Spring 6-1-2022

Towards a Computational Model of Narrative on Social Media

Anne Bailey
Anne.G.Bailey.22@Dartmouth.edu

TOWARDS A COMPUTATIONAL MODEL OF NARRATIVE ON SOCIAL MEDIA

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Bachelor of Arts

in

Computer Science

by

Anne Bailey

Advised by Professor Soroush Vosoughi

DARTMOUTH COLLEGE

Hanover, New Hampshire

June 1, 2022

Abstract

This thesis describes a variety of approaches to developing a computational model of narrative on social media. Our goal is to use such a narrative model to identify efforts to manipulate public opinion on social media platforms like Twitter. We present a model in which narratives in a collection of tweets are represented as a graph. Elements from each tweet that are relevant to potential narratives are made into nodes in the graph; for this thesis, we populate graph nodes with tweets' authors, hashtags, named entities (people, locations, organizations, etc.,), and moral foundations (central moral values framing the discussion). Two nodes are connected with an edge if the narrative elements they represent appear together in one or more tweets, with the edge weight corresponding to the number of tweets in which these elements coincide. We then explore multiple possible deep learning and graph analysis methods for identifying narratives in a collection of tweets, including clustering of language embeddings, topic modeling, community detection and random walks on our narrative graph, training a graph neural network to identify narratives in the graph, and training a graph embedding model to generate vector embeddings of graph nodes. While much work still remains to be done in this area, several of our techniques, especially the generation and clustering of graph embeddings, were able to identify groups of related and connected nodes that might form the beginnings of narratives. Further study of these or other techniques could allow for the reliable identification of full narratives and information operations on social media.

Acknowledgments

This thesis would not have been possible without the guidance and support of a number of people. First and foremost, I want to thank Professor Soroush Vosoughi, who introduced me to machine learning in his course last spring and who advised this thesis. Despite being incredibly busy, he took me on as an advisee, introduced me to a topic that would allow me to combine my interests in computer science and linguistics, guided me through my research, and even took the time to teach me deep learning as an independent study this winter. I am deeply grateful for his mentorship and kindness.

I would like to thank Professor Tim Pierson and Professor Tim Tregubov for their time and consideration in serving on my thesis committee, as well as for being wonderful mentors who shaped my Dartmouth experience through classes and software development work at the DALI lab. I am grateful to the entire Computer Science Department and Linguistics Department for introducing me to subject areas that excited my interest and giving me the opportunity to learn from dedicated and brilliant professors over the past four years.

Finally, I want to thank my friends and family for their love and support as I was working on this thesis and throughout my time at Dartmouth. I couldn't have done this without them.

Contents

Chapter 1

Introduction

Section 1.1

Motivation

As stated in the aims of the Seventh International Workshop on Computational Models of Narrative, "Narrative provides a model for organizing and communicating experience, knowledge, and culture" [14]. Narratives are central to how humans process events and share knowledge, and they can reveal prevalent attitudes, perceptions, and connections among people and ideas. A computational approach to the study of narrative can provide ways to identify societal patterns of thought and behavior that cannot always be seen through everyday interactions or through the reading of individual stories or ideas. One example of the importance of narrative analysis can be seen in the growing influence of information operations on social media.

The rise of social media has changed the way people receive news and form opinions. As media platforms like Twitter, Facebook, Reddit, and others become more widespread, more people turn to these platforms to read about and discuss current issues. These platforms enhance the widespread sharing of ideas, but they also make it easier for governments and organizations to spread false or distorted narratives.

Recent online information operations, such as attempts to influence elections or perpetuate misinformation about the COVID-19 pandemic, have motivated numerous efforts to study, identify, and mitigate such campaigns [13]. Many of these efforts, however, involve manual identification of misleading or harmful posts and accounts [7, 13, 21], a process that cannot hope to keep up with the large volume of influence efforts being produced around the world. A recent DARPA announcement requesting research in this field makes the problem clear: "adversarial information operations (IOs) have become a defining feature of the modern-day information environment, but defensive capabilities ... remain thin" [4]. The ability to automatically detect online information operations, then, would be a great help not only in the defense space, but also for any social media platforms trying to limit the spread of misinformation on their sites. As the DARPA announcement goes on to state, "the concept of narrative is central to IOs" [4], but the process of computationally modeling, analyzing, and identifying narratives is still relatively new and poorly understood. These issues motivate the development of a computational model of a narrative, specifically a narrative on social media, and eventually, a natural language processing system that can use this model to automatically detect information operations online.

Section 1.2

What is a Narrative?

Before we explore the development of a narrative model, it is important to consider definitions from multiple fields of what constitutes a narrative. Various literary, linguistic, and technical works have all created different definitions of narrative, but all these definitions center around the same concepts. Narratives are composed of events, actors (agents that cause or experience events), time, and locations [1, 2, 17, 22]. An event has been defined by different sources as an occurrence at a particular place and

time [2], an action (or verb) and the participants that that action describes [22], and a transition from one state to another state [1]. This "series of logically and chronologically related events" must also have "semantic coherence" and a "uniqueness of theme" that distinguishes the events of one narrative from the events of another [22]. Finally, all sources agree that a narrative is distinguished from a simple chronological relation of events by the order and manner in which the events are related, and by the point of view from which they are told. Any series of events can be narrated in multiple ways, and any given telling will portray the events in a different (and subjective) way. Some goals for our narrative model, then, might be identifying some form of events, actors, and the points of view from which these events are framed.

Section 1.3

Problem Statement

The goal of this thesis is to create a computational model of narratives found on social media. Specifically, we will focus on narratives created on Twitter. However, a "computational model of narrative" can mean many different things: previous studies in this field have used a wide variety of techniques, ontologies, and logic-based languages to represent narratives in a computer-suitable form [14]. Our more specific goal, then, is to identify relevant features of tweets that allow us to train a machine learning model to automatically group large numbers of tweets into distinct narratives. This thesis explores possible relevant features and possible methods of training such a model.

Chapter 2

Related Work

There have been many efforts to create narrative classifiers or ontologies in various different contexts, from literature to social media [14]. Rather than providing a systematic literature review, we will outline some motivating works and key background research that are most relevant to this thesis.

Section 2.1
Online Information Operations

Exploring Online Influence Efforts. Two papers by Diego Martin and Jacob Shapiro, "Recent Trends in Online Foreign Influence Efforts" [13] and "Trends in Online Influence Efforts" [12], provide an overview of recent foreign and domestic information campaigns on social media. Martin et al. define foreign influence efforts as "coordinated campaigns by one state to impact one or more specific aspects of politics in another state" [13], and define domestic influence efforts similarly as "coordinated campaigns by a state, or the ruling party in an autocracy, to impact one or more specific aspects of politics at home or in another state" [12]. These papers used media reports to identify and create a database of influence efforts, which they defined by the attacker, target, and political goal involved. Finally, they analyzed

these campaigns to highlight common influence strategies, such as discrediting, undermining, or supporting other political entities, or polarizing a state's politics. It is these types of online information operations, as well as the attackers, targets, and political goals defining them, that we would like our narrative model to help identify.

Identification of Misinformation in COVID-19 Tweets Using BERTweet. Though it addresses a slightly different goal than ours, a recent study on identifying misinformation in individual tweets provides helpful insight on techniques we could use to identify influential narratives on Twitter [20]. The authors of this paper used a dataset of English tweets labeled with yes/no answers to certain questions related to misinformation, such as whether the tweet had a verifiable claim, whether it contained false information, or whether it would be of interest to the public. They then fine-tuned BERTweet, a Twitter-specific language representation model (described in more detail in Chapter 3), for the task of predicting the answers to these questions. This BERTweet based model performed these predictions relatively well on new test tweets. This study demonstrates the effectiveness of identifying features relevant to the spread of information on Twitter and fine-tuning BERTweet for a specific Twitter classification task, both of which we will use in developing our narrative model.

Automatic Detection of Influential Actors in Disinformation Networks. Another recent study, very similar to the work in this thesis, proposes a system to "automate detection of disinformation narratives, networks, and influential actors" on Twitter [19]. Their framework identifies potential influence operation narratives, uses features of user behavior to train a model to identify influence operation accounts, constructs a "narrative network" of accounts propagating a certain narrative, and estimates each account's influence in spreading that narrative. This system's account classifier and account impact estimation go beyond the model we are trying to de-

velop. Its narrative identification methods and narrative network, however, suggest both potential areas for success and areas for improvement for our narrative model. This study's narrative detection is limited to topic modeling, a pre-existing method of unsupervised classification that generates "topics" associated with clusters of words in a text corpus. The authors collect tweets related to certain subjects of interest, identify accounts posting this relevant content, pass tweets from those accounts to a topic modeling algorithm, then manually select interesting "narratives" from the topics that algorithm generates. Our goal is to develop a way of modeling and detecting narratives that goes beyond probabilistic topics. Furthermore, this study's narrative network – a network of Twitter accounts connected by retweets – suggests a way to model narratives by including not just users but other elements of their tweets in the network. Our final narrative model will center around such a network.

Section 2.2

Moral Foundations

Our final narrative model will involve extracting and connecting various elements from tweets, such as their authors, hashtags, and named entities. One of these elements, however - a tweet's moral foundation - requires further background to explain.

Moral Foundations Theory. Moral foundations theory [8] is the idea that all cultures share a set of "intuitive ethics", and that their rules and ideas of morality, different as they are, are built in different ways on top of these foundations. The five commonly accepted moral foundations are:

(a) Care/harm: values related to our ability to feel, dislike, cause, and prevent pain in others

(b) Fairness/cheating: values related to reciprocal altruism, justice, and rights

(c) Loyalty/betrayal: values related to belonging in and acting on behalf of groups or coalitions

(d) Authority/subversion: values related to leadership and hierarchical social interactions

(e) Sanctity/degradation: values related to religious ideals and the psychology of disgust or contamination

Short statements such as tweets will often, although not always, center around one primary moral foundation. Part of our narrative model will involve classifying the moral foundation of each tweet and connecting it with other elements in the tweet. It is worthwhile, then, to take a brief look at other works that identify moral foundations in tweets.

Identifying Morality Frames in Political Tweets using Relational Learning. One recent study developed a structured framework for modeling and predicting moral foundations in tweets [16]. The authors introduce the concept of "morality frames", each of which includes a moral foundation and certain "typed roles"; for example, the typed roles of the care/harm moral frame are the "entity providing care," "entity needing the care," and "entity causing harm." These frames serve as a more detailed way of representing moral positions in tweets. The paper then proposes a statistical relational learning model, "modeling the dependency between [moral foundations] and moral roles", to predict moral frames in text.

