

Contextualized Conversational Network Dynamics on Social Media

Thomas Magelinski

CMU-S3D-23-101

April 2023

Software and Societal Systems Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee

Kathleen M. Carley (Chair)
Patrick Park
Osman Yağan
Renaud Lambiotte (University of Oxford)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Societal Computing.*

Copyright © 2023 Thomas Magelinski

This material is based upon work supported by the Center for Informed Democracy and Social Cybersecurity (IDeAS) Fellowship, ARCS Fellowship, Office of Naval Research, MURI Grant N000141712675: FACTIONS:Near Real Time Assessment of Emergent Complex Systems of Confederates, Office of Naval Research Grant N000141812106: Group Polarization in Social Media, Office of Naval Research, MURI Grant N000142112749: Persuasion, Identity, & Morality in Social-Cyber Environments, Office of Naval Research Grant N000142112229, Scalable Tools for Social Media Assessment and Army Grant W911NF20D0002: Scalable Technologies for Social Cybersecurity. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the ARCS, the Office of Naval Research or the United States Army.

Keywords: social networks, computational social science, social dynamics, machine learning, graph machine learning, conversational networks, online communities, contextualized networks, community prototypes

To my family.

Abstract

Network Science provides a framework to understand the large-scale discussions that happen on social media and their impact on society. However, a standard network model of a conversational network destroys the context that users are interacting within. First, the interactional context is destroyed. The interactional component of context includes the content of the conversation in which the users are interacting. When interactional context is not accounted for, separate discussions are combined into one big network, artificially inflating the number of nodes and edges in the network. This leads to inaccurate information about conversation structure and important actors. Next, the personal context is destroyed. The personal component of context includes the attributes of the users involved, as observed through their self-descriptions. Long-standing social theory of offline social communities such as self-categorization place great importance on personal context. Thus, this context needs to be accounted for to test these theories in the social media setting.

This thesis provides the theory and methodologies needed to account for both interactional and personal contexts which were previously lost in network analysis of social media conversations. Specifically, I study the importance of these contexts as they relate to community dynamics. I find that network structure is indeed dependent on interactional context, indicating that existing non-contextualized analyses could be improved. When investigating personal context, I find that the long-standing theory of self-categorization can be extended from offline social communities to massive online communities, with some important limitations. Taken together, the dynamic contextualized analysis outlined in this thesis furthers our understanding of attribute salience in online interactions. Each of these analyses is performed on multiple case studies, providing both validation and a set of examples used to detail a list of best practices for contextualized network analysis.

Acknowledgments

First, I would like to thank my advisor, Kathleen Carley for all of her support and mentorship throughout my PhD journey. Without her, this thesis would not have been possible. I'm especially grateful for her encouragement in seeking out new research directions in domains I was unfamiliar with. These directions expanded my horizons, broadened my skill set, and resulted in some of the most interesting findings of this work.

Next, I would like to thank my committee members, Patrick Park, Osman Yağan, and Renaud Lambiotte. Each of them have helped form this work through their advice, questions, and comments.

I consider myself lucky to have spent my time at Carnegie Mellon learning with and from the members of CASOS. Specifically, I'd like to thank Dave Beskow and Iain Cruickshank for teaching me how to be a successful Ph.D. student and how to develop a healthy work-life balance. I spent many hours at the whiteboard reasoning about network vitalities and trails with Mihovil Bartulovic. That time lead to crucial developments in this work. And lastly, I would like to thank J.D. Moffitt for being the best co-teaching assistant I could have asked for.

Sienna Watkins and Connie Herold have both made my time at Carnegie Mellon much easier. They have saved me countless hours in navigating through the academic bureaucracy. So to them I give thanks.

I have also been fortunate to have great collaborators beyond Carnegie Mellon. Thank you to Jialin Hou, Zachary K Stine, Thomas Marcoux, and Nitin Agarwal. I especially want to thank my collaborators in Ukraine, Tymofiy Mylovanov and Tymofii Brik. Your bravery over the past year has been an inspiration.

I wouldn't have pursued a Ph.D if it weren't for my mentors in the earlier stages of my academic career. Thank you to Sunny Jung for giving me my very first research experience, to Shane Ross for helping me develop my independent research skills and encouraging me to apply to grad school, and to Nicole Abaid for fostering my interest in social networks. Going back even further, I must thank my high school teachers Jeffrey Koegel for opening my eyes to the variety of applications of mathematics and Marc Cicchino for expanding my academic interests beyond the hard sciences.

Finally, I would like to thank my family and my partner, Jess. It is difficult to describe the gratitude I have for their constant love and support.

Contents

1	Introduction and Motivation	1
1.1	Overarching Thesis Goal	1
1.2	Literature Review	4
1.2.1	Social Cybersecurity	4
1.2.2	Community Detection and Clustering	5
1.2.3	Dynamics of Network Communities	7
1.2.4	Story and Topic Detection	7
1.2.5	Machine Learning on Networks	8
1.2.6	Networks and Identity	8
1.2.7	Community-Aware Centrality	9
1.3	Data	9
1.3.1	Reopen America	9
1.3.2	2020 US Elections	10
1.3.3	Ukraine Legislature	10
1.3.4	Captain Marvel	11
1.3.5	Coronavirus	11
1.3.6	Specialized News Discussion	11
1.4	How to Read This Dissertation	12
2	Contextualizing Social Media Conversational Networks	13
2.1	Related Work	16
2.2	Simple Labeling Approach	18
2.2.1	Interactional Contexts in the Reopen Dataset	20
2.2.2	Interactional Contexts in the Election Dataset	20
2.3	A Deep Learning Approach	20
2.3.1	Data Cleaning	21
2.3.2	Heterogeneous Network Construction	21
2.3.3	Deep Tweet Infomax	22
2.3.4	Validation	29
2.4	Automatic Labeling of Contexts	36
2.5	Impact of Contextualization on Networks	38
2.5.1	Nodeset Overlap in Tweet Contexts	38
2.5.2	Influencer Overlap in Tweet Contexts	40
2.6	Limitations	42

2.7	Discussion	43
3	Contextual Dynamics of Social Media Discussions	45
3.1	Related Work	45
3.2	Intra-Context Activity Dynamics	48
3.2.1	Categories of Intra-Context Dynamics	50
3.3	Intra-Context Network Dynamics	54
3.3.1	Methods: Snapshot-Based Change Detection	55
3.3.2	Validation	59
3.3.3	Results	63
3.4	Inter-Context Activity Dynamics	67
3.4.1	Categories of Inter-Context Dynamics	67
3.4.2	Sequences and the Temporal Order of Contexts	70
3.5	Inter-Context Network Dynamics	71
3.6	Discussion	74
4	Dynamics of Online Community Prototypes	76
4.1	Related Work	77
4.2	Methods	79
4.2.1	Network Construction and Community Detection	79
4.2.2	Prototype Measurement with Projected Modularity	80
4.2.3	Community-Level Visualizations	82
4.2.4	Prototype Construction with Projected Modularity Vitality	83
4.2.5	Multi-Modal Analysis	86
4.2.6	Computation	86
4.2.7	Relating Prototypicality and Status	87
4.3	Results	88
4.3.1	The Presence of Prototypes	88
4.3.2	The Construction of Prototypes	90
4.3.3	Relationship Between Prototypicality and Status	92
4.4	Discussion	93
5	Pipeline for Contextualized Conversation Dynamic Analysis	98
5.1	Best Practices for Data Collection	98
5.1.1	Guiding Principles	98
5.1.2	Collection of the News Dataset	100
5.2	Full Contextualized Analysis	103
5.2.1	Identifying Interactional Contexts	103
5.2.2	Characterization of Contexts	105
5.2.3	Contextualized Dynamic Network Analysis	106
5.2.4	Contextualized Prototype Analysis	108
5.3	Discussion	110

6	Thoughts and Conclusions	113
6.1	Overview of Contributions	113
6.2	Limitations and Future Directions	114
6.3	Beyond Twitter	116
6.4	Concluding Thoughts	116
A	Manual Interactional Context Annotation Details	118
A.1	Reopen	118
A.1.1	Liberate Tweets	118
A.1.2	Trump’s Job Reopening	118
A.1.3	Trump Refuses CDC Guidance	118
A.1.4	Fauci Critique	118
A.1.5	Recall Whitmer	118
A.1.6	Florida Data Scientist	119
A.1.7	Arizona Scientists	119
A.1.8	Reopen Updates	119
A.1.9	Reopen Commentary	119
A.1.10	Reopen Strategy	119
A.1.11	Economy	119
A.1.12	Worker Unemployment	119
A.1.13	Mask Orders	119
A.1.14	Schools	119
A.1.15	Reopen Satire	120
A.1.16	COVID Information	120
A.1.17	Worker Precautions	120
A.1.18	Lockdown Hypocrisy	120
A.1.19	Vaccine	120
A.1.20	Anti-Vaccination	120
A.1.21	People Testing Positive	120
A.1.22	Anti-Mask Violence	120
A.1.23	Punishment for Lockdown Violators	120
A.1.24	Reopen Protesters	120
A.1.25	Protest Coordination	121
A.1.26	Black Lives Matter	121
A.1.27	Bot Story	121
A.1.28	Reopen Criminal Cases	121
A.1.29	Petitions	121
A.1.30	Lowes Donation	121
A.1.31	Healthcare Workers	121
A.1.32	Hurricane Support	121
A.1.33	General Flynn	121
A.1.34	General Politics	122
A.1.35	Boycott China	122
A.1.36	Entertainment	122

A.1.37 Memes	122
A.1.38 Oregon Burning Aborted Babies	122
A.1.39 Miscellaneous	122
A.2 Election	122
A.2.1 Claims of Fraud	122
A.2.2 Spam	122
A.2.3 Biden Campaign	122
A.2.4 Trump Campaign	123
A.2.5 Election Updates	123
A.2.6 Biden Won	123
A.2.7 Trump Won	123
A.2.8 Vote Information	123
A.2.9 Vote Counting	123
A.2.10 Trump Has COVID	123
A.2.11 Democrat Comedy	123
A.2.12 Biden Racism	123
A.2.13 Antifa	124
A.2.14 Democratic Fundraising	124
A.2.15 A\$AP Rocky	124
A.2.16 Biden's Bus	124
A.2.17 Hunter's Laptop	124
A.2.18 Attacks on Voting Officials	124
A.2.19 Kamala Equity	124
A.2.20 Covid and Trump	124
A.2.21 Project Veritas	124
A.2.22 Voter Purge	125
A.2.23 Election Memes	125
A.2.24 USPS	125
A.2.25 Trump to Declare Early	125
A.2.26 Biden's Health	125
A.2.27 Trump Motivation	125
A.2.28 Suing Trump	125
A.2.29 Black Lives Matter	125
A.2.30 Deported Veteran	125
A.2.31 Medows Gets COVID	125
A.2.32 Alex Trebek	126
A.2.33 Anti-QAnon	126
A.2.34 Federal Workers	126
A.2.35 NBA White House	126
A.2.36 Miscellaneous	126
B Vector-Contextualized Networks	127
B.1 Development of Vector Contextualized Networks	127
B.2 Case Study on the Election Dataset	129

C Modularity Vitality	133
C.1 Introduction	133
C.2 Prior Work	135
C.2.1 Preliminaries	135
C.2.2 Modularity and Grouping	136
C.2.3 Network Centrality Measures	137
C.2.4 Evaluation: SIR Models and Network Robustness	140
C.2.5 Community Deception	141
C.3 Calculating Modularity Vitality	142
C.4 Methodology	146
C.4.1 Fragmentation-Based Evaluation	146
C.4.2 Attack Strategies	146
C.5 Network Fragmentation	148
C.5.1 Generated Networks	148
C.5.2 PA-Road Network	150
C.5.3 Canadian Election Twitter Network	151
C.5.4 Additional Experiments	153
C.5.5 Discussion	155
C.6 Community Deception	158
C.7 Conclusion	161
D Additional Prototype Results	163
D.1 Extended Results Diagrams and Tables	163
D.1.1 Community Diagram on Unfiltered Data	163
D.1.2 Salient Attributes	163
D.1.3 Prototypical Attributes	167
Bibliography	182

Chapter 1

Introduction and Motivation

1.1 Overarching Thesis Goal

Many of the central questions regarding social media and its impact on our society boil down to questions about conversational networks. How polarized are online communities? What makes fake news spread, and does it spread faster than real news? How does the incentive structure change how people communicate? These are just some of the many important questions surrounding social media, but to answer these questions and more we need to study conversational networks.

Conversational networks are social structures of users who interact with each other as they communicate online. On platforms like Twitter, Facebook, Reddit, Tiktok, Snapchat, and Instagram, users interact with each other by sending text or multi-media content in various forms, including direct messages and replies. The conceptualization of this social activity as a network enables researchers to quantitatively study the large scale behavior of users on the platforms. From these networks, groups of users or “communities” can be derived, users can be ranked by their importance, and much can be learned through the structure of the network itself. These analyses are made possible by the decades of development within the field of Network Science, which seeks to understand networks, social or otherwise.

While the Network Science toolkit has grown to be a vast and powerful set of scientific analyses, their application to social media data remains a challenge. The underlying assumption in a network analysis, stated or otherwise, is that all interactions are equivalent. In many scenarios, this assumption is easily satisfied, in others this assumption is met by careful experimental design. For example, a needle-sharing network may be used to study risk of HIV in a community of people who use drugs. There, each instance of sharing a needle-sharing is seen as equivalent in that they are equally able to spread HIV provided the initial user has it.

However, this assumption is not easily met on social media. Because of this violation, our analyses are corrupted while we are looking to answer important questions like those about polarization or the spread of fake news. This dissertation seeks to demonstrate why this is a problem and to provide a some ways forward.

A classic social network model assumes that an interaction between two people can be recorded by a simple edge, or a connection between the two. However, interactions on social media are complex and by reducing this complexity to a simple edge, information is lost. Throughout this dissertation, the “information” that is being lost is referred to as the *context* of discussion. Specifically, we define context as the information surrounding a social interaction that sheds light on its meaning. We break context down into two components: the *interactional context* and the *personal context*, as illustrated in Figure 1.1.

The *interactional context* refers to the information about the nature of a specific interaction that helps shed light on its meaning. This dissertation is concerned with conversational networks, so we also refer to this as the *conversational context*. Thus, the conversational context captures what is being said when two users interact through conversation. In figure 1.1, we see two very different types of interactions. In the first, one user asks another about a soccer game, while in the second the first user accuses the other of attempting to steal an election. The different interactional contexts give us a different understanding of why the interaction is taking place and the relationship between the two users. While this is an extreme example, it should be clear that considering these two interactions to be equivalent is problematic.

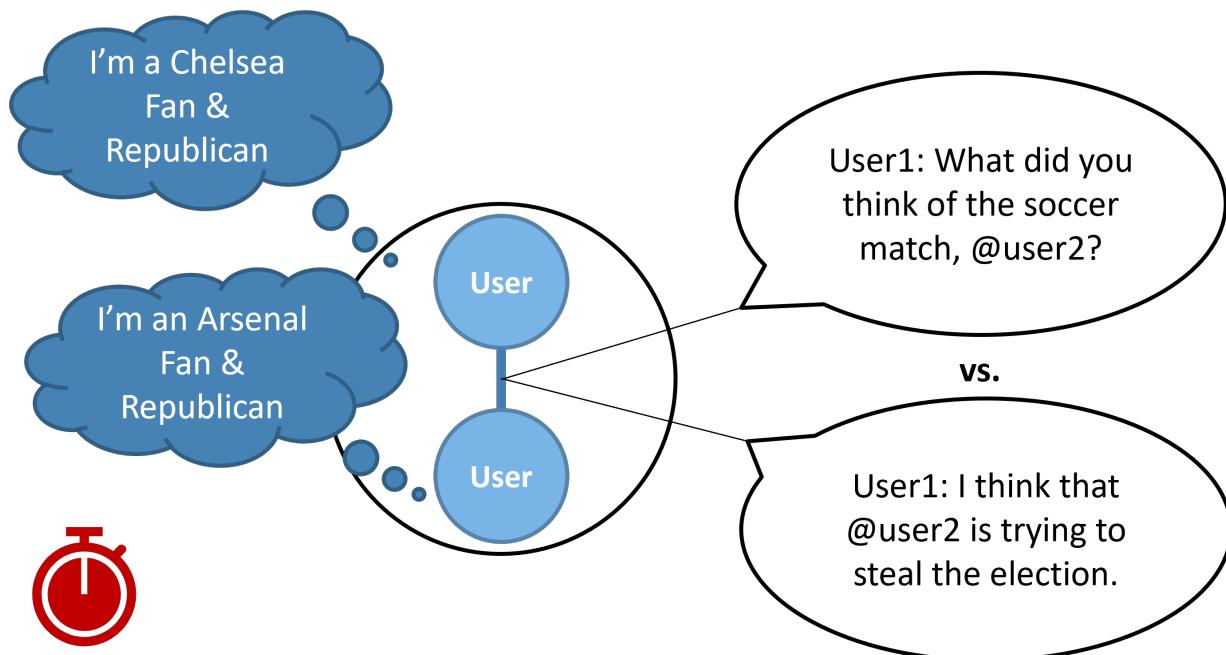


Figure 1.1: Cartoon illustration of the two types of contexts. Interactional context is shown on the right, with two very different exchanges between the users. On the left, personal context is shown. Both types of contexts are dynamic.

If conversational context is a fitting definition and is easier to understand, why call it interactional context? The tools and analyses provided in this dissertation apply to more than just conversational networks. For example, an online community may be centered

around users sharing images with each other. There, the interactional context refers to the information encoded in the images being shared. The same logic applies, that very different contexts shouldn't be combined. Thus, the use of the more general term, interactional context, serves to be a reminder that the problems and solutions identified in this work go beyond just conversational networks.

Figure 1.2 details a simple example to show how the presence of different interactional contexts in your dataset can corrupt a network analysis. In this example, we consider a Twitter dataset collected on the discussion of the Reopen America Protests of 2020, which sought to end COVID restrictions. In each network, nodes represent users who are connected to each other when they interact through replies, quotes, or retweets. For example if one user replies to another's Tweet, they are connected in the network. We see that there are three discussions, or three conversational contexts, in the dataset; one about the protests, one about strategies to reopen, and one about Black Lives Matter, which was a trending topic at the same time.

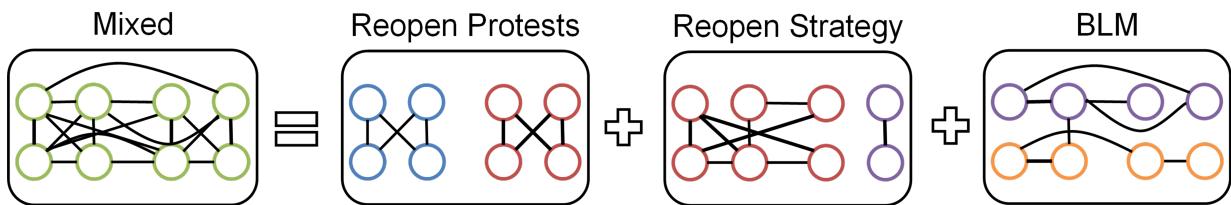


Figure 1.2: Cartoon illustration of how contextual mixing can hide community structure. Networks represent users connected through conversation. Nodes are colored by their network community. The “mixed” network includes links from all three contexts. This lack of contextualization hides the community structure seen in the contextualized networks. After contextualization, we understand that Reopen Protests and Reopen Strategy have similar structure, while the BLM conversation is very different.

The “mixed” view of the network corresponds to the current way of doing network analysis on social media; all of the interactions from different contexts are combined into a single network. Within this network, there is no discernible structure. However, when we consider each interactional context as a separate network, we see interesting structure. In the case of the protests and the strategy discussions, nodes are roughly grouped in a left community and a right community. In the case of Black Lives Matter discussion, there is top-and-bottom structure.

We refer to the networks on the right, those that each correspond to a single discussion, as *contextualized networks*, since they control for the interactional context. The first major goal of this dissertation is to develop a set of tools for uncovering these networks, which give a much more accurate view of the data compared to the “mixed” networks that current studies operate with. This toolset is developed in Chapter 2. From there, we further develop a set of tools for understanding the relationship between contexts in Chapter 3. These tools could tell us, for example, how the conversation about the protests is related to the conversation about strategies to reopen.

The second component of context in online interactions is the *personal* context, which

refers to the information about the users themselves which may shed light on their interactions. For example, in Figure 1.1, the users are members of the same political party but are fans of different soccer teams. This new layer of information can paint interactions in a new light. As we will see, there is a wide body of work in sociology and social psychology about how these attributes and their alignment colors interactions. For example, the question “what did you think of the soccer match?” seems innocent, but when we know that the users are fans of opposite teams, we see that this could be a taunting question. Again, ignoring this information when analyzing conversational networks will limit our analyses. Thus, it is the second major goal of this work to develop a set of analyses to understand the personal context of individuals interacting online, and how it relates to the broader view of online communities. These tools are developed in Chapter 4.

To summarize, online conversations are complex, and interactions between users are colored by the interactional context (what they are discussing) and the personal context (who they are). Current network analyses do not take this context into account, and as a result their conclusions are corrupted. In this dissertation, a series of tools and analyses are developed to properly account for both types of context. This begins in Chapter 2, where methods for extracting the interactional contexts are developed in order to perform the network separation observed in Figure 1.2. From there, methods for analyzing the social dynamics within and between these contexts are developed in Chapter 3. Following this, a method for studying the relationship between self-descriptions and online social communities is given in Chapter 4, shedding light on the applicability of offline sociological theories to the online domain. Lastly, Chapter 5 details a pipeline that demonstrates how all the tools developed in this work fit together.

1.2 Literature Review

1.2.1 Social Cybersecurity

Perhaps the most pressing area of research relying on a solid understanding of online communities is the area of social cybersecurity, which is defined in [39] as follows:

Social Cyber-security is an emerging scientific area focused on the science to characterize, understand, and forecast **cyber-mediated** changes in human behavior, social, cultural and political outcomes, and to build the cyber-infrastructure needed for society to persist in its essential character in a **cyber-mediated** information environment under changing conditions, actual or imminent social cyber-threats.

Thus, this area of work encompasses a number of important challenges in the information environment including the spread of disinformation and the measurement of polarization [38]. Early work on the science of “fake news,” for example, calls for further work to understand its spread and how it is received [81, 124]. At the individual level, network centrality measures are often used to determine important actors in a conversation. However, these centrality measures have been found to be sensitive to the quality of the observed network compared to the underlying network; thus, a de-contextualized network adds “noise” edges

to the point which centrality analysis may be unreliable [26]. These problems are typically studied at the community-level, and as such, community-detection is often included in information operation analysis pipelines [218].

Analysis of polarization, too, often relies on analysis of interaction networks and could thereby benefit from a contextualized approach. Specifically, distinct communities within retweet networks are often used as evidence of polarization [48, 69, 205]. However, these analyses do not *contextualize* the observed retweets. Without this contextualization, it is difficult to distinguish if communities are polarized because they are supporting opposing ideas, or they are simply involved in different discussions.

A final example is the problem of stance-detection, where social media data is used to label users' position on a topic, e.g., pro or anti-gun control. Note that stance detection is closely related to the study of polarized communities, however it is methodologically distinct in that it uses content-based approaches to label users before performing network analysis [122]. Stance detection methods such as that in [122], assume that all observed data is on-topic enough to leverage, however early results of the contextualization process developed in this thesis suggests that this is not the case. The contextual mixing that occurs in social media datasets could be harming results of these analyses, and thereby could be improved by this work.

1.2.2 Community Detection and Clustering

Community detection is the problem of dividing a network into sub-networks, or “communities” where nodes are more closely related to other nodes within the community than they are to nodes in other communities [228]. Community detection is a core problem in the study of network analysis, and as such many methods have been developed in the space with the dominant approach being modularity maximization [22, 158, 213].

Two sub-areas of community detection are of particular relevance to this thesis. First, is the application of community detection to spatial networks, or those where nodes are fixed at a location in space [60, 163]. Spatial networks are relevant but distinct from contextualized networks. While spatial networks have nodes embedded in space, contextualized networks can be modeled to have edges embedded in space, where their spatial position indicates the context occurring in that interaction. Thus, the methods from spatial networks will not be directly applicable, but may be worth considering in the development of new contextualized methods. When it comes to edges embedded in a vector space, classic clustering techniques such as DBSCAN and its variants are applicable [58, 67, 144, 145, 193].

More directly related is the area of multi-view, multi-layer, or multi-slice networks, which expands upon traditional networks with the addition of distinct edge types [6, 150]. A contextualized conversation could be modeled as a multi-view network where users are nodes and edge types represent interactions within different contexts. For example, two users might have one edge indicating their conversation about sports, and another edge indicating their conversation about politics. A series of specialized techniques for clustering multi-view networks have been developed that give a single definition of communities that combines information from all views [49, 106, 152]. This is a useful approach to incorporating contextual information to improve the quality of detected communities.

This is particularly important due to work that indicates that multi-layer cluster structure drives the diffusion of information over a multi-layer network [240, 242]. In our case, this could mean that decomposing social media conversations into a multi-layer network could uncover diffusion patterns that were obscured through contextual mixing. However, multi-view clustering’s output of a single definition of communities will not allow for the comparison of contextualized communities or the analysis of communities shifts between contexts, as is the focus of this thesis.

All of the methods discussed thus far are traditional in the sense that each node is assigned a single community. A more complex approach is overlapping community detection, or fuzzy clustering, wherein each node can be assigned to multiple clusters [232]. This can be done extending approaches used in the traditional setting, such as matrix factorization, or by alternative approaches like link clustering [4, 234]. In the case of link clustering, links are partitioned into communities rather than edges, which naturally leads to nodes being associated with all the communities their connections are. The idea of assigning a node to multiple communities is no doubt useful for online social networks; users have different social groups online. For example, a Twitter user might have connections to a group of friends he talks about sports with and a group of coworkers he talks about the economy with. Overlapping community detection may be able to sort out this distinction, but we take a different approach. We control for the context of interaction such that nodes should only be present in a single community. Using the previous example, if the discussion about sports and the economy was analyzed as two separate networks, overlapping community detection is not needed.

Regardless of the community detection algorithm applied to a dataset, the results must be scrutinized. Different algorithms that maximize different heuristics are likely to lead to poor results in at least some scenarios. Modularity maximization methods have found particular scrutiny. Because of the glassy structure of modularity, different runs on a stochastic modularity maximization can lead to very different network partitions [78]. Further, because modularity maximization methods are heuristic-based and not built off a generative model, they are unable to determine scenarios where there are no valid communities in a dataset [171]. Even when considering other approaches, the structure of large-scale communities has been called into question with a large study that suggests that much of the community structure in very large networks is occurring on a scale too small for methods like modularity maximization to pick up [127].

With this said, modularity-maximization methods like Louvain and Leiden remain the dominant approach to studying large-scale community structure because they are some of the only methods that can be easily run on networks with millions of nodes. Instead of applying more complex and less scalable approaches to find communities, we simply ask: are the communities obtained from methods like Leiden clustering valid? In terms of validity, we turn to an aspect of the social media data independent of the clustering: the identity attributes of the members of the community. As Turner has argued, shared self-definition through social attributes is more important for group membership than the structure of the group’s interactions when understanding communities [214]. Thus, by showing that the conversational clusters have a shared sense of social identity, we provide validity to the clusters.

1.2.3 Dynamics of Network Communities

Understanding the dynamics of network communities is another core area of work in Network Science, however the prevailing models are difficult to apply to social media data. A popular approach to modeling network dynamics is to use network snapshots, which model the dynamic network as a sequence of static networks, usually constructed from the edges occurring within fixed time windows [172, 173, 208]. Snapshot-based approaches have also been developed on temporal networks [101, 102, 141]. These approaches then compare the snapshots, either at the network level or based on community structure. Such comparison is not possible in social media datasets, where there may be little overlap in the users present in different snapshots, and where adjacent snapshots may have networks that differ in size by orders of magnitude. While snapshot-based approaches have some, but not enough, tolerance for the transience of nodes seen on social media, statistical methods are even more restrictive. Many statistical approaches, such as the stochastic actor-oriented model assume near perfect knowledge of node connections, measured at regular intervals, which is far from the data seen on social media [162, 201, 202].

Trails are another approach to understanding network community dynamics that is relevant for this thesis [16, 36]. Trails can be used to model nodes' transitions between semantic states while accounting for the time between these states. For example, in [36], trails modeled how terrorist organizations transitioned between different types of attacks. In this thesis, trails will be used to understand how users transition between contexts.

1.2.4 Story and Topic Detection

For the problem of content-based contextualization, the areas of topic detection and story detection are very relevant as they both make use of social media text to better understand the context of a post. These methods are slightly different than the notion of context that I will use in this thesis, as will be explained. Also, there is little work that goes beyond the detection and analysis of a topic or a story to understand how they relate to community dynamics.

Topic detection seeks to uncover patterns, or “topics” in a collection of text documents [20]. These topics are typically characterized by their most prominent and frequent words. There are many topic detection models that have been developed, including a number of methods that have been designed specifically for social media by leveraging the brief nature of social media posts and the presence of hashtags [9, 44, 61, 104, 136, 223, 226, 243].

While topics can be distilled to a series of words stories often have a notion of a topic tied to a specific event. Story detection methods build on topic-based approaches to find temporally prominent topics corresponding to events that occur during data collection [7, 8, 56, 165, 203]. Again, these models predominantly leverage social media text, but also use temporal patterns.

For this thesis, a method of accounting for context will be developed similar to the techniques used in topic and story detection. However, the method will expand on the usage of text by including both hashtags and URLs, as well as the conversational structure directly.

1.2.5 Machine Learning on Networks

Recent developments in machine learning will enable the content-based contextualization method developed in this thesis. The machine learning community has seen increasing interest in deep learning methods applicable to graphs. These approaches work by converting network-based data into vector-based data. Some approaches use random walks to generate sequences of nodes which can then be fed to a skip-gram architecture to embed nodes in a vector space based on network structure alone [82, 174]. More commonly, node attributes (in vector form) are required. Nodes can then aggregate information from their local neighborhoods to obtain their vector embedding [37, 41, 155, 224]. This has led to a large area of research into what type of aggregation scheme nodes should adopt in different scenarios [30, 87, 118, 220].

