reduce the number of random variables in the data. This is used to speed up computation and helps mitigate the common problem of overfitting where some models can classify results into groups where they

do not belong. To then find clusters in the data, an algorithm named HBDSCAN is employed which groups together points that are in close proximity, while also marking outliers.

The clusters generated by HBDSCAN are then run through a c-TF-IDF, which is essentially a proprietary implementation of TF-IDF embedding, where the frequency of a word is based on its frequency within some class, which in this case is a given cluster. The formula of this procedure is shown below.

$$W_{x,c} = \text{tf}_{x,c} \times \log\left(1 + \frac{A}{f_x}\right)$$

Where, $\text{tf}_{x,c}$ is the frequency of the term x in class c , f_x is the frequency of term x across all classes, and A is the average number of words per class.

Finally, to increase word diversity and topic coherence, MMR (Maximal Marginal Relevance) is used which is a ML algorithm that improves search results within the model, considering the relevance of the documents. This results in the removal of words which the algorithm deems to not contribute to a topic, diversifying the words in each topic. This entire process is shown in the diagram marked Figure 1.

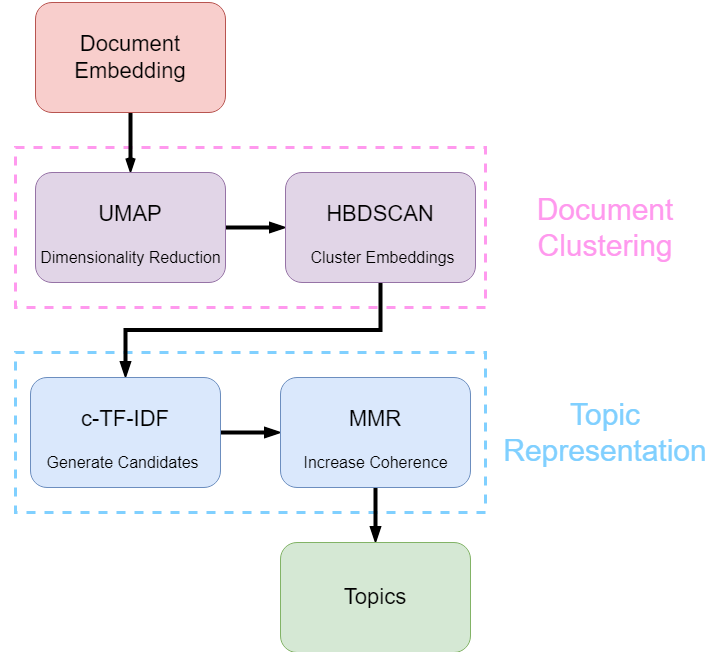


Figure 1: Diagram of the workings of BERTopic

This process produces coherent, and distinct topics from a set of textual data. The actual implementation of this is discussed in Section 4, along with the associated challenges.

2.2.3 Supervised Machine Learning

For comment political classification, supervised machine learning techniques were employed. This Section describes the multiple algorithms which were used. The selection process is discussed in the Classification Section.

2.2.3.1 Support Vector Machine Classification (SVC)

Machine learning has become one of the most widely used and useful breakthrough technologies in computer science. It has a multitude of applications, from the classification of images for facial recognition, to the classification of astronomical data, and text classification. One of the algorithms trialled was Support Vector Machines Classification (SVC) to classify the political ideology of a given piece of text.

As SVC is a supervised method of learning, labelled data is required to train the model. Each object in our training set is represented as a point in N -dimensional space. These dimensions correspond to features. For example, if we wanted to create a model which could classify whether an image was of a cat or a mouse, we may have features such as weight and tail length, which could be represented in the following 2D space, shown in Figure 2.

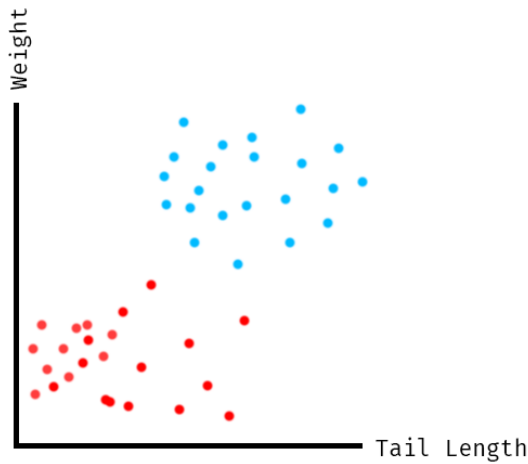


Figure 2: Example of a 2D feature-space

The goal of an SVC is to find a hyper-plane, which is a line in 2D or a plane in 3D, which aims to split the dataset into two groups. Note that in any n dimension, the hyper-plane will always have a dimension of $(n - 1)$. For our cat and mouse example, the line to best separate the data is shown in green below in Figure 3.

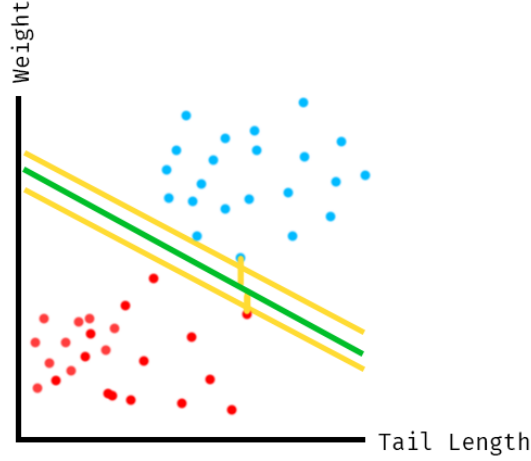


Figure 3: Example of a 2D feature-space with hyper-plane

By the 'best' line, we mean the line which maximises the space between each category. The distance (i.e. space between the two yellow lines) is called the margin and the points which fall exactly on the margin lines (with lines drawn to them in yellow) are known as the 'support vectors'.

This technique produces a classifier that can provide a certainty in the range $[0, 1]$, which can then be used to classify 'neutral' results. This could be done by marking a prediction as neutral if the certainty produced by SVM is more than 0.55 for either label.

To evaluate the accuracy of any supervised model used in this project an F1 score will be used. When measuring a result from the classifier, it can fall into one of four categories, shown below.

		<i>Actual</i>	
		<i>Positive</i>	<i>Negative</i>
Prediction	<i>Positive</i>	True Positive	False Positive
	<i>Negative</i>	False Negative	True Negative

Table 1: Table showing possible outcomes

The F1 score is then calculated as follows, and falls in the range $[0, 1]$, with 1 being perfectly accurate, and 0 being completely inaccurate.

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{F1 Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\end{aligned}$$

2.2.3.2 Multinomial Naive Bayes Classification (MNB)

Multinomial Naive Bayes is a much more probabilistic method, based on the probabilities of words occurring relative to the document set. The basis of this solution is Bayes' rule of conditional probability.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

The algorithm finds the probabilities of each word occurring given each label. Taking the word 'trump' as an example, we would have a probability for each of our prospective tags, $P(\text{trump} \mid \text{Left})$ and $P(\text{trump} \mid \text{Right})$. This is done for each word, and to make a prediction, the product probability of all words in the input is calculated for each of the labels, with the highest being declared the winner.

Naive Bayes classifiers are very efficient, making use of closed-form expressions which take linear time, rather than other more expensive options used by other classifying techniques such as iterative approximation.

Where Multinomial Naive Bayes differs from regular Naive Bayes is in the format of its feature vectors, which in this implementation represent what is essentially a histogram $\mathbf{x} = (x_1, \dots, x_n)$ with x_i being the cumulative count of the number of times term i was observed in a particular document. SVMs are currently the standard for most text classification problems yet performance can be highly dependent on the format of the data in question, with MNB potentially performing better on smaller samples of data, such as Reddit comments.

2.3 Libraries

Python is one of the most popular programming languages globally [20], and a premier reason for this is the massive collection of packages, both built-in and third-party which allow for vast functionality out of the box. This section covers the packages I have used throughout this project to collect, analyse, and draw conclusions from my data.

2.3.1 Built-In

Built-in libraries are the libraries which come as standard with any installation of Python. Some of the main built-in packages used in this project are discussed below.

The **re** library allows for the use of regex, which in turn lets us perform quite complex string manipulation. This was largely used for removing items such as punctuation, or redundant strings such as '[deleted]', which denoted a deleted comment.

The **datetime** library was used throughout the project for a variety of functions, including to both organise data by date and format dates. This was important when analysing time-sensitive data such as account creations or following data trends over time.

2.3.2 Third-Party

Data Processing	NLP	ML	Data Display
numpy	nlTK	scikit-learn	matplotlib
pandas	spacy		plotly
	bertopic		

numpy is a very useful library for analysing numerical data as it makes it very easy to work out statistics. It was mainly used to work out statistics of lists of integers, such as means, medians, variance, etc. It also includes useful facilities for random choices from lists, which was used to sample comments from the dataset.

pandas was also used widely across this project, as it allows for powerful 2-dimensional data structures inside Python. This was mostly used for reading in the comment data, manipulating it, and putting it in a useful format. An example of this usage was to group posts by comment and remove any posts with less than 10 comments.

nltk is one of the most popular NLP Python packages available. It was used in this project for some rudimentary sentiment analysis through its **SentimentIntensityAnalyzer** class, which makes use of the **vader** sentiment analysis tool, something which was specifically designed for social media analysis.

spacy is a very useful package for general NLP and, more specifically, named entity recognition. It employs word embeddings, deep learning, and many other complex technologies to create an entity tagger. This was mainly used in the exploratory stage of the project.

bertopic is a topic modelling package using a proprietary algorithm which can organise documents into specific topics based on their content. It can be used in a completely unsupervised fashion, which allows for quick, and accurate document classification without training. This was used to classify individual posts for comparison across conspiracy topics.

scikit-learn is a machine learning package which allows for the usage of complex ML algorithms, such as those discussed above including Support Vector Machines and Multinomial Naive Bayes. It also provides useful optimisation functions for fine-tuning ML parameters. This package was used during the political classification stage.

matplotlib was used for the simpler plots that were used throughout the data exploration phase. They were very useful for looking at data and identifying patterns or interesting features.

plotly is a further graphing library that provides clean and readable plots, which were mostly used for creating plots to be inserted into this report to display results or support my findings.

3 Data Collection

This project primarily seeks to analyse Reddit comment data. A central requirement thus involves collecting a significant amount of Reddit data to act as a foundation for further analysis.

3.1 Reddit Post and Comment Structure

Reddit, as a data source, is rich in terms of its scope, and quantity. As previously discussed, users join 'subreddits', or 'subs', which discuss specific topics. Users make posts in these subs which will include a title and some content, such as a link to a news article or an image. Users then discuss the post in the comment 'forest', where multiple tree data structures reside, each made up of user comments. Figure 4 shows an example of a Reddit post.

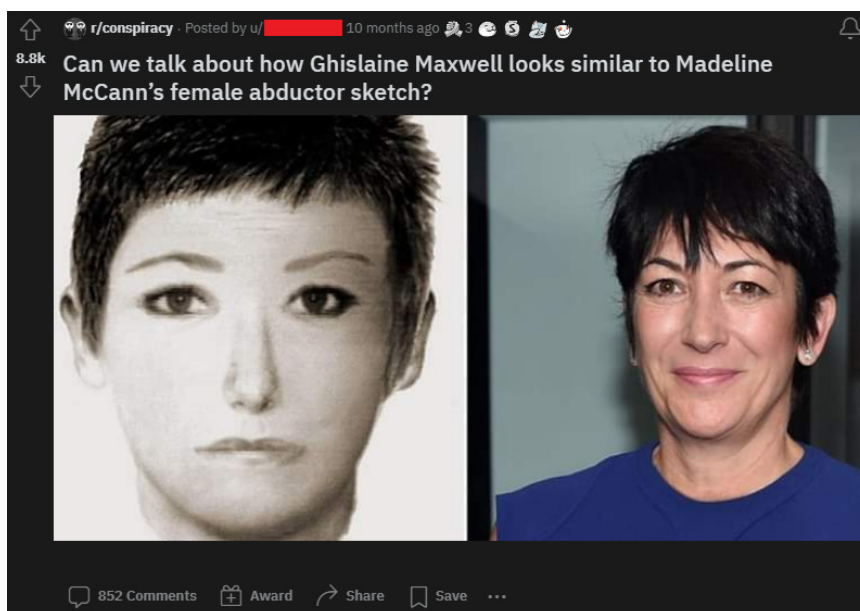


Figure 4: Example post from **r/conspiracy**

The Reddit post in Figure 4 shows the post title above an image of Ghislaine Maxwell. We can also see on-screen the current score of the post (8.8k), the subreddit, the author (redacted in red), and any awards given to the post by users. At the bottom of the image we can see the number of comments under the post, which at the time of capture was 852.

Figure 5 shows an example of Reddit comments from the same post.

