CSC110 Project: A Study of Online Conspiracy Theories Regarding COVID-19

Aaron Ma, Benjamin Liu, Vishnu Nittoor

Friday, November 5, 2021

1 Introduction

Research Question **What is the nature of the circulation and development of conspiracy theories surrounding COVID-19?**

Problem Description Since the advent of the COVID-19, the world has witnessed the circulation and development of conspiracies surrounding the origin of the virus, the efficacy and nature of the new vaccines, and the political motivations behind COVID-related policy decisions. Although these conspiracy theories may be seemingly innocuous, they originate significant public unrest surrounding the vaccine, masking policies, the origins of COVID, and more.

We observe that the anti-mask movement, the anti-vaccine movement, and nationalistic narratives that suggest that the virus was engineered are key products of such conspiracy theories. Examples of certain ideas originated by these conspiracy theories involve the perception of COVID-19 as an attempt at biological warfare, that the vaccine causes and helps proliferate diseases such as autism, and that the pandemic was part of an elaborate business strategy for pharmaceutical firms to generate abnormal amounts of revenue.

Several accounts such as Crist, 2021 report that conspiracy theories severely impact the well-being of those that believe and spread them. Further, they pose emotional and physical risks for others around them - Basit, 2021 suggests that ideologically diverse extremist groups who are especially reactive to these conspiracy theories, pose the threat of violent terrorism in the public domain. On a global scale, these theories have escalated political tensions, promoted anti-masking and anti-vaccination, and have acted against the dissemination of reliable information about the pandemic and the preventative measures that are relevant, according to Henriquez, 2021.

We are motivated by these problems to investigate the bidirectional relationship between COVID-19 case data and conspiracies surrounding the pandemic. Specifically, we would like to examine when these conspiracy theories gained prominence relative to the spread of the virus, how they are related in content to the pandemic, and how emotionally charged they are, and the popularity of certain topics over time.

2 Dataset Descriptions

We will be using two datasets for this dataset.

1. **COVID-19 Case Dataset** We used COVID-19 case data provided by Our World In Data "Data on COVID-19 (coronavirus) - Our World in Data", 2021. Out of the many options given, we used the JSON file representing daily case data segregated by country (with a worldwide aggregate). We only used the worldwide aggregate of the daily case data. Other country-relevant statistics such as median age, population density, etc. was present, but ignored by our data aggregation function.

   We used the following fields for each entry in the dataset that represented one day's worth of cases:

   (a) The field `new_cases`, an integer representing the number of new cases in the day

   (b) The field `date`, a string formatted in `''YYYY-MM-DD''` representing the date corresponding to the dataset entry

2. **Reddit Post Dataset From r/conspiracies** We are going to be investigating conspiracy theories surrounding COVID-19 using post data from the subreddit "r/conspiracy - Reddit", n.d., a page on the popular message board/forum website Reddit. Being an incredibly active page with 1.6 million members, each post contains a

title, a score representing net "upvotes" versus "downvotes", comments, and a text body. We will only consider text-based posts for this project. We will be extracting data from the subreddit using our own web scraping script.

The data output of the script is a JSON file containing data that corresponds to a variety of attributes. The API used by the script is PushShift Reddit (API PushShift, 2019), which provides a large array of attributes for each post. Out of the several, we are only considering each post's title, number of comments, score, and the created time in seconds since the Unix epoch (detailed in "What is Unix Epoch?", n.d.).

3. **1000 Most Commonly Used Words** This (deekayen, n.d.) is a dataset of the 1000 most commonly used words in English. We require this dataset to filter out common words while performing frequency analysis on text in the Reddit posts when we are determining good keywords to search for.

3 Computational Overview

3.1 Major Computations

Data Collection Since the COVID-19 case dataset is provided completely by Our World In Data, there is no additional dataset construction necessary for this portion of the data collection. A JSON file containing all the information we required was downloaded and stored in a subfolder `/data`.