A graph-based framework can be used to model Twitter posts, as each post may be connected to other posts (replies or quotes), hashtags, and URLs. Thus, a twitter dataset can be seen as a heterogeneous network connecting tweets, hashtags, and URLs. Representing this network in a vector space enables contextualization of interactions observed in Tweets. The majority of information in a tweet is encoded in the tweet’s text, which could be represented by a vectorized node-feature using a variety of different natural language techniques [23, 53, 111, 147].

Until recently, feature-based methods required supervision or some labeled training data to work with. However, Deep Graph Infomax has been developed as a framework for learning feature-based representations of graphs in an unsupervised manner using mutual information [221]. Further, this has been expanded to heterogeneous networks [183, 221]. These methods will be at the core of the contextualization model developed in Chapter 2.

1.2.6 Networks and Identity

For the other type of context considered in this thesis, personal descriptions, there is a wide area of prior work. Sociology has long been concerned with how internal processes play out at the community level. The specific theories most relevant to the connection between individual attributes and community dynamics are social identity theory and self-categorization theory [18, 89, 93, 97, 98, 99, 105, 169, 215, 216]. These theories posit that the concept of self is defined in terms of attributes and these attributes are selected with respect to the community that an individual is or wants to be a member of. Self-categorization theory outlines the idea of a “community prototype” or a collection of attributes that would belong to a prototypical member of that community.

Social theory states that members are aware of these prototypes and are aware of how their attributes compare to it. The theory posits that these relationships are key factors in tie formation and group dynamics. Specifically, people with prototypical attributes have higher potential for leadership roles. Conversely, community members who are poorly aligned with the group prototype will seek to conform to the group to improve their status. The theory of prototype adoption is quite similar to models of correlated information spread, where abstract bits of information are spreading along a network, but the adoption of these bits of information can be correlated [241]. This is similar to prototype adoption

in that a number of attributes are potentially being adopted across a network, pairs of attributes within a prototype are positively correlated, while those between prototypes are negatively correlated.

It has been found that Twitter users do signal their social identity in their biography [169]. Further, there is evidence that user self-description alignment is associated with content propagation on Tumblr [237]. These studies provide evidence that community prototypes may exist on large social media platforms like Twitter, but they offer no method for directly testing this hypothesis, as I outline in Chapter 4. Beyond testing the presence of community prototypes, further tenets of the social theory can then be tested and connections between personal attributes and contextual dynamics can be explored.

1.2.7 Community-Aware Centrality

An emerging area of research of relevance to the study of networks and identity is that of community-aware centrality [137, 178]. Traditional centrality measures, such as Pagerank, are concerned with quantifying the importance of nodes in a network [24, 166, 228]. These measures are a function of network structure only. However, it is well understood that community structure is an important feature of real-world networks. Thus, community-aware centrality quantifies each node's importance with respect to the given definition of the network's community structure [72, 73, 85, 178].

This field within Network Science is relevant to the thesis as it can allow for the measurement of how important attributes are relative to communities of users. Existing community-aware centrality measures, however, do not allow for the measurement of contribution (a signed quantity), and do not allow for the measurement of importance with respect to a specific community, instead giving a single score for the full network. Thus, Chapter 4 in part develops modularity vitality to solve these issues, building on the concepts of network vitalities and the key-player problem [24, 25]. Modularity vitality has since been published and has been verified as an important quantity by outside researchers [137, 179].

1.3 Data

This thesis makes use of 5 core datasets throughout its chapters. These each offer unique features meant to best test the methods being developed. Further, the use of multiple datasets results in multiple case studies of the community dynamics under investigation, providing a more robust understanding of the phenomena being examined. Each dataset and its purpose are now explained.

1.3.1 Reopen America

The *Reopen* Twitter dataset was collected from April 1 to June 22 in 2020 to understand the discussion of the reopen America protests [13]. The dataset was collected using a keyword search using terms such as “reopen” and “openup,” including each US state’s abbreviation

appended to the terms, e.g., “reopenNY.” One year after collection, the reply trees were crawled to get a better view of the full conversation. The resulting dataset has 10 million unique tweets across 3.3 million users. At the time of the collection, the Black Lives Matter movement became a major point of discussion and resulted in significant context mixing. The context mixing occurring in this dataset makes it a prime candidate for analysis, both to test the developed method’s ability to distinguish context and to demonstrate its importance. Thus, this dataset is used in all analysis chapters of the thesis, Chapters 2-5.

1.3.2 2020 US Elections

The *Election* Twitter dataset captures online discussion of the most contentious elections in recent US history. False claims of voter fraud and a stolen election were rife on Twitter and are present in this dataset. These claims have since been named “The Big Lie” and have had a lasting impact on American politics¹. The dataset was captured using a keyword-based stream of Twitter’s API from November 2 2020 to November 8 2020. This allowed for the capture of data one day before election night, which was November 3 2020, and one day after major news outlets declared Joe Biden the winner on November 7 2020. The keywords² were selected in order to maximize conversation around the election. This includes general hashtags, campaign hashtags, and mentions of prominent figures in the election such as Trump, Pence, Biden, and Harris. It also includes hashtags relating to anticipated election-related issues, such as the Black Lives Matter movement, US Sanctions on Iran, issues with voting-by-mail, and claims of voter fraud. The collection resulted in 4.5M tweets. Unlike the reopen dataset, there is no competing discussion present. Thus, the election dataset presents the “normal” scenario where keywords search alone provides moderately successful contextualization. Also, this dataset spans a much shorter time period than the Reopen dataset, which can give examples of dynamics occurring on different time scales. Because this dataset offers contrast to the Reopen America dataset, it is also used in all analysis chapters of the thesis, Chapters 2-5.

1.3.3 Ukraine Legislature

The *Ukrainian Legislature* dataset is the record of all Ukrainian legislative votes cast in the 22-month span of the 7th convocation of the Rada. The Ukrainian revolution of 2014 occurs midway through the convocation, drastically changing the political allegiances observed. While this network is extremely different than the data seen on social media, it provides

¹<https://www.npr.org/2022/01/05/1070362852/trump-big-lie-election-jan-6-families>

²Keywords used for data collection: #election2020, #presidential election, #democrats, #republicans, #JoeBiden, #BidenHarris2020, #Biden, #MAGA, #KAG, #VoteByMai, #USPS, #SaveTheUSPS, #voterfraud, #BlackLivesMatter, #BLM, #reopen, #reopenamerica, #IranSanctions, #QAnon, #WWG1WGA, "natural born", @JoeBiden, @realDonaldTrump, @POTUS, @Mike_Pence, @VP, @KamalaHarris, @SenKamalaHarris, @USPS, @CoryGardner, @SenCoryGardner, @Hickenlooper, @Perduesenate, @sendavidperdue, @ossoff, @joniernst, @SenJoniErnst, @GreenfieldIowa, @SenSusanCollins, @SenatorCollins, @SaraGideon, @SteveDaines, @stevebullockmt, @GovernorBullock, @ThomTillis, @SenThomTillis, @CalforNC

an example of a large, ground-truth change in communities to detect. As such, it is used to validate the community dynamics method developed in Chapter 3.

1.3.4 Captain Marvel

The *Captain Marvel* Twitter dataset was originally collected by Babcock and Carley, and aims to capture discussion around the premier of Captain Marvel, Marvel’s first female-led superhero movie [14]. The main keywords used for dataset collection include: #CaptainMarvel, Captain Marvel, Brie Larson, Alita, SJW, Feminazi, #BoycottCaptainMarvel, and #AlitaChallenge. The initial goal of this dataset collection was to study the development of misinformation and counter-narratives around the movie. Because of this, the collection procedure is more complex than that of the other datasets, and we refer to the original paper for those details. This dataset was included for the purpose of demonstrating an implicitly political dataset. While other datasets contains discussion on topics that were overtly political, such as the 2020 election and how the government should handle the pandemic, this dataset contains more of a mixture of political and non-political content. Thus, this dataset potentially provides a different set of communities to observe in Chapter 4.

1.3.5 Coronavirus

The *COVID* Twitter dataset is the largest dataset examined in this Thesis. It was collected using a keyword-based stream of the Coronavirus discussion resulting in 77 million tweets from the following keywords: coronaravirus, coronavirus, wuhan virus, wuhanvirus, 2019nCoV, NCoV, NCoV2019, covid-19, covid19, covid 19. The Twitter API does not allow for retroactive collection of a user’s profile information. Instead, a user’s profile information can only be obtained by direct query or by observation when a user’s tweet enters a collection. This makes tracking the evolution of a collection of a group of users’ attributes over time difficult. It is also effectively impossible to track the evolution of attributes over an unexpected event.

The long-standing collection of the Coronavirus discussion, however, resulted in a longitudinal picture of users’ profiles who were active in the discussion of the virus. The coronavirus discussion was general enough to include users across many different interests. Further, the dataset spans the murder of George Floyd and the subsequent rise of the Black Lives Matter Movement. Thus, this dataset is uniquely positioned to study the dynamics of user attributes at the community level, and to specifically study the adoption or lack of adoption of attributes in support of Black Lives Matter, a highly polarizing issue. This will be studied in Chapter 4.

1.3.6 Specialized News Discussion

As part of the analysis pipeline detailed in Chapter 5, a series of best practices in data collection will be provided. The first goal is to provide a set of procedures that researchers can follow to yield the best results from the tools outlined in the previous chapters. The

second goal of this dataset is to directly demonstrate the robustness of the methods as well. The collection will maximize the conversational connections between Twitter users through the new Conversation Collection feature of Twitter’s V2 API ³. This feature enables the collection of full reply trees, which were previously unobtainable. With more of the conversational structure available, it is expected that this specialized dataset will be contextualized in a clearer way than the other datasets, providing a useful case study to demonstrate the full contextualized analysis pipeline.

Initially, all tweets containing news links from 6 news agencies will be collected within one day. The six agencies have been chosen to represent different types of popular news agencies, which may drive different types of conversations, thereby acting as a robustness test for the analyses. The news agencies are as follows: two direct reporting agencies (Reuters and Associated Press), one American left-leaning (CNN), one American right-leaning (Fox), and two state-sponsored agencies (CGTV and RT). The conversation collection will then be used to obtain the full threads surrounding news-related posts on that day. This will result in large number of conversations talking about different topics from different points of view. This dataset will be collected and used in Chapter 5 which will cover application of all the tools developed in preceding chapters.

1.4 How to Read This Dissertation

This dissertation was intended to be read in a linear fashion, proceeding from one chapter to the next, where each chapter investigate the types of contexts shown in Figure 1.1. However, there are alternatives depending on the interests of the reader. Interactional context is considered in Chapters 2 and 3. Chapter 3 directly builds off of the methods developed in Chapter 2, so if interactional context is of interest to the reader these chapters should be read as a pair. Personal context is considered in Chapter 4, and can be read as a standalone chapter. Chapter 5 gives the full contextualized pipeline, which details how all of the analyses fit together. So, for those interested in applying contextualized network analysis to their own research problems, or those looking for a high-level overview of the work, it could be beneficial to start with Chapter 5 and visiting the previous chapters for more details, as needed.

³https://blog.twitter.com/developer/en_us/topics/tools/2020/introducing_new_twitter_api

Chapter 2

Contextualizing Social Media Conversational Networks

In this chapter, we demonstrate that standard social media analysis mixes many different types of interactions together, even on clean datasets. This introduces tremendous amounts of noise to the derived social networks, and network analysis is notoriously sensitive to noise [26, 66]. We propose two methods of reversing this problem, which not only provides a more accurate view of online communities, but also gives insights into the dynamics of large-scale conversations to be explored in the following chapter. The different types of interactions considered in this chapter are a result of differing *interactional context*, which refers to the data surrounding an interaction between users which gives that interaction meaning. We will use this term interchangeably with *conversational context*, since all of the data we consider are from social media conversations.

Historically, social network analysis has always relied on proper contextualization of social interactions, however this contextualization has typically been done as part of experimental design and data collection [107, 148, 238]. By the time the network analysis takes place, the contextualization has been taken for granted. For offline social networks, contextualization has been traditionally done by scoping the measurement of social interactions within a physical space: a conference, an office, a school, etc. Situating a social network within a single context provides a clean dataset and allows for interpretable analysis.

Consider a network of coworker interactions. Central members in the a network can be seen as information brokers within the office. Including information about how the workers interact outside of the office provides more information but can also muddy the analysis. Adding out-of-office connections to the initial network is likely to affect who the central actors are, what the community structure is, and the general topology of the network. While this denser network encodes more information, it also conflates two types of edges; workers will interact with each-other differently according to where they are. Thus, it is more appropriate to study the *contextualized* networks and the relationship between them. This may include studying changes in centrality and community structure from one context to another. Contextualization not only improves the specificity of the claims that can be made from the original network analysis, but also adds new information about the

relationship between contexts.

This problem is illustrated in Figure 1.2. This simple scenario details 3 different social contexts with one having much different community structure than the others. A standard or decontextualized analysis mixes all of the edges together and hides *all* community structure. Networks analyses are known to be sensitive to data quality, making this an important problem [26, 66].

Methods for contextualizing social networks to date have leveraged simple property of offline networks: people can only be in one place at one time [141]. The result of this obvious fact is that offline social contexts occur in sequences. For example, someone might go to work, then go to a restaurant to meet their friends, and finally return home to their family. This then creates 3 sequential social contexts: interactions among co-workers, friends, and family. Dynamic network analysis methods have leveraged this sequential structure can be leveraged to identify network “states,” effectively contextualizing these interactions.

Online communication is different. Social media platforms such as Twitter are designed for users to engage in vastly different discussions simultaneously¹, ruining the sequential structure of contexts. Without this sequential structure, existing dynamic network approaches are inapplicable.

Differing interactional context for social media data is illustrated in Figure 2.1. In typical models of Twitter networks, one user mentioning another user is modeled as an edge between the two. However, not all mentions are the same; they can differ both in the content and in the information they reveal about the relationship between users. In this example, the first interaction is discussing sports in a way which may indicate a friendly relationship between the users while the second interaction is discussing an election, where there is a power differential and hostile relationship between the users. Typical models collapse this rich information into a simple edge, making these wildly different interactions look equivalent.

Here, we only consider the content of interactions, leaving the implication about the pair’s relationship for future work. Though interactional context can be thought of as the topic at hand, “topic” has a fairly narrow connotation for computer scientists who are familiar with topic models applied to textual data. On social media, the context of an interaction goes beyond words. Users often interact with others using emojis, images, and videos as well.

Researchers attempt to account for this by scoping the data collection to a specific topic or event. On Twitter, the available filters for data collection include keywords, specific users, and geographical bounding boxes. After applying these filters, researchers assume that the data are reasonably contextualized about a certain event or topic. However, a related area of research, story-detection, has demonstrated that multiple events or “stories” have separate discussions occurring even within filtered datasets [203]. We will further show

¹Technically, people can only send one Tweet at a time, so they are actually oscillating between conversations rather than simultaneously being engaged in them. The distinction for online interactions is the time-scale of state changes. Online discussions play-out over hours, while users switch between conversations within minutes. This mismatch in timescales creates the ability for users to be in multiple discussions “simultaneously”

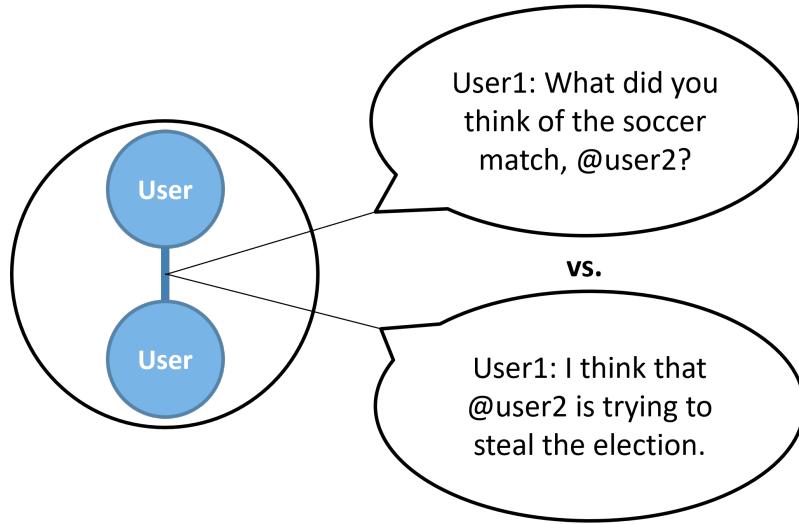


Figure 2.1: An example of two interactions with wildly different interactional contexts.

that contextual mixing is still a problem in keyword filtered datasets. A cursory proof of this issue is given by visualizing the top hashtags in the *Reopen* dataset in Figure 2.2. While many hashtags are related to the reopen discussion and protests, there are also many Black Lives Matter hashtags, indicating that there is a major contextual mixing problem.



Figure 2.2: Hashtag cloud of the Reopen dataset with hashtags sized by their frequency.

Properly contextualizing online conversational networks is critical given their importance within the field of Social Cybersecurity [38, 39]. Online conversational networks are deeply integrated into the methods for studying and understanding information operations [81, 124, 218]. Thus, accurate representations of the conversational networks are necessary to understand the information space.

In the following sections, we propose a simple labeling approach for contextualizing

Twitter networks from first principles. We then propose an unsupervised deep learning approach, which would enable the contextualization of social media conversations without requiring hours of work from human annotators. Different configurations of the model are experimentally compared so that the most expressive model configuration is chosen. Then, the results are validated against our simple labeling approach. Lastly, we demonstrate that contextualization has major impacts on network analysis in two major areas. First the nodesets of contexts are compared, where we show that the most fundamental attribute of networks can have significant differences between contexts. Then, we compare the results of contextualized vs. mixed-context centrality analysis. There, we see the detection of central nodes is severely corrupted when two different conversational contexts are mixed.

For robustness of results, we apply our methods to both the *Election* and the *Reopen* dataset throughout the chapter.

2.1 Related Work

Social connections must occur in the same context for social network analysis to work effectively. What constitutes the “same context” depends on the study. For example, if a study seeks to understand the spread of information in the workplace, the inclusion of connections outside the workplace may be inappropriate. If the study instead was looking to measure epidemic spreading, all interactions are appropriate to include. In many settings, and particularly for offline networks, this is an extremely easy requirement to meet which is easily satisfied though data collection processes such as observing connections in a specific place.

For offline networks, dynamic analysis methods have been developed to detect *sequences* of network states, finding that datasets observed over longer time periods contain multiple contexts [135, 141, 173]. In one example, changes can be observed from how students interact at lunch compared to in the classroom [141, 173]. Students interact differently in the lunch context than they can in the classroom context. In another example, changes are observed in how Ukrainian legislatures cooperate before and after the Euromaidan revolution [135]. An upheaval in socio-political context disrupted friendships and rivalries between politicians. These studies find that the community structure and central actors can be very different from context to context, and that combining contexts leads to an inaccurate representation of the network. Accounting for contexts has also led to improvements in the modeling of processes occurring on the networks [172].

For online social networks, however, contextualization is not an easy task. Two related fields have shown that social media data often contains multiple entangled contexts: topic modeling and story detection. Topic modeling seeks to uncover a selection of different semantic contexts, or “topics” which occur within a collection of documents [20]. Traditional topic models such as LDA are poorly suited for the extremely short documents in Twitter data, leading to topic models specifically designed for short texts [44, 104, 243]. Alvarez-Melis and Savesci found Tweets can aggregate information from their conversational context to improve topic representation [9]. Other topic detection models have been developed which specifically leverage the hashtag feature of Twitter data to obtain topics

[61, 136, 226]. Methods differ, but all of these works successfully demonstrate the presence of multiple semantic contexts in Twitter conversations.

Topic modeling demonstrates that entirely different things may be discussed in the same Twitter dataset, while story-detection shows that different contexts can occur even within very similar topics. Story detection seeks to uncover “stories” or discussions tied to specific events [7, 8, 175, 203]. First-story detection and event-detection are very related, as they seek to identify the first Tweets breaking the news of a story developing, compared to more general story-detection, which detects all the Tweets in the discussion of that story [165, 175, 223]. In any case, detected stories are separate contexts which could otherwise be considered the same topic. For example, story detection applied to Donald Trump’s twitter timeline can distinguish within-party arguments from between-party arguments, which both belong to the topic of federal US politics [56]. Another example applies story detection to the Twitter discussion following the police killing of Michael Brown [203]. Here, fine distinctions of context are made, such as the difference in discussion of the police-lead smear campaign against Michael Brown from the discussion of the robbery that Brown committed early in the day of the shooting. This is to say that both topic modeling and story detection develop methods of uncovering discrete conversational contexts on social media, and thereby demonstrate that these contexts exist. These works do not, however, investigate the implications of this finding for social network analytics.

While dynamic analysis can leverage the sequential structure of human movement in offline networks, this is not possible with online networks. The studies in topic modeling and story detection show that conversations within these conflicts can occur simultaneously, with users rapidly switching between contexts. And while methods from topic modeling and story detection can be used to uncover conversational contexts, existing methods don’t typically leverage all of the available indicators of context simultaneously: Tweet text, hashtags, URLs, and the conversational graph.

Advancements in graph neural networks enable us to develop a new architecture for unsupervised Tweet representation which leverages all of the available data and places Tweets in a continuous space. Older methods of unsupervised node representation relied on random walks or “surfs” to obtain local information which can be encoded in node vectors [37, 82, 174]. These methods do not rely on node features to obtain their representations, in contrast to the graph convolutional networks that are typically used in the semi-supervised or supervised setting [87, 118]. Node features are necessary for Tweet representation because they are used to represent the actual contents of a tweet, the tweet’s text.

Methods leveraging node features have been used applied to model social media *users* in a number of supervised settings, including the detection of hateful users, and the prediction of locations. [55, 168, 185]. Perhaps the closest related model to ours is that of Nguyen et al., who used unsupervised embedding methods such as BigGraph for users, hashtags, and URLs, before combining them in a supervised Retweet prediction model [126, 161].

While models leveraging node features have been developed for social media, a mechanism for training them in an unsupervised manner was not available. Deep Graph Infomax (DGI) filled this gap by outlining an unsupervised training procedure for feature-leveraging approaches through the principle of mutual information [221]. Similar to Structural Deep

Network Embedding (SDNE), DGI derives an objective function in the unsupervised setting so that the architecture has something to optimize [224].

Because DGI is a methodology for training, the specific architecture for node embedding is customizable, similar to the HARP procedure [41]. Later in this chapter, we develop a custom GCN-based architecture for representing Tweets, which uses the conversational context, hashtags, and URLs, which is then trained with DGI on a real dataset. We use the obtained Tweet representations to contextualize user-to-user interactions and demonstrate the importance of contextualized network analysis.

2.2 Simple Labeling Approach

Before diving into a complicated model with many parameters, we consider a simple initial approach. First, content is manually categorized into different contexts. Next, labels are propagated to Tweets referencing the label content. Label propagation continues until no more Tweets can be labeled.

Manual content annotation begins with URLs. URLs contain rich information about the makeup of large Twitter discussions, as they often report on the top events that are being discussed. Annotation of URLs will give the analyst a quick understanding of what the major conversational contexts are in the data. We assume that any Tweet linking a URL is a part of the same conversational context as that URL. Based on this assumption, URL labels are propagated to all Tweets that use the labeled URLs. For example, if a URL is labeled as part of the “Reopen Protest” conversation, a Tweet linking to that URL is also labeled as part of the “Reopen Protest” conversation. This assumption enables us to quickly label contexts across user communities. To most efficiently label data, URLs are sorted based on the number of times they appear in the dataset. As the annotator goes through the sorted list, dataset coverage will be increased, though there is diminishing returns since most datasets have many URLs that are posted by few users. Tweets which post URLs with conflicting labels are not considered, though they are flagged for the annotator as an indicator that some labels may need to be reconsidered.

Next, the dataset’s top Tweets are annotated. While URLs capture much of the discussion, viral Tweets are essential to label to truly understand a dataset. For example, President Donald Trump’s Tweets in support of the Reopen Protests generated a lot of the discussion surrounding the protests. While there are URLs linking to news stories that discuss his Tweets, the Tweets themselves are in the dataset and should be labeled. Similar to URLs, Tweets are sorted according to the number of likes or Retweets they received, and are then labeled.

The question of *how* annotators should label data is unfortunately subjective. However, the process is not too dissimilar from interview coding, and so we can draw from the best-practices available there [230]. These seem like separate problems, however both consider data where users are free to enter their responses how they see fit and the outcome categories are not predefined. In our case, we know that categories can generally be thought of as a conversation.

So, like in interview coding, the annotation should be done in multiple phases. In the

first phase, content is annotated in the most specific form. For example, an article about the implications of Trump tweeting “LIBERATE MINNESOTA!” is taken to be a conversation about just that. Next, an article about the implications of his Tweet “LIBERATE MICHIGAN!” will be taken as a separate annotation. The process continues. In the second phase, the annotator reviews their categories and looks for patterns to combine and resolve. In our example, clearly the two Tweet contexts are related, so they can be merged into a general “Trump’s Liberate Tweets” category. The process continues until the annotator feels that the right balance is struck between the expressiveness of the categories and their ability to combine related conversational contexts. This criteria’s vagueness feels unsatisfactory, however conversational context’s hierarchical nature makes it difficult to do much better. Researchers working on sub-story detection recognize that no matter how a “story” is defined, sub-stories can be derived [203]. It is up to the analyst to draw these lines and deal with the implications. This issue provides some motivation for the deep learning approach detailed in Section 2.3, which seeks to automatically discover contexts, leaving the decision making up to the clustering algorithm.

Lastly, the hand-labeled and URL-labeled Tweets are propagated through the conversational network. As a reminder, the conversational network connects Tweets to other Tweets when they Retweet, quote, or reply to them. Thus, if we begin with a single labeled Tweet, in the next step, all the Tweets that Retweet, quote or reply to our labeled Tweet will obtain the same label. In the next step, the propagation will continue from all the labeled Tweets. Again, Tweets with conflicting labels are permanently not labeled. The propagation terminates when labeled Tweets have no remaining neighbors to propagate to.

Label propagation is not guaranteed to reach all nodes, so in most cases a portion of the dataset remains unlabeled. To see why this is the case, consider a single Tweet which does not include a URL, and is never Retweeted, quoted, or replied to. This Tweet is an isolate in the conversational graph, so the only way for it to receive a label is if it was manually annotated, which is unlikely since it had such little engagement. More generally, if a component in the conversational graph does not have any seed labels, all of its nodes will remain unlabeled. Large but disconnected components are not common on Twitter, where linking Tweets is pervasive and a giant component forms quickly. However, it is possible when there are extremely isolated communities or there are communities with structural isolation such as those that speak different languages. In those cases, more sophisticated annotation schemes may be required than the simple rank-and-annotate method detailed here.

The approach assumes that interacting Tweets (Retweets, quotes or replies) are part of the same context as the initial Tweets. This assumption is based on the inherent nature of a conversation, where one party responds to another. However, over time conversational drift (or topic drift) may be a problem [95]. Drift occurs between interlocutors when interactions slightly change the topic of discussion. The small changes add up over the course of many interactions, leading to an entirely different topic of discussion. This concept is familiar to those of us whose conversations with relatives, no matter the starting point, end in a discussion about politics. Label propagation is unable to account for drift unless one of the new-context-interactions is in the set of manual annotations. However, empirical analysis of our datasets show that they are shallow. In the *Reopen* dataset, for example, 90% of

Tweets are within 2-hops from root Tweets, or those which post original content not linking to other Tweets. Thus, there are not enough back-and-forth interactions in these shallow conversations for conversational drift to be a problem, so our propagation is valid.

2.2.1 Interactional Contexts in the Reopen Dataset

For the analysis of the *Reopen* dataset, the 150 URLs that were Tweeted the most and 200 Tweets with the most likes² were hand annotated according to their conversational context. President Donald Trump’s Tweets garnered especially high interaction, so his 25 Tweets with the most likes were also annotated, including those which fell into the top 200. This procedure resulted in 39 conversational contexts, the top 5 of which were *Petitions*, *Liberate Tweets*, *Lowes Donation*, *Reopen Strategy*, and *Black Lives Matter*, in terms of number of Tweets labeled. The explanations of all contexts are provided in Appendix A. Through this exercise alone it is clear that there were multiple conversations mixed together in the data that need to be filtered out to perform proper network analysis. Finally, the label propagation method was applied resulting in 2.1 million labeled Tweets.

2.2.2 Interactional Contexts in the Election Dataset

For the analysis of the *Election* dataset, the 100 URLs that were Tweeted the most and 100 Tweets with the most likes were hand annotated according to their conversational context. This procedure resulted in 38 conversational contexts, the top 5 of which were *Claims of Fraud*, *Spam*, *Election Updates Biden Campaign* and *Trump Campaign*. The explanations of all contexts for this dataset are also provided in Appendix A. Again, there are clearly distinct contexts present in the data. Finally, the label propagation method was applied resulting in 756k labeled Tweets.

2.3 A Deep Learning Approach

While the previous method results in interpretable labels derived from first principles, there is a substantial burden on the annotator in terms of the time and cognitive load that manually uncovering contexts requires. An automated approach could relieve this burden by categorizing Tweets without human input. Automation also could remove the ambiguity of decide how content falls within closely related contexts.

While there are many approaches that can be taken to automating this problem, a deep learning approach has some distinct benefits. To be clear, a deep learning approach to this problem would learn a vector representation of content (e.g., Tweets), which can then be clustered into discrete contexts. The primary benefit is that a deep learning approach has the power to acknowledge that contexts can be closely related. Because content is represented in a vector space comparisons between contexts can be quantified through a distance function. A secondary benefit of a deep learning approach is that the methodology

²Sorting by Retweets gave nearly the same ranking, though Tweets are liked more than they are Retweeted, making them a more sensitive measure.

provided in this section can be used as deep learning approaches continue to advance. Thus, we can piggy-back off of advancements in deep learning to improve our representation of online conversational communities.

It should be noted that there is no free lunch here. Automation in the discovery of contexts does not free analysts from manual investigation into what the uncovered contexts represent. However, the task of interpretation is more straightforward than that of annotation and we provide some simple tools to help analysts quickly interpret contexts.

In the following subsections we will introduce a deep learning approach to automatically contextualize Twitter data.