Moral Foundations Twitter Corpus: A Collection of 35k Tweets Annotated for Moral Sentiment. This recent paper provided important labeled data for our narrative model. One of the paper's main contributions is a collection of about 35,000 tweets labeled for their moral sentiment, or moral foundation [9]. The tweets

were drawn from several prevalent social discourses – Black Lives Matter, All Lives Matter, the 2015 Baltimore protests, the 2016 presidential election, hate speech, Hurricane Sandy, and #MeToo – and labeled by trained human annotators. The authors then used this labeled data to train and compare the performance of several different machine learning models to predict moral sentiment, with several different models performing best in different contexts. Most relevant to this thesis, we used their labeled data collection to train our own classifier to identify the moral foundations of tweets.

Section 2.3

Problem Formulation

In summary, many different studies have looked at many different aspects of the problem of identifying narratives on social media. Some works have developed computer-suitable ontologies of narratives but stopped short of automatically identifying them. Others have manually identified and analyzed online information operations. Still others have automatically identified individual aspects of tweets, such as topics, actors, and misinformation, that are relevant to social media narratives. This thesis seeks to combine and build on ideas from all of this work to develop a more holistic model of narratives, specifically online information operations. Our goal is to use a variety of relevant features, such as topics, entities, moral foundations, and more to identify such narratives.

Chapter 3

Methodology and Experiments

The following sections will describe the various techniques we explored to model and identify narratives on Twitter. We will detail which approaches worked, which did not, and which led to other potential areas of exploration.

Section 3.1

Data

Our main data source for building and testing potential narrative models was the IEEE's Coronavirus Tweets Dataset [10], a collection of English-language tweets related to the COVID-19 pandemic. The dataset contains the Twitter IDs of these tweets, and we used the Twitter API to fetch the text and author of each one. As the Twitter API imposes a rate limit of 900 requests per 15 minutes, we limited our data to a sampling of approximately 1700 tweets for every 10 days between March 2020 and November 2021. In total, we collected 40,521 tweets. Our goal was to identify narratives within discussions related to COVID-19 before expanding to tweets on any topic.

Some of our later experiments involved training a named entity recognition (NER) classifier and a moral foundation classifier for recognizing these elements in tweets.

For training our named entity classifier, we used the WNUT 17 (Workshop on Noisy User-Generated Text 2017) dataset [5]. We kept this dataset's split of train data ($N = 3394$), validation data ($N = 1009$), and test data ($N = 1287$). For training our moral foundation classifier, we used the Moral Foundations Twitter Corpus [9] described in Chapter 2. Each tweet in this dataset is labeled by three to four annotators; for simplicity, we used the first annotator's label. We used an 80%-20% train-test split for this data, resulting in a training set of size $N = 8359$ and a test set of size $N = 2090$.

Section 3.2

Pretrained Language Models

We began our research with the simple approach of exploring whether pretrained language models would be sufficient to represent and find narratives within social media data. Pretrained language models are trained separately on very large sets of data to learn high-dimensional vector representations of words and phrases. These vectors map language onto high-dimensional space, capturing context so that words and phrases with more similar meanings have more similar vector representations. These language models can then be downloaded and used by others to generate such vector embeddings, or sets of features, for their own text data. Pretrained models can also be fine-tuned for a specific task, such as sentiment analysis or named entity recognition: users can slightly adjust the model by training it further on their own, more specialized data to perform that task. Following is a brief explanation of the pretrained language models we used – BERT and BERTweet – and how we used them.

3.2.1. BERT and BERTweet

BERT, which stands for Bidirectional Encoder Representations from Transformers [6], is a state-of-the-art, open source language model. Using a deep learning model known as a transformer, BERT is able to read sentence input in both directions (left to right and right to left) and therefore learn representations of words using their entire surrounding context. The BERT model can then be used to generate vector representations of new text, or fine-tuned to perform a variety of tasks.

BERT was trained using the Books Corpus (a collection of 800M words) and the entirety of English Wikipedia (2,500M words). While we could simply use BERT embeddings for our Twitter data, the words and syntax used on Twitter vary significantly from the words and syntax used in most books and Wikipedia articles. To address this problem, Nguyen et al. [15] created the BERTweet model, a language model built using the same architecture as BERT but trained instead on English tweets. It is this model that we used to generate embeddings for our tweets (section 3.2.2) and that we fine-tuned to create our NER and moral foundations classifiers (sections 3.4.1 and 3.4.2).

3.2.2. Clustering of BERTweet Embeddings

Our first experiment involved simply generating BERTweet embeddings for each tweet in our dataset. We used the HuggingFace Transformers library to load the BERTweet model, tokenize the tweets (truncating those that were too long for the model to handle), and encode each tweet.

We then attempted to cluster these tweet embeddings using scikit-learn's agglomerative clustering algorithm. This algorithm is a form of hierarchical clustering in which each data point (in this case, each tweet embedding) starts as its own cluster, then pairs of clusters are recursively merged based on how similar they are. The resulting clusterings at each step can be represented in a tree diagram known as a

dendrogram (Figure 3.1), where the diagram's height represents the distance between clusters. Our goal was to use the clusterings at different levels of the dendrogram (different steps of the agglomerative clustering process) to find tweets clustered into different groups. For example, we hoped larger clusters might be grouped by a broad agenda and smaller clusters might be grouped by specific topics or subtopics.

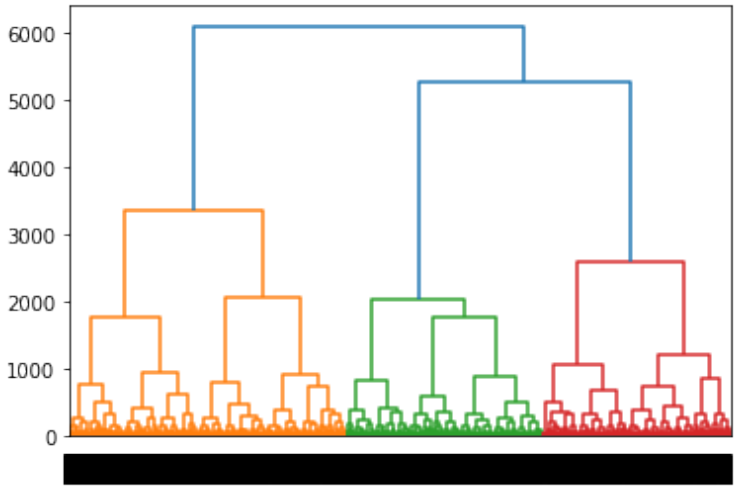


Figure 3.1: Dendrogram of clusterings of our COVID-19 tweet embeddings.

We tried creating different numbers of clusters, running the algorithm with n_clusters = 3, 5, 10, 15, 50, 100, 500, and 1000. We calculated the silhouette score, a method of evaluating how well data is clustered, for each number of clusters. None of the scores were very high, ranging from 0.07 for 3 clusters to 0.41 for 1000 clusters.

We used dimensionality reduction techniques to visualize the tweet embeddings, in order to better understand how the tweets were related and why our clusterings were not very effective. PCA (principal component analysis) and tSNE (t-distributed stochastic neighbor embedding) are both dimensionality reduction techniques, or methods of mapping high-dimensional data (data with a large set of features) into lower-dimensional space while still preserving most of the information in the original data. PCA better preserves the global structure and variance of the entire dataset, while tSNE better preserves local structure, such that neighbors in the original high-

dimensional space are also close together in the reduced dimensional space. PCA performs better on large datasets with large numbers of features, so we first used this technique to reduce the dimensions of our tweet embeddings from 768 to 50. Since tSNE better preserves local clusters in the data, we then used this method to embed the 50-dimensional embeddings in 2 dimensions. The figures below show these tSNE embeddings plotted on a scatter plot and colored by cluster for 10 clusters.

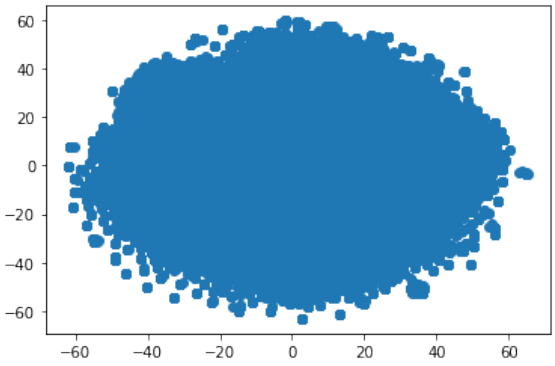


Figure 3.2: tSNE plot of BERTweet embeddings.

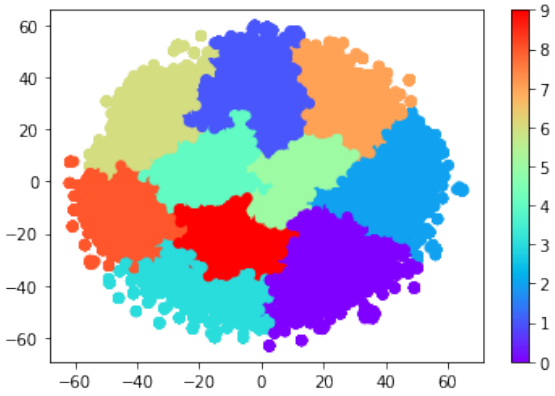


Figure 3.3: Clustered BERTweet embeddings.

As these figures show, our tweet embeddings cannot be clearly separated into defined clusters. This experiment showed us that we would need to look beyond simple language embeddings in order to group tweets into topics, agendas, or narratives.

Section 3.3

Topic Modeling

After our simple tweet clusterings displayed no clear separation by subject, we decided to try topic modeling on our collection of tweets. We hoped that grouping tweets by topic might be a step towards identifying narratives within these groups. We used Gensim's LDA model for this task.

LDA, or Latent Dirichlet Allocation, is a method of discovering abstract topics in a collection of documents. It uses a probabilistic algorithm to detect patterns such as word frequency and distance between words, group similar word patterns, and infer topics in unstructured text data. It ultimately produces a list of unlabeled topics, where each topic is defined by words with different weights (some words are more representative of the topic than others), and each word has a certain probability of belonging to each topic.