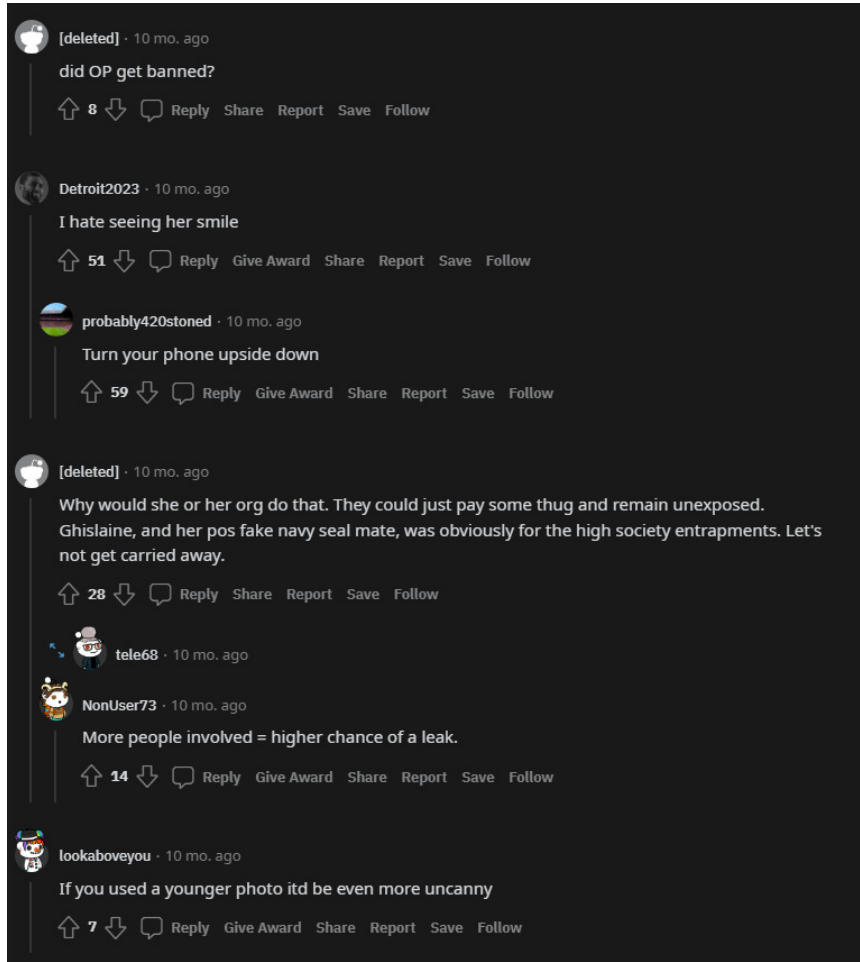


Figure 5: Example comments from **r/conspiracy**

The structure of these comments was previously described as a forest, due to its nature as a group of reverse tree data structures. The top level comments are roots which then spread out into branches of comments. The first comment in the image is just a root with no replies, under that we have another root comment which has one reply. The third group of comments shows a root comment with two replies (one collapsed), which result in two branching comment chains. Finally, there is another single root comment. The structuring of Reddit comments and their relation to this project is discussed further when training the classifier in Section 5, and discussing potential improvements to the project in Section 6. The data collected from PushShift consists of these comments, from all levels.

3.2 Gathering Data

Initially, the Reddit API was selected for this task, using **praw**, or the Python Reddit API Wrapper, to enable seamless interaction through Python. This proved to be very difficult; however, due to Reddit's rate-limiting, which greatly limits how many requests can be made to the API. This was such a restraint, that it took 15 hours to gather the top 100 posts and all their comments. This is where I started to look for an alternative data source.

As briefly mentioned in the Background Section, data was obtained from PushShift [21], which is an open-source community project, that essentially constitutes data dumps. This allowed me to gather 18 million comments, that were then narrowed down to 13 million 'clean' comments, all of which possessed the required information, such as author, body, and created time.

The information obtained with each comment included:

- **body** - The actual text of the comment
- **score** - The number of upvotes minus the number of downvotes
- **id** - The Reddit ID of the comment
- **author** - The Reddit username of the comment's author
- **author_created_utc** - The datetime when the user's account was created
- **permalink** - The absolute URL to the comment
- **post_id** - The Reddit ID of the post
- **controversiality** - A Boolean value for whether a comment is 'controversial'

This information was gathered through the use of a small tool built by a developer named Magnus Nissel. [22]. It provides a CLI for decompressing the files, finding posts in a date range, and from a specific subreddit in one action.

The below graph in Figure 6 shows gathered comments per month, without cleaning.

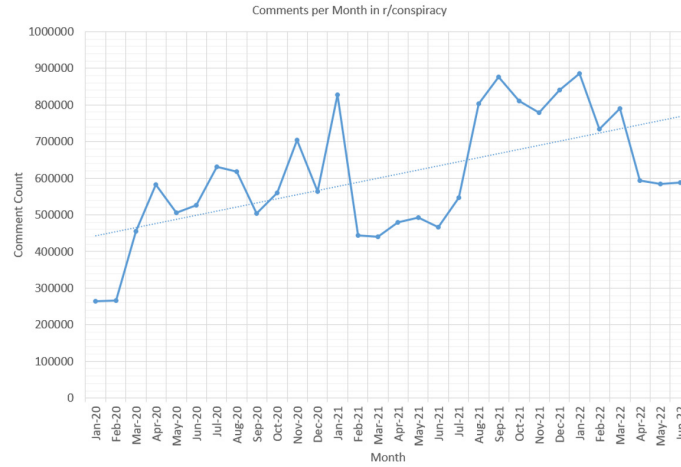


Figure 6: Graph showing the counts of monthly collected comments

The comments collected from the **r/conspiracy** subreddit were from the period of January 2020 to June 2022. This period was chosen as I wanted a focus on recent conspiratorial subjects, such as COVID, vaccinations, and 5G, and have a manageable set of data. The **r/conspiracy** subreddit also lends itself to other more historical theories such as the earth being flat, or that the moon landings were a hoax. However, these posts make up a small subset of the current content of the subreddit.

After the data was downloaded from PushShift and extracted using PS_REDDIT_TOOL, I wrote a small Python script to read the JSON files, pull out required fields, and save each month to a CSV file as shown in Appendix C. These monthly DataFrames were then collated into a single frame for all of the subreddit's data. Saving it in this format allowed for its straight-forward importation into a DataFrame for future data analysis and processing.

The data obtained from this source allowed me to explore these topics in more detail. For example, the date of user account creation could be used to determine if users created so-called 'zero-day' accounts [23], which would indicate that accounts were created to specifically talk about a certain topic. These accounts can also point to 'astroturfing' campaigns, which is when a group creates accounts to make their message appear as though it is supported by normal/grassroots users [24].

4 Data Exploration

This section contains some exploratory analysis of the collected dataset. This was done to get a better idea of the structure and composition of the data, and to discover any interesting anomalies which could be explored. Finally, the analysis conducted here provided a useful way to get used to using `pandas` for data manipulation, and working with `plotly` for graphing. This analysis included:

- Finding top URLs
- Finding top subreddits
- Exploring when users created their accounts
- Named Entity Recognition for top people, organisations, locations, and groups

The final piece of this section was the execution of topic modelling on the corpus to extract 10 topics for further analysis. As one of the main aims of the project was to organise posts into topics, this would prove invaluable for more detailed partisan analysis in the Political Classification Section.

4.1 Data Composition

As discussed in section 2, the data for this project was gathered from the PushShift project and is comprised of the vast majority of all Reddit comments in `r/conspiracy` from January 1st 2020 until June 30th 2022. This provided a set of 13,068,591 comments after cleaning and removal of incomplete data. This data took the form of a 5.3GB CSV file, which could be loaded in and out of Python using `pandas`.

4.2 Early NLP Analysis

To see what interesting insights the data could hold, some preliminary data analysis needed to be completed. This was achieved by looking through the dataset to find any anomalous occurrences, or strange patterns.

4.2.1 Top URLs

One of the first areas which potentially held interesting information was the most common URLs present in user comments. Performing this analysis shows that the most common URL by far was `www.reddit.com`, which made up 13% of all subreddit links. This is intuitive, as users will regularly

reference specific posts or comments across Reddit. This area is investigated further in the Top Subreddits Section.

Following this, links to YouTube made up around 8% of all links in the comments. Sampling these comments shows users referencing news reports, memes, or videos supporting their viewpoints.

Interestingly, the fourth most popular URL is `np.reddit.com`, which is an alternate subdomain for Reddit called 'Non-Participation' Reddit. This was created by a group of moderators from across Reddit, and aims to prevent 'brigading' or 'invading', where users will go en-masse to vote, comment, or otherwise interfere with other content on Reddit. This is strictly up to the target subreddit's moderation team, as they can use custom CSS styling to hide voting buttons, or the comment textbox. For example, the `r/conservative` subreddit displays 'I thought you were just here to read??', when hovering over the voting buttons, seen in Figure 7.



Figure 7: Brigading prevention measure on `r/conservative`

These measures are not enforced by Reddit administrators and are currently optional for subreddit moderators. This also does not necessarily stop determined actors from 'brigading', as it only hides buttons with CSS, but it does make the process harder.

The pie chart for the top 10 URLs is shown below in Figure 8.

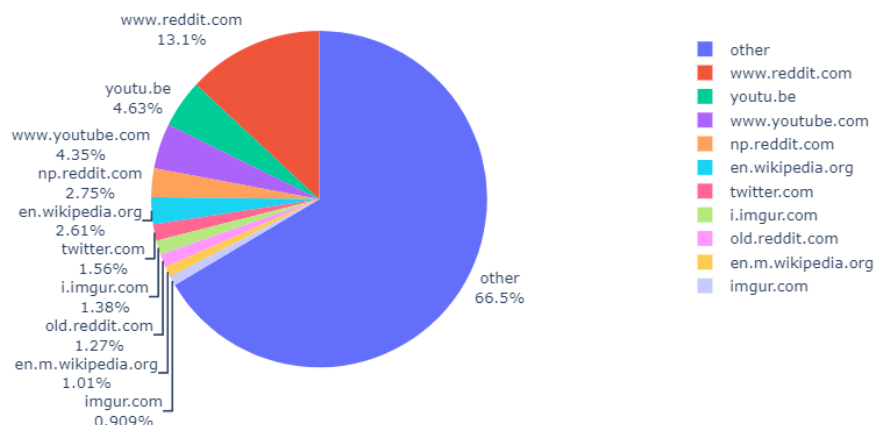


Figure 8: Pie chart showing top URLs in **r/conspiracy** comments

The other Reddit URL - `old.reddit.com` - is the old styling of Reddit that long-time users tend to prefer. Wikipedia appears twice in the top 10 - both the English web Wikipedia and the English mobile Wikipedia are present. There are also two links to Imgur, an image hosting service, which when sampled tended to be memes relating to popular conspiracy theories.

4.2.2 Top Mentioned Subreddits

While Reddit posts and comments are sometimes referenced using their absolute URL (permalink), users often reference other subreddits directly, using the syntax `r/<sub>`. This offers a clickable link to the referenced subreddit. Users will often do this when referring to a subreddit as a group of individuals, or to link to another subreddit. The pie chart for this analysis is shown in Figure 9.

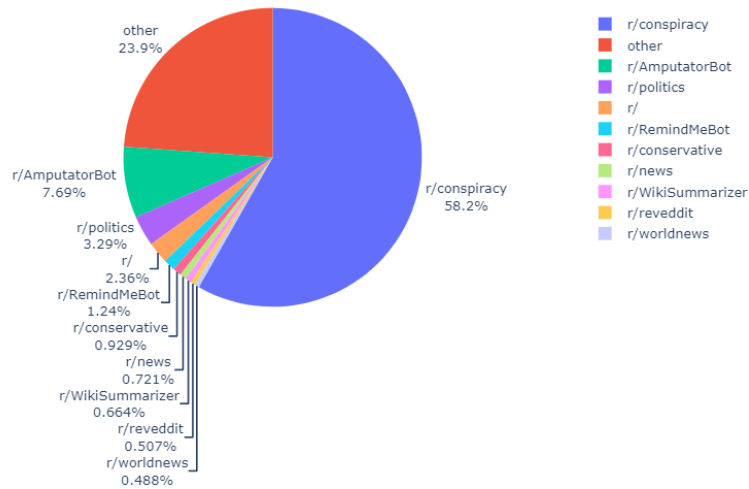


Figure 9: Pie chart showing top subreddits in **r/conspiracy** comments

The top subreddit with around 60% of all mentions is the **r/conspiracy** subreddit itself. Sampling these, it is mostly users referencing the 'sub' in a disparaging way or referring to meta-conspiracies. Some examples sampled from the dataset include:

- 'lol, you think the mod team on r/conspiracy is under any threat of turning blue? lol'
- 'They need to keep anti establishment people in r/conspiracy so they are easier to manage.'
- 'Thats the one. I guess the people in r/conspiracy trust Wikipedia now.'

The next largest individual subreddit mentioned is **r/AmputatorBot**, which when replied to a Google Accelerated Mobile Page (AMP) link, removes the AMP and delivers the original content. AMP is controversial, especially on Reddit, as it could potentially be seen as Google's attempt to consolidate their control over the internet, which the **r/conspiracy** subreddit are particularly averse to.

The third largest referenced subreddit is **r/politics**, which is the general politics subreddit on Reddit, it does however, have a reputation for having an alleged left-wing bias, and so is regularly the subject for criticism in **r/conspiracy**.

The list includes two more bots, one called **r/RemindMeBot** which sends a notification to a user after a specified time interval, and **r/WikiSummarizer** which creates a short summary for linked Wikipedia articles.

There are also two news subreddits mentioned, **r/news** and **r/worldnews**, which are both general news subreddits, with **r/news** providing more U.S.-centric news, and **r/worldnews** providing popular news stories from across the world.

Finally, **r/conservative** is also present in the top 10, making up 0.9% of all subreddit mentions. There tends to be some crossover between the two subreddits, as they commonly discuss similar topics, such as vaccinations, COVID, and political conspiracies.

Sentiment analysis was run on all comments for a subset of subreddits, with the results, shown in Figure 10, plotted with the average sentiment of all comments in the dataset. It was noted that mentions of the subreddit **r/conservative** were the most negative of the five most mentioned subs, with **r/conspiracy** being the most positive, although still negative.