The 1000 most commonly used words dataset is simply downloaded and renamed in order for our methods to later load the file into memory.

For the dataset of Reddit posts, we designed a method (provided in the file `collect_reddit_data.py`) which was responsible for querying the PushShift API described in Section 2. After querying this API every second for the post data in a 5-day window (while filtering the query for the specific fields we are interested in), the method extended an accumulator list of all Reddit posts. Then, this list was written in JSON format to the file `pushshift-reddit-extracted.json` in the `/data` subfolder.

Data Aggregation All data aggregation functions are enclosed in the file `data_aggregation.py` in the parent directory. We constructed the dataclass `RedditObject` to store all relevant attributes of a Reddit Post (described in Section 2) for ease of access, and wrote functions to load the incoming Reddit data (stored after running the previous algorithm in a JSON file), clean the post title for uniformity in sentiment analysis, and load the COVID-19 data from the JSON file (only loading the worldwide cases into memory).

In order to determine a good list of keywords and topics, we developed `freq_analysis.py`. This file loads all Reddit posts, computes the number of times each word appears in the text, and then filters out the 1000 most commonly used words. Running the file prints the 100 most frequently used words on the subreddit (discounting the common English words that we filtered out). This part is essential for the data aggregation process - one of the main components in the data aggregation is assembling keywords and topics, all of which are detailed in the file `constants.py`.

Computational methods Our main computational methods are found in `computations.py` and are enumerated as follows:

1. We calculate the popularity of a post in `calculate_popularity(post: RedditObject) -> int` by adding the post's score (net upvotes vs downvotes) to its number of comments.

2. We calculate the degree of relation of a topic to a post in the function `calculate_relation`. We scan for keywords in the post title and body, and return 1.0 if we find that one of the topic's keywords matches, and 0.0 if not. The motivation for returning a float is to allow for variability in the degree of relation between topics and posts in future iterations of the project.

3. Arguably our **most important method**, the function `calculate_sentiment_from_text` calculates the negative/positive of a Reddit post and returns a value between -0.5 and 0.5 representing this quantity.

   Sentiment detection is done using a default sentiment classifier from the Python library flair Akbik et al., 2019, a library which describes itself as "An easy-to-use framework for state-of-the-art NLP".

   Note that the confidence value given by the classifier represents the degree of confidence of a post being either negative or positive, and is bounded above by 1 and below by 0.5 - a lower confidence value would imply that the classifier is more confident in the other option, contradicting our assumption that the classifier reports the

class in which it is most confident. So, we subtract 0.5 from this confidence value for continuity between our reported valences. We also give this value a sign depending on whether the sentiment is negative or positive. Therefore, our valence levels range from -0.5 to 0.5.

The classifier is only loaded when it needs to be called; we use the Python library Joblib (Joblib Development Team, 2020) to cache our function results for inputs that we have encountered before, and we save this cache to the disk in the folder `'/cachedir'`. This is both for ease of development and for running the main sentiment analysis method without facing the large, repeated computational cost of recomputing the sentiments associated with the same text inputs.

NOTE: Despite PythonTA disallowing the use of nonconstant global variables, we found that it is required for us to have two nonconstant global variables representing the Memory and TextClassifier objects respectively - any alternative implementation (that does not include major refactoring) includes loading the TextClassifier each time the function is called (and not caching our results) which would result in our running time increasing by a massive factor.

4. We have the functions `posts_in_interval` and `new_cases_in_interval` that filter the lists of posts and cases respectively to only contain those which are in the time interval specified. A similar function filters posts by topic: unsurprisingly, the function is named `filter_posts_by_topic`.

5. We calculate the total valence of posts in an interval in the function `calculate_total_valence` by summing the valences of all posts in the interval.

6. The function `get_time_array` returns a one dimensional list containing timesteps between the beginning `start` and `end` timestamps, each spaced by the `resolution`, which is a `datetime.timedelta` object.