2.3.1 Data Cleaning

Tweet text was cleaned by first removing all URLs, hashtags, and mentions. Next, punctuation was removed. Finally, text was tokenized in preparation for the text embedding discussed in the Methodology section.

The procedure for URL normalization was as follows. First, text before the domain name was removed. Next, URL parameters were removed for links with domains other than “facebook”, “google”, and “youtube.” These parameters commonly store information about the user who shared the link, among other things. The presence of these parameters prevents direct matching between URLs. For “facebook”, “google”, and “youtube,” however, these parameters are used to point to the actual destination, so cannot be removed. “Amp” links were converted to non-amp links. Lastly, youtube.com and yout.be links were all converted to the yout.be format.

All links to twitter.com were not considered to be typical URLs, as they are either links to media or quotes of other Tweets. Links to media were not included, while the metadata from quote-links was used to add the appropriate quote-edges in the tweet-Tweet network discussed below. Hashtags were lower-cased, as case does not affect their functionality.

2.3.2 Heterogeneous Network Construction

The presented approach relies on building a *heterogeneous* conversational network from the data and using a deep learning approach to represent all of the nodes. Clearly, we want to include Tweets as they are what we are trying to contextualize. Next, we know that URLs are powerful differentiators of context, so they are also included. Lastly, hashtags are pervasive on Twitter, allowing users to connect related Tweets [61, 136, 226, 236]. Hashtags were too general to be useful in the manual annotation step, however they are a useful way of collecting related Tweets, so they are also included in the network.

Now, we construct our heterogeneous Twitter network with three node types (tweet, hashtag, and URL) and three edge types (tweet-URL, tweet-hashtag, and tweet-tweet). When a URL or hashtag is used in a tweet, an *undirected* edge is drawn between them. The selection of an undirected edge allows for URLs (and hashtags) to aggregate information from all the Tweets they appear in, while allowing Tweets to aggregate information from the URLs (and hashtags) they contain.

The third relationship, tweet-tweet, occurs through replies or quotes. While these are slightly different operations, they both create the effect of continuing the conversation with a new Tweet connected to the original. Edges between Tweets can be modeled as directed *or* undirected, as a setting within the model. A directed edge allows the reply or quote’s representation to be affected by the original tweet’s representation while keeping the original tweet’s representation isolated. This is an intuitive modeling approach; however, modeling this relation with an undirected edge allows for base Tweets to obtain some context, which can push similar but disconnected conversations closer together. The two approaches (directed or undirected), are tested and quantitatively compared in Section 2.3.4.

Retweets are simply copies of Tweets, so they will provide no additional information from a tweet-representation point of view. Worse, they are such a large fraction of the dataset that they could have adverse effects on the training process. Instead, we give Retweets the same representation as their original Tweet. Thus, Retweets will always be considered in the same context as the original tweet.

2.3.3 Deep Tweet Infomax

This sub-section develops the deep learning architecture used to automatically contextualize Tweets.

Initial Tweet Embedding

Graph convolutional networks require some form of node-features. We derive features for Tweets using the Tweet’s text. To limit the scope of analysis to our proposed architecture and to enable the use of multi-language text embedding, we used the pre-trained³ and language-aligned vectors trained using fastText on the Wikipedia corpus [23, 111]. The use of language-aligned vectors allows us to place similar Tweets in the same discussion, even if they are tweeting in different languages.

We rely on the Twitter language detection output for the classification of Tweet language. Many Tweets, however, do not have an available language label. This often occurs when Tweets do not have text, but instead only have URLs, emojis, images, and sometimes hashtags. In our case, 15.6% of Tweets in the dataset do not have an available label, and therefore cannot be embedded with this approach. We will revisit these Tweets later.

For each Tweet with a label, we perform a normalized tf-idf (term frequency \times inverse document frequency) weighting of the fastText word vectors to obtain a 300-dimensional tweet-text embedding:

$${}^l\mathbf{v}_i = \frac{1}{\sum_{t \in {}^lD} w_{i,D,t}^p} \sum_{t \in {}^lD} w_{i,D,t}^p \mathbf{v}_t \quad (2.1)$$

$$w_{i,D,t} = \frac{c_{i,D,t}}{\sum_{t \in {}^lD} c_{i,D,t}} \log \left(\frac{1}{\sum_{d \in {}^l\mathfrak{D}} t \in d} \right) \quad (2.2)$$

³<https://fasttext.cc/docs/en/aligned-vectors.html>

where left-superscript, l , indicates the language, \mathbf{v}_i is the vector representation of node i , D is the document for node i containing all its associated tokens, t , $c_{D,t}$ indicates the counts of token t in document D , and ${}^l\mathfrak{D}$ indicates the set of all documents in language l . A power-term p is introduced and set to $p = 1$ to retain the classic tf-idf weighting scheme. Lastly, the final vectors are L2-normalized, since they will be compared using cosine similarity: $\mathbf{v}_i = \|\mathbf{v}_i\|_2$

We use this procedure to embed Tweets in Arabic, English, French, German, Hebrew, Italian, Portuguese, Romanian, Russian, Spanish, and Turkish, covering over 95% of the reachable Tweets.

Finally, we use feature propagation to obtain a feature vector for the remaining Tweets [191]. Feature propagation holds known feature vectors fixed while iterative updating unknown feature vectors. In each iteration, each node with an unknown feature vector updates its vector by taking the average features of its neighbors. Nodes with unknown features which have not been reached by the propagation are not counted in the update step. After few iterations, all features converge. Rossi et al. demonstrate that this approach yields good results in downstream tasks such as node classification even in the face of extreme missing data, when 99% of nodes are featureless. The task of filling in features for approximately 15% of nodes is much less formidable. Feature vectors converged within 40 iterations on our datasets.

Initial Hashtag and URL Embedding

Now that Tweets have an initial vector representation, we need initial representations of hashtags and URLs. A straightforward approach would be to allow each hashtag or URL to learn its representation from all of the Tweets that use it. This could be done using a graph convolution and would nicely fit into the rest of the architecture, which will also leverage graph convolution. However, the structure of the heterogeneous conversational graph inhibits this strategy from working well.

The problem is that the average hashtag or URL in the dataset is used by many Tweets. So many tweets, in fact, that the majority of them are *not* informative as to what the node’s representation should be. Out of the potentially thousands of tweets that use a hashtag, for example, the top 10 or so would likely be the most useful to inform the representation. However, graph convolutions aggregate from the entirety of the nodes neighborhood. Graph attention recognizes this problem and attempts to solve it by taking a weighted aggregation. However, even this is not enough with the extreme node degrees of hashtags and URLs.

We will now illustrate this problem with a text-based approach. One way of understanding a URL or hashtag is to aggregate all of the tweets that use it into a single document. These aggregated documents have been proved helpful for topic modeling on Twitter [9, 204]. With these documents, use a tf-idf-weighted word vectors, as was done in the previous section. This is a very similar procedure to that of using graph-convolution on the Tweets, however this procedure should provide better results because it allows us to aggregate word-level information, and the tf-idf is a well validated method of determining weights in a non-learnable way.

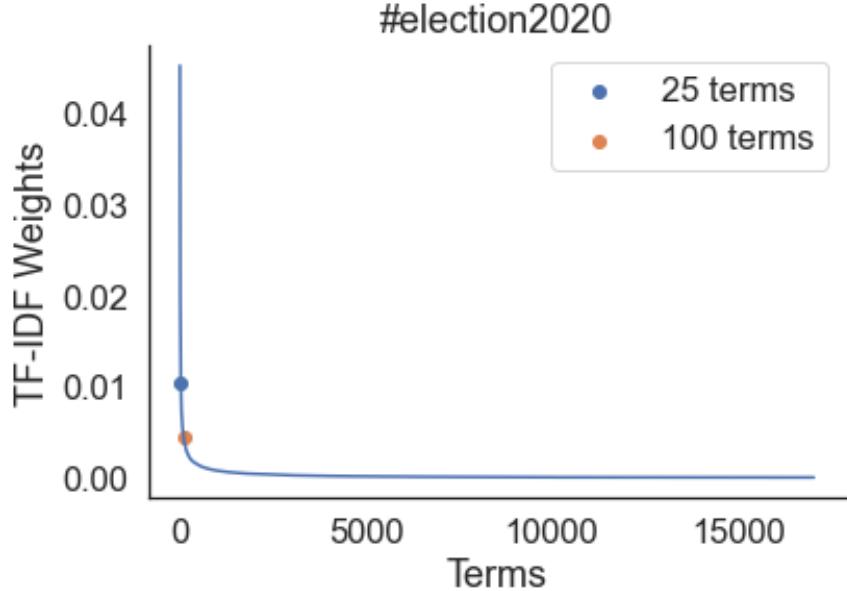


Figure 2.3: The distribution of tf-idf weights for `#election2020` in the *Election* Dataset is shown. The top 25 and 100 terms are marked for reference.

To investigate this weighting scheme, Figure 2.3 shows the tf-idf distribution of terms for `#election2020`, a popular hashtag in the *Election* dataset. We see that that only the tf-idf weights drop dramatically, with only around 25 terms having high weights. However, there is a very long tail with over 17000 terms having near-zero weights. The crux of the problem is that the top 25 terms have a much higher score than the rest, but there are so greatly outnumbered that a weighted sum washes them out. The effect is illustrated by showing the cumulative weights in Figure 2.4.

Since only the top 25 terms have a high weight, we would expect them to make up a high percentage of the final representation. This percentage is shown as the height in Figure 2.4. We see that using the normal weighting, the top 25 terms make up less than 20% of the representation. Essentially, this approach is learning a good representation (with the top 25 terms), and then averaging that with noise where the noise vector gets 4 times more weight! So, while the classic tf-idf in Figure 2.3 looks to be heavily skewed, it is actually not skewed enough. This is related but distinct from the oversmoothing problem in graph convolutional networks where nodes approach the same representation as depth is increased [40, 129, 233]. Here, nodes are approaching the same representation because of the high-degrees.

We introduce more skew by raising the weights to a higher power, $p > 1$ in Equation 2.1. The higher the power, the higher the skew, because the small terms in the distributions tail will be shrunk most aggressively. The affect is illustrated by the other lines in Figure 2.4. We assess that $p = 3$ strikes the best balance of weighting the top 25 terms heavily while not entirely removing the remaining terms.

Hashtags and URLs can be Tweeted in multiple languages. To account for this, separate

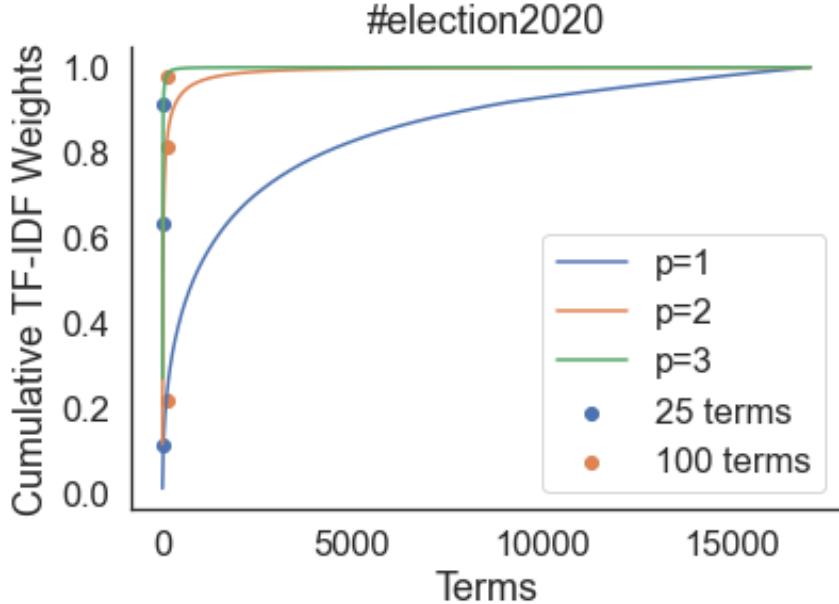


Figure 2.4: The cumulative distribution of tf-idf weights for `#election2020` in the *Election* Dataset is shown at different powers. The top 25 and 100 terms are marked for reference.

tf-idf weighted vectors are obtained for each language, and the average is taken.

The aggregation problem outlined in this subsection is a due to the degree distribution of the underlying network. So any further attempts to learn aggregations for these nodes will run afoul, even though we are beginning with strong representations. Thus, we fix the hashtag and URL representations during training.

Heterogeneous Graph Neural Network Design

Now that initial feature representations are given for all nodes, we propose an unsupervised approach for Tweet representation. The flow of information in 1 step of the graph neural network architecture can be seen in Figure 2.5. As previously described, hashtags and URLs, obtain information by aggregating from the Tweets that they are used in. A Tweet aggregates information from the hashtags and URLs that it uses, as well as all of the Tweets that it is connected to via replies, or quotes. This model is trained using Deep Graph Infomax, leading to the informal name of our approach of Deep Tweet Infomax (DTI). The architecture will now be described in detail.

Let t , u , and h represent nodes of the type tweet, URL, and hashtag, respectively. They will be indexed using subscripts, e.g., t_i corresponds to the i^{th} Tweet. Feature vectors are represented with the letter x , using subscripts to indicate the corresponding node and superscripts to indicate the layer. For example, $x_{t_i}^0$, represents the 0^{th} -layer vector (otherwise known as the feature vector) for the i^{th} Tweet. We will make use of a neighborhood function \mathcal{N} , which takes in a node and returns the set of its neighbors. Subscripts of the neighborhood function allow for the return of only a specific type of

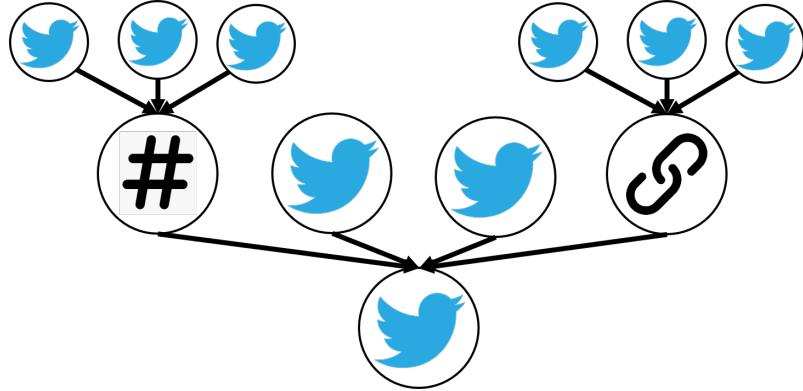


Figure 2.5: The flow of information in one layer of DTI. The Tweet being represented is shown at the bottom. It contains a hashtag, a URL, and is connected to two Tweets via reply or mention. First, the Hashtag and URL obtain a representation by aggregating from Tweets that the Hashtag and URL appear in, respectively. Then, the target Tweet aggregates information from all its neighbors, the Tweets, the hashtag, and the URL, to obtain its representation.

neighbor. For example, $\mathcal{N}_u(t_i)$ returns all of the URLs connected to the i^{th} Tweet.

Tweets aggregate from their heterogeneous neighborhoods. Separate aggregation functions are learned for the Tweets, hashtags, and URLs that a Tweet is connected to, which are then averaged, an activation function is applied, and the L2 norm is taken as seen in Equation 2.3, where AGG is a potentially learnable aggregation function, and σ is an activation function. Although that Equation 2.3 appears to weight all terms equally, weighting is actually learned, it is just absorbed in the aggregation functions.

$$\begin{aligned} x_{t_i}^1 = & \|\sigma\left(\frac{1}{3}(\text{AGG}(\{x_{h_i}^0, \forall h_i \in \mathcal{N}_h(t_i)\})\right. \\ & + \text{AGG}(\{x_{u_i}^0, \forall u_i \in \mathcal{N}_u(t_i)\}) \\ & \left. + \text{AGG}(\{x_{t_i}^0, \forall t_i \in \mathcal{N}_t(t_i) \cup \{t_i\}\})\right)\|_2 \end{aligned} \quad (2.3)$$

The process thus far defines the network over which features are passed, and the order in which to pass them. The selection of the aggregation function, AGG, is the main topic of debate within graph neural network research. As research develops on this front, AGG, can be substituted for the new state-of-the-art aggregation schemes. Here, we employ the GraphSAGE aggregation, which is the initial aggregation scheme applied in the Deep Graph Infomax work [87]. This aggregation scheme is detailed for the tweet-to-Tweet relationship in Equation 2.4, where \mathbf{W} are trainable weight matrices, and \mathbf{b} is a trainable bias vector.

$$x_{t_i}^1 = \mathbf{W}_1 x_{t_i}^0 + \frac{1}{|\mathcal{N}(t_i)|} \sum_{t_j \in \mathcal{N}(t_i)} \mathbf{W}_2 x_{t_j}^0 + \mathbf{b} \quad (2.4)$$

GraphSAGE is a relatively simplistic aggregation scheme, where all neighbors are treated equally. More recent aggregation schemes add attention, which allows for a weighted average of neighbors. Graph attention, initially developed by Veličković et al and improved by Brody, Alon, and Yahav is a popular alternative which may add expressive power[30, 220]. We add this comparison to our experiments in model selection.

Finally, we must select a nonlinear activation function. Again following the original Deep Graph Infomax work, we use the PReLU, activation function [91].

The process up to here details a single-layer of the architecture. Tweets will only obtain information from 1-hop away, and hashtags and URLs will only receive information from the initial feature vectors. Stacking these layers enables further information spread between Tweets, URLs, and hashtags. Here, we stack three of these layers. Classically, a depth of 3 is very shallow. However, in our case, Tweet networks themselves are shallow. The vast majority of Twitter replies are replies to a base-tweet, rather than replies to replies.

Lastly, we need to ensure that hashtags and URLs are embedded in the same space as tweets. This can be accomplished by applying the linear transform in the aggregation function to each of the hashtags in URLs. For example, if GraphSage is the aggregation function we have:

$$\widetilde{x_{h_i}} = \|\mathbf{W}_2 x_{h_i} + \mathbf{b}\|_2 \quad (2.5)$$

where $\widetilde{x_{h_i}}$ is the transformed representation and \mathbf{W}_2 and \mathbf{b} are given in Equation 2.4.

The architecture has now been fully defined. To train this architecture, we use Deep Graph Infomax (DGI), an approach for learning unsupervised node representations by maximizing mutual information between patch representations and corresponding high-level summaries of graphs [221]. We note that a version of DGI has been developed specifically for heterogeneous networks [183]. Because of our focus on Tweet representations and the lack of features available for URLs and hashtags, we proceed with the original formulation of DGI.

The DGI training process involves four steps. First, a normal forward pass on the data is performed, giving Tweet representations, \mathbf{x}_t . Next, a readout function is applied to give a graph-level summary vector, \mathbf{s} . Velickovic et al. applied a sigmoid function to a simple averaging of the node vectors but suggest that more sophisticated methods such as the Set2Vec method developed by Vinyals et al. could perform better on larger graphs [222]. We test both. In the case of Set2Vec we use 5 processing steps: $\mathbf{s} = \sigma(\text{Set2Vec}(\{x_{t_i} \forall t_i\}))$, where σ is the sigmoid function. Third, a forward pass is performed on corrupted data, giving corrupted Tweet representations, $\tilde{\mathbf{x}}_t$. We use the same corruption function as the original work, a shuffling of the Tweet features while keep edges intact. Finally, to classify Tweets as corrupted or not a scoring function is given as $d_{t_i} = \sigma(x_{t_i}^T \mathbf{W} s)$, where \mathbf{W} is a trainable scoring matrix and σ is the sigmoid function, providing a score between 0 and 1. Binary cross entropy loss was used on the score, d , and the label (corrupted or not) to train the model.

The model was implemented using the PyG library [63]. All hidden and output layer dimensions were set to 300 to match the FastText embeddings. The model was trained using minibatches of size 2500 due to limited GPU memory. PyG’s “NeighborLoader” was

used to handle the neighborhood sampling within minibatches, where 20 neighbors of each edge-type were sampled for 3 iterations. The ADAM optimizer was used during gradient descent with an initial learning rate of $\alpha = 0.00001$ for 25 epochs with early stopping⁴ [117].

Clustering

Once Tweets are represented in a continuous space through DTI, they can be clustered with a variety of clustering algorithms. Tweet clusters, then, are the discrete contexts that conversational networks can be studied within.

For extremely large datasets, like the ones we use here, k-Means clustering is one of the only available approaches [134]. Given a set number of clusters, k , the algorithm partitions the data based on their distance to k reference points, which are updated throughout the process. Though the algorithm is extremely efficient, the number of clusters must be manually selected. The “elbow method” heuristic is a useful way of selecting this number, wherein the mean cluster distance is plotted against the number of selected clusters, and the “elbow” or point of diminishing returns is selected [211]. The number of clusters can also be determined externally, such as if the clusters were previously hand labeled as is our case.

For smaller datasets, density-based approaches like DBSCAN can automatically determine the number of clusters [58]. Despite criticism of DBSCAN [67], the algorithm has proved flexible enough to work well under many scenarios, provided the parameters are well-selected [193]. Even for medium-sized Twitter datasets it can be extremely costly to run density-based clustering many times to select appropriate parameters, as was done in Schubert et al. To minimize the need for parameter tuning when clustering, the hierarchical version of DBSCAN, HDBSCAN, which requires less parameters, is a reasonable alternative [144, 145].

Model Selection

We have detailed 4 levels of design choices: directed vs. undirected edges, GraphSage vs. GAT aggregation, and mean vs. set2vec summarization. This leads to 8 possible configurations for our model. We evaluate these configurations based on their ability to capture the relationships in the heterogeneous conversation network. This ability can be quantitatively evaluated by first calculating the cosine similarity of neighbors in the network. Then, non-edge pairs are sampled and the cosine similarity of these pairs is calculated. Finally, the average difference is taken from edge pair similarity and non-edge pair similarity; the higher the difference the better the model. This captures our intuition that pairs of nodes that are connected in the network should have more similar relationships than those which are not connected. For example, a tweet should have a similar relationship to a hashtag that is in the tweet compared with a hashtag that is not in the tweet. This is obviously true for hashtags and URLs. This is less clear for Tweets,

⁴For reference, the model trained in about 12 hours on an Intel E5-2687W v3 CPU.

due to the prevalence of spam and broadcasting⁵ neighboring tweets may not need to have such a similar relationship [62, 189]. However, it is no doubt still useful for neighboring Tweets to have a more similar representation than non-neighboring Tweets. So we proceed searching for model configurations that score highly across nodesets.

	S-D-M	S-D-S	S-U-M	S-U-S	A-D-M	A-D-S	A-U-M	A-U-S
Tweet	0.013	-0.002	0.278	0.006	0.599	0.273	0.293	0.192
Hashtag	0.095	0.065	0.110	0.067	0.084	0.086	0.081	0.094
URL	0.248	0.152	0.249	0.158	0.216	0.215	0.185	0.223

Table 2.1: The model selection results are shown for the *Election* Dataset. For each model configuration, the mean difference the cosine similarity of edges and non-edges are shown for each nodeset. Higher numbers are better. The best results are emboldened. The configuration keys are as follows: S is GraphSage, A is GraphAttention, D is Directed, U is Undirected, M is Mean, and S is Set2Set.

	S-D-M	S-D-S	S-U-M	S-U-S	A-D-M	A-D-S	A-U-M	A-U-S
Tweet	-0.005	0.005	0.384	0.152	0.432	0.197	0.397	0.213
Hashtag	0.115	0.101	0.160	0.101	0.237	0.147	0.131	0.130
URL	0.187	0.175	0.311	0.173	0.330	0.264	0.244	0.221

Table 2.2: The model selection results are shown for the *Reopen* Dataset. For each model configuration, the mean difference the cosine similarity of edges and non-edges are shown for each nodeset. Higher numbers are better. The best results are emboldened. The configuration keys are as follows: S is GraphSage, A is GraphAttention, D is Directed, U is Undirected, M is Mean, and S is Set2Set.

The results are given for the *Election* and *Reopen* datasets in Table 2.1 and Table 2.2, respectively. Across model nodesets, configuration and datasets, mean summarization outperforms Set2Set summarization. Next, GAT outperforms GraphSage in nearly all cases. The choice of directed or undirected edges is less clear. When considering the GraphSage models, the choice has little affect on the hashtag and URL edges, but has a large affect on Tweet edges, where undirected gives better results. For GraphAttention, however, directed edges provide better results. Perhaps GraphAttention’s additional expressive power makes it capable of deriving strong representation from directed edges. Thus, we continue with GraphAttention, mean summarization. The full distributions of similarity scores for our selected model are shown in Figure 2.6.

2.3.4 Validation

Moving forward with the model selected in the previous section, we further validate our approach on the *Election* dataset in two steps. First, we use a simplistic data annotation

⁵Broadcasting is where users latch on to a popular post to gain views on their mostly unrelated post

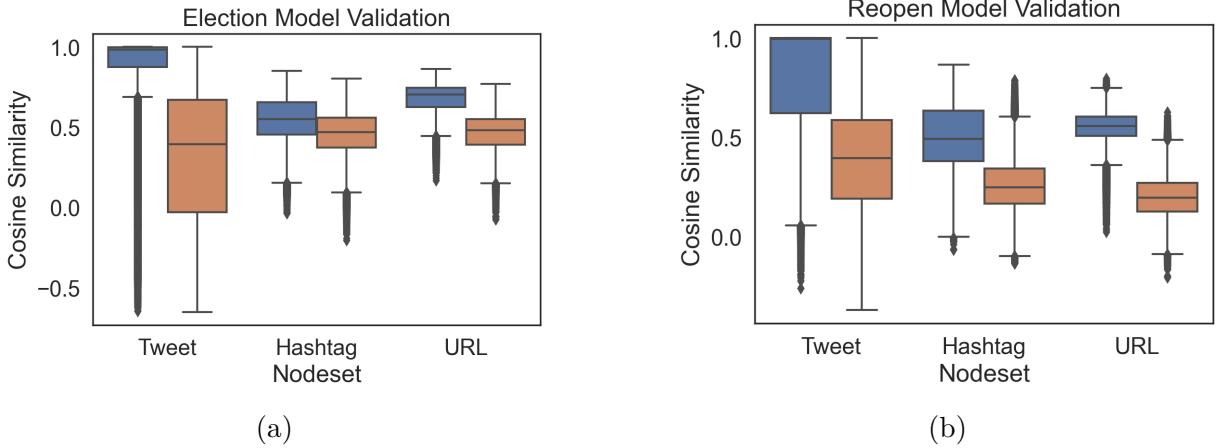


Figure 2.6: Cosine Similarity for edge-pairs (Blue) and non-edge-pairs (Orange) for the *Election* and *Reopen* Datasets using the trained Graph Attention model with directed edges and mean-summarization.

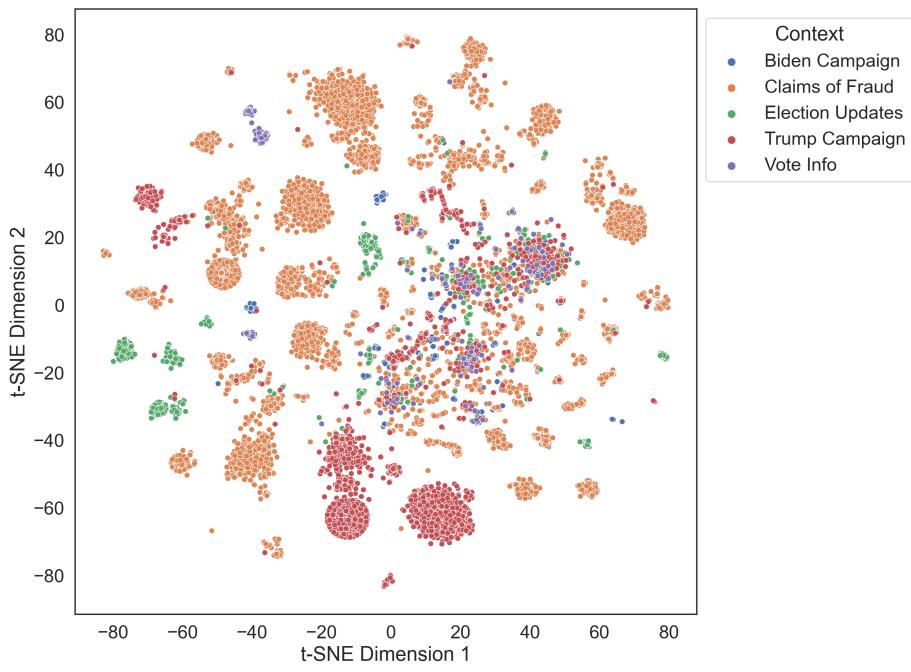
scheme and see how 5 categories of Tweets fall within the embedded space. We find that the clusters in the tweet-embedding space are well-correlated with the annotated groups. Second, we detail some of the URL and hashtag’s nearest-neighbors to demonstrate that intermediate steps within the model are working.

Cross-Validation with Simple Annotation Method

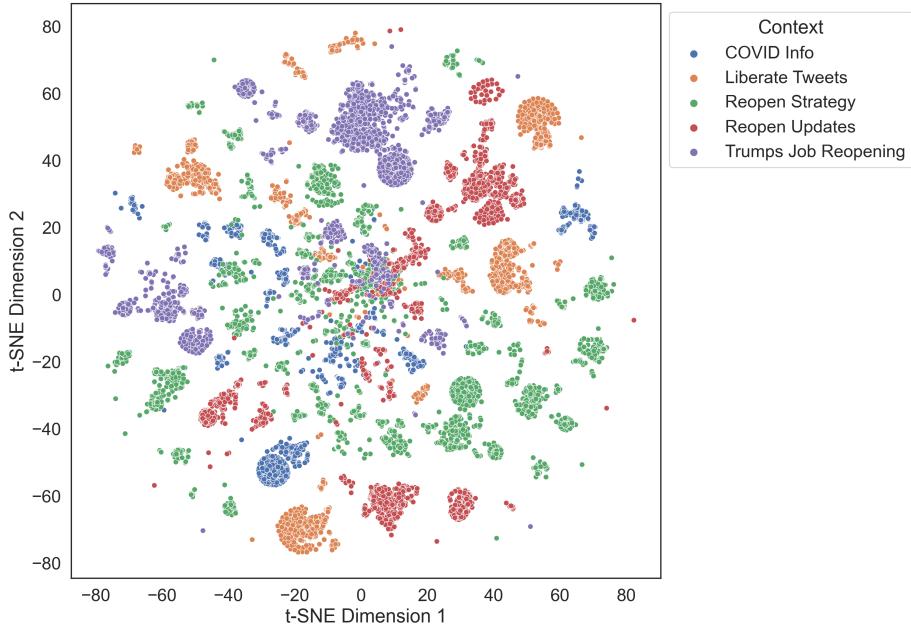
The simple approach that we outlined earlier was built on first principles, validating the results. Therefore, we test the validity of the unsupervised approach by testing its correlation with the simple label propagation. The intuition is that Tweets falling under the same conversational context should, on average, be near to each other in the embedding space. To visually inspect distance in the embedding space, the embeddings were projected into 2-dimensions using t-SNE [219].