The negative sentiment around **r/conservative** was quite intriguing. Lots of the comments referencing the right-wing subreddit were complaining that **r/conspiracy** was becoming **r/conservative**. The negative sentiment of most other communities could also be indicative of the climate of the subreddit. Previous studies have shown that belief in conspiracy theories can be an important predictor in distress, anxiety, and job and life satisfaction [25]. A study of Ecuadorian healthcare workers found that 'those who believed the virus [COVID-19] was developed in a lab were more likely to have distress disorder and anxiety disorder and had lower levels of job satisfaction and life satisfaction' [26].

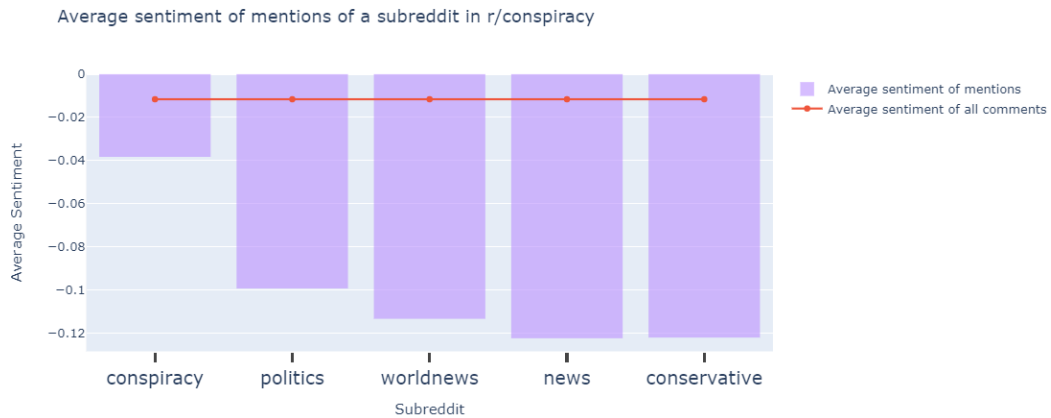


Figure 10: Sentiment of different subs in **r/conspiracy**

4.2.3 User Account Creation Dates

Another piece of early analysis completed on the dataset examined when users created their accounts. Each comment entry in the dataset includes an attribute called `author_created_utc`, which is the date and time that corresponds to when a user created their account. Grouping these by month provides the following histogram, shown in Figure 11.

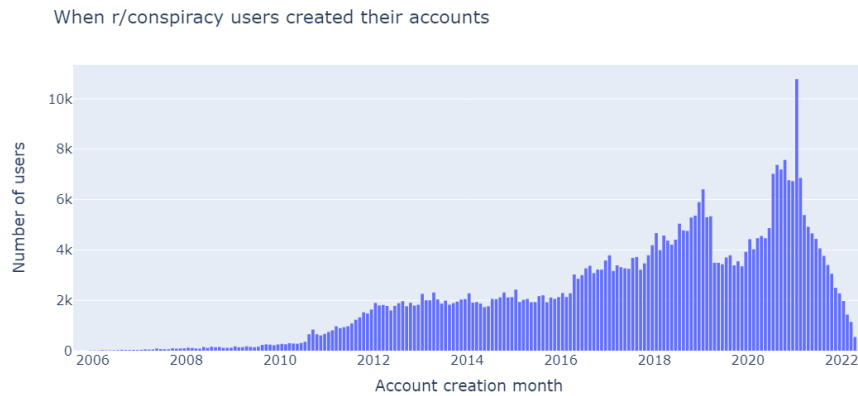


Figure 11: Users by Account Creation Month

The noticeable trend of this graph is a somewhat linear increase in users per month, which accelerates around 2010 and 2016, with a curious trough from April 2019 to June 2020. However, the most notable data-point in this graph is in January 2021, where the average monthly users almost doubles on the previous and later months. This required further investigation, so the data was split out into daily data, and zoomed to a four-month period around January 2021, shown in Figure 12.

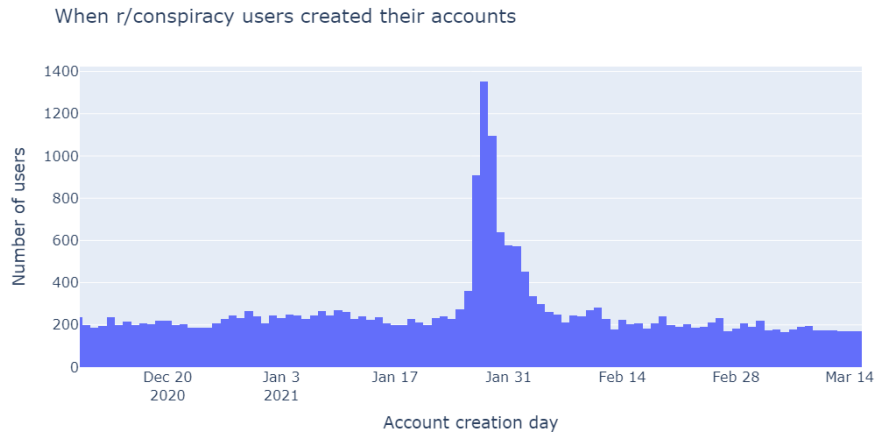


Figure 12: Users by Account Creation Day (December 2020 to March 2021)

The peak is even more evident on this graph, with the daily peak on 27th January being over six times greater than a normal data-point around this time. It was also at this time that the GameStop Short Squeeze of 2021 was occurring [27], which brought a significant amount of media attention to Reddit, as well as a large increase in users. Analysed against the stock price of GameStop (shown in Figure 13), the user increases lag behind by a couple of days, but this would be expected as news stories are written, and readers find them over the following days. **r/conspiracy** also became interested in financial conspiracy theories around this time, with the emergence of allegations of market manipulation from stock trading platforms such as RobinHood [28], which can also be seen later in analyses of topics over time. The peak in users would thus most likely be more pronounced in **r/conspiracy**, as users join to discuss the allegations.

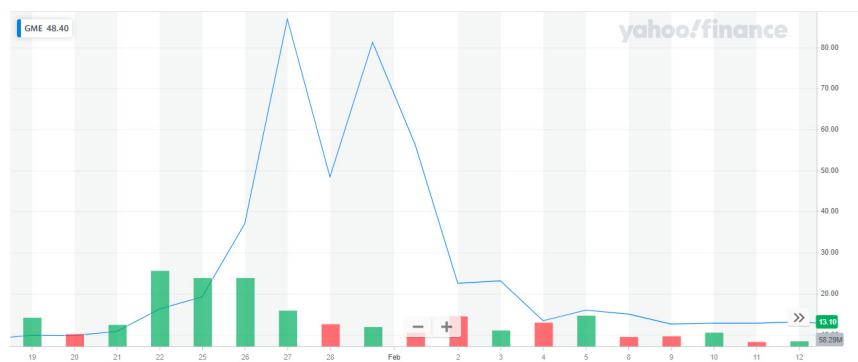


Figure 13: Price of GameStop Stock from January 15th 2021 to February 15th 2021 [1]

4.2.4 Named Entity Recognition (NER)

SpaCy is a Python library which provides numerous Natural Language Processing facilities. One of these tools is their Named Entity Recognition (NER), which has been shown to have similar accuracy scores, and faster processing times to other solutions such as StanfordNLP and NLTK [29]. This Section will look at some of the most common named entities found in the dataset and explore whether any conclusions can be drawn from them.

Before any entity recognition was performed, the entire dataset was lemmatised, since in initial executions, there would be lots of duplicates such as 'republican' and 'republicans'. This was achieved through the following code listing 1.

```
1 from nltk.stem import WordNetLemmatizer
2 from nltk.tokenize import WhiteSpaceTokenizer
3 lem = WordNetLemmatizer()
4 tok = WhiteSpaceTokenizer()
5 def lemmatise_text(text):
6     return ' '.join([lem.lemmatize(w) for w in tok.tokenize(str(text))])
```

Listing 1: Word Lemmatisation

The function `lemmatise_text` takes in a string, splits it by white-space, lemmatises each word and then joins the string back together with white-space. Then, to perform NER with SpaCy, listing 2 is used.

```
1 ents = {
2     'PERSON': [],
3     'ORG': [],
4     'GPE': [],
5     'NORP': []
6 }
7 corpus = df['lemma'].tolist()
8 for doc in corpus:
9     doc = nlp(str(doc))
10    for word in doc.ents:
11        if word.label_ in ents.keys():
12            ents[word.label_].append(word.text)
13
14    count_ents = {k: Counter(v) for k, v in ents.items()}
```

Listing 2: NER with SpaCy

The entities collected are recognised people, organisations, locations, and social groups. To do this, I first define an object to hold all occurrences of each label type, then loop through all documents in the corpus, run NER on them, and for every entity found we add it to its label's list. Finally, for every type, occurrences in the list are counted and returned as a new dictionary of tuples containing each found term and its frequency.

4.2.4.1 Top People

Figure 14 shows the top 10 recognised 'PERSON' entities found in all comments in the data capture range. It is along the lines of what would be expected, with the majority being western (mostly U.S.) political or media personalities.

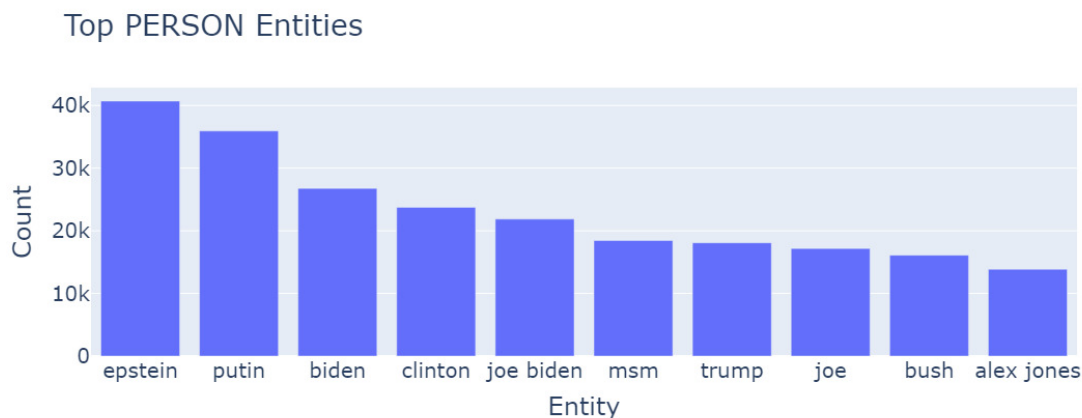


Figure 14: Top 10 PERSON Entities

The only oddity here is the inclusion of 'msm', which was mis-recognised as a name, which is in fact a reference to the 'Mainstream Media', usually used as a slur against large, well-known media outlets.

4.2.4.2 Top Organisations

Figure 15 shows the top organisations recognised in the corpus. This was included for two main interesting observations. Firstly, the FDA was mentioned much more often than expected, and had

over double the mentions of the CDC, this is most likely due to the FDA being responsible for vaccine approval in the United States, with the CDC responsible for COVID and disease response. Another interesting observation from this data is that the CIA is mentioned significantly more than the FBI. This would probably be due to the fact that while the FBI is much more topical with respect to Donald Trump and potential investigations of both Trump and political enemies such as Hillary Clinton, the CIA has allegedly been involved in more historical conspiracies, which tend to interest the subreddit.

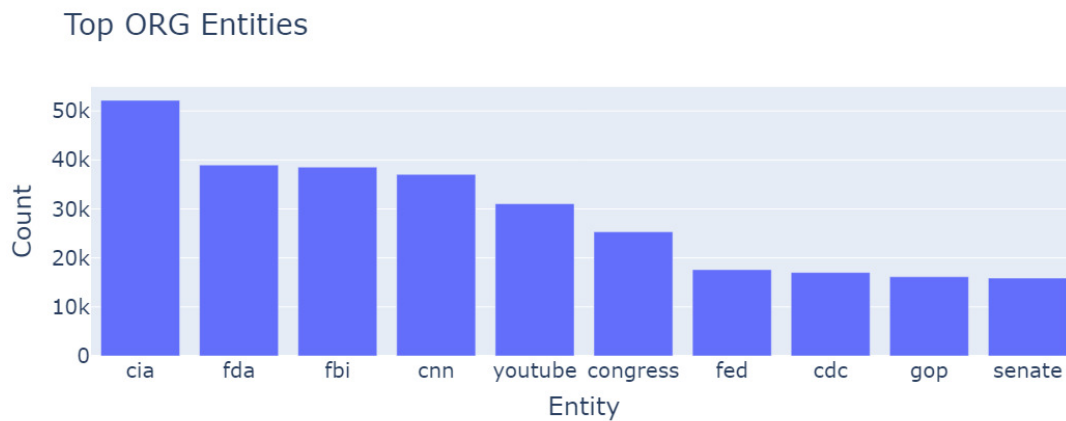


Figure 15: Top 10 ORG Entities

4.2.4.3 Top Locations

The top 10 locations mentioned in comments are shown in Figure 16. What was notable here was the inclusion of only two specific U.S. states, California and Florida which are derided for their association with democrats and republicans respectively. Curiously, MRNA, the technology underlying the COVID vaccinations, is also included in the top 10.