We first converted our data into a format that could be passed to a topic model. We created a bag of words (unordered collection of words) from our tweets, preprocessing this collection to remove stopwords (common words without much semantic meaning), remove punctuation, and lemmatize, or standardize words with the same root. We also added bigrams, or consecutive pairs of words, since they often convey meaning that individual words cannot. We passed this collection of words and bigrams to Gensim's LDA model, which gave us a list of topics (labeled simply as integers), along with the top-weighted words belonging to each topic. We tried varying the hyperparameters of the model, such as the number of topics to create and the number of passes to take through the dataset during training, but none of these variations produced significantly different results. We calculated the coherence of the model, which measures semantic similarity between high-weighted words in each

topic, after training the model various times to output different numbers of topics. All of the coherence scores were rather low; the score for a model generating 10 topics, for example, was 0.39. We also used pyLDAvis, a library for interactive topic model visualization, to visualize the topics and words associated with them.

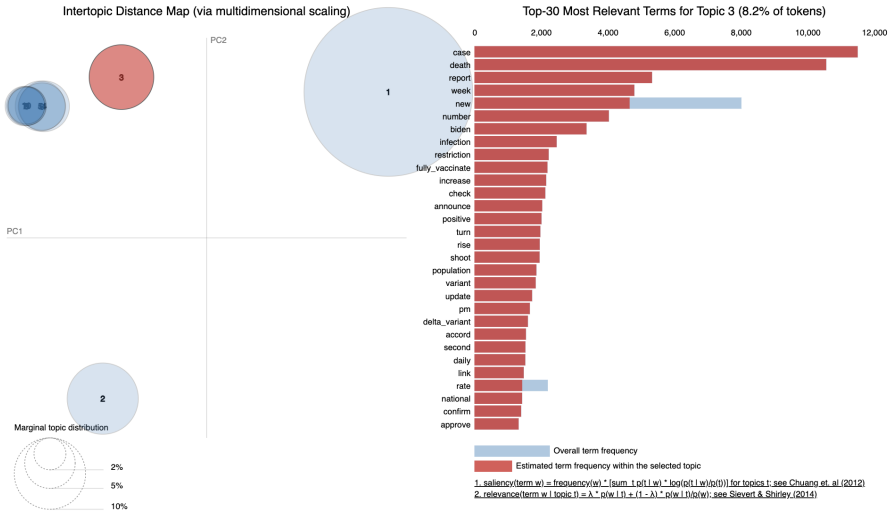


Figure 3.4: Visualization of topic model for full text of COVID-19 tweets.

The visualization showed that most topics (topics 4-10) were very overlapping, rather than distinct as we had hoped. Even between topics 1-3, which appear more separated in the diagram, there was little clear conceptual separation. Topic 3 included top-weighted words such as 'case', 'death', 'report', and 'infection', topic 2 featured words such as 'hospital', 'patient', 'fight', and 'doctor', and topic 1 seemed to focus on the pandemic overall with words like 'covid', 'pandemic', and 'lockdown'. While all these topics addressed slightly different aspects of the pandemic, they were not distinct enough that we could use them as starting points for identifying different narratives.

We then tried topic modeling on just the tweets' hashtags, in case the full text of the tweets was simply too noisy. We repeated the same process as before; the collection of words we passed to the LDA model was just limited to hashtags extracted

from each tweet. The coherence score of this model, 0.58, was better than before, but visualizing the model showed similar results.

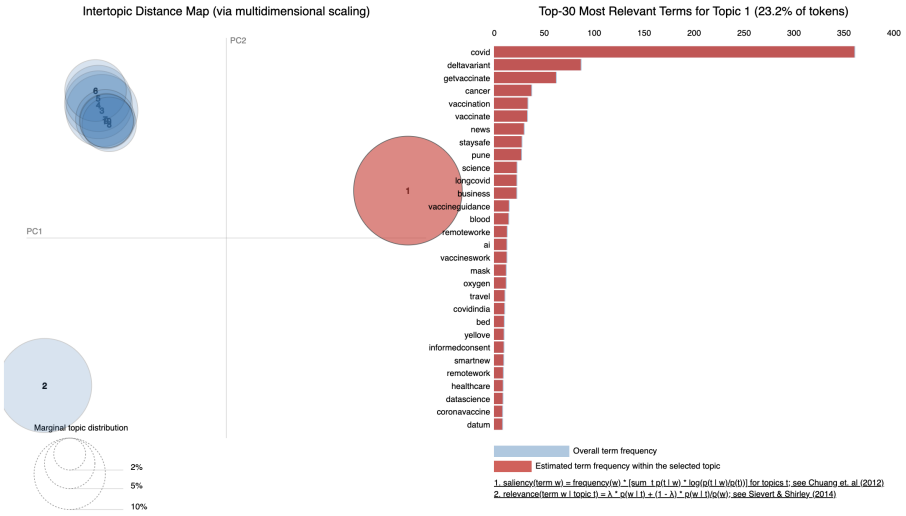


Figure 3.5: Visualization of topic model for hashtags of COVID-19 tweets.

As before, most topics overlapped, and the top-weighted words of topics 1 and 2 addressed only marginally different aspects of COVID-19.

Given that neither pretrained language models nor topic modeling could separate tweets into defined subjects, we decided that we would need a more complex way to represent tweets and the narratives they create.

Section 3.4

Narrative Graph

As discussed in Chapter 1, many definitions of narrative involve key elements like events, actors, places, and points of view. One large reason that our attempts at clustering tweet embeddings and topic modeling did not produce helpful results may be that they did not address such key elements. Our embeddings used the entire text of each tweet, and topic modeling used either the entire text or just the hashtags.

To improve upon these methods, then, we decided to create a narrative graph, in which important elements of tweets would be represented as nodes and elements that appeared in the same tweets would be connected by weighted edges. Our initial version of this graph would include named entities (people, places, organizations, etc.), hashtags, and moral foundations identified in the tweets, as well as the authors of those tweets.

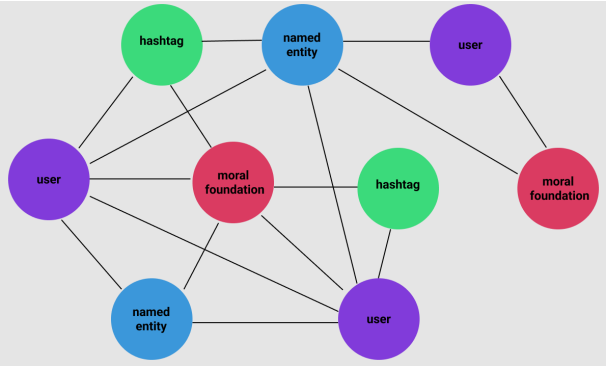


Figure 3.6: Idea for narrative graph.

Once this graph was populated, we planned to try using various graph algorithms and machine learning techniques to extract narratives from the graph.

3.4.1. Named Entity Recognition

We began by working on an NER classifier to identify named entities in each tweet so we could include them as nodes in our graph. We initially looked at a pretrained NER classifier that had fine-tuned BERT to recognize locations, people, organizations, and other miscellaneous entities in text.[1] From an initial qualitative assessment – running the model on a small group of tweets in our dataset and manually looking to see if it correctly identified all named entities – the model seemed to perform very well. However, since the model had fine-tuned BERT instead of BERTweet, and

[1]The model, dslim/bert-base-NER, can be found here: https://huggingface.co/dslim/bert-base-NER. It is a fine-tuned BERT model [6] trained on the CoNLL-2003 Named Entity Recognition dataset [18].

was trained on normal English text rather than on Twitter data, we decided to try training our own model by fine-tuning BERTweet to see if we could get an even better performance.

As mentioned in Section 3.1, we trained this model using the WNUT17 dataset [5], with a train/validation/test split of 3394/1009/1287. We used the HuggingFace Transformers library to tokenize the tweets and to load and fine-tune the BERTweet model. We trained the model for 4 epochs, using a batch size of 32 and weight decay of 0.01. For optimization, we used Adam with a learning rate of 3e-5. All other hyper-parameters were set to their default values according to HuggingFace's implementation.

The model performed quite well on the validation set, with a loss of 0.29 and an F1 score of 0.94. However, when we tested the model on a small group of tweets in our dataset, it failed to identify many entities, leading us to suspect that the model was overfitting to the training data. Before taking the time to improve the model, we tested the pretrained NER classifier's performance on the WNUT17 dataset. It also had an F1 score of 0.94, and it correctly identified many more of the named entities in the small group of tweets we had used to test our fine-tuned model. Rather than taking the time to fix our fine-tuned model, then, we decided to simply use the pretrained model to identify named entities in our tweets.

3.4.2. Moral Foundations

As mentioned in Section 3.1, we used the Moral Foundations Twitter Corpus [9] to train a classifier to predict the moral foundations of our tweets. As with our (unused) fine-tuned NER classifier, we used the HuggingFace Transformers library to tokenize the tweets and to load and fine-tune the BERTweet model. We again trained the model for 4 epochs with a batch size of 32, and used Adam for optimization with a learning rate of 1e-5. We also set a random seed of 17 for training. All

other hyper-parameters were set to their default values according to HuggingFace's implementation. The model had a validation loss of 0.32 and an F-1 score of 0.65. We used this model to assign a moral foundation to each tweet and connect these tweets with their moral foundations in our narrative graph.

3.4.3. Hashtags and Users

Besides named entities and moral frames, we had two other types of nodes in our graph: hashtags and users (authors). These were much simpler to add. We extracted hashtags (words beginning with the # character) from tweet text, and used the Twitter API to fetch the usernames of the users who had written each tweet.

3.4.4. Graph Construction

We used NetworkX, a Python library for graph construction and manipulation, to create our narrative graph. We populated the graph with the named entities, hashtags, moral foundations, and users we had identified in our tweets. Each of these items was a node in the graph, and we connected nodes that had appeared in the same tweets with edges. Edge weights represented the number of times those items had appeared together. For example, the hashtag '#covid19' and the named entity 'Biden' were mentioned together in 12 different tweets, so they were connected by an edge with weight 12. We had only five moral foundation nodes in the graph (for each of the five foundations), and hashtags, users, and named entities were connected to a certain moral foundation node if they were mentioned in a tweet that centered around that foundation. Users were connected not only to the named entities, hashtags, and moral foundations used in their tweets, but also to each other if they retweeted each other. Our final graph contained 61,482 nodes, with edge weights ranging from 1516 to 1.