3.2 Computation and reporting of results

We compute our results through various functions given in `main.py`. They are detailed as follows:

1. The function `run_popularity_vs_negatively_charged` loads the Reddit posts, calculates popularities and sentiments for each post, and plots them on a scatter plot by calling the corresponding plotting function. We colour each post on the scatter plot according to its valence (a solid red represents completely negative, a solid green represents completely positive).

2. The function `run_frequency_over_time` calculates the number of posts in intervals spaced a week apart each, starting 1st December 2019 and ending at the current date, and plots the number of posts per week over time by calling the corresponding plotting function. It also loads the COVID-19 case data, counts the number of new cases within the timeframe of a week at the relevant points in time, and plots this case frequency data alongside the post frequency over time.

3. The function `run_valence_over_time` calculates the total valence of posts in intervals spaced a week apart each, starting 1st December 2019 and ending at the current date, and plots the total valence of posts in the week over time by calling the corresponding plotting function. It also loads the COVID-19 case data, counts the number of new cases within the timeframe of a week at the relevant points in time, and plots this case frequency data alongside the valence over time.

4. The function `overall_valence_histogram` plots the valence data of all posts on a histogram by calling the corresponding plotting function. The motivation of this function is to observe the distribution of the post valences over the interval from -0.5 to 0.5.

5. The function `run_topics_vs_time` takes in a list of topics to track. Similar to the approach of the previous functions, it looks at posts within a timeframe of a week at different points in time, and calculates the aggregate relation of the relevant posts to a topic by summing outputs of the function `calculate_relation` over the relevant posts. It also loads the COVID-19 case data, counts the number of new cases within the timeframe of a week at the relevant points in time, and plots this case frequency data alongside the total topic relation over time. The plotting is done by calling the corresponding plotting function.

6. The functions `most_negative_posts` and `most_positive_posts` serve to print the 10 posts with the most extreme valence for our analysis.

3.3 New libraries used

The most important new libraries that the project uses are as follows:

1. `flair`

   This library (Akbik et al., 2019) powered our sentiment analysis; we were able to load a pre-existing text classifier using

   `TextClassifier.load('en-sentiment')`, and then use this classifier to predict whether post titles were positive or negative. We also had to use the `Sentence` class to initialize a sentence containing our post title, on which we ran the `classifier.predict` method to obtain valence scores. More details on how these scores are used and transformed is given in Section 3.1.

2. `joblib`

   We used this library (Joblib Development Team, 2020) to cache the function results of `calculate_sentiment_from_text` in order for ease of development and for running the main sentiment analysis method without facing the large, repeated computational cost of recomputing the sentiments associated with the same text inputs, as mentioned above.

   We used the decorator `@memory.cache` above the function to indicate that we would like to cache the outputs of this function to disk, and specified the caching location while initializing an instance of the `Memory` object in the line `memory = Memory("cachedir")`.

3. `requests`

   This library ("Requests: HTTP for Humans™ — Requests 2.26.0 documentation", n.d.) was used in order to query the PushShift API given the r/conspiracies subreddit and the appropriate timeframe.

4. `json`

   We used this library to load json files. The function `json.load` was used for this purpose.

5. `matplotlib.pyplot`

   We used this library to do all of our plotting. The most critical functions used were `plt.plot` and `plt.scatter`.

4 Instructions

The entire project repository is found at https://github.com/nitvishn/csc110-project, if needed.

4.1 Obtaining datasets

To ensure that the datasets used by our TA are *exactly* the ones we used, we will provide links to a GitHub repository on which our project is hosted. We will be submitting the file used to generate the extracted Reddit dataset, but we do not require the TA to execute this file to query the API and generate the dataset themselves.