Each Tweet with a label-propagation label was plotted in the embedding space as a dot and was colored by its label. The results are shown in Figure 2.7. For a baseline comparison, the initial untrained embeddings are shown in Figure 2.8, as these embeddings only accounted for Tweet text.

The text embedding in Figure 2.8, is similar to Sia et al.’s approach to topic modeling and is used as a baseline [198]. We observe that the text embedding is unable to recover the conversational contexts we set out to find. This is likely due to the facts that these contexts have similar word distribution, and that a text-only approach cannot leverage replies, hashtags, or URLs. Next, we observe in Figure 2.7 that there are a number of well-formed Tweet clusters which correspond to different conversational contexts. We observe that some of these clusters form tight balls, almost perfect circles in the embedded space. Investigation into these regions finds that this occurs when many Tweets reply or quote a popular Tweet. The original Tweet anchors the conversation, while the additional information in replies or quotes move these secondary quotes in different directions within

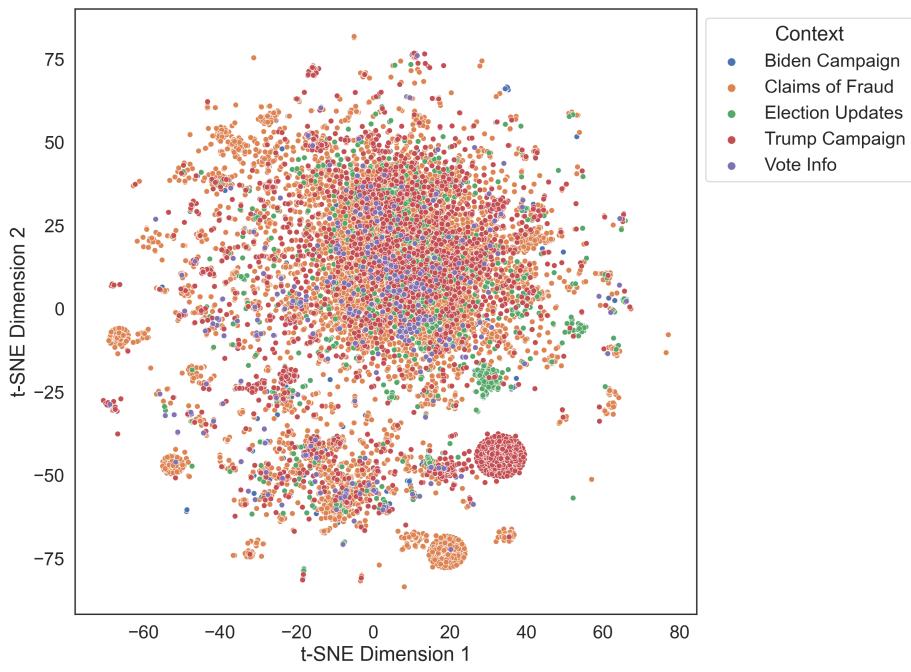


(a) Election

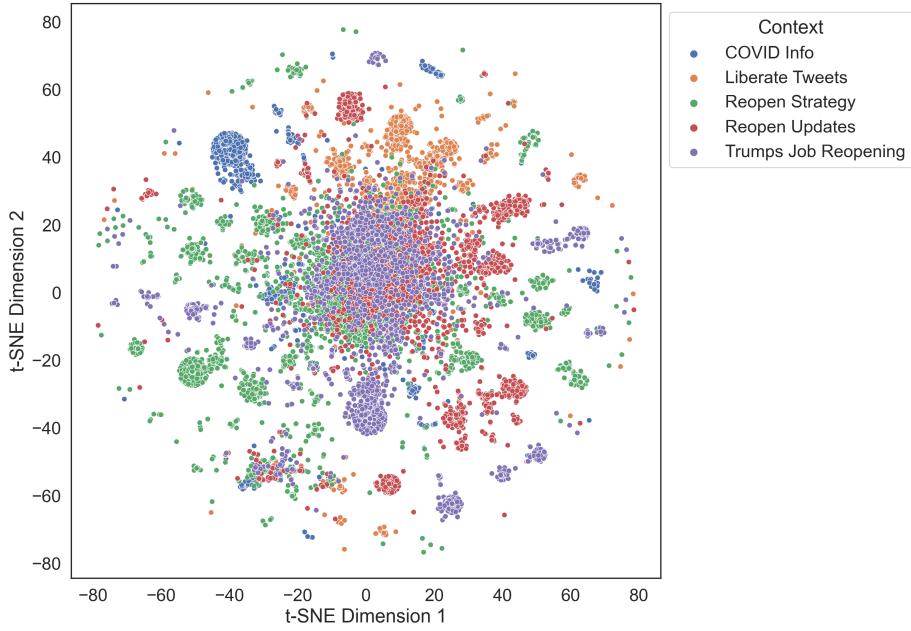


(b) Reopen

Figure 2.7: 2-dimensional t-SNE representation of DTI Tweet embeddings on the *Election* and *Reopen* Datasets. These are the Tweets within 2-hops of a hand-annotated URL or Tweet. The Tweets are colored with their associate hand-annotation.



(a) Election



(b) Reopen

Figure 2.8: 2-dimensional t-SNE representation of un-trained DTI Tweet embeddings for Comparison with Figure 2.7. Again, these these are the Tweets within 2-hops of a hand-annotated URL or Tweet. The Tweets are colored with their associate hand-annotation.

the embedded space, but not far from their neighboring Tweet. We also see that not all of the clusters are so simple, pointing to more interesting contextual structure.

Importantly, the majority of Tweet clusters have homogeneous labels. We see many clusters with 100% label agreement labeled with “Vote Trump,” and “Vote Info.” There are clusters with noticeably higher than average density of “Election Updates” and “Vote Biden” Tweets, but they are surrounded by “Pro-Trump Conspiracy” Tweets, and occasionally “Vote Trump” Tweets. This follows from the observation that many Tweets about the election or statements in support of Biden were replied or quoted with lies about voter fraud and the Democrat’s efforts to steal the election. At the same time, there are often multiple clusters with the same homogeneous labels. That is, there are multiple clusters all labeled with the *Trump Campaign* context. This shows that the unsupervised approach can operate with a different level of granularity than a human annotator, for example saying that two parts of a human-annotated context are actually distinct sub-conversations.

Nearest Neighbor Evaluation

To dive deeper into the specifics of the embeddings, node’s nearest neighbors were analyzed. Specifically, the top-5 closest pairs of hashtags and URLs are displayed in Tables 2.3 and 2.5, respectively for the *Election* dataset, and in Tables 2.4 and 2.6, respectively for the Reopen dataset. The nearest-neighbor analysis only considers the top-500 nodes in terms of their cumulative Retweets in the dataset. Distance is calculated as cosine distance in the 300-dimensional space.

In both datasets, we observe tight triangles of nodes with very similar representations. For hashtags, there is a trio of hashtags, `#returnoftheusa`, `#japanisready`, and `#presidenttrumpwins` that were used in a Japanese discussion about how Trump won and Japan has prepared for the moment. Other than this, we see pairs of hashtags that are used very similarly. For example `#jewsfortrump` and `#womenfortrump` are both discussing groups of people beyond Trump’s base supporting him. Similarly, `#bidencrimesyndicate` and `#laptopfromhell` are both discussing conspiracy theories related to Biden.

Hashtag 1	Hashtag 2	Distance
<code>#bidencrimesyndicate</code>	<code>#laptopfromhell</code>	$5.96 * 10^{-8}$
<code>#jewsfortrump</code>	<code>#womenfortrump</code>	$5.96 * 10^{-8}$
<code>#wethepeople</code>	<code>#wwg1wgaworldwide</code>	$5.96 * 10^{-8}$
<code>#japanisready*</code>	<code>#returnoftheusa</code>	$1.19 * 10^{-7}$
<code>#presidenttrumpwins</code>	<code>#returnoftheusa</code>	$1.19 * 10^{-7}$

Table 2.3: Pairs of hashtags that are closest in the embedded space in the *Election* Dataset. Only the top 500 hashtags are considered, as they are the most important for Tweet representation and are the cleanest. Perfect matches are excluded. A star indicates translation from Japanese.

For URLs, we also see tight triangles. In the *Election* dataset, for example, there is a trio of NBC news election dashboards, each showing the live results for different states

Hashtag 1	Hashtag 2	Distance
#americaortrump	#trumpdictatorship	$1.79 * 10^{-7}$
#americaortrump	#trumpmustresign	$1.79 * 10^{-7}$
#trumpmustresign	#trumpdictatorship	$1.79 * 10^{-7}$
#antifa	#seattle	$1.79 * 10^{-7}$
#wwg1wgaworldwide	#opencalifornianow	$2.38 * 10^{-7}$

Table 2.4: Pairs of hashtags that are closest in the embedded space in the *Reopen* Dataset. Only the top 500 hashtags are considered, as they are the most important for Tweet representation and are the cleanest. Perfect matches are excluded. A star indicates translation from Japanese.

URL 1	URL 2	Distance
https://www.nbcnews.com/politics/2020-elections/arizona-results?cid=sm_npd_nn_fb_ma	https://www.nbcnews.com/politics/2020-elections/arkansas-results?cid=sm_npd_nn_fb_ma	$1.19 * 10^{-7}$
https://www.nbcnews.com/politics/2020-elections/california-results?cid=sm_npd_nn_fb_ma	https://www.nbcnews.com/politics/2020-elections/arkansas-results?	$1.19 * 10^{-7}$
https://www.nbcnews.com/politics/2020-elections/california-results?cid=sm_npd_nn_fb_ma	https://www.nbcnews.com/politics/2020-elections/arizona-results?cid=sm_npd_nn_fb_ma	$1.19 * 10^{-7}$
https://www.vote.org/polling-place-locator/	https://vote.gop/	$1.19 * 10^{-7}$
https://www.newsweek.com/why-i-will-vote-trump-opinion-1543803	https://vote.gop/	$1.19 * 10^{-7}$

Table 2.5: Pairs of URLs that are closest in the embedded space in the *Election* Dataset. Only the top 500 URLs are considered, as they are the most important for Tweet representation, and are the cleanest. Perfect matches are excluded.

(Arizona, Arkansas, and California). It is natural for these to be represented together since they are from the same source and reporting on very similar information. In a similar vein, there are two pairs in the *Election* dataset which are different links to the same place. First, there is a pair of links to a petition for Justice for Tamir Rice. Second, there is a pair of links to an AP article about the Trump administration hiding the CDC’s reopening advice. The other closest-pairs are also similar in content.

The neighbors in both URL and hashtag space are observed to be well-matched pairs. This gives further validity to the methods ability to represent information using the network’s connections. It also gives indirect validity to the Tweet embeddings, because

URL 1	URL 2	Distance
https://www.change.org/p/department-of-justice-investigate-the-killing-of-tamir-rice?recruiter=945350819&recruited_by_id=80cbaad0-4f2b-11e9-a703-ab4db1866a60	https://www.change.org/p/department-of-justice-investigate-the-killing-of-tamir-rice?recruiter=849468015&recruited_by_id=aa29a640-f85e-11e7-b19d-75c28132848a	$2.07 * 10^{-2}$
https://www.washingtonpost.com/world/asia_pacific/china-signals-plan-to-take-full-control-of-hong-kong-realigning-citys-status/2020/05/21/2c3850ee-9b48-11ea-ad79-eef7cd734641_story.html	https://www.bbc.com/news/world-asia-china-52759578	$2.73 * 10^{-2}$
https://apnews.com/article/virus-outbreak-health-us-news-ap-top-news-politics-7a00d5fba3249e573d2ead4bd323a4d4	https://apnews.com/article/virus-outbreak-health-us-news-ap-top-news-politics-7a00d5fba3249e573d2ead4bd323a4d4	$4.48 * 10^{-2}$
https://www.justice.gov/opa/page/file/1271456/download	https://www.washingtontimes.com/news/2020/apr/27/william-barr-orders-legal-action-against-governors/	$5.35 * 10^{-2}$
https://txdshs.maps.arcgis.com/apps/opsdashboard/index.html	https://thehill.com/homenews/coronavirus-report/497509-texas-sees-1000-new-coronavirus-cases-for-5-days-in-a-row/	$5.86 * 10^{-2}$

Table 2.6: Pairs of URLs that are closest in the embedded space in the *Reopen* Dataset. Only the top 500 URLs are considered, as they are the most important for Tweet representation, and are the cleanest. Perfect matches are excluded.

they rely on the representations of URL and hashtags.

Lastly, the closest pairs of different language Tweets in the embedding space are given in Tables 2.7 and 2.8, for the *Election* and *Reopen* datasets, respectively. Off-language pairs were chosen to highlight the method’s ability to work in the multi-lingual setting.

In the case of the *Election* dataset, all the closets off-language tweets occur around the same source Tweet. That initial tweet, in the upper-left on the Table, details an accusation of mail fraud. Then, a someone quoted this in Japanese. Other quote Tweets then received a very similar representation to the Japanese Tweet, despite being written in

Tweet 1	Tweet 2
Breaking: QT: BREAKING: Michigan USPS Whistleblower Details Directive From Superiors: Back-Date Late Mail-In-Ballots As Received November 3rd, 2020 So They Are Accepted “Separate them from standard letter mail so they can hand stamp them with YESTERDAY’S DATE & put them through” #MailFraud	At a post office in Michigan, a bureau clerk said, Whistleblowing. My boss instructed me to postmark the ballot that arrived at the post office today with yesterday’s date. In this regard, it looks like an investigation will begin. As soon as I called my boss, I was cut off. (ja)
this is outright voter fraud. twitter will no doubt block direct video evidence.	”
where is the DOJ???	”
this is outright voter fraud. twitter will no doubt block direct video evidence.	”
President Trump needs to talk about this. Game changer.	”
election fraud alleged by whistleblower in michigan. btw, do we still have a justice department?	”

Table 2.7: Pairs of different-language Tweets that are closest in the embedded space in the *Election* Dataset. Only the top 1000 Tweets which had did not have “undefined” language were considered. Translated with Google Translate from languages codes appended to the quote. The ” symbol indicates the cell is the same as that above it.

English. Similarly, the off-language pairs in the *Reopen* dataset were due all due to quote-tweet interactions with 3 of the 5 pairs quoting a Tweet about the Bomboclate music festival.

These tables highlight the methods ability to give similar representations to Tweets that are close in the conversational graph, even when source languages are differing.

2.4 Automatic Labeling of Contexts

Now that the unsupervised model has been selected, trained, and validated in comparison to the human annotated model, this section will close the loop by providing a computer-assisted interpretation of the unsupervised contexts.

The first step is to determine discrete contexts from the DTI embeddings. Due to the size of the datasets, this is done with k-Means clustering. Our hand-labeling approach converged on roughly 40 clusters for each dataset. We leverage this expert knowledge and set $k = 40$ to obtain the conversational contexts.

As with any cluster classification, our goal is to label the contexts based on their distinguishing attributes. Here, we distinguish contexts based on their n-grams, due to their well-known expressive power [187, 199, 225]. N-grams are sequences of n tokens

Tweet 1	Tweet 2
i feel like uni students are dying for universities to reopen. i know right now we are doing classes from the comfort of our own homes tapi interaction from lecturers first hand tu is more effective. taknak cakap odl ni susah tapi itulah itu kshdjska	agree (in)
i need a group of friends that's down to do stuff like this	bomboclate (du)
And Vietnam did. Extraordinary. (in)	if everyone stayed home... we could be in the same place
kids: “what happened in 2020??” us:	bomboclate (du)
my tia coming up to me at a family party trying to get me to dance with her	bomboclate (du)

Table 2.8: Pairs of different-language Tweets that are closest in the embedded space in the *Reopen* Dataset. Only the top 1000 Tweets which had did not have “undefined” language were considered. Translated with Google Translate from languages codes appended to the quote.

or phrases separated by punctuation or whitespace. Specifically, we consider 3-grams, as 3-grams are significantly more interpretable than 2-grams, but are still computationally tractable compared to 4-grams.

For each English-language tweet in each context, we count the instances of each 3-gram. We only consider the English-language tweets, in order to get English-language. The counts are denoted as $f_{i,g}$, which indicates the count of 3-gram g in context i . The total counts of g are given as $f_g = \sum_i f_{i,g}$. The total number of 3-grams are also useful and are denoted as $d_i = \sum_g d_{i,g}$ and $d = \sum_i d_i$.

Lastly, a method is needed to balance the popularity of a 3-gram with how well it distinguishes a context. This is accomplished by subtracting the frequency within the context by the *expected* frequency given its popularity in other contexts:

$$r_{i,g} = f_{i,g} - \lambda d_i \frac{f_g - f_{i,g}}{d - d_i} \quad (2.6)$$

where λ is a parameter that that analyst to balance popularity and the ability to distinguish contexts; the higher the λ value, the less popularity is weighted. We only consider $\lambda = 1$. The top 3-gram according to $r_{i,g}$ can be taken as a context’s label, though the top 3 are reported for the *Reopen* and *Election* datasets in Tables 2.9 and 2.10, respectively.

Tables 2.9 and 2.10 show the top 3-gram is a useful way of understanding the context. However, looking at the next one or two 3-grams can provide additional context to give an even better label. For example, context 29 in the *Reopen* dataset could be labeled by *Liberate the White House*, the counter-movement to Trump’s “liberate” tweets, calling for him to be voted out of office. However, given the second and third 3-grams are about calls

Context	1	2	3
5	need reopen economy	want reopen economy	people want reopen
2	want schools reopen	safe reopen schools	reopen social distancing
26	liberate hong kong	hong kong revolution	gym stop corona
29	liberate white house	sandra bland case	reopen sandra bland

Table 2.9: Top 3-grams for select contexts in the *Reopen* dataset.

Context	1	2	3
4	mail ballots counted	electoral college votes	votes registered voters
20	stop counting votes	voter fraud claims	prime minister slovenia
29	proven putin puppet	trump proven putin	putin puppet vote
24	ballot drop box	polling place ballot	voting plan election2020

Table 2.10: Top 3-grams for select contexts in the *Election* dataset.

to reopen investigations of Sandra Bland’s death in a Texas jail, we see that the context is more specifically a rebuke of the current government’s issues with racism. Another example is context 20 in the *Election* dataset, which details calls to stop the vote due to claims of fraud, and refers to the Prime Minister of Slovenia’s move to prematurely call the election in Trump’s favor⁶. Additionally, there is a need interpret the 3-grams and fill-in the stop words or characters that were removed to achieve clean results. This way “Want Schools Reopen” can be translated to “Calls for Schools to Reopen.”

2.5 Impact of Contextualization on Networks

Now that the contextualization approach has been introduced and validated, we apply it to the datasets to demonstrate the impact that contextualization has on Networks. Referring again to the context cartoon in Figure 1.2, we expect the separation of networks to have major impacts on the network analysis. To assess this impact, we consider two basic aspects of a network: its nodeset and its central nodes working with the automatically extracted contexts from Deep Tweet Infomax.

2.5.1 Nodeset Overlap in Tweet Contexts

Perhaps the most basic aspect of a network is its nodeset. For social media analysis, the nodeset tells us which users are active in a conversation. Mixing conversational contexts likely means that we are mixing our nodesets. This means that a traditional analysis would say that some users are active in a conversation when they were talking about something else entirely. This could have important implications for critical conversations. For example, a researcher interested in misinformation could collect a dataset on misinformation

⁶<https://twitter.com/JJansaSDS/status/1323913419200864256>

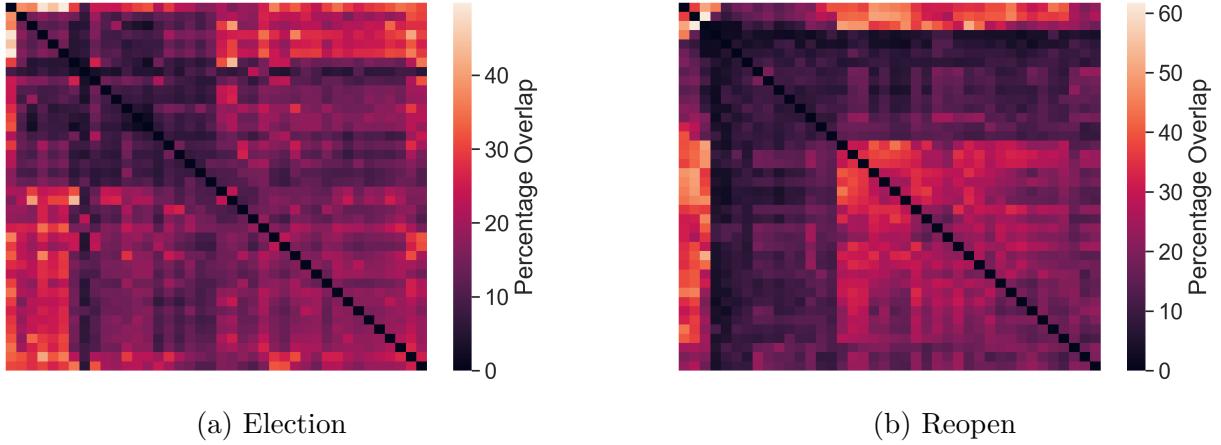


Figure 2.9: Overlap of active users in the conversational contexts. The diagonal is set to 0% for readability.

hashtags. As we have seen, this dataset is likely to have very different conversations embedded within it due to the shortcomings of data filtration techniques. A traditional analysis, then, could say that some users were engaged in a misinformation discussion when they actually were not. In this section, we quantify the extent of the impact that context has on the nodesets.

We consider users to be active within a conversational context if their Tweet is in the context, or they Retweet one that is. For each dataset, we calculate the pairwise percentage of overlap in membership and plot the results in Figure 2.9, normalizing levels to that of the smaller context. We see that the overall levels of overlap are low across both datasets, often below 20%. However, both datasets have a small cluster of contexts with high overlap with many other contexts (40-60%). Further investigation shows that these are smaller contexts, where it is simply easier to have high overlap.

This finding has important ramifications for conversational network analysis. The presence of non-overlapping contexts highlight that *global* properties of conversational networks are being affected by context. Placing users from one context in the same network as those in another context is a misleading representation of the data, which could affect the vast majority of nodes in each context. It is possible that users from these different contexts may even be placed in the same component of a decontextualized network. As the number of active users increases, it becomes more likely that the two contexts will be merged into a single component under decontextualized analysis.

More importantly, most of the context-pairs have low but not negligible overlap, around 15%. This means that the *local* properties of the de-contextualized network are affected. With 15% overlap, we can expect that about 15% of users will have connections from users in both contexts with no way of distinguishing them. This has negative effects on every aspect of network analysis. Path-based centralities, for example, will be calculated on paths that could not occur in the data because they span contexts. The impact of this is further studied in the following section.

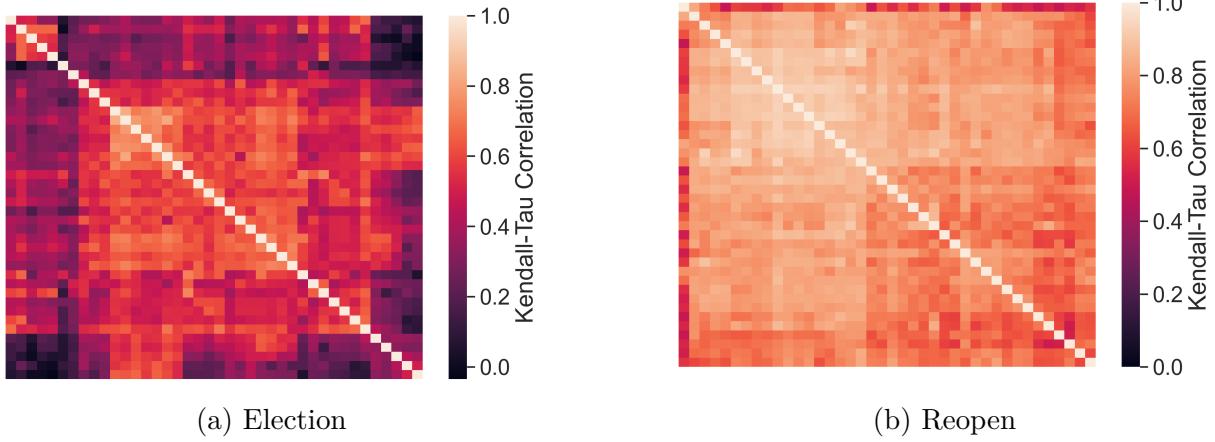


Figure 2.10: Kendall-Tau correlation of PageRank-sorted users across contexts.

2.5.2 Influencer Overlap in Tweet Contexts

Next, we study the impact of contextualization when analyzing central nodes. We do so by comparing the centrality rankings of different contextualized networks. To be clear, these are user-user networks, where connections are drawn if a user is mentioned, Retweeted, replied-to, or quoted within Tweets of a specific context.

To fairly compare across contexts, we only consider the intersection of the nodesets. This is important especially given the results of the previous section. If context A has a network with 10000 nodes, and context B has a network with 10000, and there is only 5000 nodes in both, we account for that by taking the subgraphs of each context so that only the 5000 overlapping nodes are considered. We note that this is a generous comparison. Because we are removing the variability in the nodesets, the results will indicate that the networks are more similar than we know them to be. We will show that even under these conditions contextualized networks have considerable differences.

Given two contextualized network subgraphs with matching nodesets, we rank the nodes in each based on their weighted and directed PageRank centrality [166]. We add a small tolerance parameter, $\epsilon = 10^{-9}$ to ensure that ranking comparison is not affected by noise. To quantify the similarity between the rankings, we calculate using Kendall's Tau ranking correlation [114]. In each case, non-significant results (those with $p < 0.05$ were set to 0. The results are given in Figure 2.10.

The mean correlation between contexts is 0.472 for the *Election* dataset and 0.784 for the *Reopen* dataset. Clearly, these are substantial correlations, which is intuitive. The most influential users are those with verified badges and many followers. This influence should carry across contexts. Still, the frequent correlations below 0.5 show that non-contextualized analysis is severely corrupting centrality rankings, especially when we consider the fact that our results already control for the potentially vastly different nodesets and the additional edges that come with them.

To illustrate how this corruption plays out for the most central nodes a dataset, we consider a simplified example. First, we take the two largest contexts in a dataset. Within

each contextualized network, we find the 10 highest ranked nodes, again according to weighted and directed PageRank. Then, for each of these nodes, we calculate their ranking in the *mixed-context* network, which is that constructed from Tweets in both contexts. The point is to show how highly ranked users may move down in rankings when other contexts are mixed in. The results are shown in Table 2.11.

Election		Reopen	
True Rank	Corrupted Rank	True Rank	Corrupted Rank
1_1	1	1_1	31
2_1	5	2_1	50
3_1	2	3_1	43
4_1	4	4_1	79
5_1	21	5_1	75
6_1	26	6_1	82
7_1	24	7_1	78
8_1	27	8_1	86
9_1	29	9_1	84
10_1	31	10_1	71
1_2	1	1_1	1
2_2	2	2_1	2
3_2	4	3_1	6
4_2	18	4_1	3
5_2	7	5_1	4
6_2	8	6_1	5
7_2	9	7_1	7
8_2	10	8_1	8
9_2	11	9_1	9
10_2	12	10_1	10

Table 2.11: Pagerank centrality ranking for the top-10 most central users in two contexts. True rank indicates their contextualize rank, with the subscript indicating the corresponding context they belong to. Corrupted rank indicates their rank in the uncontextualized setting.

From this experiment we see 3 key ways that centralities are corrupted from context mixing. First, we see that important users within a context can be relegated to much lower rankings when contexts are mixed. For example the 10th node in context 1 of the *Election* dataset moves to 31st, while the 8th node in context 1 of the *Reopen* dataset moves to 86th. Second, we see that the relative rankings are not preserved within-context. For example, the 10th node in the context 1 of the *Reopen* dataset is ranked higher than nodes 4-9 in that context when the data is mixed. Third, we see that the rankings from one context can dominate another. The second context of the *Reopen* dataset are consistently ranked above that of the first context, forcing all of their rankings downwards.

Taken together with the overall correlations, we see that the deviations in node ranking

from contextualized networks to non-contextualized networks are considerable and can arise in multiple ways. Especially given that real-world non-contextualized data is mixing many contexts of varying size, there is no straightforward way of understanding *how* the overall centrality rankings are affected. Thus basic network analyses like centrality analysis are severely corrupted when context is ignored, and the best way to account for these errors is to study the underlying contextualized network instead.

2.6 Limitations

There are a few key limitations to consider for this work. First, the initial feature representation of Tweets are derived from a relatively simple language embedding scheme which does not include attached images or video. The scheme was selected due to its scalability and its ability to embed Tweets written in different languages within the same vector space. This approach embedded Tweets from 11 languages, but 4% of the reachable Tweets were still not reached. The lack of image or video representation is the more important limitation, particularly because many Tweets with images or video do not have text. While this is largely the case for replies and not original Tweets, the full space is affected due to the transfer of information from reply to base Tweet and vice-versa. Even though a pre-trained model can be used to obtain image or video representations, the process of including this information in initial Tweet feature representation is unclear. Most Tweets lack images, so feature concatenation will not be effective. Combining the feature vectors is also not straightforward because the vector spaces are not aligned. A process which gives a feature representation of both text and images is an open but active area of work within multi-modal learning [43].

All Tweets are treated equally in current methods, however, social signals such as the number of Retweets or favorites a Tweet gets could inform more sophisticated aggregation schemes for GNNs. This is left for future work. Further, methods which incorporate URLs domain name could improve embeddings but are also left for future work.