3.4.5. Community Detection

Once our graph was constructed, we decided to look for narratives in communities of that graph. Graph communities are subsets of nodes that are densely connected to each other and loosely connected to other nodes in the graph. In this case, groups of users, hashtags, named entities, and moral frames might create a community if the users were discussing a specific issue with each other in depth, but not discussing other topics or responding to other users as much. Such clustered discussions could easily form a basis for narratives being created or pushed via Twitter. Various algorithms exist for detecting graph communities; we used NetworkX's greedy_modularity_communities function, an implementation of the Clauset-Newman-Moore greedy modularity maximization algorithm [3].

Running this algorithm on our graph generated 969 communities. A qualitative overview of these communities showed some patterns or general topics in a number of them. For example, one community seems to represent a discussion criticizing Trump's promotion of white supremacy, with hashtags '#WhiteSupremacy', '#KKK', '#racewar', '#TrumpsAmerica', '#TrumpChaos', '#TrumpDeathToll185', '#dictator', '#republicanlie', and '#RNC2020', and the named entity: 'Yu Americans'. Another community discusses making vaccines accessible in African nations, with hashtags such as '#Africa', '#endcovid', '#africansarenotlabrats', '#VaccineEquity', and '#sustainabletourism', and the named entity 'Vaccine Equity Africa', among other individual names. (Note: We've omitted usernames in both examples since, while users discussing topics with each other form the foundations of online narratives, their names don't contribute to the meanings of those narratives or topics.) However, most communities, including the examples above, included at least a few nodes that didn't relate to the main topic, and many communities had no clear narrative at all. Their size varied significantly, so some communities were too large

to contain only one narrative, and some only contained a few nodes. We concluded that while some communities seemed to form the bases of different narratives, the results of community detection overall were too variable and noisy to use for broader narrative identification.

3.4.6. Random Walks

After seeing the results of community detection, we tried to approach the same goal – finding meaningful clusters of closely-related nodes in the graph – with a different method. We tried running random walks on the graph: a random walk would start at a random node, then probabilistically move to one of that node's neighbors. That neighbor would be selected with a probability depending on the weight of the edge connecting those two nodes, so the walk would be more likely to move along edges with higher weights. This process could be repeated a given number of times, forming multiple 'steps' in the walk, and multiple walks could be run by starting at a different random node each time. In this way, we hoped to find densely connected subsections of the graph that might represent different narratives. However, after running a number of these walks, we found that they all quickly converged on the same one to three nodes, rather than expanding to reach more entities, hashtags, or users. It seemed that our graph consisted of enough very small neighborhoods that random walks could often get stuck in these neighborhoods instead of reaching broader narratives.

3.4.7. Classification Using Triads

Community Classification. Although random walks did not allow us to find narratives in our graph, they gave us an idea for how we might start to classify graph communities as narratives. Rather than using long random walks to extract large groups of nodes from the graph, we used a similar method to extract triads (groups of three nodes), each made up of one named entity, one hashtag, and one moral

foundation. We found each triad by randomly selecting a named entity node, then selecting the moral foundation and hashtag connected to that node with the highest-weighted edges. Each named entity node would have at least one moral foundation connected to it, since we had classified the moral foundation of every tweet. If the named entity didn't have a hashtag connected to it, we would not save that entity and moral foundation as a triad. We ran this triad-extraction process 100,000 times, making sure not to save each triad more than once, and identified 5977 named_entity-hashtag-moral_foundation triads in the graph.

We used these triads to label the graph communities we had found using community detection (Section 3.4.5). When we had initially explored these communities, we had found that many of them were too small to contain a full narrative, or had very small edge weights between their nodes, indicating that those nodes had only been mentioned in one or two tweets. If a community contained a triad with sufficiently high edge weights, however, that might indicate that the triad's hashtag, entity, and moral foundation were being discussed at length within that community, and the nodes in that community could then form the basis of a narrative. We sorted our triads by the sums of their edge weights, and planned to label communities that contained triads whose edge weights were above a certain cutoff as 'narratives', and communities without such high-weighted triads a 'non-narratives'. We ultimately made a few modifications to this plan. The first was that, rather for looking for an entire triad in each community, we looked only for a named entity that was part of a triad. This was because we only had five moral foundation nodes, so we could only expect five communities to contain a moral foundation. If a community contained the named entity of a triad, the hashtag and moral foundation that the named entity was connected to would at least be closely connected to that community. The second modification was that we classified communities as 'narratives' if they contained a triad at

all, not just if they contained a high-weighted triad. We had planned to set the triad edge weight cutoff so that approximately half of our communities would be classified as narratives and half wouldn't, but we found that this ratio was achieved only when triads with any total edge weight were included. In summary, after this process, we had a set of graph communities that were labeled as 'narratives' or 'non-narratives'.

We then wanted to use this labeled data to train a classifier to recognize narratives in narrative graphs like ours. To do this, we used a graph neural network (GNN), a deep learning model specifically trained to capture graph data and relationships, such as node data, edge data, and global structure. We used PyTorch Geometric, PyTorch's implementation of GNNs, for this task. Since PyTorch tensors can only take in numbers, we began by using scikit-learn's label encoder to encode the text of our named entities, hashtags, and moral foundations as integers. We did not include usernames when training our GNN, since they did not contribute to the meaning of any community's potential narrative. We converted our label-encoded communities from NetworkX subgraphs to PyTorch Geometric graphs. We then randomly shuffled these graphs and split them into a training set of size 775 and a test set of size 194. Finally, we used PyTorch Geometric's GCN (graph convolutional network) class to train a narrative prediction model. Our model used three GCN layers, with a final linear output layer for binary classification (predicting whether a given graph community was a narrative or not). We used dropout with a keep probability of 0.5 for regularization. We trained the model for 10 epochs, using a batch size of 64 and 64 hidden channels. For optimization, we used Adam with a learning rate of 0.01. We also used a random seed of 12345 for training. All other hyper-parameters were set to their default values according to PyTorch's implementation.

The model performed very poorly, showing no improvement over the course of training. Its train and test accuracy converged to 0.5497 and 0.4897, respectively,

within 4 epochs. With an accuracy of approximately 50%, the model did no better than randomly guessing whether a community included a narrative or not. We concluded that our graph communities were too noisy for a model to detect clear patterns separating narratives from non-narratives.

User Neighborhood Classification. Since graph communities had proved too noisy, we decided to try looking for narratives in more specific subsections of our graph. Since users, or groups of users, are the driving force behind narratives being spread on social media, we decided to explore small, user-centered sections of the graph. We extracted two-hop neighborhoods around each user (all of that user's neighboring nodes, and all of those nodes' neighbors). As with graph communities above, we labeled user neighborhoods as 'narratives' if they contained named_entity-hashtag-moral_foundation triads with edge weights above a certain cutoff, and as 'non-narratives' if they didn't. For this experiment, we used a weight cutoff of 50, because we found that if we labeled user neighborhoods as narratives only if they contained a triad whose edge weights summed to 50 or more, we had an approximately equal number of neighborhoods labeled as 'narratives' as 'non-narratives'.

As with our community narrative classifier, we used PyTorch Geometric's GCN class to train a model to predict whether a user neighborhood contained a narrative our not. As before, our model used three GCN layers with a linear output layer, used dropout with p=0.5, and was trained for 10 epochs with a batch size of 128 and 64 hidden channels. We again used Adam with a learning rate of 0.01 and a random seed of 12345. All other hyper-parameters were set to their default values according to PyTorch's implementation. The model once again performed very poorly. Its train and test accuracy started at 0.5549 and 0.5628, respectively, after the first epoch, and did not change over the rest of training. Again, the model was essentially randomly guessing whether a user neighborhood included a narrative or not.

Classification with a Heterogeneous Graph Neural Network. When training our PyTorch Geometric classifiers for the community classification and user neighborhood classification experiments above, we used graph neural networks that were meant for homogenous graphs, or graphs whose nodes and edges are all of the same type. Our narrative graph, however, is a heterogeneous graph, with different types of nodes: named entities, hashtags, users, and moral foundations. Different types of nodes play different roles in the graph and in a narrative, so a classifier that could recognize these differences might be better able to recognize a narrative. We realized that PyTorch Geometric also supports heterogeneous graphs, so we decided to try creating a heterogeneous GNN classifier for our labeled graph communities. Besides this change from a homogeneous to a heterogeneous model, the setup and training were almost identical to the community classifier described above. Our preliminary experiments with implementing this model, however, were unsuccessful, and time constraints kept us from working on the model in more depth.

The poor performance of our community and user neighborhood narrative classifiers indicated that our data was too complex for supervised learning (machine learning to predict a defined set of outcomes using labeled data). We thought that, although supervised learning with a heterogeneous graph framework would likely be better than without it, the increase in performance would probably not be enough to make the classifier successful at identifying narratives, since the initial performance had been no better than random guessing. We instead decided to use our remaining research time to explore unsupervised learning, in which we would feed our narrative graph without labels to a model that would simply try to detect patterns and information within the graph. We hoped that such a model would detect patterns that would allow us to group our graph nodes into potential narratives. We leave the completion and improvement of a heterogeneous GNN classifier to future work.

3.4.8. Unsupervised Learning

For our next experiment, we used Facebook Research's PyTorch-BigGraph [11] to generate graph embeddings for our narrative graph. Like the word embeddings described in Section 3.2, graph embedding models learn vector representations of the nodes in a graph. Similar nodes or nodes with edges between them are given more similar vector embeddings than unrelated or unconnected nodes. Most notable for our use case, PyTorch BigGraph supports heterogeneous graphs: when passing our graph data to the model, we could specify whether nodes were named entities, hashtags, users, or moral foundations, and what types of nodes each edge connected. This additional information would hopefully make the embeddings more accurate, since different types of nodes would play different roles in creating a narrative. Unfortunately, PyTorch-BigGraph does not yet support weighted edges in graphs. As a proxy, if an edge had weight n in our original narrative graph, we added that (unweighted) edge to our PyTorch-BigGraph graph n times. This is not a perfect proxy, since graph data is split into batches for training, so two edges that connect the same node might be put in different batches and the model therefore might not recognize that those nodes were connected with a higher weight. However, this was the best proxy we could find. We trained to model for 7 epochs, with a learning rate of 0.01, regularization coefficient of 1e-3, and a softmax loss function. We set the dimension of the output embeddings to be 100. After training, we saved the embeddings in four different files, one for each type of node, for further analysis.