1. The COVID-19 dataset may be obtained by downloading this JSON file (hosted on our GitHub repository). It must be placed in a subfolder named `/data`.
2. The Reddit post dataset may be obtained by downloading this JSON file (hosted on our GitHub repository). It must be placed in a subfolder named `/data`.
3. The commonly used words dataset may be obtained by downloading this TXT file hosted on our GitHub repository. It must be placed in a subfolder named `/data`.

4.2 Running the program

It is important to create the empty directory `/figures` before running the program in order for the program to store all its visualizations correctly.

After installing all requirements given in `requirements.txt`, and after ensuring that the files `/data/pushshift-reddit-extracted.json`, `data/owid-covid-data.json`, and `data/commonly_used_words.json` are present, you may run `main.py`.

Expect the download of the flair text classification model for sentiments to take 2-5 minutes - once this file is downloaded, the model will be executed on all Reddit posts. On first run, logs will be printed by the `joblib` module indicating that the function outputs are being cached for `calculate_sentiment_from_text`.

The **main output** from our program is the set of images that represent the various figures that we plot our final results on. These figures will be present in the `/figures` directory after `main.py` terminates.

5 Changes from proposal

There were a few changes to the computational plan to the project. We have documented them below.

1. The OSRSBox Reddit scraper did not obtain the date posted for each post, so we swapped it out for our own Reddit scraping script. This script is detailed in Section 3.1.
2. Our choices of keywords was not sufficient to categorize the majority of Reddit posts, so we instead performed a frequency analysis to determine our list of keywords.
3. We initially assumed that the relationship between post sentiments and popularity would be modelled by a polynomial relationship - it is not. Instead, it is to our benefit to plot these on a scatter plot and intuitively analyze the behaviour of posts at the extremes. More information is present below in Section 6.
4. The VADER sentiment algorithm did not seem to effectively determine the sentiment of many Reddit posts - a lot of seemingly aggressive posts were classified as neutral. Instead, we used the `flair` library for sentiment analysis. More details are present in Section 3.1.
5. We added a few more visualizations to the original list given in the proposal: a histogram of the valence of the Reddit posts, and a graph representing the development of total post valence per week over time.

6 Discussion

6.1 Discussion of Results

We will analyze each of the figures individually, and then begin to formulate hypotheses that address our research question below.