Another limitation of the current analysis is the lack of mention representation. Mentions are a core feature of Twitter, allowing for users to directly tag other users in their Tweets. Incorporating mention nodes into Deep Tweet Infomax should improve Tweet representation, since mentions are so commonly used to tag major players in a discussion. This was not done in the present work because of the quality of the data available. In the first version of Twitter’s API, replying to a Tweet adds a “mention” of the user that is being replied to, and often adds “mentions” to several other user higher in the conversation tree. These are not actual “mentions,” just artifacts of the already modeled Tweet-Tweet graph, so their inclusion could harm our results.

The last major limitations of the approach are that the method still derives *discrete* conversational contexts and that these are compared with noisy human annotations. Specifically, the fact that our annotations were given by a single annotator poses a limitation. Next, we have seen that interactions can be represented as occurring in a *continuous* context space. However, all existing network approaches assume discrete context spaces. Given the appropriate methods, the continuous context space could be used to measure things

such as conversational drift and contextual persistence of links. Thus, methods directly operating in continuous space are of interest, and are considered in Appendix B.

2.7 Discussion

The first major finding of this Chapter is that keyword-based data collection is too coarse to achieve properly contextualized social media data. In both of the cases studied, there were several major discussions present in the dataset that were distinct from the topics of interest. Perhaps most notably, a substantial portion of the *Reopen* dataset was actually discussion about Black Lives Matter, mostly due to the use of “reopen” as in “reopen this investigation”. Even for the discussion that is related to the topics of interest, we observe several inter-related conversational contexts. For example, in the *Election* dataset there are distinct discussions about information on *how* to vote versus who to vote for.

The Tweet representation method, Deep Tweet Infomax, automatically uncovers these contexts. The multi-step validation procedure shows that this model follows our intuition by clustering Tweets of the same hand-annotation⁷ together. The validation efforts also show that the model gives similar representation to similar hashtags, similar URLs, and similar Tweets. Lastly, the model’s representations are capable of differentiating edges from non-edges in the heterogeneous conversational network.

The second major finding was that conversational contexts extracted with Deep Tweet Infomax have largely different nodesets. Often, pairs of contexts in the two datasets only shared 20-40% of their users, though this was occasionally as high as 60%. This means, for two different discussions going on in the same dataset, only about a third of the users can be expected to be active in both discussions. This findings has major implications for how we analyze social media datasets. The lack of overlap between conversations indicate that a non-contextualized approach will mislead analysts into thinking that users are active in discussions that they never Tweeted in, and possibly never even read. When analyzing critical discussions, such as those about disinformation, hate speech, or extremism, this can have major consequences. Further, the nodeset is the most fundamental part of a network; such drastic differences in the nodesets of contextualized networks indicate that there will be even more drastic results for network analysis such as community detection or centrality ranking.

Lastly, we find that there are major differences in centrality rankings of users between conversational contexts. This analysis was performed after controlling for the differences in nodesets, so the true differences in rankings are likely to be much larger. The rankings were positively correlated, with $\tau = 0.472$ for the *Election* dataset and $\tau = 0.784$ for the *Reopen* dataset, on average. However, the lack of one-to-one, or even close to one-to-one ranking of central actors between conversations demonstrate that non-contextualized analysis will give misleading results about who is most important in a dataset. We further show that these results can be corrupted in a number of ways, including central actors from one conversation dominating the others, mixed central actors, and even re-orderings

⁷and those annotated with label propagation from hand annotations.

of the within-context rankings. The fact that the central actor rankings can be corrupted in several ways makes the non-contextualized rankings even harder to interpret appropriately..

Contextualizing data allows for more accurate representation of user's importance within a discussion. Social media analysis can have high stakes when it is used to determine the importance, or presence, of users within information operations. While this work moves closer to properly attributing users to the conversations that they are actually active in, there is a question of interpretability. The move towards deep graph neural networks makes interpretability challenging, though the computer-assisted techniques developed in this chapter make this easier. Considerable validation steps have been taken, however if this work were to be applied to qualitative work looking to attribute accounts a high-stakes setting, much more in-depth checks about how specific users fit into a conversation must be taken.

In the following chapter we will build off static contextualized network analysis by exploring the dynamics within and between contexts.

Chapter 3

Contextual Dynamics of Social Media Discussions

In Chapter 2, we demonstrated that Twitter data collections attempting to study a conversation actually contain many conversational contexts. We saw that mixing these many contexts together harmed our network analysis because fundamental network properties like the nodesets differed greatly between contexts. The initial solution is to use the methods in Chapter 2 to separate out contexts so that they can be properly analyzed using static network techniques.

Now, we demonstrate that contextualized network analysis can help us discover new aspects of our data, not simply enable classic analyses with greater precision. Specifically, we show that the interactional context dynamics give new insights into the nature of online communities and the conversations they have. We break down these dynamics into four parts. The first division is between activity dynamics and network dynamics. We consider activity dynamics to be those which are primarily characterized by the number or order of posts (i.e. activity) over time. On the other hand, network dynamics consider the structure of conversational networks, paying attention to the dynamic relationships between users and contexts. The second division is between intra-context and inter-context dynamics. Taken together, these four dynamic analyses give a rich portrait of online communities previously inaccessible with a non-contextualized approach.

3.1 Related Work

With the initial detection of conversational contexts being so closely related to the literature on event and story detection, it is natural that contextual dynamics are related to event and story dynamics. Prior work in both of these areas consider the activity dynamics, and are not interested in networks. For event dynamics, the focus on primary start times in order to construct a timeline in the dataset, is a central starting point for our work, where we similarly construct context-based timelines [61, 223, 229]. However, the overlapping and inter-related nature of contexts makes them more similar to that of story and sub-story analysis [56, 203]. Within that analysis, we rely heavily on the “Dynamical Classes of

Collective Attention” of Lehmann et al., which enables us to categorize activity dynamics based on the shape of the activity curve [125].

When considering the intra-context network dynamics, the most important literature to draw on is that of dynamic community detection. The problem of community detection in complex networks has received a tremendous amount of attention, resulting in many popular algorithms that have been empirically verified [22, 156, 160]. However, these algorithms all assume that the network being analyzed is *static*. If this assumption is violated, different communities may have been averaged together over time, resulting in obscured or misleading results. While dynamic aspects of communities are still often ignored in practice, many potential methods of dynamic community detection have been proposed. Rossetti and Cazabet posit that this is due to a disconnect between researchers in the field, and a lack of visibility [190]. Here, we discuss how prior work in dynamic community detection has motivated our approach.

Using the terminology of Rossetti and Cazabet, there are two types of dynamic network models: snapshots, and temporal networks. Snapshots segment the data into networks that are assumed to be static, while temporal networks assign a birth and death timestamp to each edge. The temporal network model is pure in that it does not aggregate links, and the network is never assumed to be static. While this is more accurate than the snapshot approach, it comes with limitations. Namely, the analysis for such objects require more computational power, and a set of tools separate from those created for static networks. Network snapshots, however, have access to the large toolset of network science. Furthermore, it is critical that community detection can extend to streaming data. Snapshots are a natural way of handling this: aggregate links in the stream until the snapshot length has been reached, then analyze it.

Snapshots are limited, however, when the goal is to find fine-grained evolutions in a network. In this case, temporal networks are more appropriate. Here, we are looking for *events*, or large changes in communities, rather than evolution patterns, so we have chosen to use the snapshot modeling approach. Given this, our partitioning techniques will work best under the assumption that network communities undergo rapid change, meaning in few time slices, rather than communities undergoing constant structural evolution. This assumption is often met when major events occur in the network’s timeline.

Using network snapshots, it is common to take an “instant optimal” or a “two-step” approach, wherein snapshots are grouped statically and then compared [12, 190]. Some others have criticized this approach, claiming that it is too vulnerable to noise and the snapshot groups do not use valuable historical data [132, 133]. These concerns have merit. It has been shown that static grouping algorithms are unstable, and can give very different result under only small perturbations to a network [11]. Given that slice groupings are expected to be noisy, we compute pairwise comparison for *all* time slices. Comparing all slices addresses both of the concerns voiced in [132, 133]; historical data is used when finding similar segments, and pairwise-noise will be present, but should average out when considering an entire block of similarities. It seems that only Goldberg et al. have used all slices in the comparison step of a two-step approach [77]. Our work differs from Goldberg et al.’s in two ways. First, our goals are different. They sought to identify evolutionary patterns in groups, while we aim to find disruptive events with respect to communities, in

hopes of aggregating network snapshots into a smaller series of networks with meaningful divisions. Second, our approaches differ. They identified common community cores across time, while we calculate the overall correlation between communities.

A very similar approach is given by Masuda and Holme, who use hierarchical clustering to label slices as “states” of the system, which are expected to recur [141]. Their work differs from ours in two major ways. First, they take a “one step” approach, where states are decided based on the network rather than its communities. Second, no temporal continuity is imposed for states. For our purposes, this is essential. Without temporal continuity, states cannot be collapsed and analyzed as a static network. Also, we rely on a different comparison mechanism: product-moment correlation. The fact that pairwise temporal similarity operations find success in network aggregation (our work) and chain-like state changes (Masuda and Holme), shows the power of the approach for solving new problems in temporal networks.

Our approach is a special form of link aggregation. The problem of link aggregation has been studied in [209]. Taylor, Caceres, and Mucha examine the effect that aggregation has on community detection. They look at both aggregation over network modes and over time slices. It was concluded that aggregation can enhance or obscure communities depending on their size and persistence. Matias and Miele recognize a similar problem, questioning the assumption that most nodes do not change groups [142]. While Taylor et al suggest analysis on multiple scales and Matias and Miele attempt to control for short term group-switching, we take a different approach: only aggregate slices that have similar community structure.

In summary, work has been done on simplistic community-based event detection, however the focuses has been on community evolution patterns, rather than shocks to the overall structure. We develop a method for uncovering such events to better understand intra-context network dynamics.

Turning to intra-context dynamics, we draw on work from sequence and trail analysis. In both cases, there are a number of states in which the unit of analysis can travel between. In our case, we are studying users (unit of analysis) traveling between contextual contexts (states). Sequence analysis, then, is concerned with uncovering patterns in the transitions between states [1, 31]. Methods of sequence analysis have been developed from researchers across scientific disciplines due to the near universal emergence of sequential data. Some of the biggest impacts have come from sociology, demographic science, and biology [19, 47]. The primary methodology of interest here is the Markovian model of sequence generation, which assumes that the next step in a sequence is only determined by the current state. Analysis of this model applied to conversational contexts enables us to show the overall flow of users in a conversation, and further characterize the contexts.

While sequence analysis is a powerful lens for analyzing contextual dynamics, it omits part of the dynamic data: the time that states are activated in. The analysis of time-dependant sequences is known as *trail* analysis [16]. The added level of detail can help uncover behaviors that are specific to a moment in time. For example, this approach has been used to study the actions of Jihadist groups, whose next action depends on their previous action, but also the timing between actions [35]. Trail analysis applied to conversational data will show *when* users or groups of users move between conversations,

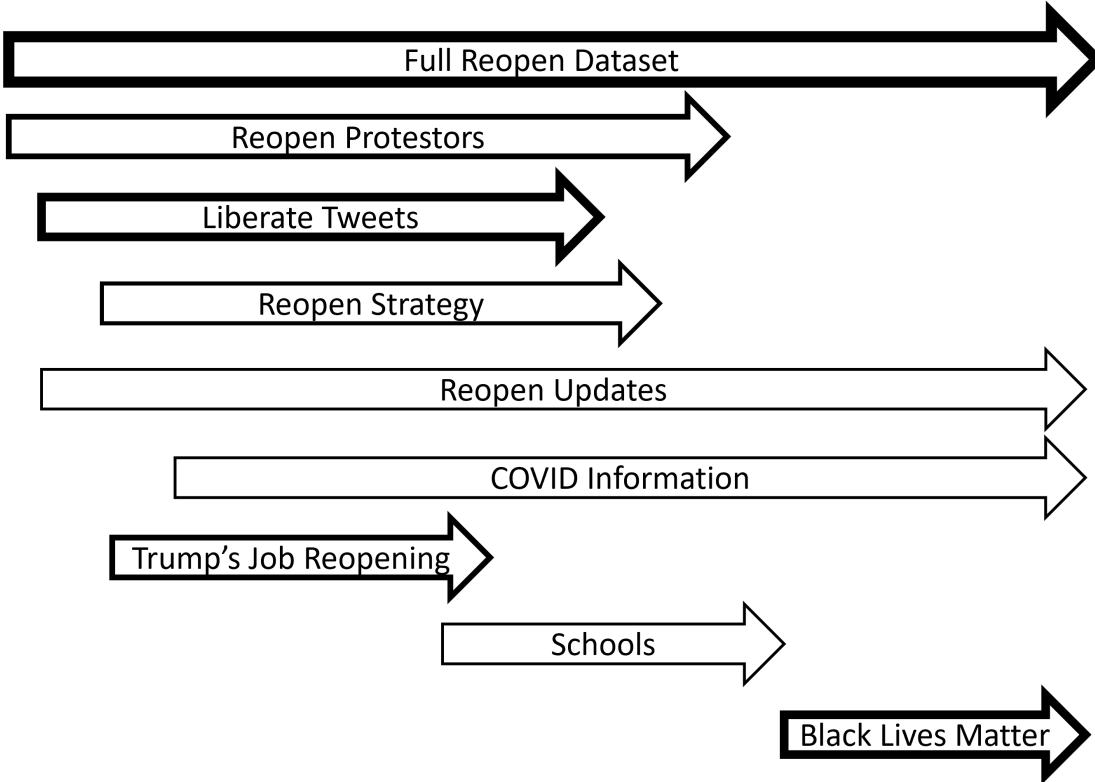


Figure 3.1: The rough start and end times are shown for the top interactional contexts in the *Reopen* Dataset.

giving us a more granular view of dynamics than that given from the sequential models.

3.2 Intra-Context Activity Dynamics

In this and the following section, we explore the *activity* dynamics within contextualized social media data, which refers to the number of Tweets posted within a context over time. We begin by studying the *intra-context* activity dynamics, which we distinguish from the intra-context *network* dynamics studied in Section 3.3. The patterns in the activity time series can help understand the timeline of discussions in the dataset, and can help categorize each of the contexts.

First, the high-level activity dynamics can be uncovered by calculating the approximate start and end time of a conversational context. In a strict sense, the start time is the time that the first Tweet in that context was posted. Similarly, the end time is the time that the last Tweet in that context was posted. In practice, it may make sense to set a small threshold to determine when a context starts and ends, to avoid outliers from skewing the data. For contexts with many Tweets, we exclude the first and last 100 Tweets to calculate these times.

The start and end times of the top contexts in the *Reopen* Dataset are shown in Figure 3.1. This diagram highlights the fact that the start and end time of the full dataset obscures

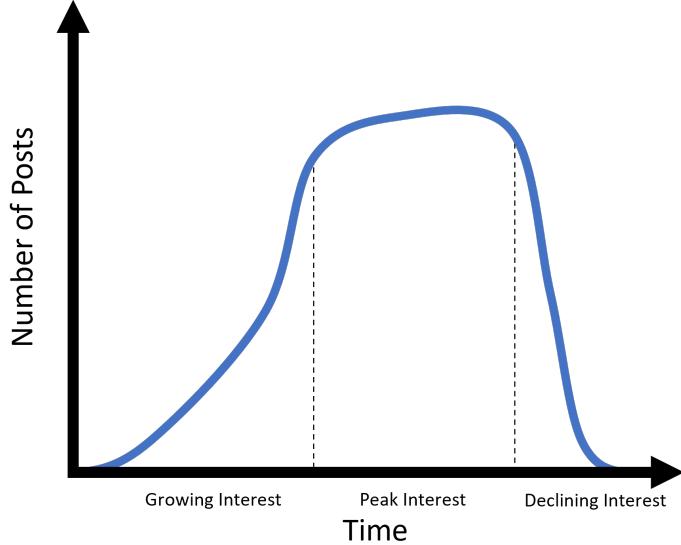


Figure 3.2: An example of an activity curve with the three major sections labeled.

the fact that there are many conversations which are starting and stopping within the full timeline. We also see that conversations vary in length. For example, discussion about COVID and schools is a relatively short conversation in the dataset while discussion and dissemination of COVID information (infection rates, best practices, etc.) is carried out through most of the dataset.

A more powerful analysis goes beyond the start and end time of each context. Here, we study the shape of the activity curve (time series of the number of Tweets posted) to categorize the type of conversation based on the “Dynamical Classes of Collective Attention” [125]. Curves are categorized based on their shape at the start, peak, and end. An example of such a curve with its three sections labeled is given in Figure 3.2. As is indicated in the cartoon, what is considered the “peak” is up to some subjective interpretation. In practice, some percentage of the maximum of the curve could be taken to find times that have near-peak activity, but the selection of that percentage is also somewhat arbitrary. Nevertheless, each section of the curve is visually inspected to match one of the potential patterns. We will outline these patterns now.

The start of the curve can have one of two behaviors. First, there can be a gradual slope upwards over a prolonged period of time. This indicates a building of user interest in the discussion. As time goes on, more people enter the discussion. In terms of events, this pattern occurs when events are anticipated. For example, a major sporting event like the World Cup is planned in advance, so people start to talk more and more about it in the days leading up to the start of the event. Outside of events, this pattern may occur when an idea or concept gradually builds interest over time.

Alternatively, there may be an abrupt or sudden burst of activity. The lack of build up indicates that the topic of discussion was unanticipated, usually because of an unexpected event. We can consider this contrast in the case of natural disasters. Certain natural

disasters, like hurricanes, are forecast in advance, so it is expected that users will Tweet about it in the days leading up to the event. Others, like earthquakes, are unanticipated so there will be no build-up and once the event takes place there will be a burst of activity.

The activity peak has two characteristics of interest: the length of the peak and its magnitude. The length of the peak, or the amount of time that the context sustains maximum engagement, gives insight into how much sustained attention the conversation received, which can help distinguish transient discussions from those that are long-lasting. At the same time, the magnitude, or number of Tweets, at the peak can indicate how important or far-reaching the conversation is overall. Typically the peak analysis does not change the category of the discussion, but nevertheless helps to understand it.

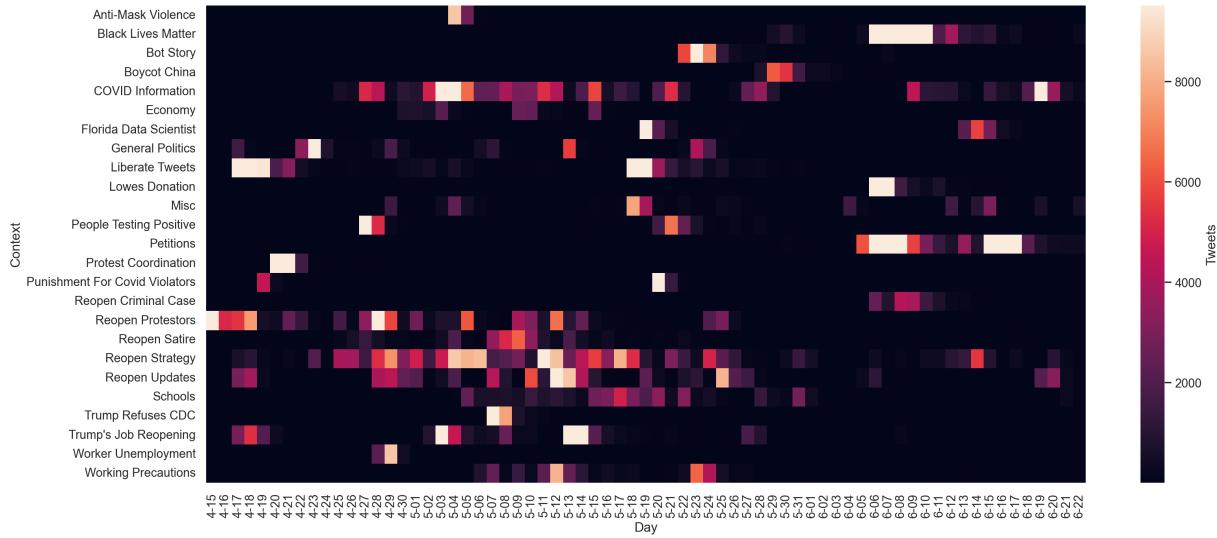
Lastly, the tail of the activity curve indicates the impact of the conversation at hand. Mirroring the start of the curve, the tail will either have a short and sudden drop, or a slow descent. A slow descent indicates lasting impact where users are interested in participating in the conversation after the peak, though this interest wanes. On the other hand, a short or sudden drop indicates that the discussion at hand had a limited impact. This is often the case with conversations surrounding scheduled events which may generate discussion during but not after the event.

3.2.1 Categories of Intra-Context Dynamics

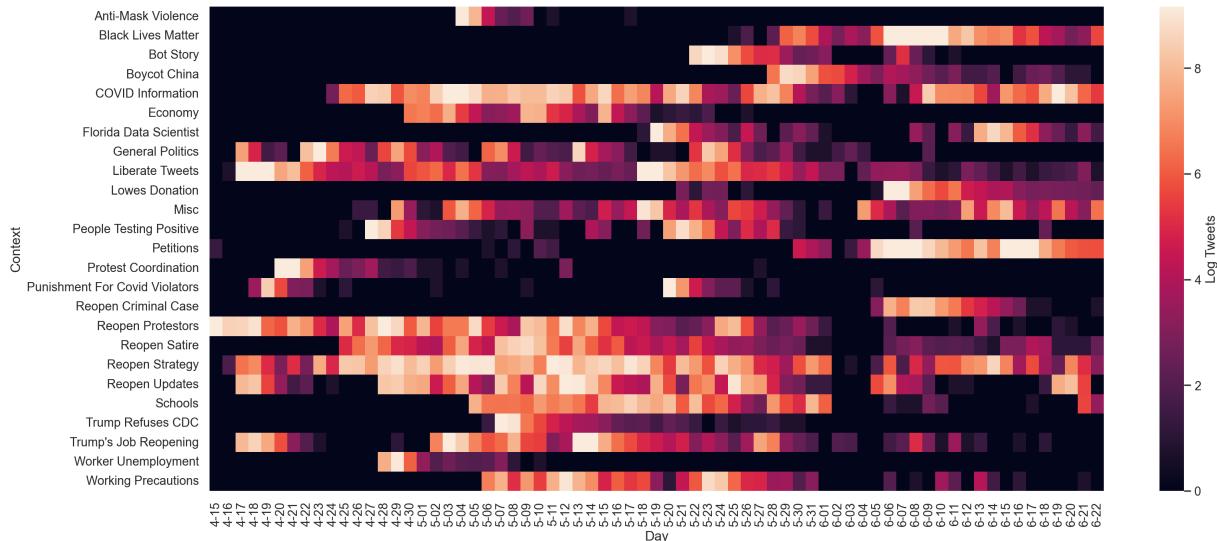
Now that the general patterns have been described we will piece them together to show the full set of categories. We will also show actual examples of each category using conversational contexts in the *Reopen* and *Election* datasets. These distinctions will be based off of the empirical analysis of the activity data for the *Reopen* and *Election* Datasets as shown in Figures 3.3 and 3.4, respectively. The activities are shown as raw counts and as log-counts, which have different advantages. It is easier to identify peak behavior in the raw count plots, whereas it is easier to identify tail behavior in the log-count plots. Now, contexts can be derived from either of the methods developed in Chapter 2. Generally, the automated method was developed for use in scenarios where there are not resources to use human annotation. Because we have already expended these resources, we proceed with the human annotated data, which are generally more interpretable. Through this exercise, we demonstrate how activity categorizations can be used to understand more about the types of conversational contexts we derive from social media discussions.

Build to Peak With Sudden Drop

This category is defined by its gradual gain in interest followed by a sudden drop. The combination of these characteristics is usually met for discussions of scheduled events that do not have lasting importance, or where the implications of the event itself do not warrant much discussion. The classic example given by Lehmann et al. is a sporting event like the Masters [125]. Fans' excitement build leading up to an event, peaks during the event itself and perhaps during the award ceremony. After the event, however, there is not much left to discuss. Neither of the datasets contained events that we would expect to follow

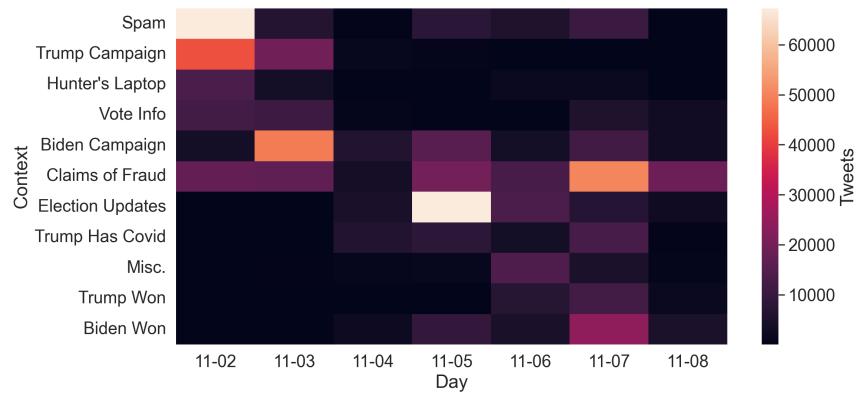


(a) Tweets

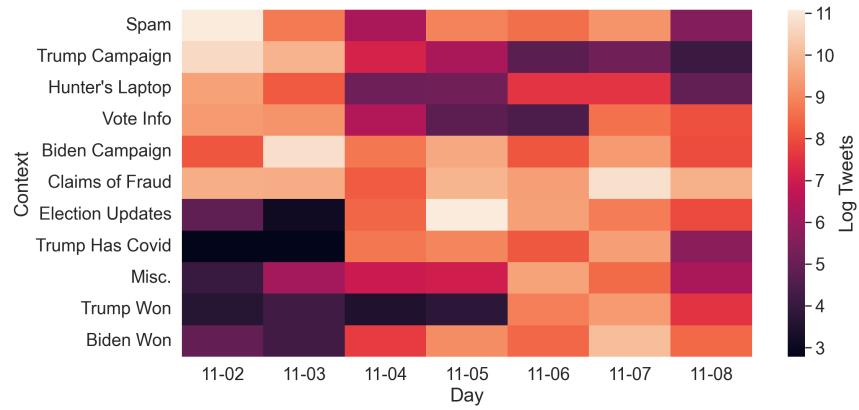


(b) Log Tweets

Figure 3.3: Detailed activity plots for the *Reopen* Dataset.



(a) Tweets



(b) Log Tweets

Figure 3.4: Detailed activity plots for the *Election* Dataset.

this behavior, so naturally none of the conversations found in the datasets neatly fit this category.

Build to Peak With Gradual Decay

Starting out in a similar way to the previous category, this category finishes with a gradual decay. This is an extremely common pattern for discussions on social media around events with lasting implications, like elections. The distinction from the previous category comes from the gradual decay, indicating that the conversation still warrants discussion after the peak, however interest naturally dies out eventually. In the *Reopen* dataset, we see this pattern most clearly in the *Reopen Strategy* conversation, which gained interest as the protests went on, and faded out of interest afterwards. This behavior can also be seen in the *Schools*, *Reopen Updates*, and *Working Precautions* contexts which follow similar patterns. The *Claims of Fraud* context in the *Election* dataset also falls under this category. Fraud claims built as the votes were being tallied, resulting in increased interest, especially as results tilted towards Biden. These waned after the election was officially called, but the discussion did not die out immediately and clearly had a lasting impact.

Sudden Peak With Sudden Drop

Here, the sudden peak is characteristic of an unexpected event, while the sudden drop indicates a lack of impact. However, the sudden peak can also be an artifact of an event that is not important enough to garner attention before it is actually occurring. Often, this occurs in conversations surrounding trivial time-related events. For example this might happen on holidays or days of the week. A common example is #MondayMotivation, where users post motivational Tweets to start their week. The discussion is isolated on Monday, since it does not make sense for users to post before or after the day itself. A clear example of this is the *Florida Data Scientist* context, which was unexpected and quickly garnered a lot of attention, however it was also quickly abandoned as its own conversation, though it may have transitioned to become a talking point in other conversations. The *Trump Refuses CDC* context follows the same pattern.

Sudden Peak With Gradual Decay

A sudden peak with gradual decay is one of the most popular categories of activity on Twitter, as it is indicative of viral moments. These moments are unexpected, so achieve quick acceleration in interest, the length of the peak is typically longer for more important viral moments, but the length of time that it takes for attention to decay is also a good indicator. The primary example of this in the *Reopen* dataset is the *Liberate Tweets* conversational context. No one could anticipate that Trump would Tweet in support of the protests, however once he did it generated a tremendous amount of discussion. This discussion naturally decayed until he followed up with additional Tweets of support. The *Black Lives Matter* context is another clear example, where the murder of George Floyd was an unexpected event with massive implications for the country. In the *Election* dataset,

the best example is the *Election Updates* context, which drastically rose as the polls closed, and then slowly declined afterward.

Sustained Peak

The word “peak” is used for continuity with the other categories, however this could also be described as a lack of peak. The activity curve in this category appears to be a flat line; steady interest is shown. The magnitude of the curve differentiates pervasive and important conversations from low-level background conversations. Generally, this category is often seen around non-event focused discussions, like those about ideas or problems. In the *Reopen* dataset, *COVID Information* and *General Politics* are examples of this category. The overall level of activity in the *COVID Information* category is much higher than that of *General Politics* indicate its relative importance.

3.3 Intra-Context Network Dynamics

We when we first considered intra-context network dynamics in Chapter 3.2, we recognized that there are rich dynamics within a conversational network, but limited the analysis of these dynamics to that of activity charts. Here, we will dive deeper to explore how these dynamics take place at the network level, and how they may be accounted for to get a clearer view of communities. This section can be seen as dynamic community detection applied to a contextualized network.

While a vast body of literature addresses the problem of community detection in networks, only a small portion considers their dynamic aspects. Recently, this problem has received more attention. The majority of networks are truly time-varying, and community detection should reflect that. Often, large periods of time are aggregated into a static network, or they are aggregated at regular intervals and analyzed individually. At best, this smooths over any interesting temporal features of the network data, at worst, combining links from old and new communities can yield misleading results. This problem is quite similar to the original network context problem detailed with the cartoon in Figure 1.2, except now, the different networks being combined are those from different time periods.