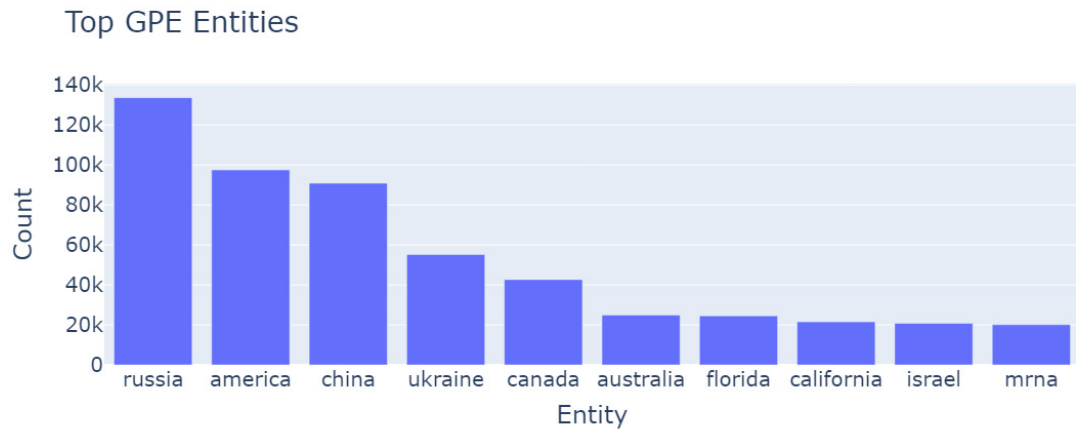


Figure 16: Top 10 GPE Entities

4.2.4.4 Top Groups

Finally, Figure 17 shows the top 10 groups (e.g. nationalities, political groups, etc.) in the corpus. The two notable inclusions here are 'Nazi', and 'Christian', with the others being nationalities or popular political groups. It is also noteworthy how equal mentions of republican and democrat are.

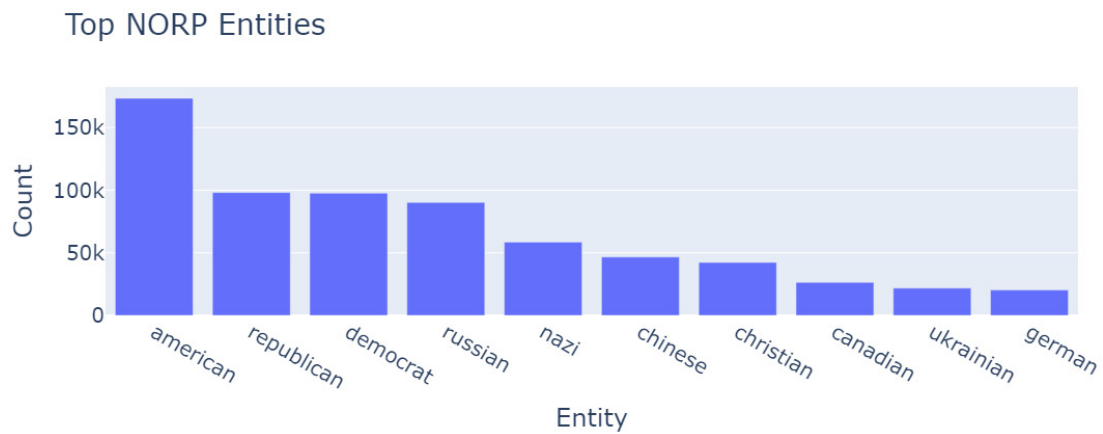


Figure 17: Top 10 NORP Entities

4.3 Topic Modelling with BERTopic

4.3.1 Model Creation

One of my initial aims in this project was to sort the comments (documents) into groups based on the topic they were discussing. For example, I intended to collate a set of comments discussing COVID, and another set discussing politics, etc. At first, I wanted to create a classifier, but as this is a supervised method of learning, it would require labelled data.

Initially, I did not view this as a problem. In fact, I created a small web application with ReactJS that would display posts from the subreddit and allow me to label them. This is shown in Figure 18 below.

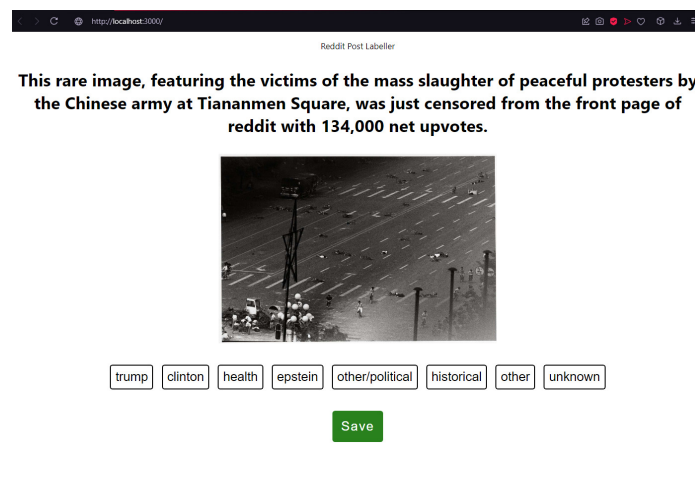


Figure 18: Screenshot of the labelling web application

This application showed both the post title, and any content such as an image or link, and allowed me to select multiple topics for a given post. I quickly realised; however, that I would need far more labelled posts than I would be able to manually tag myself.

I then started investigating alternative methods, and more specifically, methods which involved unsupervised learning methods. One of the methods that I researched, but eventually decided against, was Principal Component Analysis (PCA). I ultimately decided against PCA due to the components generated by the algorithm being notoriously hard to interpret, relative to other components. It is also very susceptible to small changes across input data [30].

I then read about topic modelling methods such as Latent Dirichlet Allocation (LDA) and BERTopic. Topic modelling is an NLP technique for discovering so-called topics in a collection of documents.

Topic models can also be referred to as probabilistic topic models, which helps describe how they achieve latent semantic analysis. One of the main advantages of topic modelling for my application is that it can be completely unsupervised, so it will take in raw data and produce a set of topics.

My goal with topic modelling was to organise individual posts in the **r/conspiracy** subreddit, so that I could then analyse trends across different theories and topic areas, to see if any interesting conclusions could be drawn.

I initially tried Latent Dirichlet Allocation, a method which had very long execution times due to the complexity of the algorithm. It did, however, have some advantages, such as the ability to easily select a specific number of topics. Upon further research I found BERTopic, which is a topic modelling technique which 'leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics' [31].

When researching BERTopic I found, as with most topic modelling methods, that the more data is put in, the better and more representative the topics are. For this reason, I grouped comments by posts and treated each post's comments as a document, disregarding any post with less than 10 comments. For comment-level analysis later, I could reliably make the assumption that the discussion under a post would be generally associated to the post topic, which was derived from the comments.

Another issue encountered when creating the model was the execution time. Ideally, the model would be transformed and fitted (trained) on all the data available. This proved inefficient; however, and was swapped for fitting on a subset of the data, and transforming on the rest. The number of posts chosen for fitting was 25,000, which was approximately 15% of the total post dataset. This number was chosen as it provided both manageable computation times, and a large sample of representative data from the document set.

The initial model generation produced 203 topics, which was reduced to 10 using BERTopic's built-in API methods to create a workable group of topics to analyse comparatively. The nature of BERTopic's reduction function is that it combines closely related topics. This leads to more general topics, with some small areas of crossover. For example, topics relating to flat earth theories may combine with theories of the Apollo 11 moon landing being fake, as there is lots of crossover in keyword usage (e.g. 'earth', 'planet', 'flat', 'moon').

The remainder of the document set can then be fit to the model to gather predictions.

4.3.2 Model Application

Now that a model has been trained on a subset of the corpus, it can be applied to the rest of the data to label each post with a tag, and as the model was trained on the comments of each post, it is safe to assume that the conversation under each post is generally about the topic at hand. To apply the model to the rest of the corpus the following snippet was used (Listing 3).

```
1 # DF of just post comment string loaded above
2 docs = df['comments'].tolist()
3
4 # gather prediction indexes
5 topics, probs = model.transform(docs)
6
7 predictions = []
8
9 # convert prediction indexes to proper name
10 for index, topic in enumerate(topics):
11     predictions.append(model.get_topic_info().CustomName[topic + 1])
12
13 # add predictions to dataframe, and select just the post ID and prediction
14 df['predicted_topic'] = predictions
15 df = df.filter(['post_id', 'predicted_topic'])
16
17 # generate dict where keys are post IDs, and values are predictions
18 post_topic_dict = dict(zip(df_post_topics.post_id, df_post_topics.predicted_topic))
19
20 # write to file
21 import pickle
22 with open('labelled_post_ids.pickle', 'wb') as f:
23     pickle.dump(post_topic_dict, f)
```

Listing 3: Application of the Topic Model

Applying the model to the rest of the dataset yielded the following distribution of topics, shown with and without the 'other' category in Figure 19.

Posts which have been modelled as 'other', will likely contain either a topic unrelated to any in the topic model, or a close combination of several topics, leading to no prevailing topic. Leaving these out of other topic groups is useful because the model has decided that they do not fit one topic well enough to be a good representation for it. If data was tagged with multiple topics, it could have led to the pollution of any conclusions made about a specific topic, as it would have contained

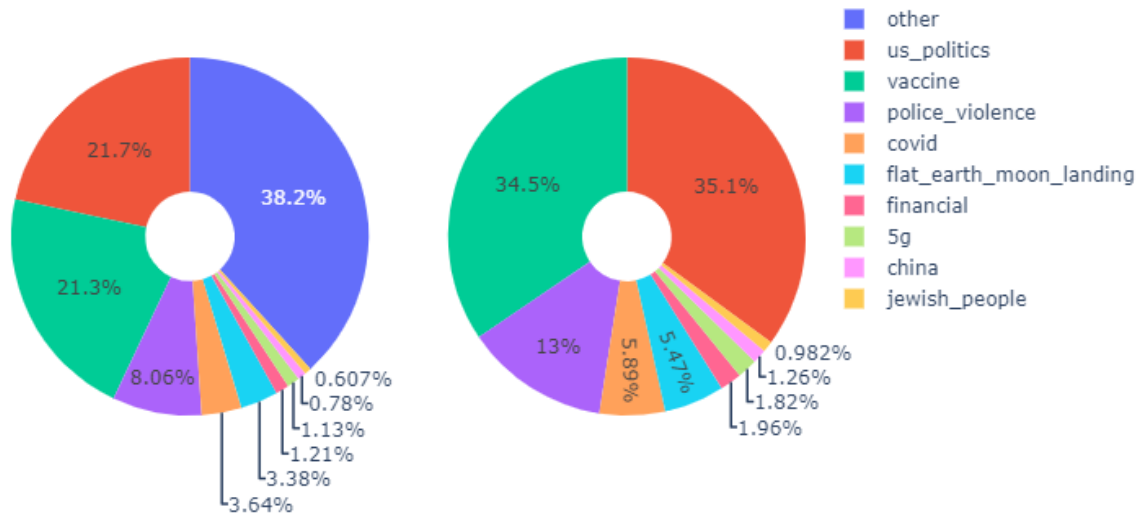


Figure 19: Pie chart of topic distribution with and without 'other'

information relating to other topics.

This was particularly problematic since a topic was generated for each post, rather than for every comment, before each post's comments were then concatenated into a single string and analysed. All comments under this post were then tagged with the resulting topic. A mixed topic distribution would thus mean the comments of a post talk about multiple topics. Putting these posts into the 'other' topic, means any conclusions made about a topic are based on comments which specifically relate to that topic, with a higher degree of certainty.

A script labelled all post IDs and saved them to a CSV file. This file was then loaded into another script to tag all comments with their post's tag and saved to a separate pickle file.

```

1 with open('labelled_post_ids.pickle', 'rb') as f:
2     labelled_post_ids = pickle.load(f)
3
4 df['label'] = df.apply(lambda x: labelled_post_ids[x['post_id']] if labelled_post_ids
    .get(x['post_id']) else 'other', axis=1)

```

Listing 4: Post Labelling

These labelled comments are now very useful, since we can now compare topics over time, allowing us to analyse the topics against political partisanship, which will be modelled further into this report. The topics generated by BERTopic were:

- US Politics
- Vaccinations
- Police Violence
- COVID
- 5G
- Flat Earth / Moon Landing
- China
- Jewish People
- Financial

4.3.3 Model Evaluation

While it is hard to precisely measure the accuracy of an unsupervised model, we can formulate a general idea of how closely different terms relate to different topics. For example, we would assume a search term such as 'george floyd' would have a large footprint in the `police_violence` tag, potentially a footprint in `other`, and little-to-no footprint in the rest of the topics, which is shown in Figure 20.

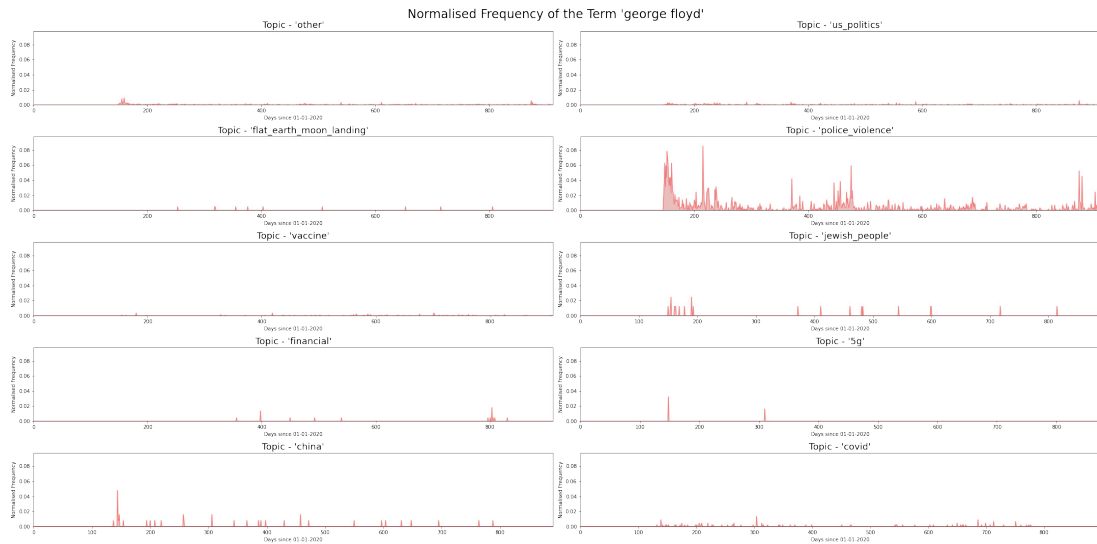


Figure 20: Normalised frequency of the n-gram 'george floyd' across topics

As can be seen above in Figure 20, the normalised frequency of the bigram 'george floyd' is most consistent and common in the `police_violence` topic, with some representation across other topics such as `china` and `jewish_people`. The main factors contributing to the peaks in unrelated topics would be due to mislabelling from the topic modelling, which is exacerbated by the normalisation of data, as the topics with unrelated peaks tend to have much less data overall, leading to mislabels being perceived as more important. However, normalisation is used because of how many posts are tagged in `us_politics` and `other`, where nearly all n-grams are mentioned at a higher total frequency than across the smaller topics.