1. Case frequency against time

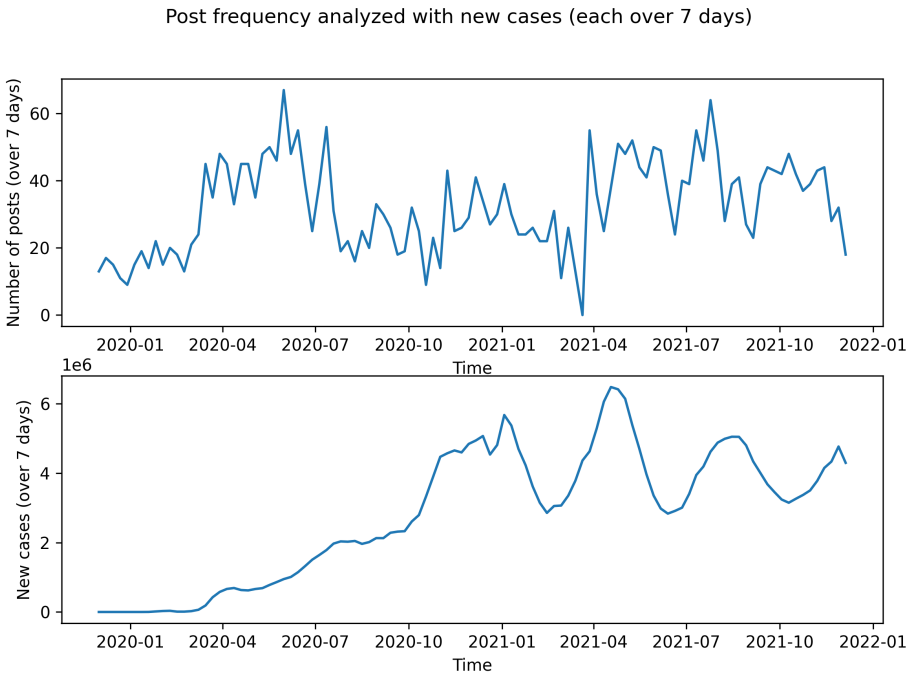

Figure 1: `figures/frequency_over_time.png`

We observe a chaotically oscillating post frequency-time curve , noticing that the points at which case frequencies are highest coincide with significant increases in post frequencies. However, this relationship is best characterized by looking specifically at topic-related posts, which is done below.

2. Popularity of the topics 'covid' and 'vaccine' over time

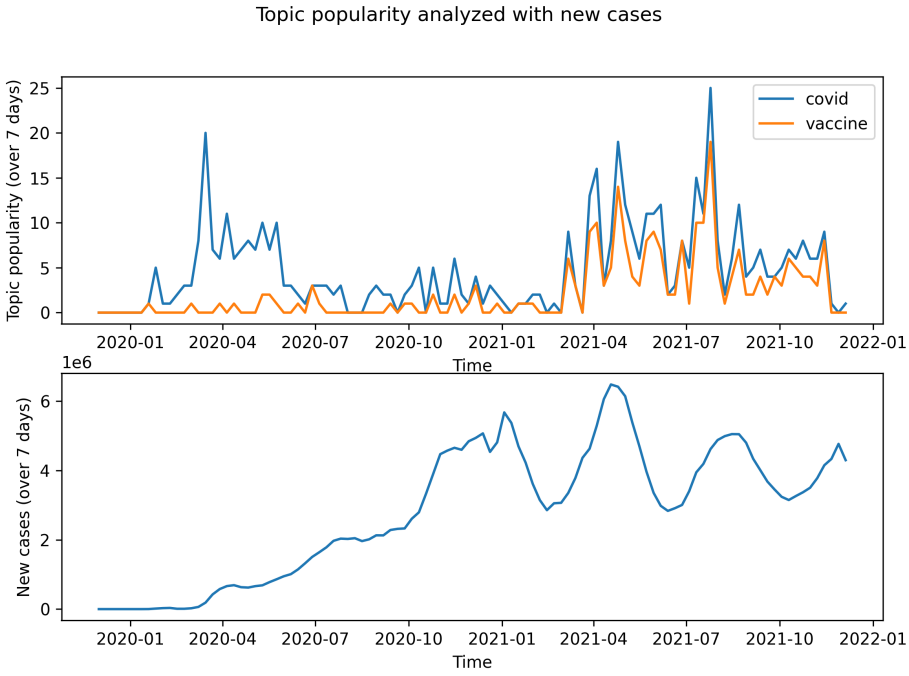

Figure 2: `figures/popularities-covid-vaccine.png`

This figure makes it abundantly clear that the upticks in post frequencies around the times at which COVID-19

case frequencies were highest is attributable to COVID-related posts. It is also observed that COVID-19 is a very popular topic during the initial stages of the pandemic, and also that the nonexistence of a vaccine at that point of time yielded low popularities in the topic 'vaccine'.