Although infrequently used in practice, there has been work in this area which is well summarized by Aynaud et. al and Rossetti and Cazabet [12, 190]. Much of this work focuses on community evolution, or how communities change in time [3, 80]. However, an easier problem has been understudied. In the first part of this chapter, we seek to answer the question: “How can we segment a dynamic network, such that static analysis of the segments will be representative of the underlying dynamic communities?” Effectively, temporal partitions should be change points for community structure. This problem is easier in that we assume communities are static until an event changes them noticeably. Segmenting the network in this way gives us access to all the tools of static network science, and simplifies the interpretation of results. “Partitions” are also used to describe the grouping of nodes into communities. In this work, we are describing *temporal* partitions, which define the ends of time segments.

We compare with one of the main methods of segmenting networks in practice: Generalized Louvain. The first shortcoming is its dependence on a specific grouping algorithm. As is well known, each grouping algorithm has strengths and weaknesses, and should therefore not be used as a universal tool. It also has the disadvantages of relying on user-defined parameters, and needing a static nodeset.

We propose a simple method for placing temporal partitions in dynamic networks such that static community analysis accurately represents the dynamic communities. This method is parameter free, works with any grouping algorithm, and is somewhat robust to noise, as demonstrated through trials with synthetic datasets and a case study.

The initial framing of the method relies on static nodesets, where the Ukrainian Parliamentary voting network is analyzed as a case-study. The Ukrainian Parliamentary voting network is for two reasons: it provides an example of a fixed nodeset, and it has a known change-point (the Euromaidan Revolution). Known change-points are rare for online social media datasets, making an outside dataset required for validation. After validation, the static nodeset assumption is relaxed to semi-static, and the approach is applied to the contextualized communication networks derived in Chapter 2.

3.3.1 Methods: Snapshot-Based Change Detection

Informally, the goal of this section is to partition the time dimension of a network such that each segment forms *cohesive groups*, and adjacent segments are *noticeably different* from each other. This section will formalize a procedure that achieves this goal.

We start with a fairly restrictive assumption: the nodeset is static. That is, every node is present in every time slice. While many datasets violate this assumption, this is a natural starting point. Dynamic nodesets introduce complications such as the problem of comparing groups with different nodes. We relax this assumption to semi-static nodesets (those where the majority of nodes are present in every time slice) after validation of the initial approach.

Each community detection algorithm has strengths and weaknesses, and is thus more or less suitable for certain types of networks. For example, Louvain grouping has proved to be extremely successful, but is ill-defined for dense weighted networks. Further a “no free lunch theorem” for community detection networks has been proven, stating that no algorithm is optimal for all community structures [170]. As such, a temporal partitioning procedures which are independent of grouping algorithm are preferred over those that rely on a specific method, such as Generalized Louvain [151].

A partitioning method independent of community detection algorithm can be constructed using the co-group network. That is, the network between nodes where links represent shared group membership. After calculating the co-group matrices, we will compare them to find natural partitions in time. This way, the user can choose any algorithm they like to group the slices before the temporal analysis begins. It is important to note that comparing co-group networks relies heavily on the static nodeset assumption. One way of comparing two networks is through the Product-Moment Correlation [120]. The

correlation, ρ , between two co-group matrices A , and B is given by:

$$\rho(A, B) = \frac{\text{cov}(A, B)}{\sqrt{\text{cov}(A, A)\text{cov}(B, B)}} \quad (3.1)$$

$$\text{cov}(A, B) = \sum_{i,j} (A_{i,j} - \mu_A)(B_{i,j} - \mu_B). \quad (3.2)$$

Throughout the work we define $A_{i,i} = 1$ in co-group matrices, to indicate that a node is always in its own group.

We take what is known as a “snapshot” approach to dynamic networks in the language of Rossetti and Cazabet [190]. That is, the initial temporal network can be defined as a series of adjacency matrices, A_1, \dots, A_T , at every time step, $1, \dots, T$. After these are all grouped there is a series of co-group matrices, CG_1, \dots, CG_T . Pairwise similarity is calculated between all of these slices to obtain the similarity matrix, S :

$$S_{t_1, t_2} = \rho(CG_{t_1}, CG_{t_2}). \quad (3.3)$$

Under the assumption that community change occurs rapidly, S will have near block-diagonal structure. Communities will remain roughly unchanged between events, causing diagonal blocks in S with high similarity. After changes, two segments will be fairly different, resulting in low similarity in S ’s off-diagonal blocks. We now formally define a method for locating each block’s boundaries, which correspond to community change points.

Now that similarity is defined for our temporal network, we can formalize the goal of “cohesive groups” as *high internal similarity*. For example, if a time segment begins at $t = 10$ and ends at $t = 20$, the values within the square similarity matrix $S_{10:20, 10:20}$ should be high. As such, we define the time partitioning problem to maximize this term. The general temporal partitioning problem is to find a list, \mathbf{b} , which contains start and stop points, or boundaries, of time segments. We will assume there are P partitions, and \mathbf{b} is strictly increasing with fixed ends $\mathbf{b}_1 = 0$ and $\mathbf{b}_P = T$, so that the partitions are well defined. Then, the problem is stated as:

$$\arg \max_{P\mathbf{b}} s_{internal}(S, {}^P\mathbf{b}) \quad (3.4)$$

$$s_{internal}(S, \mathbf{b}) = \frac{1}{n_i} \sum_{k=1}^{P-1} \sum_{i=b_k}^{b_{k+1}-1} \sum_{j=b_k}^{b_{k+1}-1} S_{i,j}, \quad (3.5)$$

where n_i is the number of entries in all the internal blocks, and i, j are the indices of the sub-matrices, and ${}^P\mathbf{b}$ enforces that \mathbf{b} be length P . The left-superscript indicating the length of ${}^T\mathbf{b}$ is dropped in other calculations for convenience. Given P , this problem can be solved quickly, especially considering that P will typically be small in practice. This could potentially be written as a dynamic program, though with the scale of data used here such an improvement is unnecessary.

However, P should not have to be given. Domain knowledge might give a reasonable guess as to what P should be, but there is no telling whether or not the structure of groups would have actually changed due to outside events. There could also be unknown events.

A natural criteria for determining P comes from the second stated goal: find adjacent segments which are noticeably different from each other. Now, P can start at its minimal possible value, 3, and be iteratively increased to meet both goals. Just as segment similarity is encoded in S , so is segment difference. Using the example from before, where a segment begins at $t = 10$ and ends at $t = 20$, add the fact that the next segment ends at $t = 30$. If these segments are very different the values within $S_{10:20,20:30}$ should be low. Calculating this difference for a full time segment with length P , the external similarity is:

$$s_{\text{external}}(S, \mathbf{b}) = \frac{1}{n_e} \sum_{k=1}^{P-2} \sum_{i=b_k}^{b_{k+1}-1} \sum_{j=b_{k+1}}^{b_{k+2}-1} S_{i,j}, \quad (3.6)$$

where n_e is the number of entries in all the external blocks, and i, j are the indices of the similarity matrix, and k indexes the boundary list \mathbf{b} . Naturally, there is a tradeoff between these values. We can expect the internal similarity to increase even beyond the optimal partitions, albeit more slowly. This happens by splitting already cohesive sections into more cohesive segments. When this happens, the external similarity should begin to increase. Since similarities will be calculated iteratively, a subscript will be used to denote which iteration the similarity was calculated on. For example, s_{internal}^1 is the initial internal similarity.

Increasing external similarity is a sign of over-partitioning, so is naturally a good stopping criteria. However, we cannot expect real-world data to exhibit perfect block-diagonal structure. Thus, we propose a stopping criteria based on the rate at which each similarity changes: $\Delta s_{\text{internal}}^t, \Delta s_{\text{external}}^t$:

$$\Delta s_{\text{internal}}^t = \frac{s_{\text{internal}}^t - s_{\text{internal}}^{t-1}}{s_{\text{internal}}^1}, \quad (3.7)$$

replacing “internal” with “external” yields the equation for $\Delta s_{\text{external}}^t$. If $\Delta s_{\text{internal}} > -1 * \Delta s_{\text{external}}$, too many partitions have been placed. The negative sign is due to the expectation that $\Delta s_{\text{external}}$ is negative. Thus, the final algorithm is given as Algorithm 1.

This two-step procedure first ensures that the segments are as similar as possible, and then stops when increasing P fails to yield additional segments that are meaningfully different. Combining the two goals into a single objective function sometimes led to inaccurate partitions, since segments were blended in order obtain low external similarity.

The algorithm could be forced to continue, which will give more breakpoints. These could also be interesting but it should be remembered that they are breaking the trade-off that we set out to balance, that between internal and external similarity, so they are likely to be less meaningful than initial break-points.

Adjustments for Semi-Static Nodesets

As stated, the above method of finding optimal time segments requires that nodes be present in each time slices. However, many datasets do not meet this requirement. In

Algorithm 1: Dynamic Partitioning

Result: List of temporal partitions, \mathbf{b}

- 1 $\mathcal{P} \leftarrow 3$ the number of partitions, including endpoints;
- 2 ${}^P\mathbf{b} \leftarrow \arg \max_{P\mathbf{b}} s_{internal}(S, {}^P\mathbf{b})$;
- 3 **while** $\Delta s_{internal} > \Delta s_{external}$ **do**
- 4 4 $P \leftarrow P + 1$;
- 5 5 ${}^P\mathbf{b} \leftarrow \arg \max_{P\mathbf{b}} s_{internal}(S, {}^P\mathbf{b})$;
- 6 **end**
- 7 $\mathbf{b} \leftarrow {}^{P-1}\mathbf{b}$;

some datasets, such those derived from social media, key actors are present in *most* of the time slices. Before applying the methodology, we will now make changes that allow nodes to be missing in some slices.

Again, each time-slice is grouped, and the co-group matrix is calculated. However, now that nodes in our nodeset can be “out of the network” completely, isolate nodes should not be allowed. Instead isolate nodes will be entered as “not a number” in the co-group matrix. Then, the correlation function is adjusted to ignore these values in calculation. Basically, the correlation now is calculating the similarity between groups of nodes present in both time slices.

Additionally, an added assumption can fill in some missing data: nodes do not change group affiliations in slices they are not present for. That is, if a node is not present until $t = 4$, is present in $t = 5$, and then is not present again, it will have “not a number” co-groupings until time $t = 5$, then it will retain the time $t = 5$ co-group ties for the rest of the dataset. To be clear the assumption is that nodes only change group through forming other links, or the lack of a link cannot be used to change a node’s group. Theoretically this seems like reasonable way to fill in missing data. In practice, however, this added assumption actually blends the time slice networks together, making it harder to establish temporal partitions. Thus, this assumption is not made in this work.

Now, nodes can enter and leave the dataset. However, large sets of infrequent nodes can disrupt results. For example, if many nodes are present in an early slice, but never return, it is possible that subsequent slices appear more correlated than they should. Issues like this can be resolved in two steps. The first part of this issue stems from the concept of time slices itself. What is a time slice? It depends on the dataset, but often it is up to the user to define a sensible time slice. Slices are arbitrarily selected as regular intervals like days, weeks, or months. Ideally, the length of the time slices should be 4-10 times shorter than that which meaningful change might occur for that network [206]. Given the somewhat arbitrary nature of the current slice length choice practices, we suggest that the node frequency should also be taken into account. Again considering an email network, it is not reasonable that everyone emails every single day, but most people send at least an email a week, so this may be a better time slice. A month would include even more users over a frequency minimum, but offers less resolution for temporal changes. These trade-offs must be looked at on a case-by-case basis.

In the case of semi-static nodesets, many nodes in the network may appear infrequently. If a researcher would like to answer the question “how do *core* members of this network change their community structure?” Infrequent nodes may want to be filtered out. This is not a necessary step, as nodes will only have an impact on results when they *are present*, however, large numbers of less important nodes can obscure results and make analysis more challenging. As such, a node frequency threshold can be introduced. After the network slices are created, define n_i as the number of slices node i is present for. Then, define λ as a threshold such that all nodes with $n_i < \lambda$ are removed from the analysis. For example, $\lambda = \frac{3}{4}T$ retains nodes that are present three quarters of the time. This node-filtering step is prevents large numbers of sporadic nodes from skewing the analysis of our segments.

This is not a parameter to be tuned or adjusted to get better results. Rather, it is an optional pre-processing step which may be helpful for researches interested in the evolution of nodes that are frequently present. In the case of social media discussions, we are most interested in how communities change around influencers or those central to the discussion at hand.

3.3.2 Validation

Synthetic Networks

Network datasets with ground truth communities are rare. Rarer still, are network datasets with ground-truth community disruption. Therefore, we test the validity of our approach using a series of experiments on synthetic datasets. With synthetic datasets, we can impose change points and access our ability to recover them. Throughout these experiments we chose a nodeset size of 500. Additionally, we are only considering Erdős-Rényi random networks of varying density between experiments. Random networks typically have low modularity, and as such are a good test case. Further, Peel et al. have concluded that verification on embedded communities is flawed, so we do not rely on manually embedded communities here [170]. If our algorithm can detect changes in weak communities, it should perform well in the easier case of strong communities. For all tests, a temporal network with 20 slices was considered. The ground-truth breaks were placed at $t = 4, 8, 17$. Every experiment was repeated 100 times.

First, a very basic set of tests were performed. At time $t = 0$, a random network was constructed. Then, each slice up until the first break was set to this network. When a break occurs, a new random network was generated and the process repeats. Basically, slices are identical within breaks, but completely different random networks between breaks. In this case, then, the ideal internal similarity score is 1, and the external similarity will likely be close to 0. This construction method was tested with density 0.2, 0.1, and 0.5. For each experiment the algorithm gave the exact set of breaks for all 100 repetitions.

Second, a more realistic set of tests were performed. Again, a random network was generated and slices were set to that network up until a break. This time, when a break occurred some fraction, f , of the links were randomly rewired. In these experiments, internal similarity is still 1, but the network is retaining some of its original structure throughout the timeline, giving higher external similarity. For these experiments, density

was held to a constant 0.1, but f was varied: $f \in [0.5, 0.1, 0.05, 0.01]$, to measure how the algorithm performs on changes of varying scales. For $f \in [0.5, 0.05]$ the exact partitions were recovered in all 100 trials. For $f \in [0.1, 0.01]$, the partitions were recovered in 99 trials. In the remaining trial the break at $t = 4$ was not detected, while the other breaks were. As f decreases, it is increasingly likely that the underlying communities do not change, so it is expected in some instances we will not see breaks. Given our results up to changes in only 1% of links, it seems that our method is extremely accurate in a noiseless environment.

Lastly, two additional experiments were conducted to study the effect of noise. Now, since the networks in question have low modularity, random changes in links can potentially have a large impact in group structure. To test this we followed the same procedure as in the second step of experiments, but adding the additional set of swapping some fraction, n , of the links *at each time slice*. In these experiments we held density to a constant 0.1, f to a constant 0.5, and tested n values of 0.01 and 0.05. The exact partitions were recovered in 88 and 66 trials, for 1% and 5% rewiring, respectively. Typically, the algorithm had only one of the following errors: one of the partitions was misplaced by 1 slice, one of the partitions was missing, or an additional partition was added.

Given the extremity of the experimental conditions (testing on networks with low community structure and high sensitivity to noise), and the results ($>99\%$ accuracy in noiseless scenario, $>65\%$ accuracy under extreme noise), these results bolster the method's validity.

Ukrainian Legislature

It is known that the Ukrainian Parliament, the Verkhovna Rada, has interesting political groups within it called “factions” [112]. Factions are interesting as they extend beyond party boundaries and change dynamically. Prior work shows that factions can be obtained from network analysis of voting data [176, 177]. Further, there is known to be a large disruption of alliances in convocation 7, spanning from 2012 to 2014, in which a revolution took place.

Thus, in this validation case study we analyze the Rada voting data from convocation 7, which is available publicly¹. The dataset from this convocation analyzed included 493 bills, over 91 time slices. Time slices are defined using the day in which bills are registered. Some time slices may have multiple bills, some may not. There are six voting options in the Rada: for, against, did not vote, no vote, absence, and abstain. Domain experts have suggested that votes other than for and against are all used to mean the same thing: they are not in favor of the bill, but do not want to send a strong signal against it. As such, these votes are not considered as ties between MPs. The voting network, then, is the network constructed so that nodes are parliamentarians and the weighted links are the instances of co-voting between two parliamentarians in the given time period.

One method of determining factions from this network is using Louvain grouping [22]. Thus, we use Louvain grouping at each of the 91 slices, and obtain the similarity matrix. Note that this procedure can lead to a significant number of isolates at each time slice.

¹<http://rada.gov.ua/en>

It was determined that there is only 1 partition in the networks timeline, occurring on February 6, 2014. Figure 3.5a shows the similarity matrix with the induced temporal partitions. It can be seen that the “internal similarity” of the partitions is high, often around 0.8, and the “external similarity” is low, often around 0; specifically, the average value internal similarity is 0.25, and the average external similarity is 0.03. While there are many entries in S with low similarity during segment 2, they are not pervasive. Meaning, since the overall block has high similarity, these intermittent entries of lower similarity are normal variations in the communities. If a meaningful change was occurring, slices would not be highly similar to entries within the block on average, i.e. a new diagonal block would form in the matrix.

The Ukrainian revolution started in February of 2014, culminating in the overthrow of the Ukrainian government and the removal of President Victor Yanukovych. As expected, our algorithm returns this as the most significant change point. Forcing our partitioning to continue revealed another date: May 15, 2014. This is interesting in that it is the last slice before the presidential election, occurring on May 25. Again, groups are highly correlated before and after these events, so there was not much of an impact on the communities overall.

We compare results to a popular alternative method, Generalized Louvain [151]. Generalized Louvain requires two resolution parameters: ω, γ . Currently, there is no way of objectively selecting these parameters. So, we perform a grid search over the parameters suggested in their initial work, adding additional values making the total space: $\omega \in [0.25, 0.5, \dots, 4, 5, 6, \dots, 10], \gamma \in [0, 0.1, 1, 1.5, 2, \dots, 6]$. Out of these 286 possible combinations, only 14 partitioned the data less than 10 times. Finally, one partition was best in terms of all score metrics (internal similarity, external, ratio, difference), which was obtained from $\gamma = 2.5, \omega = 8$. This parameter combination yields one partition in the voting data, on April 4, 2018. This partition is visualized in Figure 3.5b. Clearly, this result is sub-optimal and does not accurately correspond to the known disruption to the network.

Now that the temporal network has been segmented, we can analyze the resulting two static networks statically. First, we visualize the network in Figure 3.6. This figure shows that the initial two communities within the Rada relied on only a few parliamentarians to bridge the gap between them. Then, we see that after the event, the community structure cannot be discerned visually, indicating poor grouping after the event. This result is confirmed quantitatively using Louvain modularity; the initial time segment had modularity 0.139 while the second only had 0.024.

To better understand how the groups have changed, a Sankey Diagram is displayed in Figure 3.7. We see that group A transitions from holding a majority to holding only a third of the seats. This is due to a large number of its constituents joining members from group B to form a third group, and a smaller number of constituents joining the opposing group. Both before and after there is a small number of MP’s failing to cast a significant vote in each time period. Here, significant refers to the fact that “co-voting” relies on non-abstention, and some MP’s strictly cast abstention-type votes.

This validation examples highlights the need for dynamic community detection: aggregating the entire convocation 7 data into one network would have lost the two very different behaviors seen in the data. Our methodology partitions the temporal network at

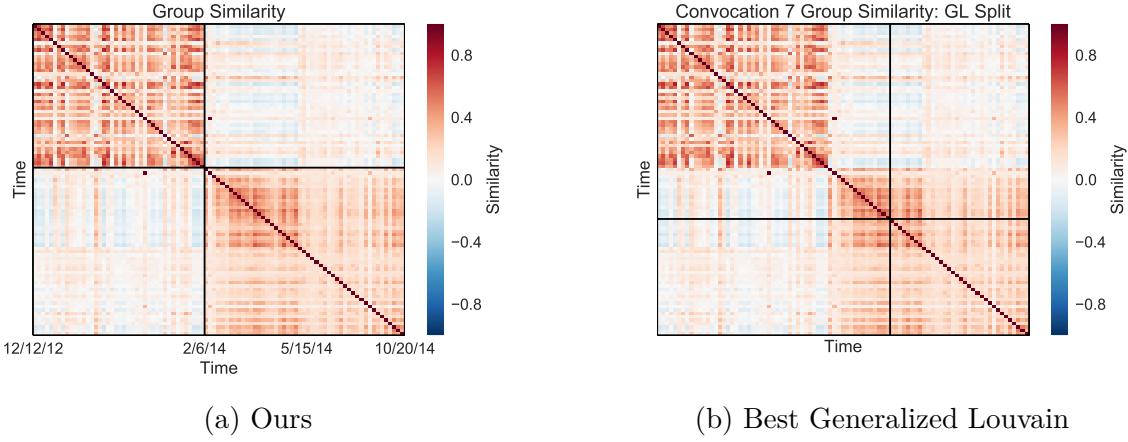


Figure 3.5: The Rada’s temporal similarity matrix for convocation 7, created using Louvain grouping. The temporal partitions are drawn in black. The partition found with our method occurs on February 6, 2014, which is the start date of the Ukrainian Revolution. The best-fit partition found with Generalized Louvain occurs on April 4, 2014.

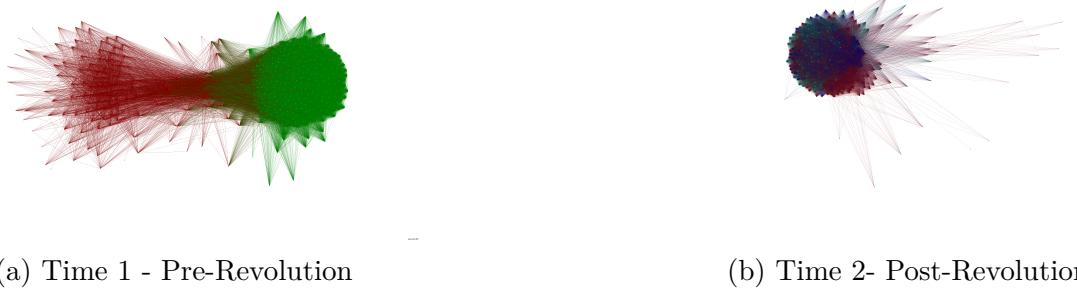


Figure 3.6: Network visualization of the Rada in the first time segment in (a) and the second in (b). Links below the mean link value are not shown, nodes are colored by Louvain grouping for the individual time segment. The initial time segment shows clear group structure in the first time segment, with a minority group detached from a central core. The second time segment does not have clear groups, because the groups found are all inter-related.

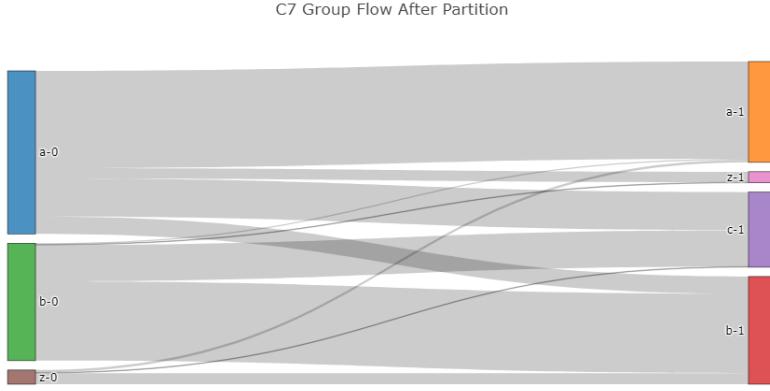


Figure 3.7: Sankey visualization of the flow from Rada groups in time segment 1 to time segment 2. Groups in the “z” category are isolate nodes that were lumped together for convenience. The main finding here is the birth of a third group, which drains the controlling faction of its majority, giving the opposing faction the most power after the revolution.

exactly the point that the legislature Ukrainian revolution began, giving empirical validation to our results. Additionally, allowing our algorithm to over-partition the data revealed two other change points, centering around the election. Finally, we show that performing Generalized Louvain with 286 parameter combinations led to only 14 usable partitions, all still with sub-optimal results. Additionally, the results from Generalized Louvain could not easily be compared without introducing some similarity measures, as we have done here.

3.3.3 Results

We now apply the dynamic partitioning approach to the contextualized networks. The first major point is that dynamic partition quickly proves to be inappropriate for contextualized networks in the Reopen dataset. The contexts in that dataset were either spread out too thinly across time, or too concentrated. The largest context, Liberate Tweets, was heavily concentrated within 6 days, 3 days in April and 3 days in May. Clearly, these sections should be considered separately, but within those two blocks, there was not enough data to warrant a full dynamic partitioning. On the other hand, discussions like COVID Information were spread out over nearly the full timeline, and had very few persistent nodes, which also made dynamic partitioning inappropriate.

On the other hand, the contexts in the *Election* dataset were concentrated over the 7 days surrounding the election, making them ripe for dynamic partitioning. The temporal similarity matrices for the two largest contexts, Biden Campaign and Claims of Fraud, are given in Figure 3.8. There, we see that the group structure similarity between snapshots in both contexts is quite low, with values from 0.1-0.2, though correlation is generally higher in the Biden Campaign context, indicating that group structure was more stable there, though it was still unstable.

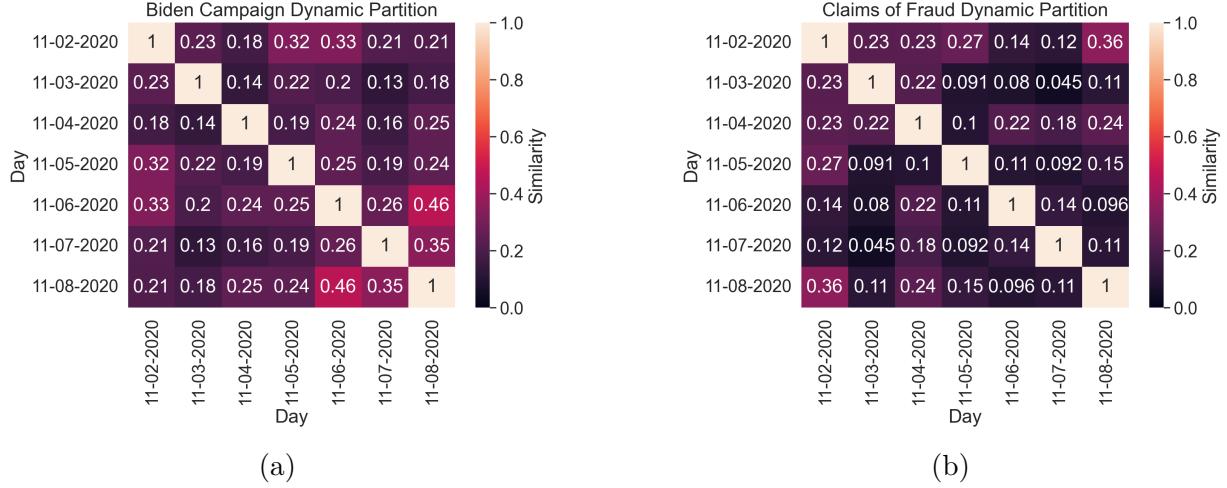


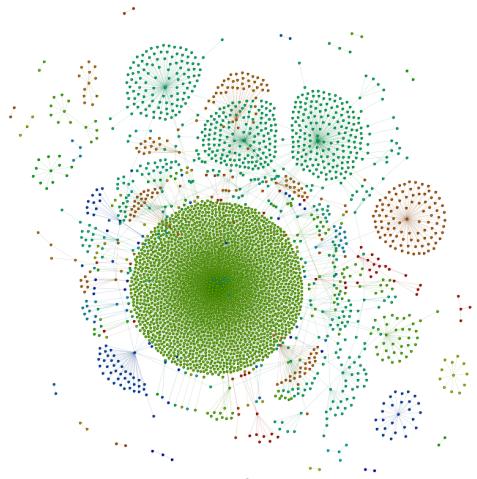
Figure 3.8: Temporal similarity matrices are shown for the Biden Campaign and Claims of Fraud contexts. In each case, the optimal partition considers each day independently.

A visual inspection of the dynamic partitions in Figure 3.8 would suggest that the snapshots are too unrelated to be grouped together. That is, every day should be considered its own network. The partitioning algorithm confirms this intuition by selected the optimal partition as each day in its own snapshot.

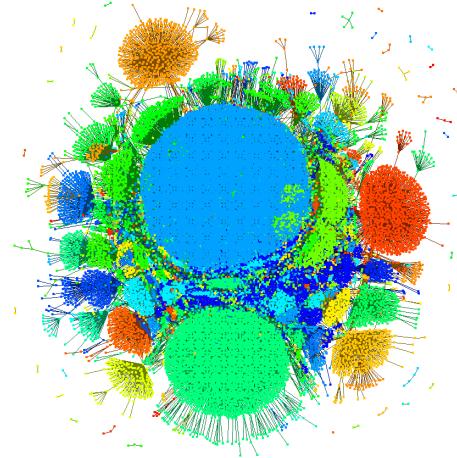
Clearly there is different network structure in each of the network snapshots in the contextualized networks, since their similarity is typically very low. To further emphasize this point, the network snapshots for the Biden Campaign context are visualized in Figures 3.9 and 3.10. Each snapshot shows a user-user network, where users are connected on the basis of any interactions they had within the Biden Campaign context. Network communities were uncovered on each snapshot using the Leiden method, and are used to color the nodes.

Visual investigation of the network snapshots show that there are major differences in network structure from one day to the next. All days have a large sub-graphs with hub-spoke structure, which is extremely common on social media and occur when many users interact with only one or two viral posts. However, the number of hubs, their size, and their membership clearly changes from day to day. For example the largest community in Day 2 is much larger than all the other communities, while there are 3 large communities of similar size on Day 3. We also see changes in network sparsity. Day 1 has not nearly the same amount of activity as Days 2-6. Even within the high-density days there is variability; density decreases from Day 4 to Day 5, before increasing again.