As mentioned in the discussion of our narrative classifier for user neighborhoods, groups of users discussing and promoting similar subjects create narratives on social media. We decided, therefore, to look at the graph embeddings for user nodes and see if the embeddings had captured enough information from the nodes' neighbors to reveal the topics these users were discussing. We first tried finding and plotting

the two-dimensional tSNE embeddings of our 100-dimensional graph embeddings, as we had with our BERTweet embeddings (Section 3.2.2). Plotting these embeddings showed no clearly defined clusters. We tried tuning the tSNE hyperparameters (perplexity, early exaggeration, and number of iterations), as well as plotting random samples of 1000 data points in case the entire set was too large to see clusters in a small plot, but we had no more success.

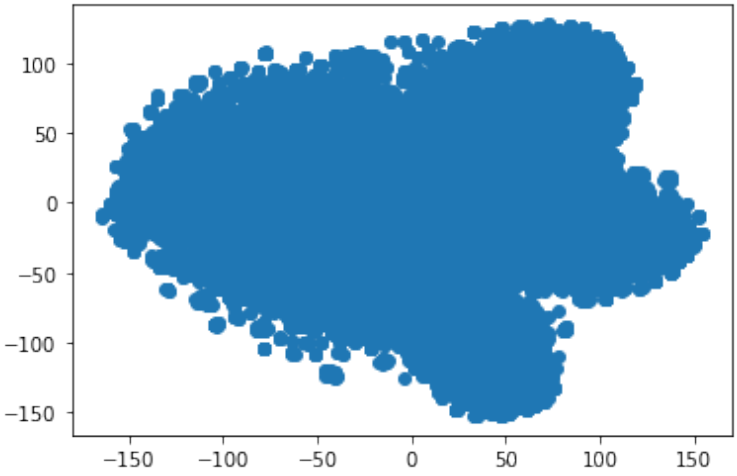


Figure 3.7: tSNE plot of user node embeddings.

We then tried generating two-dimensional embeddings with UMAP, another dimensionality reduction technique that claims to preserve both local and global data structure (while tSNE focuses on local structure). Plotting these embeddings did show us four clearly defined clusters.

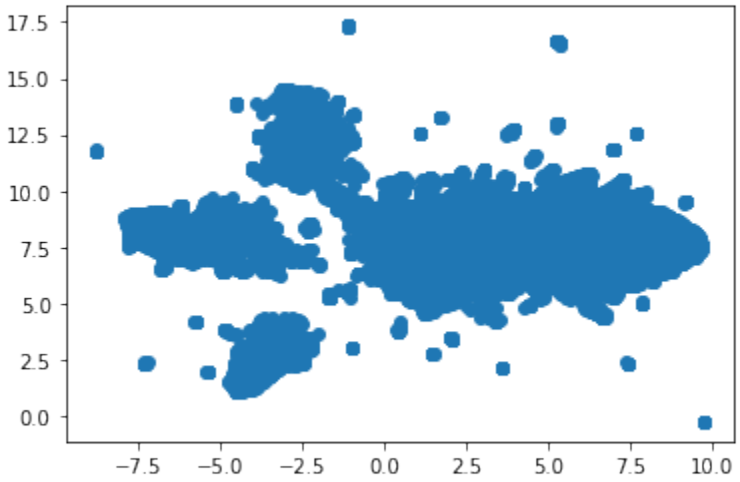


Figure 3.8: UMAP plot of user node embeddings.

We analyzed the tweet information associated with the users in each cluster and found that the user embeddings seemed to be grouped by the moral foundations of their tweets. The four clusters corresponded to the moral foundations of care/harm, fairness/cheating, loyalty/betrayal, and authority/subversion; there were not enough tweets classified as sanctity/degradation for this foundation to make a significant appearance in the graph or graph embeddings. The fact that the node embeddings were more similar for users who wrote tweets with the same moral foundations indicated that these embeddings did accurately recognize some patterns and context in the narrative graph. We decided to further analyze these user embeddings by finding hierarchical UMAP embeddings, or separately generating UMAP embeddings for the users in each moral foundation cluster. We hoped that, within each moral foundation, the embeddings might be grouped by some other characteristic, such as named entities or hashtags. Such subclusters (e.g. a group of users talking about the same entities with the same moral framing) might allow us to start identifying narratives.

The new UMAP embeddings for each moral foundation cluster did not group into subclusters as neatly as the original clustering, and the subclusters that did appear showed no clear patterns in terms of what named entities or hashtags they contained.

However, we created bar charts of the 15 most common named entities and hashtags in each subcluster in order to show the types of narratives that might be created in that subcluster. These charts demonstrated what aspects of a narrative could and could not be identified from graph embeddings of Twitter users, and are shown and discussed as part of our results in Chapter 4.

Section 3.5

Framework

After all these experiments, we concluded that using unsupervised learning on a narrative graph was the most promising direction for identifying Twitter narratives, as our graph embeddings seemed to successfully incorporate the context around each node in the graph. We will now summarize the final process used to produce and analyze the results of our narrative model.

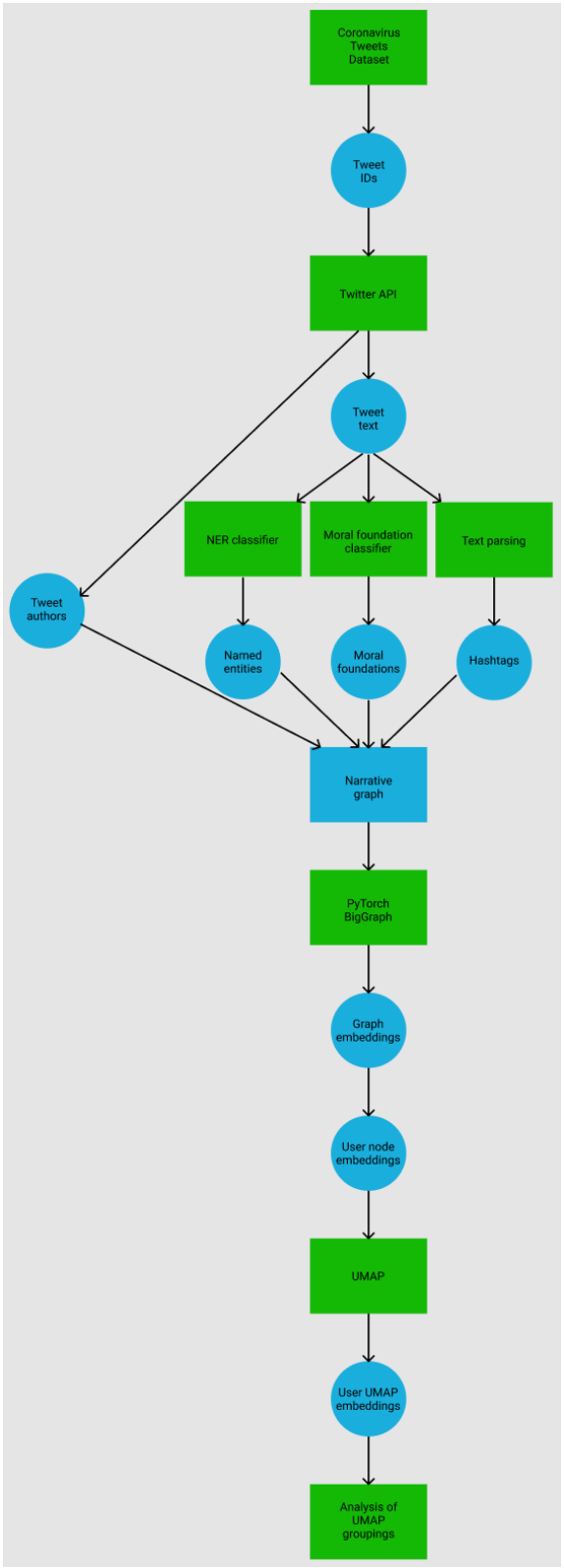


Figure 3.9: Process for creating and analyzing our narrative model.

We began by downloading tweet IDs from the IEEE's Coronavirus Tweets Dataset [10] and using the Twitter API to fetch the text and author of each tweet. We used a pretrained BERT-based NER classifier[2] to identify named entities in each tweet, and we trained our own BERTweet-based classifier to assign a moral foundation to each tweet. We also extracted the hashtags, all strings beginning with the '#' character, from each tweet. We compiled all this information into a pandas dataframe with a row for every tweet, where each row contained the tweet's named entities, hashtags, moral foundation, and user (author). We used this dataframe and the Python NetworkX library to populate a narrative graph with the named entities, hashtags, moral foundations, and users as nodes. Nodes in the graph were connected with an edge if they appeared in the same tweet, with the edge weight corresponding to the number of tweets they appeared in together. User nodes were also connected if they retweeted each other. We then used PyTorch BigGraph to generate embeddings for each node in the graph and used UMAP to visualize the PyTorch BigGraph embeddings of user nodes in two dimensions. Finally, we looked for patterns and in those UMAP embeddings, first finding clusters divided by moral foundation, then analyzing the distribution of named entities and hashtags in each moral foundation cluster.

[2]The model, dslim/bert-base-NER, can be found here: https://huggingface.co/dslim/bert-base-NER. It is a fine-tuned BERT model [6] trained on the CoNLL-2003 Named Entity Recognition dataset [18].

Chapter 4

Results and Discussion

There are no clear metrics we can use to evaluate the narrative model we have developed. The development of computational models of narrative is a relatively new field of exploration, and while numerous studies have explored the concept from a variety of directions [14], no approaches have directly paralleled ours, and we therefore have no clear baseline against which to measure our model. We will therefore provide a qualitative and visual evaluation of our results, describing the narrative graph we built and the effectiveness of graph embeddings for identifying narratives.

Section 4.1

Narrative Graph

Our graph of 40,521 tweets contained 61,482 nodes and 205,547 edges. We used pyvis, a Python library for network visualization, to display samples of the graph; the entire graph was too large to load, but we created visualizations of a sample of 2000 tweets. The images below show a graph of just the named entities and hashtags of these tweets as well as the graph of all types of nodes (named entities, hashtags, users, and moral foundations) for these tweets.