3. Popularity of other topics over time

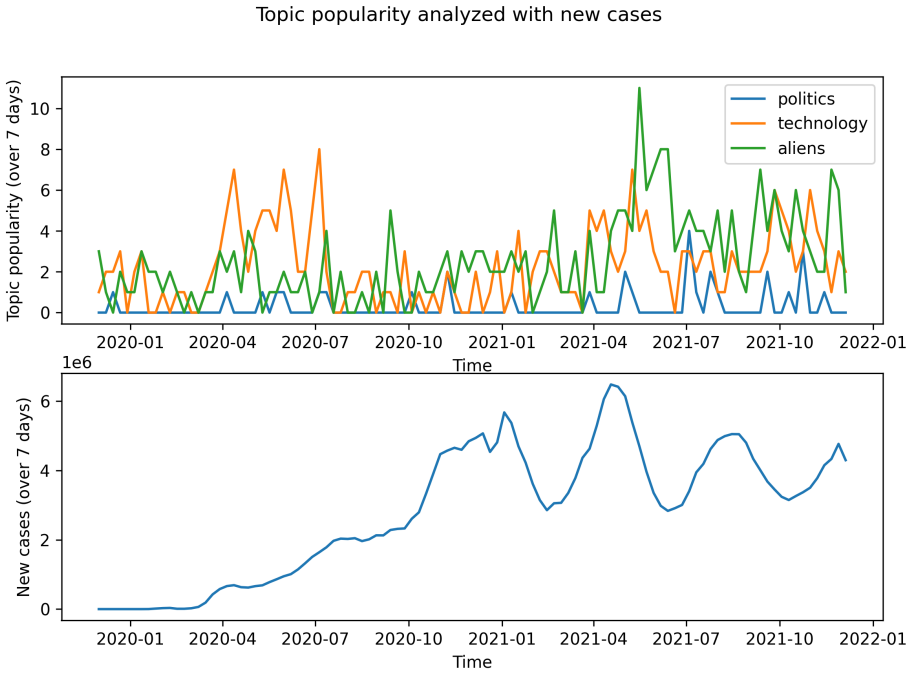

Figure 3: `figures/popularities-politics-technology-aliens.png`

4. Total valence per week over time

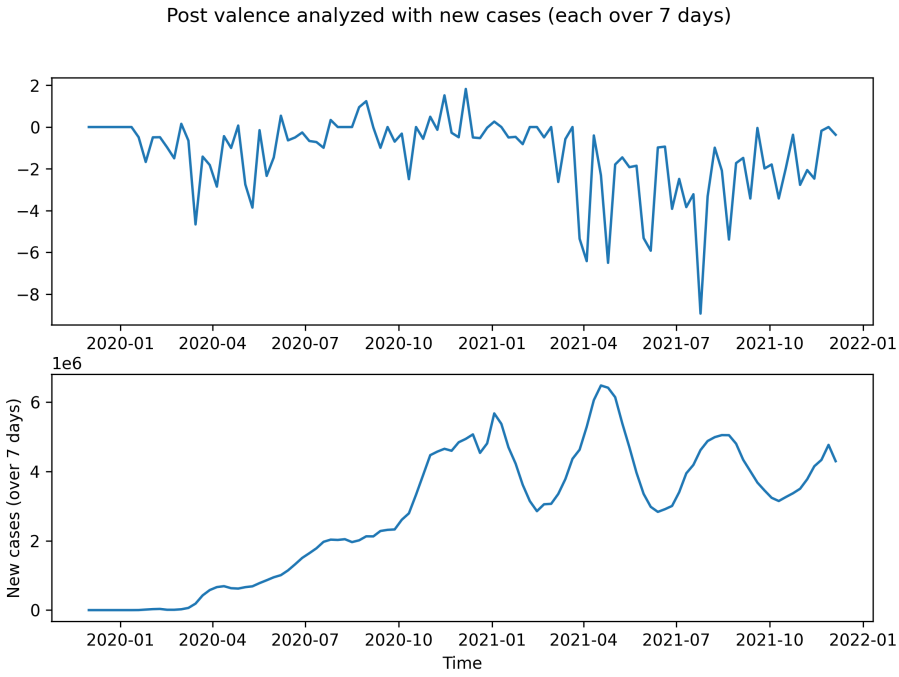

Figure 4: `figures/valence_over_time.png`

One of the most significant indicators is the large drop in total post valence around the time of the third rise in case frequencies around July 2021. This fall in net valence shows that in addition to the number of posts being high, the vast majority of them were negatively charged.