This pattern repeated for most terms specific to a topic. They would have the highest frequency in their specific topic, and a smaller footprint in the 'other' topic. There would also typically be some peaks in unrelated topics, as discussed above, which would usually coincide with days where that topic was abnormally frequent. To counteract this in the data visualisation, the term frequencies were normalised by the mean comments per day for each topic. This helped to normalise data both across topics, and across an individual topic.

Another way we can generally judge the accuracy of the model is to look at the top words for each topic. The top eight words for each topic are shown in Figure 21.

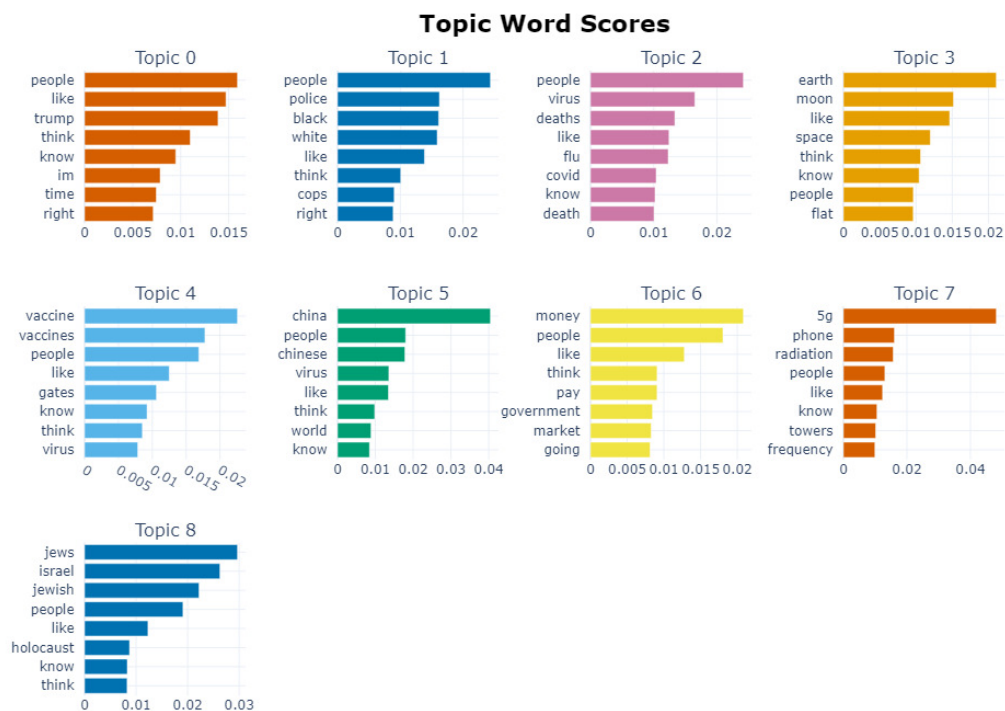


Figure 21: Top Eight words for each topic in the model

As can be seen in Figure 21, each topic has several clearly defined words in its topic set. Note that there are only nine topics present in this visualisation. This is because to have an even 10 topics, any document which does not fit into any of these topics, or fits into many equally, is marked as 'other'. We can clearly make out the 9 non-other topics in this set from their top words. This shows that if provided a suitable document, the topic model will reliably label posts from the topics correctly as long as the important words are present in the document. It also demonstrates that there is not much cross-over between topics.

It is noticeable however, that there is some cross-over between topics relating to stop-words such as 'people', 'like', 'think', etc. This problem, and its impact are discussed in the final Section of this report.

4.3.4 Model Findings

4.3.4.1 Topic Analysis Over Time

Analysis of the topical makeup of discussion over time paints a fairly clear picture of what was in the public conversation for a given time period. As can be seen, the catch-all topic 'other' stays fairly consistent, representing around 40% of all posts on the subreddit each month.

Intuitively, this tracks quite well with the graph shown in Figure 6. When we when we have more posts, we would expect more to have few comments or interactions, especially compared to when there are big events unfolding, leading to more activity. In such circumstances, there are also many duplicate posts, which will either be locked by moderators, removed by mods/users, or ignored by users.

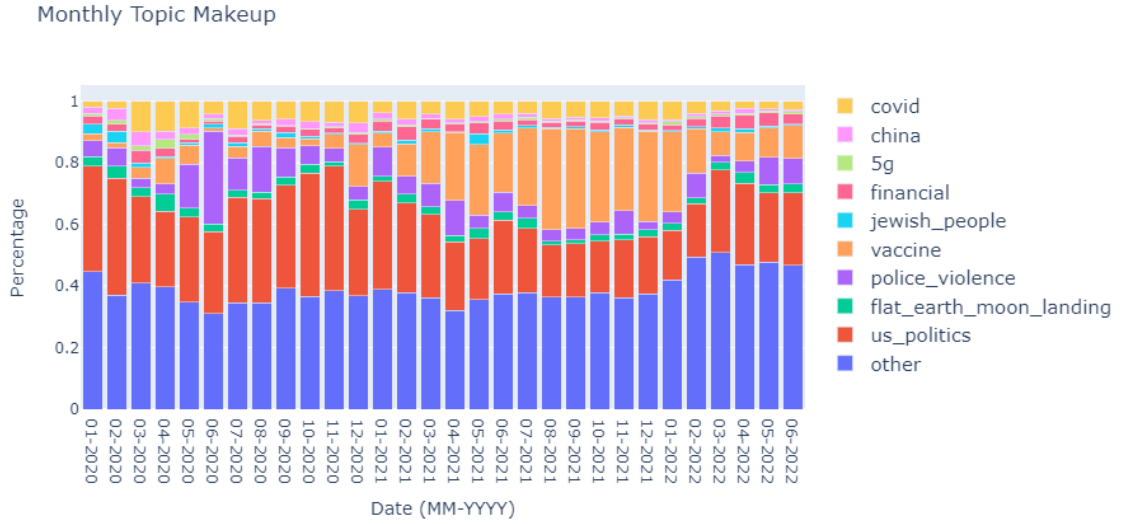


Figure 22: Evolution of all topics over time

Figure 23 shows the topic makeup of conversation over a month for every month in the dataset (without the 'other' category). A function was written, shown in Appendix A, for removing certain topics from the graph.

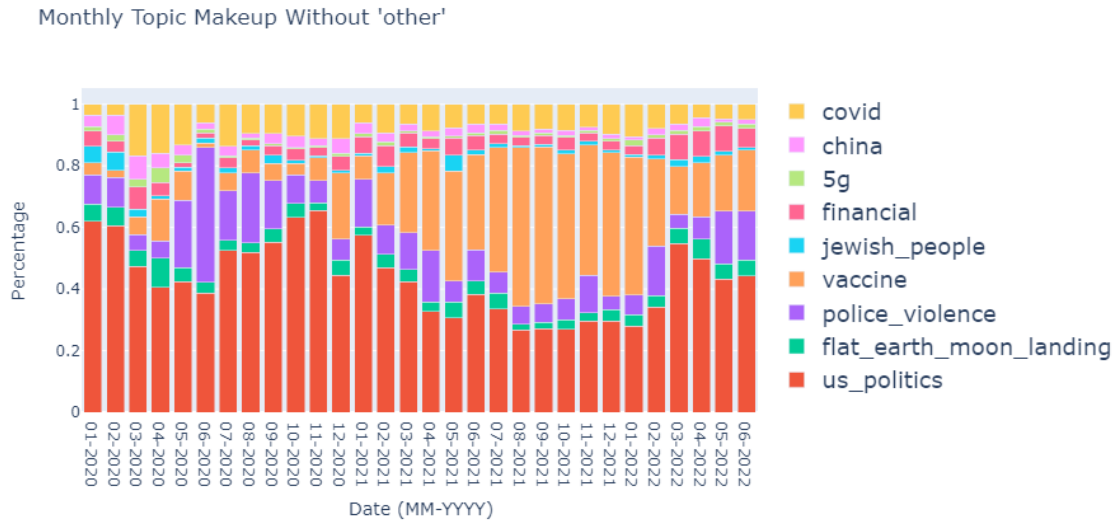


Figure 23: Topics over time without 'other'

The `us_politics` tag dominates the graph, especially prior to the surge in vaccine conversation. The peaks are around the build-up of COVID in January and February 2020, the U.S. election in November 2020, and near the capitol riot in January 2021.

There is a noticeable dip in discussion around U.S. politics around the summer through to the winter of 2021. This is mainly driven by an increase in conversation about other topics, notably vaccines. There is also a prominent rise in discussion from March 2022 to the end of the capture period, which could be accounted for by President Biden's first state of the union address, and the debate around U.S. abortion laws.

Removing the `us_politics` tag from the visualisation shows a better picture of the other topics, as well as their distribution across the data capture period, displayed below in Figure 24.

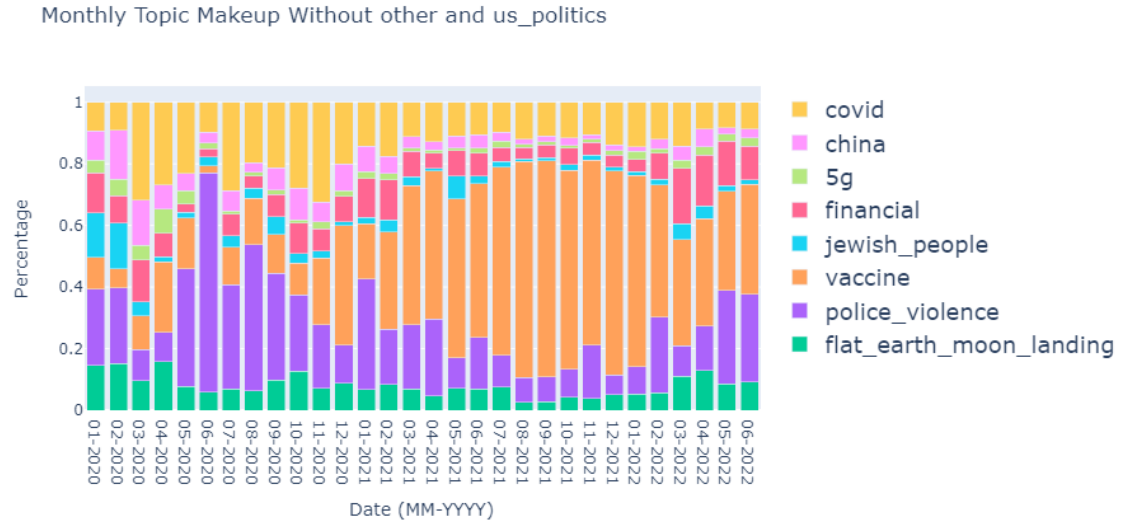


Figure 24: Topics over time without 'other' and 'us_politics'

This graph more clearly shows the trends of the 'smaller' topics in the dataset. The most noticeable change with this graph is the way in which vaccine discussion overwhelms the subreddit from late 2020 through to mid-2022. In fact, these discussions represented the dominating topic of the entire subreddit around this time, surpassing `us_politics`, consistently making up 30% of all discussion on the subreddit, over a six-month period.

It is also interesting to note the steep increase in discussion around `police_violence` from May 2020 into late 2020 and early 2021. This is mainly due to the death of George Floyd, which was the subject of approximately 30-40% of all discussion on the subreddit in June 2020. A better name for this topic may just be 'police' or 'law enforcement', as other noticeable peaks around January 2021 (capitol riot [32]), and May and June 2022 (Uvalde School Shooting [33]) relate to issues concerning law enforcement more generally, rather than violence perpetrated by police.

COVID was also a relatively large topic from early 2020 until early 2021, where discussion around the topic was seemingly replaced with the vaccine topic, reflecting a shift in the subreddit's zeitgeist from COVID conspiracy theories to vaccine theories. This could potentially also be a quirk of the topic modelling system, as COVID and vaccines would be quite similar topics, with potentially similar keywords. Since vaccines did not enter the conversation until late 2020, any post mentioning

COVID would then typically mention vaccines, categorising that post under the vaccine topic. We also see increases in discussion around `jewish_people` whenever there were flair-ups in the Israel-Gaza conflict, for example around January and February 2020, May 2021, and March 2022. Finally, as mentioned previously, the GameStop short squeeze was a big talking point around January 2021, something which can be seen in the increase in interest around the financial topic in early 2021.

4.3.4.2 Single-Issue Users

An initial interesting piece of exploratory work completed with these new topics was looking into single-issue users. By this, I mean users who only discuss one topic 75% or more of the time. This was achieved through DataFrame manipulation using `pandas`, and enabled the creation of the following graphic, shown in Figure 25.