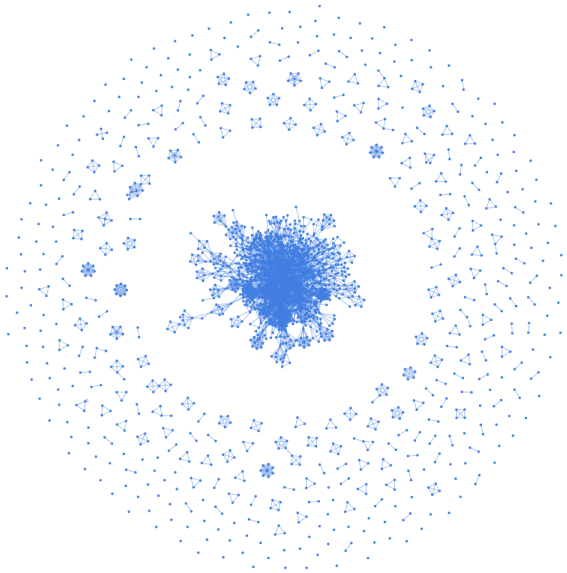


Figure 4.1: Graph of named entities and hashtags for a sample of 2000 tweets.

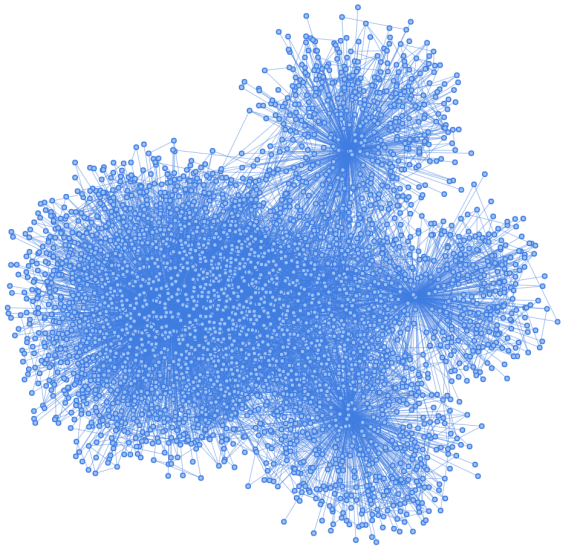


Figure 4.2: Full narrative graph for a sample of 2000 tweets.

In Figure 4.1, we see that the named entities and hashtags, which make up most of the narrative 'meaning' in our graph (in terms of providing information on what is being discussed), form very closely-connected clusters at the center of the image, with many smaller, separate topics around the periphery. Once user nodes are added in Figure 4.2, however, all sections of the graph become much more densely connected.

This addition shows how fluid and interconnected social media narratives can be. Although we can pick out separate subjects being discussed (Figure 4.1), most users discuss multiple topics and connect them with each other (Figure 4.2). While this dense, complex graph may make it difficult to identify fully separate narratives or information operations, some coherent narratives still emerge. One such example is shown below.

In order to provide a more interpretable sample of our graph, we extracted a small, interconnected section of the graph with hashtags and named entities related to COVID-19 and American politics. This section provides a demonstration of how the entire narrative graph is structured.

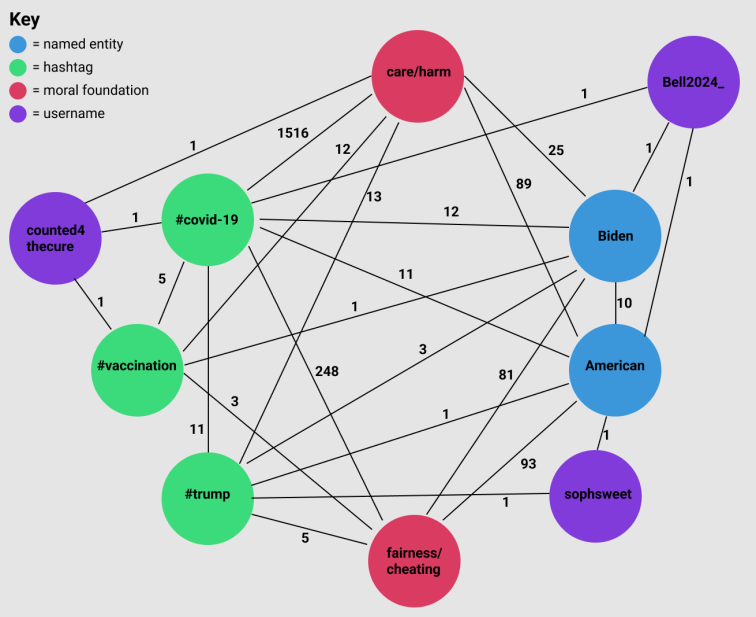


Figure 4.3: Small section of our narrative graph.

As expected, since there are only five moral foundation nodes in the graph, the edge weights connecting hashtags and named entities to moral foundations are the highest. The topics of COVID-19 and vaccinations are more connected with thoughts of care and harm, while Biden and American are more connected with thoughts of fairness and cheating. The connections between the different hashtags and named

entities, however, were perhaps lower than expected. In a collection of 40,521 English-language tweets centered around COVID-19, we expected the topics of covid, vaccination, and the political leaders of a large English-speaking country to be heavily connected, but the edge weights indicate that these of pairs entities and hashtags occurred together in only 1 – 12 tweets each. These low edge weights could perhaps provide some insight into why our models had trouble identifying narratives that we thought would be prominent, like American politicians' actions around COVID-19. Nevertheless, each of these hashtags, named entities, and moral foundations is connected to all the others, usually with edge weights well above 1, making it clear that some narrative around the pandemic and American politics does exist. This small yet densely connected portion of our graph can demonstrate how interconnected the nodes and narratives are within the entire graph.

Section 4.2

Clustering of Graph Embeddings

As discussed in Section 3.4.8, when we analyzed the UMAP embeddings of the user nodes in our graph, we found that these embeddings were generally grouped by the moral foundations of their tweets. The image below shows these embeddings colored by the moral foundation of the user's tweet. (If the user wrote multiple tweets, we simply picked the moral foundation of one of their tweets.) The size of each data point is determined by the number of tweets that user wrote.

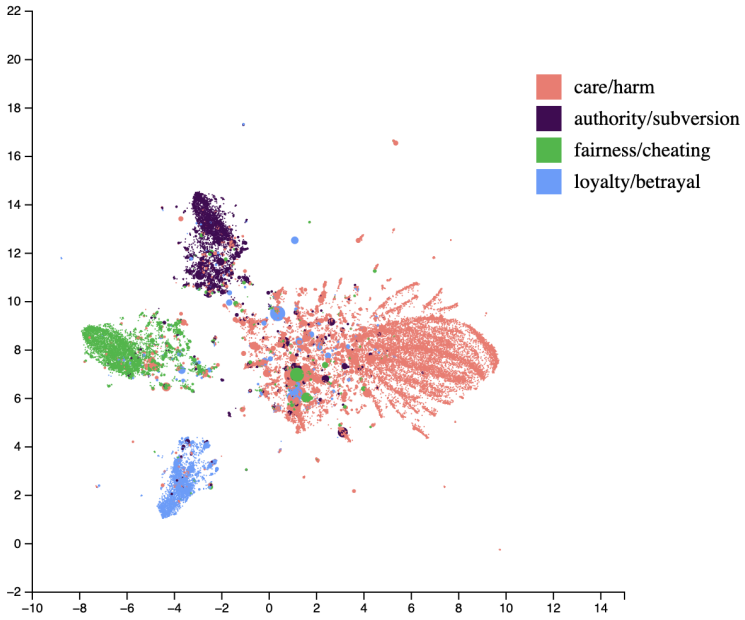


Figure 4.4: UMAP plot of user node embeddings, colored by moral foundation.

We then explored hierarchical UMAP embeddings, in which we separately generated new UMAP embeddings for the users in each moral foundation cluster. These new embeddings are plotted below (Figures 4.5, 4.8, 4.11, and 4.14). Although these embeddings did not show patterns that separated them clearly by named entity, hashtag, or general topic, they did show some separation into new clusters. We created bar charts of the 15 most common named entities and hashtags in each subcluster in order to show the types of narratives that might be created in that subcluster. These bar charts are shown below their respective embeddings.

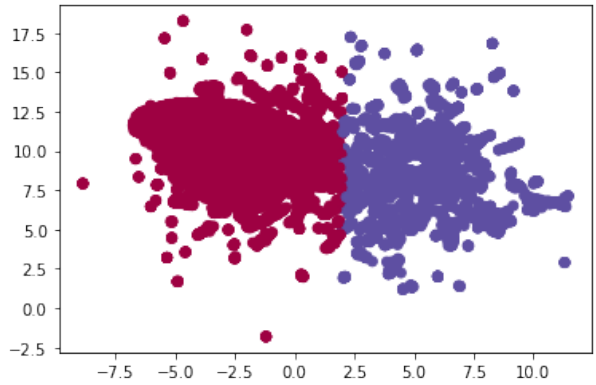


Figure 4.5: UMAP embeddings of the care/harm user cluster in Figure 4.4, colored by subcluster.

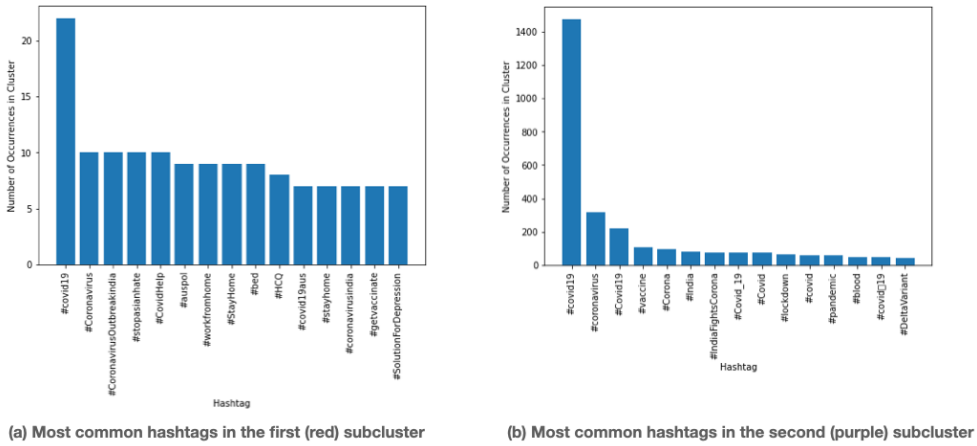


Figure 4.6: Most common hashtags in each of the care/harm subclusters of Figure 4.5.