5. Distribution of valences

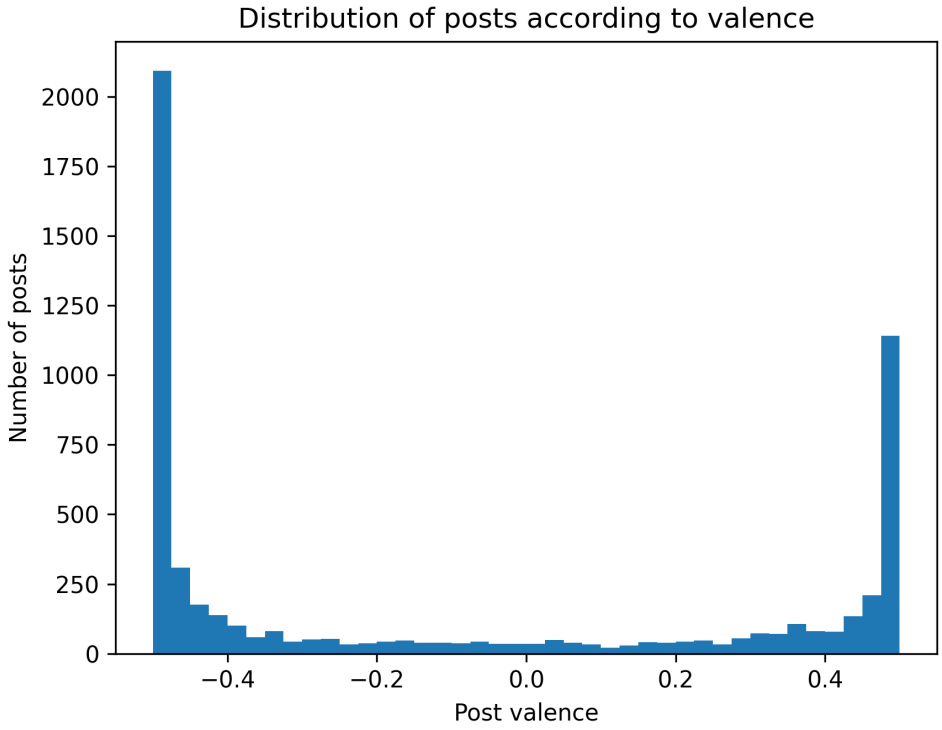

Figure 5: `figures/valence_histogram.png`

This figure allows us to build upon the inferences we made from the previous one; indeed, we see that most posts are negative. More than anything else, this figure suggests that the majority of posts lie on the extreme ends of the negativity/positivity spectrum.