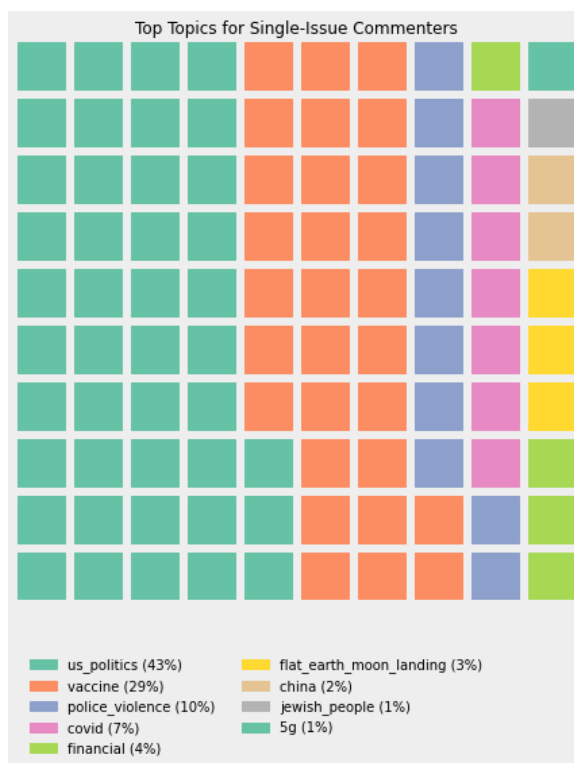


Figure 25: Top Topics for Single-Issue Users

This shows that the majority of single-issue users only discuss US politics and vaccinations. The groups then slowly decrease down to Jewish people and 5G which each make up 1%. This somewhat

lines up with the overall topic makeup of the subreddit, but with a larger portion of US politics. This is also in line with prior hypotheses which postulate that the **r/conspiracy** subreddit is more of a political subreddit than an actual conspiracy subreddit.

4.3.5 Next Steps

This section was used to increase familiarity with the dataset, and with the methods in use throughout the project. Now that a group of coherent and sound topics had been generated, a major aim of the project had been met. This enabled me to move on to the final task of this project - creating a political classifier, which will be used to analyse partisanship across the subreddit, and across the topics generated above.

The following Section explores the creation of this classifier, as well as its applications.

5 Political Classification

One of the main objectives of this project was to create a classifier which could classify comments based on whether they were in support of a political ideology, for example left-wing, right-wing, or neutral. This section discusses data collection, data pre-processing, classifier training, and classifier application for this political classifier.

The classifier created here is used to analyse the subreddit as a whole, specific topics, and the subreddit over time. This Section is where the major goals of this project are achieved.

5.1 Data Gathering

As this classifier’s goal was to produce two labels for its predictions, my earlier usage of BERTopic would likely generate unsatisfactory results, since the author has stated with any model with less than 10 topics, there is too little data available to generate any meaningful or coherent topics [34]. This led me to decide to use a supervised method. Several algorithms were trialled, including Multinomial Naive Bayes and Support vector Machines (SVMs), which are discussed in the Background Section of this report. The drawback of using a supervised method is that labelled data is required, meaning that a large dataset of labelled political discourse would be needed to either be created or found.

The strategy for data collection would again make use of PS_REDDIT_TOOL. I decided to gather more Reddit data from other communities, dedicated to certain branches of political discourse. For more right-leaning conversation, **r/conservative** was chosen, which yielded 5,746,843 comments. There is no subreddit of comparable size for left-wing discussion, but one example could potentially be **r/politics**, which while it does have an alleged left-wing bias, contains too much of a mix of conversation to be classified as purely left-wing. Therefore, for a sample of left-wing discussion, a combination of the subreddits **r/joebiden** and **r/liberal** were used, which provided 527,446 comments and 65,149 comments respectively. The polarised nature (and keywords) of these communities lends itself well to the task of text classification.

Once the data had been collected into a CSV file over the period of January 2020 to June 2022, comments with a negative score would be discarded to avoid comments from opposing ideologies being tagged incorrectly.

After removing comments with a negative (or zero) score, there were 4,918,559 comments from **r/conservative**, 58,491 comments from **r/liberal**, and 505,534 from **r/joebiden**.

Political subreddits are notoriously heavily-moderated, as they usually have rules against 'hate speech, trolling, and personal armies [brigading]' [35]. In combination with the score threshold should increase the accuracy of the training set greatly.

A table showing the breakdown of corpus sizes, along with the sizes when certain constraints are imposed is shown in Table 2.

	Collected	Defined	Positive Score	> 100 Characters	All Combined
r/conservative	5,746,843	5,746,825	4,918,546	2,993,398	2,524,781
r/liberal	65,149	65,038	58,399	24,196	21,634
r/joe Biden	527,446	526,300	504,461	175,049	166,365

Right Corpus	5,746,843	5,746,825	4,918,546	2,993,398	2,524,781
Left Corpus	592,595	591,338	562,860	199,245	187,999

Table 2: Classifier Training Corpus Sizes

5.2 Pre-Processing

Before any text classification, the raw data needs to be processed for multiple reasons. One such reason is to reduce the dimensionality of the vectorised data. Some methods of performing this include removing stop-words (such as 'and' or 'the') and removing punctuation.

This is also done to standardise data. For example, if we had a name such as 'Joe' present in the data, then the word 'joe' has the same meaning for the purposes of text classification. In fact, this helps the classifier as it will treat the occurrences as the same word, as opposed to a capitalised version meaning something different.

For this classifier, pre-processing consisted of:

1. Removing punctuation
2. Converting entire corpus to lower-case
3. Removing stop-words

The removal of punctuation and text-lowering was achieved with basic Python string manipulation while to remove stop-words from the corpus, the `stop_words` property was used in `scikit-learn`'s `CountVectorizer` as described below.

5.3 Text Vectorisation

The next stage was to create a numerical representation of the corpus. This is essential because most if not all machine learning techniques rely on numerical data. There are many accepted ways of performing vectorisation, but for this classifier, I used a count vectoriser and TF-IDF.

Initially, the count vectoriser converts the data into a matrix of token counts per document, which takes the form $TOKENS \times DOCS$. This is why it was essential to reduce the dimensionality of the data, as the relationship between tokens and documents is multiplicative, thus when one token is added, a new row is added to the matrix, which can be millions of rows for large text corpora.

Now that I had a matrix of token counts, I could apply TF-IDF which provided me with a matrix which aimed to reflect how important a word is to a document, in relation to the rest of the corpus. It makes use of 'inverse document frequency' to reduce the weight of very commonly occurring terms, which would include stop-words such as 'the'.

5.4 Training

To train the model, a trial-and-error approach was used, tweaking values such as corpus constraints, the **n-gram** range of the vectoriser, and the algorithm used for classification. To pick the parameters for the vectoriser, TF-IDF transformer, and classifier, **scikit-learn** provides a helpful optimisation tool, which runs the classifier pipeline multiple times with different parameters, and returns the best result achieved, with the used parameters. This resulted in the classifier setup shown below.

```
1  from sklearn.pipeline import Pipeline
2  from sklearn.feature_extraction.text import CountVectorizer
3  from sklearn.feature_extraction.text import TfidfTransformer
4  from sklearn.naive_bayes import MultinomialNB
5
6  clf = Pipeline([
7      ('vect', CountVectorizer(
8          ngram_range=(1,2),
9          stop_words=stopwords.words('english'),
10         strip_accents='ascii'
11     )),
12     ('tfidf', TfidfTransformer()),
13     ('clf', MultinomialNB()),
14 ])
```

Listing 5: Classifier Definition

The following table shows the F1 scores which were achieved for each combination of parameters. Note - The dataset size is split 50/50 for each tag, this is done by selecting the size of the smallest corpus, and trimming the other corpus to that length.

N-Gram Range	Algorithm	Restrictions	F1 Score	Training Time (s)
(1, 1)	SVM	Defined	0.7615	50.674
(1, 1)	MNB	Defined	0.7814	23.703
(1, 2)	SVM	Defined	0.7556	120.820
(1, 2)	MNB	Defined	0.7955	70.898
(1, 3)	SVM	Defined	0.7536	214.088
(1, 3)	MNB	Defined	0.7964	185.880
(1, 1)	SVM	Defined Positive Score	0.7620	48.512
(1, 1)	MNB	Defined Positive Score	0.7800	23.661
(1, 2)	SVM	Defined Positive Score	0.7567	115.052
(1, 2)	MNB	Defined Positive Score	0.7956	69.136
(1, 1)	SVM	Defined Positive Score > 100 Chars	0.8242	20.834
(1, 1)	MNB	Defined Positive Score > 100 Chars	0.8266	12.655
(1, 2)	MNB	Defined Positive Score > 100 Chars	0.8282	42.725
(1, 3)	MNB	Defined Positive Score > 100 Chars	0.8276	79.679

Table 3: F1 scores for different classification configurations

The highest F1 score achieved was 0.8282, with an n-gram range of (1, 2), and using Multinomial Naive Bayes classification. When using the same parameters but with an n-gram range of (1, 3), the score decreased, which is useful, as whenever a prediction is made using the resulting classifier, the input data needs to be transformed in the same way as the training data. Therefore, for any higher n-gram ranges, more processing is needed when predicting values, but this is minimised with a range of (1, 2), especially since the increase of 0.002 in score could potentially be a swing of nearly 3000 labels over the 13 million comment corpus.

As the model has an F1 score of over 0.8, it is accurate enough to make general observations on large text corpora. However, when performing fine-grained analysis on single comments, it may prove more unreliable. This model was used going forward to make observations on the **r/conspiracy** dataset.

The scores of the classifier could potentially have been improved further, with more data, but the availability of this data was a big limitation, as the returns of adding a new subreddit are diminishing as they are usually smaller and smaller communities. This is discussed further in the next Section regarding future work.

5.5 Model Application

`sklearn`'s implementation of a Multinomial Naive Bayes classifier has a method called `predict_proba` which allows us to predict a tag, returning the probability prediction for each tag.

Using this, any given comment that had no probability for either tag greater than 0.55 was marked as neutral. This would typically include very short comments, mostly replies, and in some cases, genuinely neutral comments.

Using the probabilities predicted, tags were generated and attached to individual comments, before being saved using `pandas` for further analysis.

The actual application of the classifier is shown below in Listing 6.

```
1 # time prediction time
2 from time import time
3 t0 = time()
4
5 valid_df = df.loc[df.body.str.len() > 0]
6
7 predictions = clf.predict_proba(valid_df['body'])
8
```

```

9 print(f"Made {len(predictions)} predictions in {round(time() - t0, 3)}s")
10
11 # assign letters for partisanship depending on probability
12 neutral_threshold = 0.55
13 for prediction in predictions:
14     if prediction[0] > neutral_threshold:
15         predictions_cleaned.append('L')
16     elif prediction[1] > neutral_threshold:
17         predictions_cleaned.append('R')
18     else:
19         predictions_cleaned.append('N')
20
21 valid_df['partisanship'] = predictions_cleaned

```

Listing 6: Classifier Application

5.6 Findings

5.6.1 Overall Distribution

Applying the model to the **r/conspiracy** dataset gives us the following distribution of tags, shown in Figure 26.

Political Tag Distribution

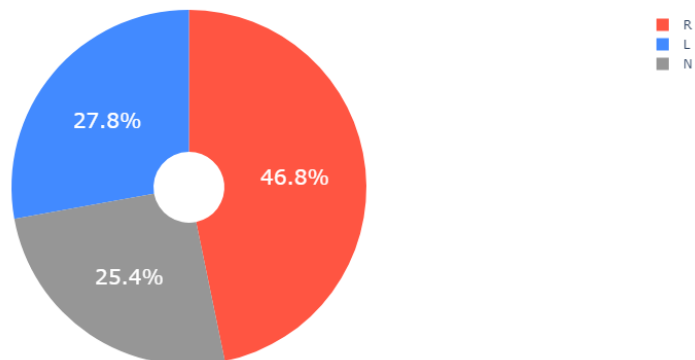


Figure 26: **r/conspiracy** comment partisanship distribution

As clearly shown in Figure 26, the plurality of discussion on the **r/conspiracy** subreddit is right-wing. More specifically, the comments of **r/conspiracy** are heavily similar to **r/conservative**, and are quite dissimilar to **r/joe Biden** and **r/liberal**.

This can also be seen when looking at the partisan composition across different topics, shown in Figure 27.

5.6.2 Partisanship Across Topics

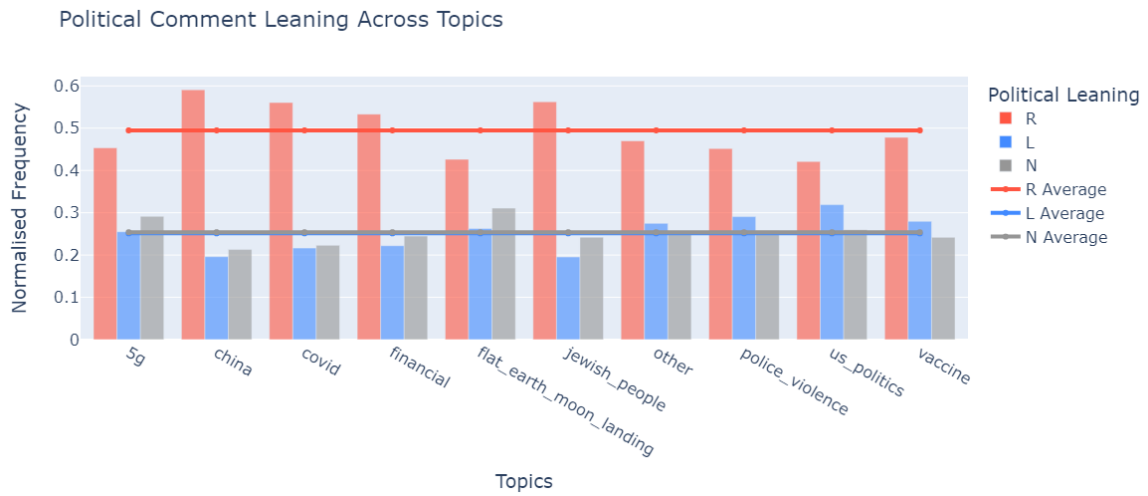


Figure 27: Partisanship across topics

This graphic is quite interesting as it shows several intriguing observations. Firstly, on posts relating to COVID-19, Jewish People, and financial conspiracies, there is much more right-wing discussion than expected, namely content which would typically be found in more conservative areas of Reddit. In addition to this, topics such as police violence, vaccinations, and general U.S. politics, have more discussion using left-wing language.