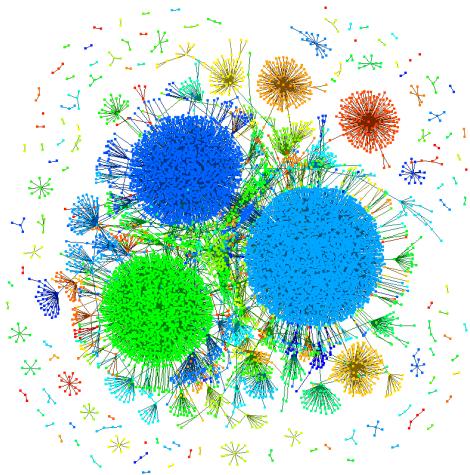
Visualizations of the snapshots from the Claims of Fraud context are not shown, but their correlation tells the same story; different days have very different network structure. Through our dynamic partitioning approach we can now separate these structures out and more accurately analyze them. If we were to mix these snapshots together, the underlying community structure would be harmed, in much the same way as contextualized mixing harms our view of networks.



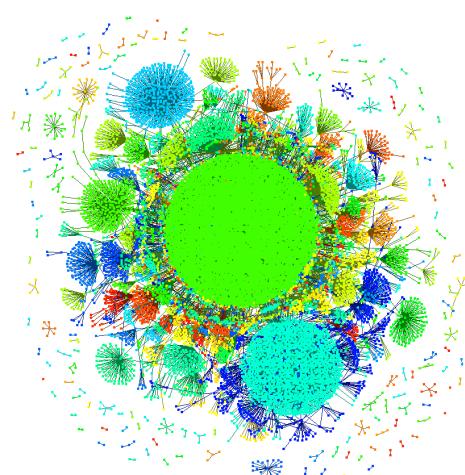
(a) Day 1



(b) Day 2 (Election Day)

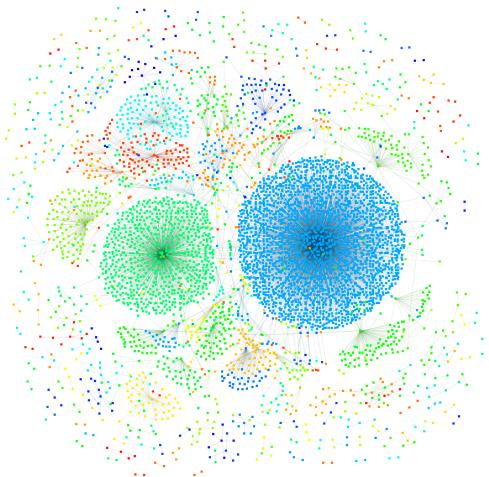


(c) Day 3

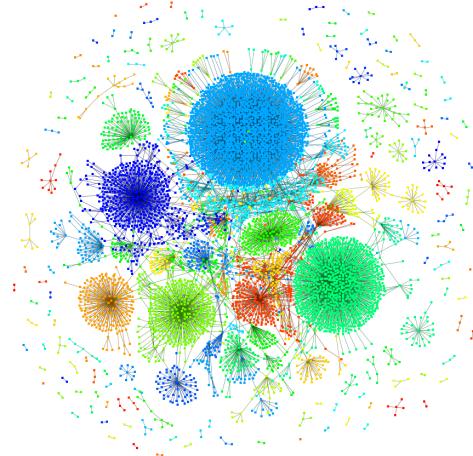


(d) Day 4

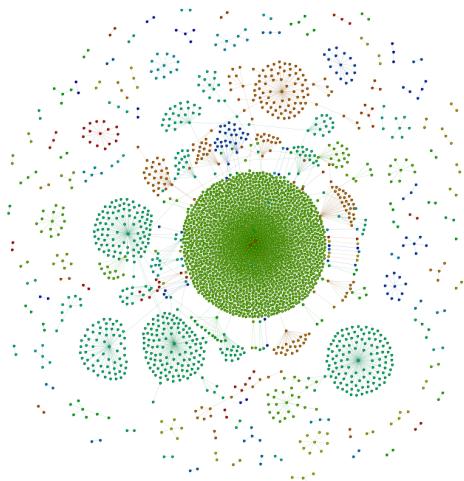
Figure 3.9: Snapshots 1-4 of the dynamic all communication network in the Biden Campaign context. Each node is a user, colored by its Leiden community for that snapshot. Similar colors do not indicate similar community across snapshots. Users are connected on the basis of all communication.



(a) Day 5



(b) Day 6 (Election Called)



(c) Day 7

Figure 3.10: Snapshots 5-7 of the dynamic all communication network in the Biden Campaign context. Each node is a user, colored by its Leiden community for that snapshot. Similar colors do not indicate similar community across snapshots. Users are connected on the basis of all communication.

3.4 Inter-Context Activity Dynamics

We now move to an exploration of *inter-context* activity dynamics, which we distinguish this from the inter-context *network* dynamics studied in Section 3.5. Here, we demonstrate that the movement of users between contexts uncovers a roadmap of the datasets conversational dynamics, showing how users flow from one context to another.

The inter-context activity dynamics are analyzed by calculating the probability that a user will move from one state to another. We model this as a Markov chain. The Markov chain model makes two assumptions. First, it assumes that the probability of changing from one state to another state is constant in time. Next, it is “memoryless,” meaning that the user’s next move only depends on the current position, not where they came from. The probabilities are stored in a transition matrix, P , where $P_{i,j}$ gives the probability of a user transition from context i to context j . Note that the probability matrix is not symmetric.

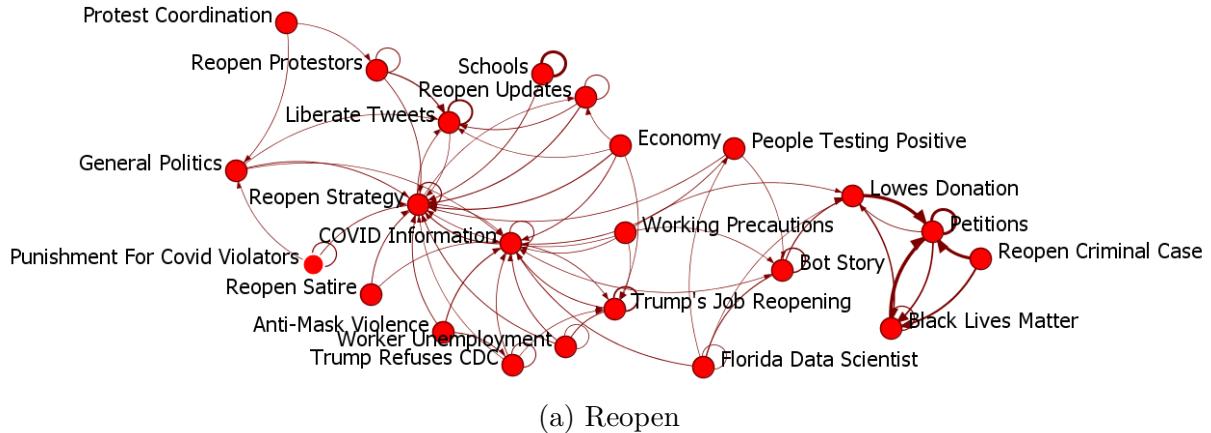
The transition matrix is calculated on real data by the following process. First, the time-ordered sequence of Tweets is constructed for all users in the dataset. Next, the Tweet labels are used to convert the previous sequence to a sequence of conversational contexts for each user. Following this, the total sum of *transitions* is calculated and stored in a total-transition matrix, T , where $T_{i,j}$ indicates the number of instances that a user began in the i context and transitioned to the j context. Finally, the total-transition matrix is row-normalized to give the transition probability matrix, P .

Self-loops indicate the probability that users continue tweeting in the context that they are in. Thus, the strength of a self-loop in the transition matrix can be considered the “stickiness” of a conversational context. While this may be useful information, self-loops make it harder to illustrate the between-context dynamics, which are of primary interest here. Thus, we exclude self-loops in the following analysis. By excluding self-loops, the transition matrix indicates the probability that a user transitions from one state to another, given that they are forced to transition somewhere (they cannot stay put).

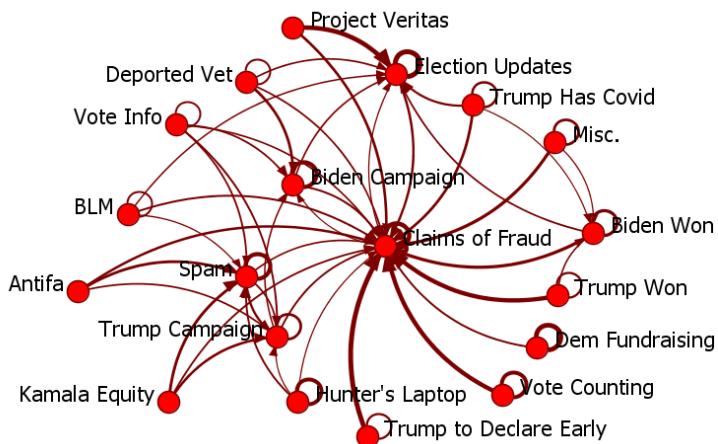
The conversational context transition matrices for the *Reopen* and *Election* datasets are shown in Figure 3.11. For a large dataset, there is usually some low but non-zero probability that a user will transition from any given context to any other. This creates a fully-connected graph that is hard to interpret. Trimming the low probability transitions from the network provides a much more useful diagram of the dynamics. Note how this compares to the static nodeset relationships between contexts shown in Figure 2.9.

3.4.1 Categories of Inter-Context Dynamics

We will explore the categories, or motifs, of inter-context dynamics. For each category, the transition networks of the *Reopen* and *Election* dataset, as shown in Figure 3.11, will be referenced to demonstrate the real examples of each pattern.



(a) Reopen



(b) Election

Figure 3.11: The transition networks are shown where arrow width indicates the probability that a user who is actively Tweeting in the source context will transition to Tweeting in the target context. Probabilities below 0.1 are removed

Sinks

Conversational sinks, or attractors, are conversational contexts with a high in-degree in the transition network. That is, many contexts have a high probability of transitioning to the sink. They are visually identified by having many strong arrows pointing to them in transition diagrams like that in those in Figure 3.11. Sinks occur when a conversational context is so important that they draw many users to it from different conversations.

In the *Reopen* dataset, we see that *Reopen Strategy*, *COVID Information*, and *Liberate Tweets*, are all sinks. These discussions are all core aspects of the larger Reopen discussion, as they draw user from many periphery conversations to more centralized ones. For example, we see that the discussion about strategy for reopening draws in users from discussions about politics, anti-mask violence, schools, worker unemployment, and many other discussions.

Conversational sinks are even more clear in the *Election* dataset, with *Claims of Fraud* being the dominant sink, but *Election Updates* and *Spam* also showing the behavior. While the votes were being counted, Trump supporters made claims of fraud while other debunked them. Because there were no results, there was not much else to discuss, allowing the fraud conversation to pull many different users into it.

Sources

Conversational sources play a similar role to conversational sinks. Sources are contexts with a high out-degree in the transition network. This means they are conversational contexts that disperse users to many different contexts. At first, this appears to be a very different behavior than sinks. It may seem that sources are less important, because they appear to be repelling users. However, this is just an artifact of the ordering of discussions.

Consider a simple example. If a very compelling conversation develops at the end of a dataset's timeline it is likely to draw in users from all sorts of conversations. Thus, it will appear as a sink. However, if that same conversation occurred at the *beginning* of the dataset the diverse set of users would begin at the conversation of interest before dispersing to many other side discussions. In this case, it would appear as a source.

This problem means that we must be very careful when attributing importance based on a context's status as a source or sink. The simple fact that a context occurs at the beginning or end of a dataset's timeline makes it more likely to be a source or sink, respectively. This is to say that consideration of the qualitative details of a context must be taken in conjunction with its position within the transition matrix to fairly assess its role in the greater discussion.

Both of our example datasets start out dispersed and become more centralized, as observed by the general flow of conversational contexts with low in-degree to those with higher and higher in-degree. So, in this dataset, strong examples of sources are not present.

Cycles

A conversational cycle occurs when users bounce back and forth between two or more discussions. This occurs because two conversations are active at the same time and they

are related. They are observed in the transition network as graph-theoretic cycles. In simple terms, they are observed when you can start on one context, follow certain arrows, and end up on the original context.

In the *Reopen* dataset, the strongest cycle is between the *Black Lives Matter* and *Petitions* conversations. This pattern shows us that users are oscillating back and forth between these related discussions.

Splits

Splits in the transition network occur when a context has a high probability of transitioning to two other contexts. Three-way or higher-level splits could be of interest, but this behavior is hard to distinguish from that of a source. Splits are notable because they indicate a conversational context that have users with different priorities or interests.

For example, the *Project Veritas* context splits to *Election Updates* and *Claims of Fraud*. This split indicates that there is a joint interest in the two conversations by those in the *Project Veritas* conversation. As more events unfold and the original discussion fizzles out, the groups split to conversational contexts that are more interesting to them. Similarly, *Reopen Criminal Case* splits into *Petitions* and *Black Lives Matter*. Often, the ends of a split will themselves be related, as is the case in both of our examples. If the two ends were observed to be unrelated, that would indicate that the initial conversation had two distinct groups that were only temporarily engaging with each other.

Sequences

Lastly, sequences appear as “lines” in the transition network, where one context only transitions to the next context. The simple and linear nature of this pattern indicates that it represents a single, evolving conversation. Users are likely to continue on interacting as the conversation drifts to other topics.

The dynamics in each of the datasets are too complex to show sequences greater than length 2. With that said, an example sequence is the transition from *Trump to Declare Early* to *Claims of Fraud*, as the users who were sounding the alarm about these claims before hand turned to discussing them as they happened.

3.4.2 Sequences and the Temporal Order of Contexts

As was briefly discussed while introducing conversational sources, the order of contexts does affect the potential roles that contexts take on in the transition network. Because contexts overlap, there is not necessarily a clear ordering. However, there may be contexts which have much more activity later in the dataset compared to those earlier. Referring to the simplified timeline in Figure 3.1, we could see that the Black Lives Matter context generally occurs after the others. Thus, Black Lives Matter can't have out-transitions to the bulk of conversations in the dataset, because they are already over.

This is to say that, to some extent, the ordering of events is built-in to the transition networks, and they can be accessed through the dominant flow of transitions. Combining

this analysis with that of the activity plots should give the best results.

The transition diagrams shown presuppose that a user makes a transition. That is the continue to be active in the conversation. It is possible that many users in one conversation are only active in that one conversation, though this cannot be accessed from the transition plots shown. Instead, a null or empty state can be added to the transition network, indicating when users are likely to leave the discussion. By definition this is a sink, since you cannot leave the empty state. The probabilities of leaving a dataset are generally very high due to the heavy-tailed engagement of users in discussion. Because of this, the inclusion of a empty state does not yield transition networks that are as visually interpretable, so they are excluded here.

3.5 Inter-Context Network Dynamics

The inter-context network dynamics are the last aspect of the interactional contexts dynamics left to consider. While we previously looked at the inter-context activity dynamics to understand more about the contexts themselves, here we will learn more about the relationships between users. Specifically, we will look to find users that make similar transitions between contexts.

Users making similar transitions between contexts can be useful for two primary reasons. At a high level, this can be seen as a form of dynamic topic groups [17]. In previous work, topic groups have been seen as groups of users who engage with similar topics of discussion. Expanding on this, groups of users who transition between discussions together could be seen as more tightly related. The second use case is in the identification of co-ordinated actors. Prior work has shown that some groups seek to manipulate discussions by synchronizing their actions [138], though this work only considered simple actions like tweeting a specific hashtag. Here, sets of users making similar transitions could be a group that is working together, deliberately moving from conversation to conversation. This is our primary use-case.

We study these dynamics under the trails framework detailed by Bartulovic, where a trail is a sequence of state-time tuples [16]. Trails capture the sequences studied in the previous section, while adding a dynamic component which indicates that the *time* a transition is made is important. The trail data considered in the literature were relatively slow moving, where transitions between states took days or sometimes weeks. This allowed for the assumption that multiple transitions would not occur within the same time window, which is clearly not the case for our data.

To get around this assumption, we construct hyper-states, another concept from Bartulovic. Now, we consider a context-day pair as a state, meaning “Biden Campaign - 11 / 2” is a distinct state from “Biden Campaign - 11 / 3.” Because of this difference, transitions between these states also encode information about when they occurred. Specifically, the occurred on the date of the target state.

To find clusters of users with similar trails, we need to formally define trail similarity. To do so, we make use of Normalized Trail Similarity, or NTS [16]. For convenience, the

NTS equation is given here as:

$$NTS(u_1, u_2) = \frac{\sum_{i,j \in \mathfrak{s}} \min [\Phi_{u_1}(s_i \rightarrow s_j), \Phi_{u_2}(s_i \rightarrow s_j)]}{\max(|A_{u_1}|, |A_{u_2}|) - 1}, \quad (3.8)$$

where u_1 indicates user 1, \mathfrak{s} indicates the set of all states, $\Phi_{u_1}(s_i \rightarrow s_j)$ indicates the number of transitions from state i to j for user 1, and $|A_{u_1}|$ indicates the length of the full trail for user 1. Trails were constructed for each user in the same manner as sequences in the previous section, except now the hyper-states were considered instead of the original contexts. With trail similarity in place, we take $1 - NTS$ to be a distance and use DBSCAN to find clusters [58]. Lastly, we only consider users with at least 10 transitions in order to make stronger conclusions about their grouping.

We now propose that cluster behavior can be analyzed by considering the frequency of transitions within the cluster. So, for all transitions made by members of a cluster, we count the number of cluster members making that transition. For example, a cluster of 30 users might have 5 members who transitioned from “Liberate Tweets 4-17” to “Liberate Tweets 4-18”, so that transition will be recorded with weight 5/30. This decision prioritizes collective action over total action. Meaning, if two members of the group make the same transition 100 times, it is no different than if they did so once. For other applications this decision might be inappropriate. However we are predominantly concerned with the behavior of the group, so we proceed.

We note that there are other clustering methods available, such as that proposed by Bartuolovic or by Cadez et al [32]. The goal of these methods is to find overall classes of transition behaviors, which is different from our purpose. Here, we are interested with tight clusters, or groups of users with very high similarity. As we will show, classes of transitions are less likely to be uncovered due to each user’s fairly unique trails. Density-based clustering is all that is needed for us to find groups of suspicious users.

We apply this methodology to the *Reopen* dataset, due to its more interesting transition structure as seen in Figure 3.11. We construct hyper-states at the week level. The above procedure lead to 498 clusters of 13175 users. 10206 users were not clustered, and there were only 7 clusters with above 10 users. The largest cluster had 1698 users and a mean NTS similarity of only 0.10. The mean NTS similarities of the remaining clusters were 0.25, 0.26, 0.37, 0.34, 0.49, and 0.36, with the largest having only 17 members.

Following our intuition, the vast majority of users are not transitioning from contexts in a synchronized or even correlated way. Our approach correctly separates out this behavior from the small groups of users who are moving together. To dig deeper into the behavior of these groups, the group transition diagrams for the two most well-clustered groups in Figure 3.12.

We see that there are a number of very strong transitions for both clusters. In the first cluster, we see that the users are transitioning back and forth between Petitions and Black Lives Matter, first on the 23rd week and then the 24th, with over two thirds of the group making all of these transitions. In the second cluster, we see more complicated behavior, but nonetheless strong transitions stick out. These strong transitions make a backbone that we can use to track the group through both context and time. While there are many

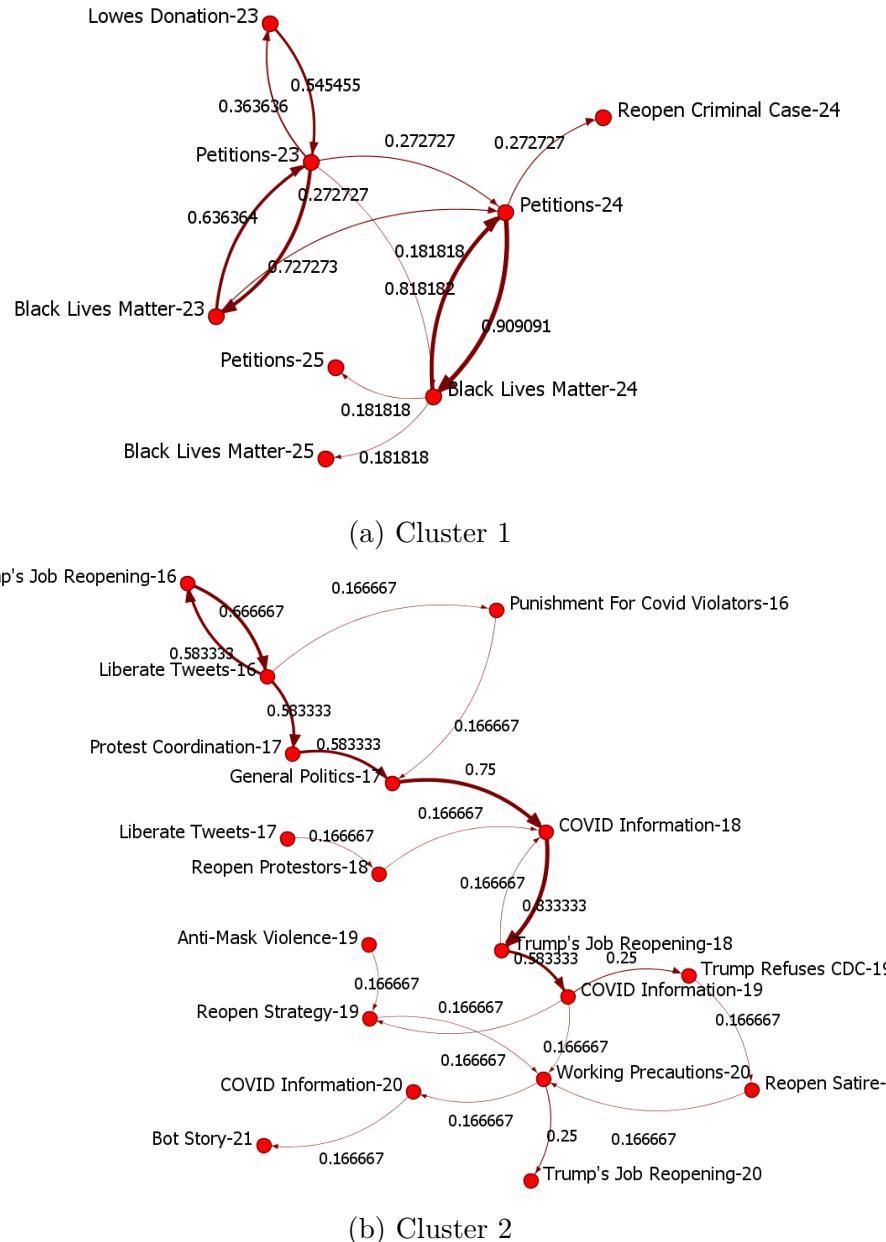


Figure 3.12: The transition networks are shown where arrow width indicates the fraction of cluster members making that transition. Fractions below 0.1 are removed

less frequent transitions, the dominant pattern of the group was oscillate between *Trump’s Job Reopening-16* and *Liberate Tweets-16* before moving to *Protest Coordination-17*, to *General Politics-17*, to *COVID Information-18*, to *Trump’s Job Reopening-18*, to *COVID Information-19*. Given the large fraction of the group making all of these transitions and the fact that these movements take place over multiple weeks, the behavior is suspicious and possibly coordinated.

In summary, the trail clustering method developed and tested in this section allows us to go beyond sequences and consider when transition occurred. Further, we now relate this to users, where we identify clusters of users who create similar contextual trails according to NTS. Investigating the tightest of clusters points to suspicious groups who make highly coordinated moves through conversational contexts. Relaxing the investigation to less-similar clusters may give insight into dynamic topic groups. However, our results indicate that the majority of users show unique trail patterns indicating that the presence of such dynamic topic groups may be rare.

3.6 Discussion

In four parts, we have detailed methods to uncover the rich interactional context dynamics on social media. In doing so, we have shown that contextualized network analysis not only gives us a clearer view of conversational networks, but they also allow us to characterize large online discussions in ways previously unavailable.

The initial building block of this dynamic analysis is the activity curve of a contextualized discussion, which we showed can be used to categorize them by relating them to the categories of collective attention. This analysis allowed us to break down the dataset into sub-timelines and to categorize each sub-timeline. The categories of each sub-timeline were useful in the next phases of dynamic analysis.

The categories of context based on the intra-context activity analysis were used to identify contexts suitable for intra-context network dynamics. Specifically, we needed those which had consistently high levels of activity for at least a few days. In the language of activity curve analysis, these are contexts with prolonged peak periods. Our first finding was that very few of the contexts met this criteria. Thus, while dynamic community detection is often applied to longitudinal social media data, our findings suggest that much of the dynamics are really occurring due to users movements between contexts. Within these contexts, static network analysis is often appropriate.

For the cases where dynamic network analysis is appropriate, we proposed a partitioning algorithm which models the network as a series of static network snapshots, and looks to combine snapshots together. Applying the approach to contextualized networks in the *Election* dataset showed that, for long-lasting contextualized discussions, there can be many phases of discussion. In both of the cases we studied, each day’s snapshot had very different density and social structure. Thus, we have uncovered two dimensions in which a simple static network analysis will misrepresent network structure: context and time.

Moving to the analysis of inter-context dynamics, we first see that Markovian models of contextual sequences, or lists of the contexts that users are active in over time, detail

a conversational map. With this map, the overall flow of large scale conversations can be seen. Additionally, each context's structural position in the transition network gives further insight to its role in the discussion. For example, conversational sinks (those with many in-connections), draw users in from many conversations and often occur later in the overall timeline. Combining these insights with the intra-context activity dynamics gives a full picture of a conversation.

Lastly, we demonstrated that contextual trail clustering can be used to identify groups of user which not only participate in similar discussions, but move through those discussions in a correlated manner. We saw that the contextual trails of typical users are not correlated with others. That is, most users are fairly unique in how they move about through a conversation. Those who are not unique, then, become interesting. Through trail clustering we identified small groups of users who were making very similar transitions between contexts. The highly related transitions, especially in contrast with the normal user's lack of trail similarity, is suspicious and could be indicative of coordinated behavior more complex than current methods have been able to detect.

Overall, it is clear that not only are each of these dynamic analyses useful on their own, but that they work together to characterize a large online discussion. The simple activity curves enable dynamic network analysis within contexts, which in turn color our understanding of contexts while analyzing the transition network. The methods outlined in this chapter provide a framework for linking the dynamics both on and between contextualized discussions. This powerful approach has multiple potential areas of extended investigation, which are discussed in the concluding chapter of this dissertation.

Chapter 4

Dynamics of Online Community Prototypes

In this chapter, we turn our focus to a different type of context in social interaction: personal or identity context. Personal context refers to the social identities of the users interacting, which in turn refers to qualities, beliefs, personality traits, appearance, and/or expressions that characterize a person or group. Figure 4.1 illustrates this concept. In this simple example, the two users belong to the same political party, the US Republican party, but they are supporters of different soccer clubs, Chelsea and Arsenal.

There is a rich literature on social identity and how it relates to social interactions. Much of this theory focuses on the how these interactions lead to group-level behavior known as group processes.

While there is much to learn from the wide literature in this field, there has been a lack of methodology for applying and testing the theories to large scale social media communities. In this chapter, we begin to fill this gap. We do so by providing a method of measuring the presence of community prototypes, a key ingredient guiding group processes. From there, we uncover the prototypes in real datasets and show that they shed light on the nature of online communities beyond just their membership. Finally, we test the theoretical claims that members of social groups who conform to a groups prototype tend to have higher status, and as a result non-prototypical members attempt to become more prototypical.

We operational expressions of social identity through Twitter biographies, the part of a user's profile that enables them to signal who they are. Again, we consider conversational communities by clustering the communication network. Although the previous chapters uncovered the importance of interactional context, this chapter takes the standard non-contextualized approach in order to show how personal identity plays out in communities in terms of the standard approach for their study. In the following chapter, however, we will investigate the interplay between interactional and personal context.

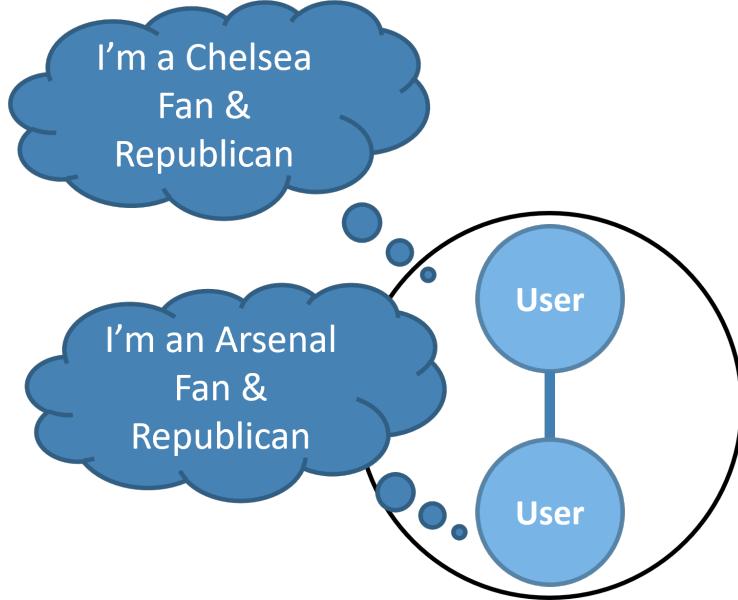


Figure 4.1: Illustration of personal context. The interacting users have differing social identities, which provides insight into their interaction.

4.1 Related Work

Group-level processes are at the heart of many pressing problems on social media such as polarization [108, 153], radicalization [57, 143], and the diffusion of misinformation [52, 74, 146]. For offline networks, the social identity perspective has been taken to make headway on these problems. Social identity theory and self-categorization theory have been validated and used to understand intergroup conflict in organizations [92], Islamic Extremism [5], American political polarization [88, 99, 109], and hostile media perceptions [182, 194]. These successes suggest that the social identity perspective has great potential for understanding these pressing issues in the social media setting. However, the social identity perspective relies on an understanding of the relationship between individuals' social identity and the communities they are a part of. While prior work has found that social media users signal their social identity, the connection to online communities has thus far been unclear.