6. Popularity of posts vs valences

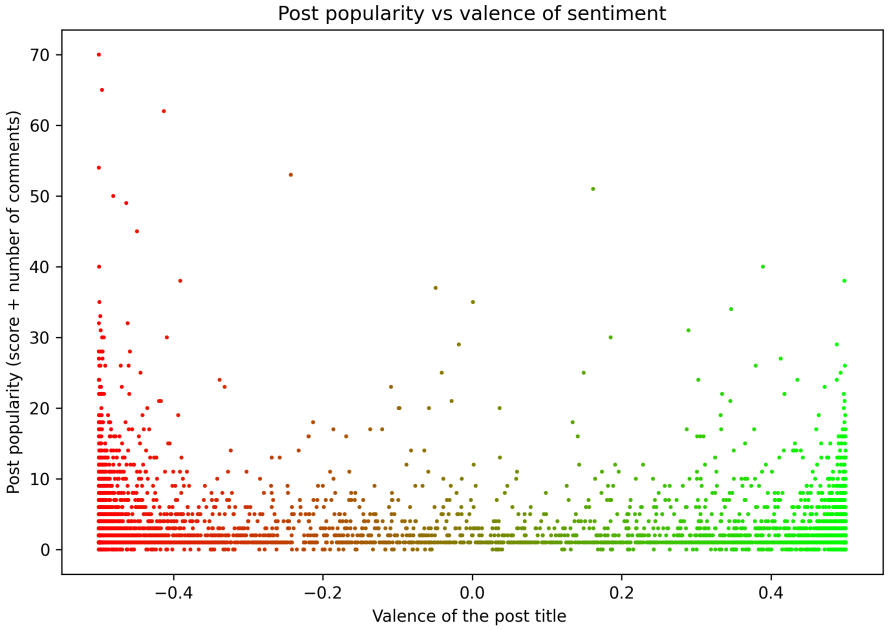

Figure 6: `figures/sentiments_popularity.png`

Here, we observe that posts on the extremes of negativity/positivity tend to have the greatest popularity, with the extremely negative posts being more popular and frequent in popularity than the extremely positive posts.

Returning to our research question, we may now begin to develop plausible explanations for the trends observed in the above results.

6.2 Addressing the Research Question

Our research question was **What is the nature of the circulation and development of conspiracy theories surrounding COVID-19?**

From the above discussion, we can conclude the following:

- COVID-19 conspiracy theories are discussed with the greatest frequency during the beginning of the pandemic in early 2020, and the two major waves in April and August 2021. This early spike can likely be attributed to the novelty of the coronavirus subject.
- After March 2021, the time around which the vaccine roll-out began to take shape, vaccines are discussed in these theories very similarly in frequency to COVID-19 conspiracy theories.
- Very negative posts represent a large portion of both popular and total posts. With very positive posts making up the second largest category. Less emotionally charged posts are usually less popular and rarer.
- Posts which are either very positive or very negative tend to do well; posts which are very negative have the largest popularity.

The next spike in covid-related conspiracy posts happened around the start of 2021. This happened along with a large amount of vaccine related posts being made. Therefore, it is likely that these posts were made in response to the possibility of a covid vaccine being administered.

6.3 Limitations

Our analysis was limited by the following :

- We only linked topics to posts in a discrete manner when we could have used a more advanced intent detection algorithm to analyze the topic of each post.
- We did not analyze comment data on the posts which were processed - this could greatly aid in assessing the valence of a post and computing an overall compound popularity.
- Our list of keywords was large, but not exhaustive.

6.4 Further work

The following further work is suggested:

- Develop a more sophisticated algorithm that determines a more precise degree (ie. floats between 0 and 1) of relation between topics and posts.
- Incorporate comment data into our analysis.
- Incorporate more subreddits into our analysis.
- Compile a larger list of keywords.

References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 54–59.

Basit, A. (2021). Conspiracy theories and violent extremism: Similarities, differences and the implications. *Counter Terrorist Trends and Analyses*, *13*(3), 1–9. https://www.jstor.org/stable/27040260

Crist, C. (2021). People who believe in covid-19 conspiracies more likely to catch virus: Study [(Accessed on 11/03/2021)].

Data on covid-19 (coronavirus) - our world in data [(Accessed on 11/03/2021)]. (2021).

deekayen. (n.d.). 1,000 most common US English words. Retrieved December 14, 2021, from https://gist.github.com/deekayen/4148741

Henriquez, G. (2021). Covid-19 conspiracy theories deter some from getting vaccine - montreal — globalnews.ca [(Accessed on 11/03/2021)].

Joblib Development Team. (2020). *Joblib: Running python functions as pipeline jobs*. https://joblib.readthedocs.io/

PushShift. (2019). Pushshift/api: Pushshift reddit api documentation [(Accessed on 11/03/2021)].

R/conspiracy - reddit [(Accessed on 11/03/2021)]. (n.d.).

Requests: HTTP for Humans™ — Requests 2.26.0 documentation. (n.d.). Retrieved December 14, 2021, from https://docs.python-requests.org/en/latest/

What is unix epoch? [(Accessed on 11/03/2021)]. (n.d.).