There is also an interesting spike in the China topic, which has above average right-wing discussion on posts relating to it. This could potentially also be linked to COVID, as in the data analysis time period (Jan 2020 to June 2022), COVID-19, and its origins, were the subjects of numerous conspiracy theories relating to China. Many of the China-related posts on Reddit were also discussing topics which were somewhat 'meta', for example, posts about censorship on Reddit after a small stake in

Reddit was purchased by Tencent, a Chinese technology company [36]. Finally, the top post of all time on the subreddit is a post commemorating the Tiananmen Square massacre, with a decently sized contingent of posts relating to China being of this type, criticising the leadership of the Chinese Communist Party.

Splitting out the graphs into their individual partisan labels provides a less-crowded view of the data. This is included in Appendix B.

5.6.3 Partisanship Over Time

Another area for analysis was partisanship over time. Figure 28 shows the daily breakdown of partisanship for all comments.

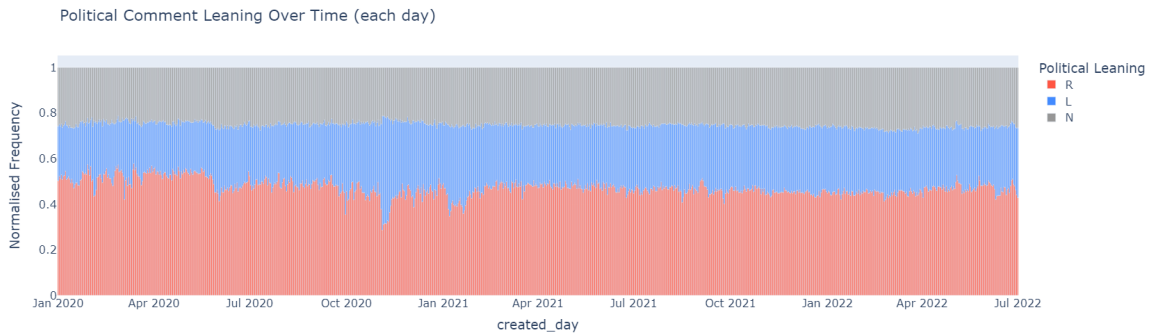


Figure 28: Partisanship Over Time (Daily)

The neutral tag stays relatively stable throughout the whole collection period. Particularly interesting are the other two tags. There is a period from late 2020 through to early 2021 where the R and L tags become rather unstable. Figure 29 shows a subsection of the graph around this time.

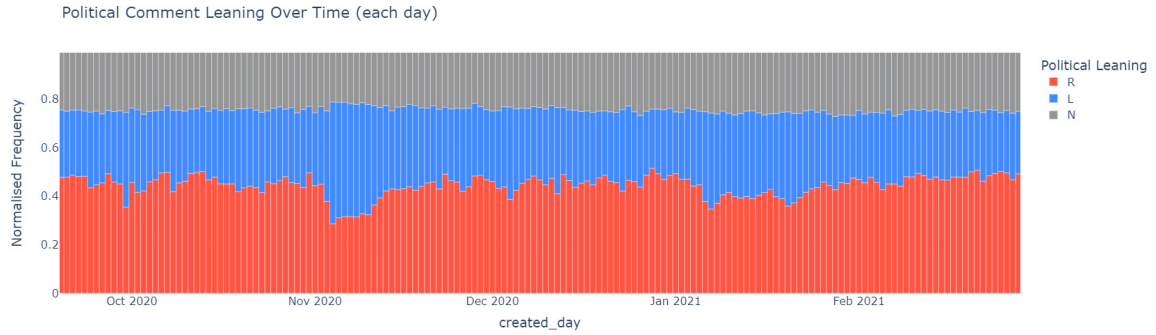


Figure 29: Partisanship Over Time (Daily) - Subsection

As can be seen in Figure 29, there is a big trough in right-wing discussion in early November, immediately after the 2020 U.S. election on November 3rd. This trough could also be seen as a peak in left-wing discussion, as many were uncomfortable with Donald Trump’s assertions that the election was fraudulent, with one poll showing 29% of respondents stating that they believed the election was fraudulent [37].

We also see similar peaks/troughs on September 30th, which was the day following the first presidential debate, the period of 6th to the 9th of January, the days including and following the 2021 Capitol riot, and around the date January 20th 2021, when Joe Biden was inaugurated at a fortified capitol building.

6 Conclusion

6.1 Summary of Findings

- The plurality of discussion in **r/conspiracy** is classed as right-wing
- There was more than expected right-wing discussion around topics including China, COVID, financial conspiracies, and conspiracies relating to Jewish people
- There was more than expected left-wing discussion around topics including 5G, flat earth/moon landing, police violence, US politics, and vaccinations
- Large increases in left-wing discussion whenever an event damaging to the right-wing occurs, but the reverse is much less common

6.2 Discussion of Aims

The goal of this project was to analyse the political and topical make-up of the **r/conspiracy** subreddit. This was achieved through both topic modelling with **BERTopic** and political classification with Multinomial Naive Bayes.

The first aim was to gather comments from the **r/conspiracy** subreddit over the period of January 2020 to June 2022. This was achieved using the PushShift data repository, and resulted in 13 million comments for analysis.

BERTopic was successfully executed on the corpus, generating 10 topics that were then analysed, both on their own and alongside the political classifier. As the method employed was unsupervised, methodologies were discussed for testing the accuracy of the model, which demonstrated that the model was providing coherent topics, as well as suitably fitting posts to those topics.

A classifier was then created to differentiate left- and right-wing text. This classifier was able to determine that the plurality of conversation in **r/conspiracy** was right-wing, with the rest evenly being made up by neutral and left-wing discussion. This classifier was also able to identify that there was more extensive-than-expected right-wing discussion around topics such as China, COVID, financial conspiracies, and conspiracy theories relating to Jewish people (e.g., holocaust denial). Similarly, it was also used to identify that there was more extensive-than-expected left-wing discussion around topics such as 5G, flat earth and moon landing conspiracies, police violence/incompetence, general U.S. politics, and vaccinations. Finally, the classifier was also used to analyse partisanship over time, showing that during events which were damaging to the image of the Republican Party in the

United States, left-wing discussion increased greatly, whereas right-wing discussion stayed relatively stable during events damaging to the Democratic Party, such as news surrounding Hunter Biden in the build up to the election in 2020.

6.3 Future Work

The results of this dissertation, while useful and successful, could be improved in several areas. This section will discuss the ways in which the results of this paper could be improved upon or taken further for additional analysis.

6.3.1 BERTopic Topic Modelling

One option for greater analysis, would be to perform more fine-grained topic modelling over a wider range of topics. My investigation, while yielding interesting results, took topics very generally. For example `'us_politics'` encapsulates a very large scope for posts on the subreddit. This could be split into topics such as `'trump'`, `'qanon'`, `'hunter_biden'`, etc.

Another clear area for improvement would be in data pre-processing. The most obvious example of this appears in the word structure of the topic model shown in Figure 21. Words such as `'like'` and `'people'` appear frequently across topics, so more robust wordlists are needed to filter out stop-words. However, this once again depends on the context of the word's usages. This does not prove to be a large issue for this project, as the topic model was sound and accurate as shown through the topic over time charts in Figures 22, 23, and 24, as well as normalised n-gram frequency such as in Figure 20.

This may enable a much more detailed view of topics over time and allow for the analysis of partisanship across more topics, something that would facilitate the drawing of further conclusions.

6.3.2 Political Classification

The classification of political partisanship constituted a very difficult part of this project, mostly because of the nature of text processing. It is usually best applied to problems with very distinct classes. For example, classifying science literature from religious literature, where the words used across the classes are generally distinct. This is especially true for probabilistic models such as Naive Bayes.

When we have a problem, such as classifying a subset of a text class; for example, right and left speech as a subset of political speech, most of the keywords in use are not exclusive to one group.

The words 'Trump', 'government', and 'Biden' would be present in most of the corpus. Logically, a classifier for this purpose would perform better on n-grams, looking for how these special keywords are used in the context of other words. For example, right-wing individuals may use the bigram 'sleepy biden', whereas a left-wing individual may not. This; however, then presents the problem that only a minority of comments will include speech like this, and this is where the nature of Reddit also comes into the equation.

Reddit uses a nested comment 'forest' system, where there are root (top) level comments, and a tree spans out from this comment to its replies, which can in turn have their own comments in reply. This type of system lends itself to having a detailed top comment, which is in direct response to the post, and the comments in reply to it are more centered on that specific comment. This can lead to short comments with no useful context for the classifier, which when the length of a comment is short enough, means that the classifier is essentially guessing. This structure also leads to comments which are more general statements containing mostly stop-words or words which hold no specific value to the classifier. An example may be a sentence such as 'Yes, i agree. They are a disgrace.', which we know expresses an opinion on something, but we cannot know what it is in isolation, without further context from the post or parent comment.

A potential solution to this would be to only consider top-level comments. However, this diminishes the size of the dataset, and therefore potentially limits the scope of any topic modelling or classification. This is possible with PushShift as it makes an attribute `parent_id` available, which lists the parent of the current comment. Another potential solution may be to score candidate sentences, possibly through TF-IDF scoring, to see if they contain enough content to be classified, and to discard comments without enough context.

The problem of creating an accurate political classifier has been visited before, with a group of students from Stanford managing to get to an aggregate F1 score of 0.684 [38], so it is clearly no easy feat to create a reliable political classifier.

As with any classifier, improvement would also come with more data. The training data used for the classifier was all comments with a positive score from the subreddits **r/conservative**, **r/liberal**, and **r/joe Biden**. Data from before the capture period is available for collection from PushShift. However, vast quantities of data are only available going back a few years, as before this these subreddits were much, much smaller, and did not have the political makeup that they have today. This would also present an issue as different topics would have been discussed at these times, which would not improve the analysis over the capture period.

In summary, while the classifier created is suitable for the purposes of general analysis of a wide range of data, if more granular analysis was required, it may fall short due to its inaccuracy. This could be improved by implementing a variety of methods, including more strict data collection parameters (e.g., minimum length, minimum sentence score), or different text transformation methods, such as using an alternative text vectoriser, or more specific stop-word lists.

Appendices

A plot_without Listing

```
1 import numpy as np
2
3 def plot_without(tags=[]):
4     x = list(counts_by_month['other'].keys())
5
6     current_tags = [tag for tag in list(monthly_percentges.keys()) if tag not in tags
7 ]
8
9     new_monthly_totals = {}
10
11     for tag in current_tags:
12         for month in x:
13             if new_monthly_totals.get(month, False):
14                 if not counts_by_month[tag].get(month, False): continue
15                 new_monthly_totals[month] += counts_by_month[tag][month]
16             else:
17                 new_monthly_totals[month] = counts_by_month[tag][month]
18
19     sorted_dates = sorted(new_monthly_totals.keys(), key = lambda x:datetime.strptime
20 (x, '%m-%Y'))
21
22     new_monthly_totals = {k: new_monthly_totals[k] for k in sorted_dates}
23
24     new_monthly_percentges = {}
25     for tag in current_tags:
26         new_monthly_percentges[tag] = {}
27         for month in sorted_dates[:-1]:
28             new_monthly_percentges[tag][month] = counts_by_month[tag][month] /
29 new_monthly_totals[month]
30
31     for tag in current_tags:
32         new_monthly_percentges[tag] = list(new_monthly_percentges[tag].values())
33
34     x = sorted_dates[:-1]
35
36     fig = go.Figure()
```



```

34
35     for tag in current_tags:
36         fig.add_trace(go.Bar(x=x, y=new_monthly_percentges[tag], name=tag,
37                               marker_color=tag_colours[tag]))
38
39     topic_string = ''
40
41     if len(tags) == 1:
42         topic_string = f'Without \'{tags[0]}\''
43     elif len(tags) > 1:
44         topic_string = f'Without {" and ".join(tags)}'
45
46     fig.update_layout(
47         barmode='stack',
48         title=f"Monthly Topic Makeup {topic_string}",
49         xaxis_title="Date (MM-YYYY)",
50         yaxis_title="Percentage",
51         legend=dict(
52             font=dict(
53                 size=15
54             )
55         )
56
57     fig.show()
58
59 plot_without(['other'])