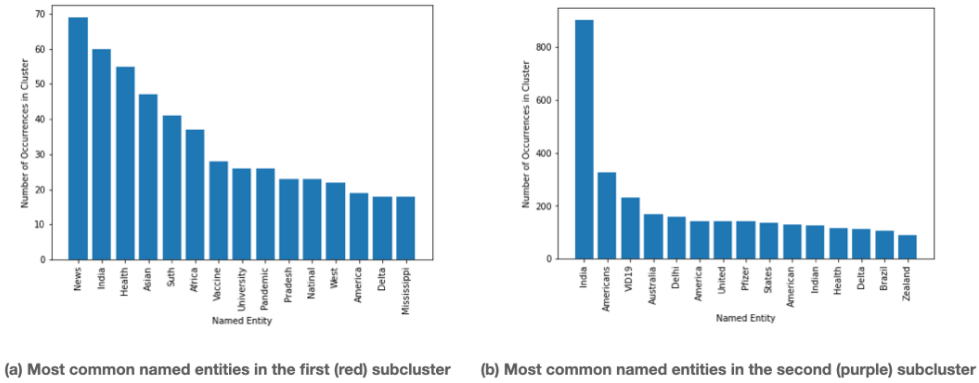


Figure 4.7: Most common named entities in each of the care/harm subclusters of Figure 4.5.

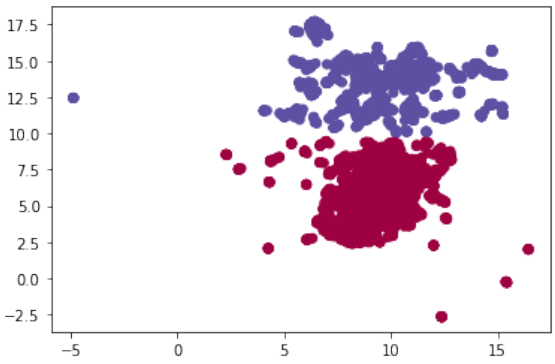
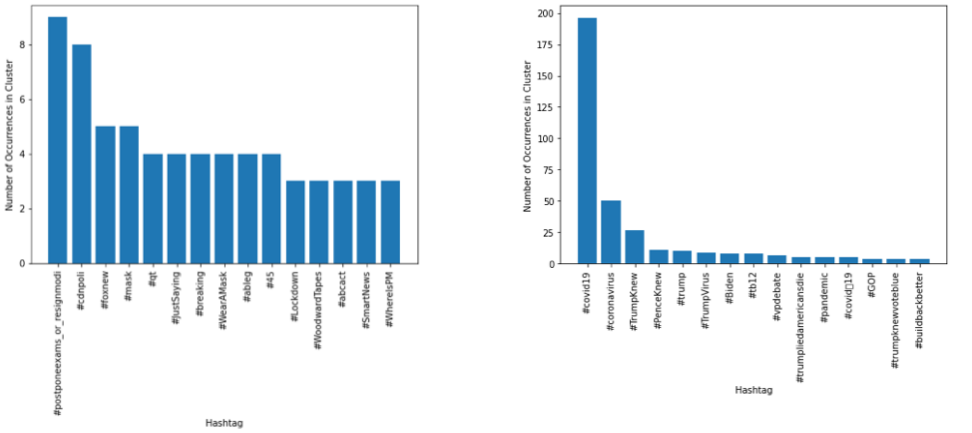
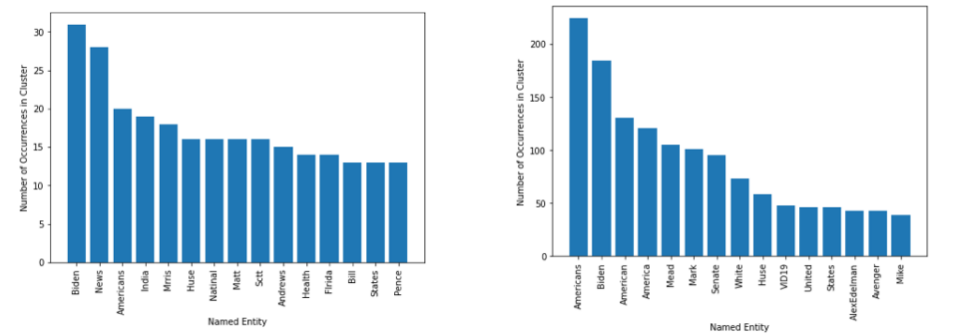


Figure 4.8: UMAP embeddings of the authority/subversion user cluster in Figure 4.4, colored by subcluster.



(a) Most common hashtags in the first (red) subcluster (b) Most common hashtags in the second (purple) subcluster

Figure 4.9: Most common hashtags in each of the authority/subversion subclusters of Figure 4.8.



(a) Most common named entities in the first (red) subcluster (b) Most common named entities in the second (purple) subcluster

Figure 4.10: Most common named entities in each of the authority/subversion subclusters of Figure 4.8.

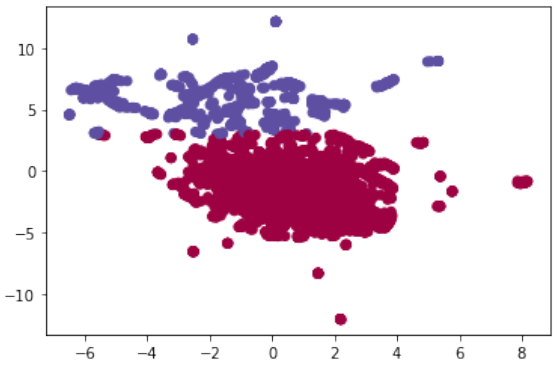
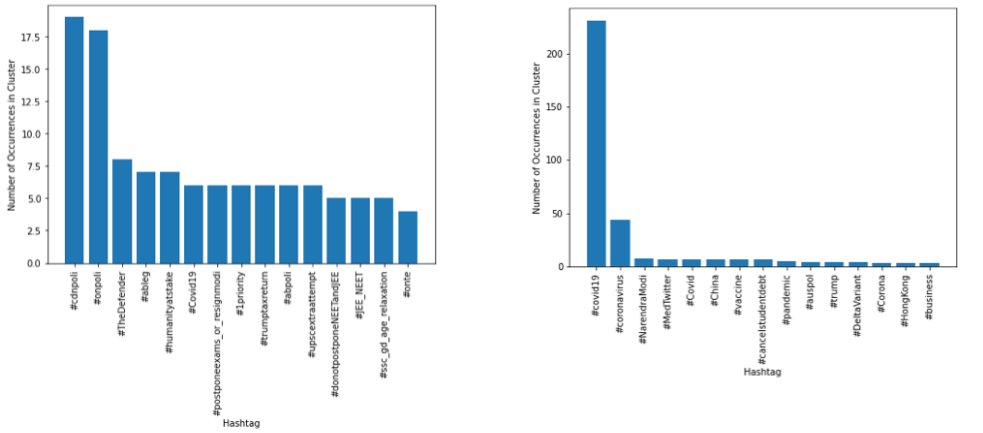
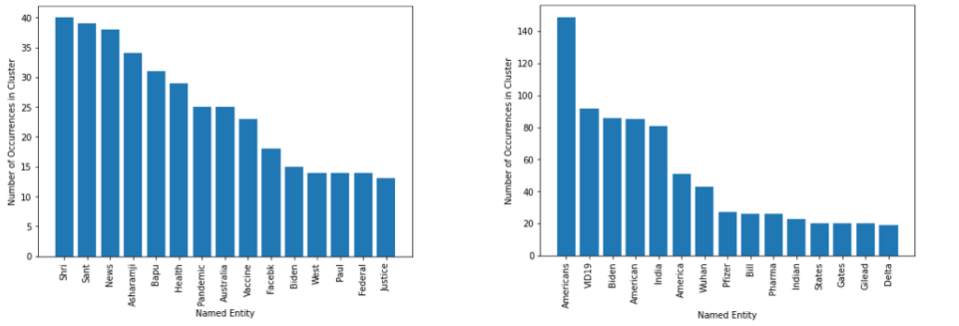


Figure 4.11: UMAP embeddings of the fairness/cheating user cluster in Figure 4.4, colored by subcluster.



(a) Most common hashtags in the first (red) subcluster (b) Most common hashtags in the second (purple) subcluster

Figure 4.12: Most common hashtags in each of the fairness/cheating subclusters of Figure 4.11.



(a) Most common named entities in the first (red) subcluster (b) Most common named entities in the second (purple) subcluster

Figure 4.13: Most common named entities in each of the fairness/cheating subclusters of Figure 4.11.

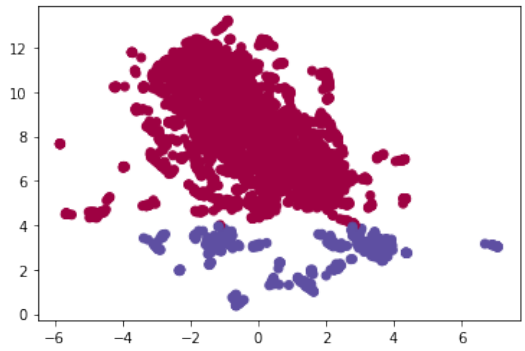


Figure 4.14: UMAP embeddings of the loyalty/betrayal user cluster in Figure 4.4, colored by subcluster.

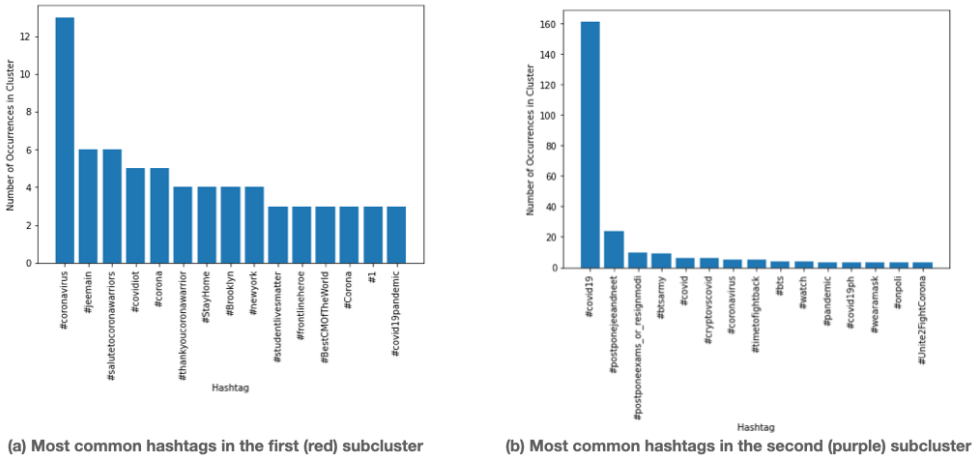


Figure 4.15: Most common hashtags in each of the loyalty/betrayal subclusters of Figure 4.14.