```

Listing 7: Definition of the `plot_without` Function

B Separated Partisan Topic Graphs

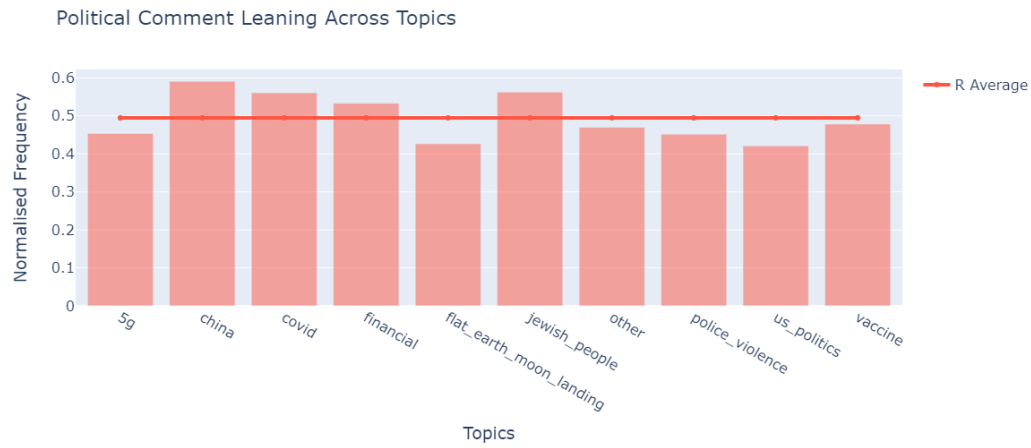


Figure 30: Right-Wing Topic Breakdown

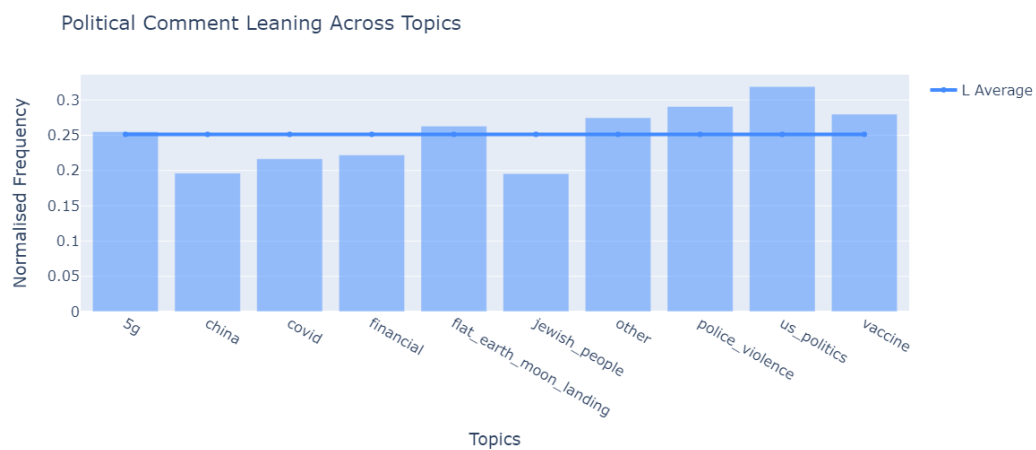


Figure 31: Left-Wing Topic Breakdown

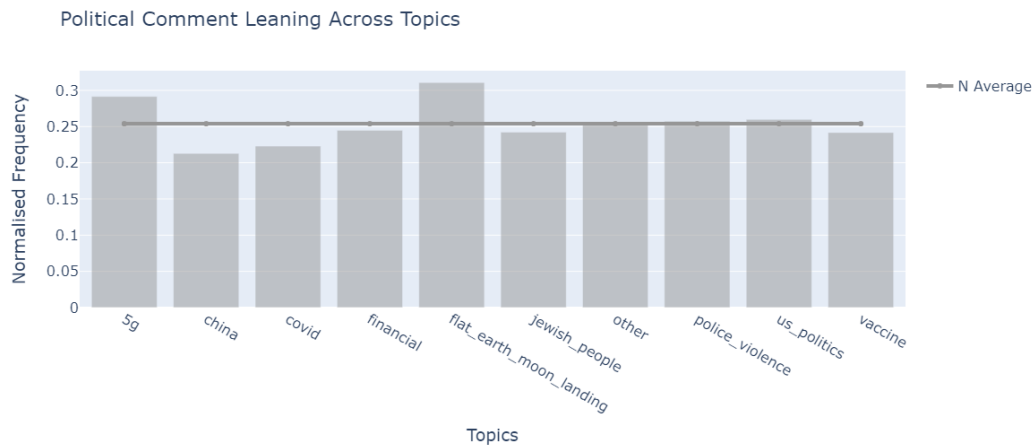


Figure 32: Neutral Topic Breakdown

C JSON Dictionary to DataFrame Function

```

1 import pandas as pd
2 from nltk.corpus import stopwords
3
4 sws = stopwords.words('english')
5
6 def dict_to_df(dict):
7     df = pd.DataFrame.from_dict(dict)
8
9     # keep needed columns
10    df = df[['id', 'author', 'author_created_utc', 'body', 'score', 'created_utc', '
    controversiality', 'author_flair_text', 'permalink']]
11
12    # remove rows with NaN created dates
13    df = df[df.author_created_utc.notnull()]
14
15    # remove rows with author of AutoModerator
16    df = df[df.author != 'AutoModerator']
17
18    # convert permalink to title
19    df['truncated_title'] = df['permalink'].apply(lambda x: ' '.join(x.split("/")[5].
    split("_")))
20
21    # get post_id from permalink

```

```

22 df['post_id'] = df['permalink'].apply(lambda x: x.split("/")[4])
23
24 # format unix datetimes to standard format
25 df['author_created_utc'] = df['author_created_utc'].apply(lambda x: datetime.
    fromtimestamp(x).strftime('%d-%m-%Y %H:%M:%S'))
26 df['created_utc'] = df['created_utc'].apply(lambda x: datetime.fromtimestamp(x).
    strftime('%d-%m-%Y %H:%M:%S'))
27
28 # remove any non-alphanumeric characters
29 df['body'].str.replace('[^A-z0-9]', '')
30
31 def process_comment(comment):
32     # to lowercase
33     comment = comment.lower()
34     # remove stopwords
35     comment = ' '.join([token for token in comment.split() if token not in sws])
36     # remove punctuation
37     comment = comment.translate(str.maketrans('', '', punctuation))
38     return comment
39
40 df['body'] = df['body'].apply(lambda x: process_comment(x))
41
42 return df

```

Listing 8: Definition of the dict_to_df Function

References

- [1] Yahoo Finance. Gamestop corp. (gme), 2021. <https://finance.yahoo.com/quote/GME/history?p=GME>.
- [2] Jake Widman. What is reddit? 2021. <https://www.digitaltrends.com/computing/what-is-reddit/>.
- [3] Danny Dover. Reddit, stumbleupon, delicious and hacker news algorithms exposed! 2008. <https://moz.com/blog/reddit-stumbleupon-delicious-and-hacker-news-algorithms-exposed>.
- [4] P Türkmen. Disinformation: One of the greatest threats to european democracy. 2021. <https://pathforeurope.eu/disinformation-one-of-the-greatest-threats-to-european-democracies/>.
- [5] Office of the Director of National Intelligence. Assessing russian activities and intentions in recent us elections. 2017. https://www.dni.gov/files/documents/ICA_2017_01.pdf.
- [6] U.S. Department of State. Fact vs. fiction: Russian disinformation on ukraine. 2022. <https://www.state.gov/fact-vs-fiction-russian-disinformation-on-ukraine/>.
- [7] W Jennings, G Stoker, H Bunting, V Valgarsson, J Gaskell, D Devine, L McKay, and M Mills. Lack of trust, conspiracy beliefs, and social media use predict covid-19 vaccine hesitancy. 2021. <https://www.mdpi.com/2076-393X/9/6/593>.
- [8] A Voela. Lockdown, conspiracy theories: Inaction, transmission, stupidity. 2022. https://link.springer.com/chapter/10.1007/978-3-030-80278-3_14.
- [9] Victoria A. Goodyear. Social media use informing behaviours related to physical activity, diet and quality of life during covid-19: a mixed methods study. 2021. <https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-021-11398-0>.
- [10] Joseph Guzman. More than 70 cell phone towers in the uk have been set on fire due to 5g coronavirus conspiracy theory. 2020. <https://thehill.com/changing-america/well-being/longevity/496502-more-than-70-cell-phone-towers-in-the-uk-have-been-set/>.
- [11] Salesforce. How ceos are leading companies in times of uncertainty, 2020. <https://www.youtube.com/watch?v=5qo-33xH1QM&t=1410s>.

- [12] R Gweryina, C Madubueze, and F Kaduna. Mathematical assessment of the role of denial on covid-19 transmission with non-linear incidence and treatment functions. 2021. <https://www.sciencedirect.com/science/article/pii/S2468227621001150>.
- [13] A Enders, C Farhart, J Miller, J Uscinski, K Saunders, and H Drochon. Are republicans and conservatives more likely to believe conspiracy theories? 2022. <https://link.springer.com/article/10.1007/s11109-022-09812-3>.
- [14] Ben Wasike. Framing social news sites: An analysis of the top ranked stories on reddit and digg. 2011. https://www.researchgate.net/publication/311068594_Framing_Social_News_Sites_An_Analysis_of_the_Top_Ranked_Stories_on_Reddit_and_Digg.
- [15] N Shrading, C Alm, R Ptucha, and C Homan. An analysis of domestic abuse discourse on reddit. 2015. https://www.researchgate.net/publication/301446050_An_Analysis_of_Domestic_Abuse_Discourse_on_Reddit.
- [16] N Proferes, N Jones, and Gilbert S. Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. 2021. <https://journals.sagepub.com/doi/full/10.1177/20563051211019004>.
- [17] Reddit Staff. Reddit recap 2021. 2021. <https://www.redditinc.com/blog/reddit-recap-2021>.
- [18] C Davies, J Ashford, L Espinosa-Anke, A Preece, L Turner, and R Whitaker. Multi-scale user migration on reddit. 2021. <https://orca.cardiff.ac.uk/id/eprint/142389/>.
- [19] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. 2022. <https://arxiv.org/pdf/2203.05794.pdf>.
- [20] StackOverflow. Stack overflow developer survey 2022. 2022. <https://survey.stackoverflow.co/2022/#most-popular-technologies-language>.
- [21] Jason Baumgartner. pushshiftio, 2019. <https://github.com/pushshift/api>.
- [22] Magnus Nissel. ps_reddit_tool, 2021. https://github.com/magnusnissel/ps_reddit_tool.
- [23] David Rogers. A multi-disciplinary co-design approach to social media sensemaking with text mining. 2021. <https://orca.cardiff.ac.uk/id/eprint/143726/2/David%20Rogers%20-%>

20Thesis%20-%20A%20Multi-Disciplinary%20Co-Design%20Approach%20to%20Social%20Media%20Sensemaking%20with%20Text%20Mining.pdf.

- [24] J Zhang, D Carpenter, and M Ko. Online astroturfing: A theoretical perspective. 2013. https://www.researchgate.net/profile/Darrell-Carpenter/publication/286729041_Online_astroturfing_A_theoretical_perspective/links/56df195908ae979addef5103/Online-astroturfing-A-theoretical-perspective.pdf.
- [25] A Cichocka. To counter conspiracy theories, boost well-being. 2020. https://kar.kent.ac.uk/84701/12/Cichocka_To%20counter%20conspiracy%20theories.pdf.
- [26] X Chen, S Zhang, A Jahanshahi, A Alvarez-Risco, H Dai, and J Li. Belief in conspiracy theory about covid-19 predicts mental health and well-being: A study of healthcare staff in ecuador. 2020. <https://www.medrxiv.org/content/10.1101/2020.05.26.20113258v2>.
- [27] R Davies. Gamestop: how reddit amateurs took aim at wall street’s short sellers. 2021. <https://www.theguardian.com/business/2021/jan/28/gamestop-how-reddits-amateurs-tripped-wall-streets-short-sellers>.
- [28] S Cabral and LaCombe A. Robinhood, reddit, and gamestop: What happened and what should happen next? 2021. <https://www.scu.edu/ethics/focus-areas/business-ethics/resources/robinhood-reddit-and-gamestop-what-happened-and-what-should-happen-next/>.
- [29] S Vychegzhanin and E Kotelnikov. Comparison of named entity recognition tools applied to news articles. 2019. <https://ieeexplore.ieee.org/abstract/document/8991165>.
- [30] S Lee. Drawbacks of principal component analysis. 2010. <https://arxiv.org/pdf/1005.1770.pdf>.
- [31] Maarten Grootendorst. Bertopic, 2022. <https://maartengr.github.io/BERTopic/index.html>.
- [32] Capitol riots timeline: What happened on 6 january 2021? 2022. <https://www.bbc.co.uk/news/world-us-canada-56004916>.
- [33] S Dey. 21 killed at uvalde elementary in texas’ deadliest school shooting ever. 2022. <https://www.texastribune.org/2022/05/24/uvalde-texas-school-shooting/>.

- [34] Maarten Grootendorst. Force bertopic to create n topics #443, 2022. <https://github.com/MaartenGr/BERTopic/issues/443>.
- [35] C Fiesler, J Jialun, J McCann, K Frye, and J Brubaker. Reddit rules! characterizing an ecosystem of governance. 2018. <https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17898/16998>.
- [36] J Constine. Reddit confirms \$300m series d led by china’s tencent at \$3b value. 2019. <https://techcrunch.com/2019/02/11/reddit-300-million/>.
- [37] Quinnipiac University. 85% of republicans want candidates to agree with trump, quinnipiac university national poll finds; americans support early cut to federal jobless benefit. 2021. <https://poll.qu.edu/poll-release?releaseid=3810>.
- [38] M Bhand, D Robinson, and Sathi C. Text classifiers for political ideologies. <https://nlp.stanford.edu/courses/cs224n/2009/fp/7.pdf>.