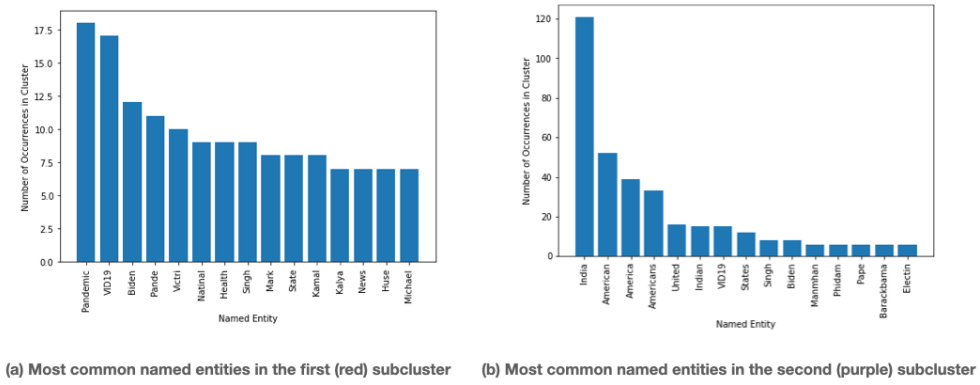


Figure 4.16: Most common named entities in each of the loyalty/betrayal subclusters of Figure 4.14.

Overall, we did not find clear clusterings of users by the hashtags or named entities they discussed. While the user embeddings for each moral foundation did separate into somewhat distinct sub-clusters, the hashtags and named entities were consistently similar across sub-clusters. These results indicate that different users discuss enough of the same named entities and hashtags that the graph data is too complex to separate out clear narratives among users based only on these entities and hashtags, or at least that users' discussions of those elements are very interconnected.

Section 4.3

Limitations

Our work on a narrative model, specifically our narrative graph, shows some promise for identifying narratives. Unsupervised learning seems to be able to pick up some relevant patterns in a narrative graph, as the graph embeddings for user nodes were able to incorporate and separate by the moral foundations of those users' tweets. Other graph analysis methods such as community detection were also able to identify general topics in certain cases. Overall, though, while our graph lets us identify some related discussions on Twitter, we cannot yet identify these discussions reliably or identify anything complex enough to call a narrative. Our model still lacks the ability to reliably identify all the necessary components of a narrative - actors, events, time, points of view, and more - and extract these elements specifically from a narrative graph.

Section 4.4

Lessons Learned

The process of working on a narrative model taught us what directions might show potential as well as what would not work and why. Our attempt at clustering

BERTweet embeddings demonstrated that narratives need more detailed representations than pretrained language models can provide. Our work with graph neural networks showed that groups of tweets are too noisy and narratives are too complex for Twitter narratives to be labeled and identified using supervised learning. Our experiments with graph communities showed how, rather than forming into clearly-separable discussions as we had hoped, social media posts about slightly different topics can still be closely connected through the users who post them or other elements that tie the topics together. Throughout all our attempts at grouping tweets into narratives – clustering of language embeddings, topic modeling, graph communities, and UMAP plots of graph embeddings – we saw large, densely connected groups of tweets discussing numerous, interrelated narratives, with some very small, unrelated discussions on the periphery. Overall, future work will likely need to focus on a method of extracting narrative elements from multiple interwoven discussions about different topics.

Chapter 5

Future Work

There are many possible steps that could be taken for future work in this area. One unfinished experiment was classifying graph communities as narratives or non-narratives using a heterogeneous graph neural network classifier (section 3.4.7). We did not finish implementing this model due to time constraints and our decision to focus on exploring unsupervised learning. Since our initial homogeneous GNN classifiers' performance was no better than random guessing, simply changing to a heterogeneous GNN model might not improve performance enough. However, a narrative classifier that made full use of a heterogeneous narrative graph would likely perform better than the homogeneous classifiers we trained, and examining this performance could provide helpful insight.

Our narrative graph and identification process could be improved by expanding the types of nodes, or narrative elements, being explored. We used PyTorch BigGraph to generate embeddings for all types of nodes in our graph, but we only explored user node embeddings in depth. Exploring the embeddings for named entity and hashtag nodes, both on their own and in conjunction with user embeddings, could provide more helpful information than the user embeddings on their own. The narrative graph itself could also be expanded upon. We used named entities, hashtags, users,

and moral foundations as a minimum version on which to start testing different graph analysis methods, but we could also add other tweet information to the graph. We could build sentiment and/or emotion classifiers and add emotions as nodes, or use sentiment to add some type of positive or negative information to edges. We could use topic modeling (Section 3.3) to identify the topic of each tweet and add topic nodes to the graph. Space and time are also important elements of narrative, so we could use tweet timestamps and any location information in the text to add time and space nodes as well. Any of this additional information could be important in framing narratives and could help identify them in the graph.

Finally, this thesis looked at tweets related to the COVID-19 pandemic to avoid being overwhelmed by the huge variety of subjects discussed on Twitter, but it would be important for any model to be able to generalize to other topics or scenarios. Initially, it could be useful and informative to train similar models on tweets related to other specified topics to compare their performance with our COVID-19 models. In the long term, though, narrative models should be trained on a broader range of tweets to be able to identify narratives in any scenario. It is also possible that our set of COVID-19 tweets was too narrow for finding many clear, separate narratives. Looking at a broader range of tweets could help clarify what narratives or general topics our model, or future models, are capable of identifying, and where they still need to improve.

Chapter 6

Conclusion

In this work, we have begun the development of a computational model of narrative on social media. Beyond the general importance of narratives for organizing and communicating thoughts and experiences, creating a representation for narratives on social media could allow for the identification of online information operations. We have explored a range of potential machine learning and graph analysis methods for identifying narratives on Twitter: clustering of embeddings from pretrained language models, topic modeling, graph community detection, supervised learning using graph neural networks, and unsupervised learning to generate and analyze graph embeddings. One of our main contributions is the representation of a narrative as a graph. Representing narrative elements in tweets as nodes in a graph, and using weighted edges to connect nodes used in the same tweets, opens up a variety of graph and machine learning techniques (including community detection, graph neural networks, and graph embeddings) that can be used to detect narratives. For this thesis, we identified named entities, hashtags, moral foundations, and users in tweets to populate the nodes of our graph, but other narrative elements, such as times, places, sentiments, or even topics could also be included. None of our work on the narrative graph thus far has allowed us to perfectly identify full narratives, but some methods,

such as community detection and clustering of graph embeddings, have allowed us to start finding potential narratives or learned groupings of some segments of the graph. Our hope is that this work can be further explored and improved upon, providing a baseline for future work on the identification of narratives and information operations on social media.

Bibliography

[1] Mieke Bal, *Narratology: Introduction to the Theory of Narrative*, 3rd ed., pp. 3–13, University of Toronto Press, 2014.

[2] Valentina Bartalesi, Carlo Meghini, and Daniele Metilli, *Steps towards a formal ontology of narratives based on narratology*, 7th Workshop on Computational Models of Narrative (CMN 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[3] Aaron Clauset, Mark EJ Newman, and Cristopher Moore, *Finding community structure in very large networks*, Physical review E **70** (2004), no. 6, 066111.

[4] Defense Advanced Research Projects Agency (DARPA), Information Innovation Office (I2O), *Influence campaign awareness and sensemaking (INCAS)*, Tech. report, October 2020.

[5] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham, *Results of the WNUT2017 shared task on novel and emerging entity recognition*, Proceedings of the 3rd Workshop on Noisy User-generated Text (Copenhagen, Denmark), Association for Computational Linguistics, September 2017, pp. 140–147.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).

[7] Alliance for Securing Democracy and Graphika, *Information Operation Archive*, Accessed Apr. 30, 2022 [Online].

[8] Jonathan Haidt, *Moral Foundations Theory*, Accessed Apr. 30, 2022 [Online].

[9] Joe Hoover, Gwenyth Portillo-Wightman, Leigh Yeh, Shreya Havaldar, Aida Mostafazadeh Davani, Ying Lin, Brendan Kennedy, Mohammad Atari, Zahra Kamel, Madelyn Mendlen, et al., *Moral foundations twitter corpus: A collection of 35k tweets annotated for moral sentiment*, Social Psychological and Personality Science **11** (2020), no. 8, 1057–1071.

[10] Rabindra Lamsal, *Coronavirus (COVID-19) Tweets Dataset*, 2020.

[11] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich, *PyTorch-BigGraph: A large scale graph embedding system*, Proceedings of Machine Learning and Systems **1** (2019), 120–131.

[12] Diego A Martin, Jacob N Shapiro, and Julia Ilhardt, *Trends in online influence efforts*.

[13] Diego A Martin, Jacob N Shapiro, and Michelle Nedashkovskaya, *Recent trends in online foreign influence efforts*, Journal of Information Warfare **18** (2019), no. 3, 15–48.

[14] Antonio Miller, Ben an Lieto, Rémi Ronfard, Stephen G. Ware, and Mark A. Finlayson, *Proceedings of the 7th Workshop on Computational Models of Narrative (CMN 2016)*, 2016.

[15] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen, *BERTweet: A pre-trained language model for English Tweets*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 9–14.

[16] Shamik Roy, Maria Leonor Pacheco, and Dan Goldwasser, *Identifying morality frames in political tweets using relational learning*, arXiv preprint arXiv:2109.04535 (2021).

[17] Marie-Laure Ryan et al., *Toward a definition of narrative*, The Cambridge companion to narrative **22** (2007).

[18] Erik F Sang and Fien De Meulder, *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition*, arXiv preprint cs/0306050 (2003).

[19] Steven T Smith, Edward K Kao, Erika D Mackin, Danelle C Shah, Olga Simek, and Donald B Rubin, *Automatic detection of influential actors in disinformation networks*, Proceedings of the National Academy of Sciences **118** (2021), no. 4.

[20] Ankit Srivastava, Naman Jhunjhunwala, Raksha Agarwal, and Niladri Chatterjee, *Narnia at nlp4if-2021: Identification of misinformation in covid-19 tweets using bertweet*, Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, 2021, pp. 99–103.

[21] Twitter, *Information Operations*, Accessed Apr. 30, 2022 [Online].

[22] Gian Piero Zarri, *Representing and managing narratives in a computer-suitable form*, (2010).