The core idea of the social identity perspective is that people construct their self-concept in part from the communities and categories that they belong to [100, 207]. Group-level processes then arise from individuals' social cognitive processes based on their social identities. Under self-categorization theory, social identities can be understood through community prototypes, or fuzzy sets of attributes which define a group and distinguish it from others [105, 217]. Attribute's maximization of in-group similarity and out-group difference is known as the meta-contrast principle. Individuals' feelings about themselves and others are functions of their alignment with the group prototype. Thus, developing an understanding of community prototypes on social media is crucial to the application of social identity-related theories.

Researchers have observed multiple mechanisms that social media users leverage to project their social identity. Hashtags are perhaps the most popular tool, enabling users to signal parts of their social identity and community membership in a searchable way [54, 94, 200]. Their searchable nature allows users to find others aligned with their social identity, and form online communities [76, 236]. Beyond hashtags, Twitter users have been observed to signal their social identity in their profile descriptions using “personal identifiers” or phrases that refer to an individual’s social group, category, or role [169, 188, 237]. Users also commonly use emojis to indicate their political beliefs and interests [79, 86, 113, 130]. Subtleties in emoji usage such as emoji sequencing and the use of skin color modifiers have been observed to signal user identity with greater granularity [70, 186].

While prior works have observed social media users signaling social identity through a variety of mechanisms, they have not tied these identity attributes to specific online communities, short of communities that are themselves defined through the use of a single hashtag. These works could, for example, identify users describing themselves as “mothers”, however, they cannot identify whether or not the social attribute of “mother” is salient for some observed community. Meaning they cannot say if there is a community of mothers or if there happen to be mothers in communities that are divided based on other attributes. In other words, identity attributes have been used to understand individual users, but have not been used to understand communities at scale.

The missing link has been a mechanism for quantifying an identity-based attribute’s meta-contrast, or its ability to distinguish a particular community. We develop a multi-view network methodology to tackle this problem. First, we propose using multi-view projected modularity to quantify the *overall* strength of prototypes in the dataset. A low modularity value indicates poor separation between user communities based on their attributes, implying that prototypes are not present. On the other hand, a high value indicates that attributes strongly separate communities and a prototype is present. Then, we develop modularity vitality for bipartite projections in order to quantify an *individual attribute’s* meta-contrast for each community. For a given community, the set of attributes with the highest values are considered the community’s prototype, because those are the attributes which maximally contribute to the community’s in-group similarity and out-group differentiation. The multi-view approach enables the detection of prototypes across the different modalities that users have been observed to signal their social identity such as putting hashtags, emojis, and personal identifiers in their biography. The multi-view approach enables the study of the different mechanisms or modalities of identity signaling seen in prior work.

One of the main outcomes under the social identity perspective is a strong relationship between social identity and within-group status. Under this perspective one of the core internal driver of an individual’s behaviour is their self-esteem, or their self-evaluation [2]. More specifically, they are driven by a need for positive self-esteem. The display of in-group cohesion and out-group distinctiveness increases the positive perception of the individual by the other group members [96]. The positive perception of prototypical individuals makes them more likely to be within-group leaders, or those with higher status within the group. Conversely, those who go against the group’s prototype are treated harshly by the group, often even more harshly than out-group members. This social-control dynamic is referred

to as the “Black Sheep Effect” [139].

We seek to measure the relationship between a social media user’s prototypicality and their within-group status empirically. Methods of measuring within-group status have begun to be developed by Network Scientists under the sub-field of “community-aware centrality” [180]. While classic centrality measures are invariant to the partition of a network, community-aware centrality measures give a ranking for a specific partition. While much of the development in this space has been from a theoretical perspective, these measures have shown promise in identifying spreaders of disease and for understanding metabolic networks [73, 84]. A thorough review of these measures is given in Appendix C.2.3. The specifics of how we use community-aware centrality to quantify within-group status are given in the Methods section.

4.2 Methods

4.2.1 Network Construction and Community Detection

Communities can be defined in several ways. For the investigation of personal context on social media, we are again interested in conversational communities, or groups of users who are more frequently engaged in conversation with each other than with users outside of their community. Following communities, or communities based on following-relationships could also be studied, and there is evidence that these communities hold similar interests and beliefs [18, 110, 130, 131, 239]. However, these communities are roughly static compared to conversational communities, making the salience of attributes difficult to demonstrate. Further, conversational communities are commonly studied when measuring polarization and information diffusion making them most relevant to study for future applications [48, 83, 149, 235].

Communication communities were derived as follows. First, a communication network between users was constructed. This network recorded each interaction between users, with the following actions counting as interactions: reply, mention, retweet, and quote. Combinations of actions were also considered. For example, if a user retweets a reply, that user is connected to both the original author, and the user that was being replied to. These interactions were combined into an undirected user-to-user network, where edge weights indicate the number of interactions between a pair of users. Network statistics for each dataset can be found in Table 4.1. Finally, the Leiden algorithm maximizing modularity was used to uncover communication communities [213]. We note that while the practice of modularity maximization is an extremely popular method it has been criticized from the point of view of inferential network analysis due to its inability to distinguish statistically significant communities from noise, and due to its glassy nature as an objective function [78, 123, 171]. The size of the communication networks prohibits the use of some powerful inferential techniques developed to tackle these problems. Because we are not aiming to make statistical claims about the structure of the online communities, only about their prototypes, we continue with the Leiden approach.

Dataset	Tweets	Users	Edges
Reopen	10,131,537	3,495,506	11,032,399
Election	4,248,125	1,814,513	7,611,473
COVID	29,498,233	9,888,775	35,288,357
Captain Marvel	5,455,142	1,642,434	4,981,094

Table 4.1: Basic networks statistics for each dataset.

4.2.2 Prototype Measurement with Projected Modularity

To measure the presence of community prototypes, we begin by modeling the user-attribute relationship as a multi-view network. Each view of the network corresponds to a different attribute-type. Based on the literature on Twitter users’ ability to signal social identity through multiple modalities, we consider 6 attribute types. From a user’s free-text biography we consider hashtags, mentions, personal identifiers [169], and emojis. We also extract hashtags within a user’s name, and unigrams in their location field. For each attribute type, a user-attribute bipartite network is constructed, where users are connected to the attributes they exhibit. Each bipartite view is projected onto the user nodeset, such that a user-to-user network is obtained. Connections in these views indicate the number of attributes that a pair of users have in common.

More formally, we begin with a bipartite network G . This network connects users to the attributes that they exhibit, only considering a single “attribute-type” for now. The information from this network is encoded in the adjacency matrix, B , where $B_{i,j} = 1$ if user i exhibits attribute j and is 0 otherwise. We fold this network to obtain a user-to-user network where edges are weighted by the number of attributes that the two users have in common. This network’s information is encoded in an adjacency matrix, $A = BB^T$. Examining the folded network allows us to study the relationship between attributes and user communities without clustering the attributes. This process is repeated for each attribute type, v , which results in a multi-view network which each attribute type’s view encoded in an adjacency matrix, A_v .

This framework enables us to quantify the presence of prototypes with the well-known network measure, modularity. While other measures have been proposed to understand the community structure in networks, such as the map equation, modularity uniquely fits the theory from which we are working [192]. More specifically, modularity in this case quantifies the *meta-contrast* exhibited by the communities. Under self-categorization theory, community prototypes are constructed with attributes maximizing *meta-contrast*, referring to the dual goal of simultaneously maximizing in-group similarity and minimizing out-group similarity [217]. The higher the meta-contrast, the stronger the prototypes.

From a network perspective, in-group similarity can be quantified by the number of shared-attribute connections within a community, known as internal edges. Internal edges represent pairs of same-community users sharing an attribute; the more of them there are, the higher the group’s cohesiveness. Given a vector of node communities \mathbf{c} , where c_i indicates the community of node i , the number of internal edges in the network is given by $\frac{1}{2} \sum_{i,j} A_{i,j} \delta(c_i, c_j)$, where δ is an indicator function equaling one if the two arguments

are equal and equaling zero otherwise. Similarly, out-group similarity can be quantified by the number of edges falling between communities, known as external edges.

The balance between internal and external edges can be captured by using the fraction of internal edges. For the fraction to be high, there must be many internal edges *and* few external edges. Thus, the higher the fraction, the larger the meta-contrast, and the stronger the evidence for prototypes. Figure 4.2 illustrates this. It shows that user communities which are well-separated by attributes will have a high fraction of internal links in the projected networks, while user communities which are poorly separated will have a low internal link fraction.

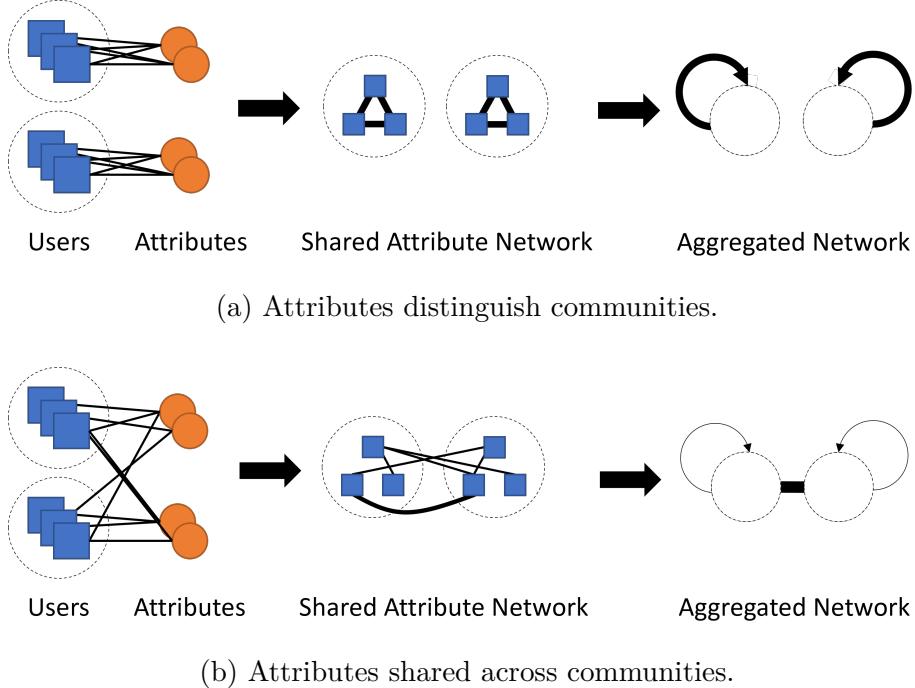


Figure 4.2: An illustration that modularity of shared-attribute networks is indicative of how well-separated user communities are by their attributes. Users are represented by blue squares while attributes are represented by orange circles. The dotted circle around clumped squares indicates that they belong to the same community; there are two user communities in each example. The black arrow indicates the process of projecting the bipartite user-attribute network into a user-to-user network based on shared attributes. When attributes distinguish communities, as in Figure 4.2a, the communities are still well-defined in the projected shared-attribute network. This can be seen by the strong links within the community boundaries in Figure 4.2a, and the absence of links between communities. Figure 4.2b illustrates the case where attributes are shared across communities. This is easily seen in the shared-attribute network, where there are many inter-community edges and few intra-community edges. Projected modularity can quantify this phenomenon, resulting in a perfect score for Figure 4.2a, and a low score for Figure 4.2b.

Modularity was developed to quantify the fraction of internal edges appearing in a network while accounting for those that would be expected by chance under a null model

[158, 159]. The most common form of modularity is Newman Modularity, which considers a unipartite network (not a projection) and using the configuration model as a null model. Though multiple instantiations of the configuration model exist, the attribute networks studied here are considered “simple” where these differences are inconsequential [65]. Barber adapted this to bipartite networks, and Arthur built on this adaptation to develop a modularity for bipartite projections, which is given in Equation 4.1 [10, 15]. The set of communities is given by \mathfrak{C} , F is the number of bipartite edges $F = \sum_{i,j} B_{i,j}$, M is the sum of weighted projected edges $M = \frac{1}{2} \sum_{i,j} A_{i,j}$, the sum of weighted internal edges for a given community is calculated as $M_c^{int} = \sum_{i,j} A_{i,j} \delta(c, c_i) \delta(c, c_j)$, and strength of a community is given by $l_c = \sum_{i,j} B_{i,j} \delta(c, c_i)$.

$$Q^P(G, \mathfrak{C}) = \sum_{c \in \mathfrak{C}} \underbrace{\left(\frac{M_c^{int}}{M} - \left(\frac{l_c}{F} \right)^2 \right)}_{Q_c^P} \quad (4.1)$$

A high value of Q^P means that a high fraction of edges in the projected network are falling within the communities, after accounting for those which would fall there by chance under the bipartite configuration model [10]. Applying this to the user-attribute network, a high Q^P indicates that many more of the shared-attribute relationships are happening between users that are in the same community than we would expect by chance. Thus, high values of Q^P are indicative of communities exhibiting prototypes.

We calculate Q^P for each user-attribute network using the communities obtained by community detection in the *interaction network*. Again, this chapter considers the standard, non-contextualized edges, while the following chapter will add that layer of complexity. The value observed for each user-attribute network indicates the extent to which prototypes are observed using that type of attribute. Along with classic interpretation of modularity, we consider values above 0.3 to give moderate evidence for the observation of prototypes, and values above 0.5 to give strong evidence [159]. We note that we are *not* performing modularity maximization in this step, as the communities have been derived separately on data that does not explicitly include any attributes.

4.2.3 Community-Level Visualizations

To visually depict the modularity values, we plot the expected number of shared attributes from members of different communities (subtracted those expected due to chance) for the top 20 communities of all datasets in Figure 4.4. The process to construct these diagrams is as follows.

For every attribute type, a community-to-community shared attribute network was constructed, after filtering 2% of the non-salient attributes according to the Modularity Filtering method discussed later in this Section. The adjacency matrix, A_v for each view v of the network was calculated as $A_v = (C^T B_v)^T (C^T B_v)$, where C is the user-community indicator matrix ($C_{i,j} = 1$ when user i belongs to community j and is 0 otherwise), and

where B_v is the filtered user-attribute bipartite adjacency matrix for attribute type v ($B_{v,i,j} = 1$ when user i exhibits attribute j and is 0 otherwise).

Next, the expected number of shared attributes across communities under the configuration null model was calculated for each view. The expected adjacency matrix, $\mathbb{E}\|A_v\|$, was calculated as $\mathbb{E}\|A_v\| = 2M_v L_v^T L_v / F_v$, where L_v is the vector of community strengths l_c for view v as previously defined, and M_v and F_v are the sum of edge weights in for view v in the projected network and the bipartite network, respectively. This is the same number of expected internal links as used in Equation 4.1.

Then, the total number of shared-attributes between communities, above which is expected by chance, was calculated as: $\mathcal{A} = \sum_{v \in \mathcal{V}} A_v - \mathbb{E}\|A_v\|$. This collapses the views of the network, while accounting for the differing degree distributions of each of the views.

Lastly, the total number of shared-attributes between communities was normalized to indicate the number of expected shared-attributes (above chance) between any pair of users in the two communities. This is accomplished by divided the number of shared-attributes by the number of users in community 1 times that of community 2. The number of users in community c , N_c , can be calculated as $\sum_i C_{i,j}$. The normalization matrix is then NN^T . This normalization is applied element-wise to obtain the final adjacency matrix: $\bar{\mathcal{A}} = \mathcal{A} \oslash (NN^T)$. To be clear, the symbol \oslash indicates element-wise division, such that $\bar{\mathcal{A}}_{i,j} = \mathcal{A}_{i,j} / (NN^T)_{i,j}$. The community-to-community visualization is finally drawn using edge weights corresponding to the entries in $\bar{\mathcal{A}}$. Edge weights below zero, indicating that there are *less* shared attributes between users of the communities than expected by chance, are not drawn.

4.2.4 Prototype Construction with Projected Modularity Vitality

Prototypical attributes are defined to be those which maximize meta-contrast; that is, attributes that help define the group and differentiate it from others. In network terms, prototypical attributes are those which maximally contribute to community structure. We have quantified the overall level of attributes association with group structure using projected modularity. Now, we develop projected modularity vitality to quantify the contribution of individual attributes to community structure. Sorting attributes by this value then gives prototypical attributes, or those which are signaled by many members within a community and few outside of it.

Network vitalities, or induced centrality measures, are used to quantify a nodes contribution to a global network value [59, 119]. Initially, we developed modularity vitality to quantify node contribution to community structure in the unimodal case. That development is detailed in Appendix C.

Now, we develop projected modularity vitality to quantify how much a node in the projected nodeset of a bipartite network contributes to communities in the opposing nodeset. Applied to our data, this will measure each attribute's contribution to each user community. Vitalities of other community-based network metrics such as the map equation vitality could be used, however they would not correspond to the meta-contrast theory as

well as modularity [21].

Network vitalities simply compare the original global network value to what it would be if a node and all its associated edges are removed from a network, as shown in Equation 4.2 where G is the network, i is the node to be removed, F is the function giving the quantity of interest and $G - \{i\}$ is the network with node i and its associated edges removed.

$$V_F(G, i) = F(G) - F(G - \{i\}) \quad (4.2)$$

We select F in equation 4.2 to be Q^P from equation 4.1. Similar to [137], we need to derive a computationally efficient form of Projected Modularity Vitality in order to apply it to large real-world network data. We do so by simply recognizing the impact of removing a node on the four terms in Q_c^P .

First, we define an attribute's degree in the bipartite network as $d_j = \sum_i B_{i,j}$, where j indicates the attribute of interest. The total number of edges in the bipartite network goes from F to $F - d_j$ when node j is removed. We also define the community degree, $d_{j,c} = \sum_i B_{i,j}\delta(c, c_i)$, which gives the number of users in community c displaying the attribute of the network. The strength of community l_c becomes $l_c - d_{j,c}$ when node j is removed. A property of network projection is that a node with degree d_j will yield edges whose weights sum to $\frac{1}{2}d_j^2$ in the projection. Thus, M becomes $M - \frac{1}{2}d_j^2$ after the removal of node j . Similarly, M_c^{int} becomes $M_c^{int} - \frac{1}{2}d_{j,c}^2$. These results give the equation for projected modularity vitality and its computation in Equations 4.3 and 4.4, respectively.

$$V_{Q^P}(G, \mathfrak{C}, j) = Q^P(G, \mathfrak{C}) - Q^P(G - \{j\}, \mathfrak{C} - \{j\}) \quad (4.3)$$

$$Q^P(G - \{j\}, \mathfrak{C} - \{j\}) = \sum_{c \in \mathfrak{C}} \left(\frac{M_c^{int} - \frac{1}{2}d_{j,c}^2}{M - \frac{1}{2}d_j^2} - \left(\frac{l_c - d_{j,c}}{F - d_j} \right)^2 \right) \quad (4.4)$$

$$V_{Q_c^P}(G, \mathfrak{C}, j) = \left(\frac{M_c^{int}}{M} - \left(\frac{l_c}{F} \right)^2 \right) - \left(\frac{M_c^{int} - \frac{1}{2}d_{j,c}^2}{M - \frac{1}{2}d_j^2} - \left(\frac{l_c - d_{j,c}}{F - d_j} \right)^2 \right) \quad (4.5)$$

Finally, we note that projected modularity vitality is naturally broken up into community terms, allowing for the quantification of a node's contribution to each community individually. This contribution, for a node j is given in Equation 4.5. For each community, the terms with the highest values of $V_{Q_c^P}$ are taken to be prototypical.

The modularity vitality approach identifies attributes which are mostly exhibited by a single community, and which are popular within that community. This is a necessary improvement over, for example, relative frequencies, which are likely to identify less common attributes.

Modularity Filtering

We previously stated that if communities exhibit prototypes, “members within a community will share a set of attributes with each other and they will not share these attributes

with other communities.” We note that it is still possible for a set of non-prototypical attributes to be shared among members of all communities.

Consider the social circles on a college campus. Each social circle has its own set of prototypical attributes, yet all people involved share the attribute that they are a student of the same college. This is not a *salient* attribute in the present definition of communities, so it does not affect whether or not prototypes are present. However, under our modularity framework, the inclusion of the college attribute would decrease the fraction of internal edges and thus lower the perceived strength of prototypes. If many of non-salient attributes are present, prototypes may be effectively drowned out. An illustration of the effect is given in Figure 4.3.

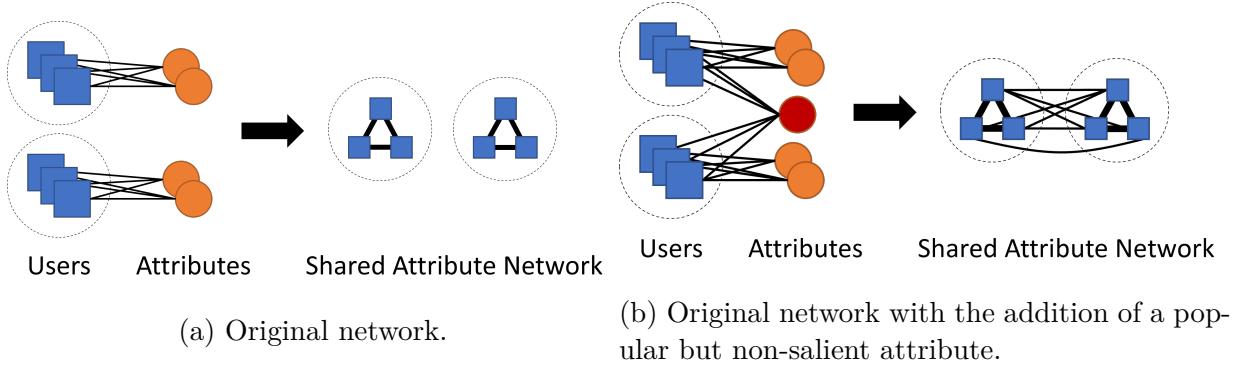


Figure 4.3: An illustration that the addition of a non-salient attribute can lower the perceived modularity. In Figure 4.3a, the communities are well-separated by attributes. In Figure 4.3b, an additional attribute has been introduced (red circle). This attribute is shared by all members, so is non-salient, however results in many external connections in the projected network, thereby lowering the modularity.

We can identify non-salient attributes as those with negative modularity vitality scores, which are known as “community-bridges” because they create shared-attribute edges between users of differing communities. Removing the top non-salient attributes is akin to the modularity filtering approach applied in [136, 137] or the modularity-vitality backbone approach in [181]. It can also be seen as an “initial” network attack when viewed from a network robustness perspective [103].

We apply this procedure to the top 2% of non-salient nodes to uncover a more accurate measurement of the presence of prototypes in the data. We have shown the top 5 attributes removed in each category, as well as the top 5 *most* salient attributes in each category. Unlike the previous analysis, this is a type of modularity-maximization procedure, so needs to be statistically tested against a null model. Thus, we used the bipartite configuration model as our null model using the real degree sequences observed in our dataset and performed filtering on each one of these randomly generated networks. The process was completed 250 times to provide confidence similar to that of using p-values at the value of $p < 0.004$. The filtered modularity value on the real data was compared to the distribution of scores obtained from the generated networks.

4.2.5 Multi-Modal Analysis

The previous methods detail our approach for a single attribute type. We extend this to the analysis of multiple attribute types through multi-view modularity [50]. Under this framework, each attribute type creates a “view” in a multi-view user-to-user shared attribute network. Multi-view modularity simply takes a weighted average of the modularities of individual views: $Q = \frac{1}{\sum_{v \in \mathbb{V}} w_v} \sum_{v \in \mathbb{V}} w_v Q^v$, where \mathbb{V} is the set of views, Q^v is the modularity for view v and w_v is the weight of view v . The weights may be manually set to enforce the importance of some views over others. In our case, we equally weight views with $w_v = 1$. Using this framework, quantifying the contribution of attributes across attribute types is now possible and attributes can be directly compared across views. To make values more comparable across communities, they are normalized according to their multi-view modularity, which is just the average of the modularity values across all views

$$\overline{MV}_{c,j} = \frac{\|\mathbb{V}\|}{\sum_{v \in \mathbb{V}} Q_c^{P,v}} V_{Q_c^{P,j}}, \quad (4.6)$$

where \mathbb{V} is the set of attribute views.

4.2.6 Computation

While mathematically we are operating on the projected shared-attribute networks, this is often not computationally feasible. Though real-world networks tend to be sparse, this is not necessarily the case with real-world projected networks. Consider an example where 1 million users share a single attribute. The bipartite network only has 1 million links, whereas the projected network will be fully-connected and contain 1 trillion links. The *COVID* dataset is the largest dataset we examine, making it the most computationally expensive to analyze. The sum of projected edge weights ranges from roughly 6.5 million (name hashtags) to 167 billion (location unigrams).

Thus, all computation is performed on underlying bipartite networks. This is made possible by the fundamental property of projection: an attribute exhibited by d users (attribute has degree d in the bipartite network), will result in $\frac{1}{2}d^2$ in the projected network. This property allows for the computation of M and M_c^{int} without the construction of the projected network: $M = \sum_j \frac{1}{2}d_j^2$ and $M_c^{int} = \sum_j \frac{1}{2}d_{j,c}^2$. This trick makes the computation of Q^P feasible even in the case that the projection yields an extremely large and dense network. Computation of modularity vitality is even more costly than modularity. Thus, the same computational trick is used and all calculations are made on the underlying bipartite networks. Refer to the accompanying code for full details¹.

¹Prototype construction of Twitter conversation communities has been implemented in the ORA-Pro network analysis software <https://netanomics.com/ora-pro/>. The original code for this work will be made available at <https://github.com/tmagelinski>

4.2.7 Relating Prototypicality and Status

To quantitatively assess the theoretical relationship between prototypicality and status, we first need a method of quantifying both. A simple method of determining the prototypicality of an individual is by summing the prototypicality of all of their attributes, where the prototypicality of a user’s attribute is dependent on the community that the user is in, as is given in Equation 4.5. For example, if a user has two attributes, “Republican” and “Arsenal Fan” with corresponding scores of 0.2 and 0.001, their prototypicality score will be $0.201 = 0.2 + 0.001$. The purpose of this approach is to account for the fact that some attribute signals are much more meaningful than others, but also that signaling multiple prototypical attributes is more meaningful than signaling one. It also enables us to balance users who have conflicting signals. For example if a user had attributes “Republican,” “Gun-Owner,” “MAGA” and “Union”, with scores 0.2, 0.1, 0.3, and -0.1, their overall score (0.5) would still reflect their strong prototypicality. For our analysis, we are concerned with whether or not a user is prototypical more than we are concerned with the magnitude, so we binarize the scores based on their sign. That is, users whose prototypicality-weighted sum of their attributes is greater than 1 are considered prototypical, and others are not.

We consider 2 methods of quantifying of status. A classic indicator of status is the number of followers that a user has. This indicates their popularity and reach on the platform. Unlike network-level measures, the number of followers metric is not subject to data quality issues. However, the followers metric measures overall status, not status with respect to their in-group for the discussion at hand. To measure in-group status, we turn to the community-aware-centrality literature. Ghalmane et al. proposed that in-group centrality and out-group centrality can be measured by applying classic centrality measures to the sub-graphs of a network that only consider in-community and out-community edges [72, 73]. Following this approach, we consider the weighted degree of a node in the within-community sub-graph to be a measure of in-group status. This corresponds to the number of interactions that a user has with members of their community.

For the dynamic analysis, we take a snapshot approach, with each snapshot being one day. On the first day, we record each users prototypicality and status. On the next day, we record any changes they made to their account, and record their new prototypicality according to the prototypes on the previous day. For example, consider a community has 2 attributes, “Vote Blue” and “Biden2020.” On day 1 these attributes have prototypicality scores of 0.1, and 0.2, respectively. On day 1 a user has “Vote Blue” in their biography, giving them a score of 0.1. On day 2, they substitute this attribute for “Biden2020” to give them a score of 0.2. Note that the day 1 scores are used to evaluate the change because we assume that users are reacting to the information they have at hand, and they do not know how the prototype will change. This is computed on a rolling basis, so the changes in day 3 will be calculating using the prototypes from day 2, and so on.

Lastly, this analysis requires a specialized dataset. Users update their profiles, but they do so infrequently. To ensure that we have enough data to draw conclusions from, we turn to data we have one of the largest Twitter movements, Black Lives Matter. The resurgence of the Black Lives Matter movement started on May 25th 2020 with the murder of George Floyd. This sparked wide-spread discussion, including many people updating

their biographies with signals like #blm. A portion of this discussion was captured within the *COVID* dataset. So, for this analysis we consider the *COVID* dataset beginning on May 25th 2020, and continuing for one week.

4.3 Results

4.3.1 The Presence of Prototypes

Results for each dataset are presented in Table 4.2. First, we consider the raw, or unfiltered data in column 3. Across all datasets, we see that user communities are strongly separated by attributes, at least across some attribute types. Hashtags and mentions in user biographies are the strongest indicators of community, with modularity values ranging from roughly 0.18 to 0.63. Hashtags in usernames are also strong indicators of community. Personal identifiers, emojis, and location unigrams have low to moderate values.

Dataset	Attribute	Modularity	2% Filtered Modularity
Reopen	Bio Personal Identifiers	0.1717	0.2655
	Bio Mentions	0.3794	0.6862
	Bio Hashtags	0.5655	0.7168
	Bio Emojis	0.1011	0.1768
	Name Hashtags	0.4299	0.5132
	Location Unigrams	0.0859	0.2426
Election	Bio Personal Identifiers	0.0795	0.2294
	Bio Mentions	0.3323	0.4509
	Bio Hashtags	0.2909	0.3987
	Bio Emojis	0.1385	0.2116
	Name Hashtags	0.1008	0.2216
	Location Unigrams	0.0885	0.1607
COVID	Bio Personal Identifiers	0.1326	0.1988
	Bio Mentions	0.3648	0.6981
	Bio Hashtags	0.6304	0.7631
	Bio Emojis	0.0368	0.0796
	Name Hashtags	0.5860	0.7476
	Location Unigrams	0.2770	0.5633
Captain Marvel	Bio Personal Identifiers	0.0522	0.0863
	Bio Mentions	0.1889	0.4385
	Bio Hashtags	0.3542	0.5755
	Bio Emojis	0.0173	0.0346
	Name Hashtags	0.3005	0.3470
	Location Unigrams	0.0562	0.2301

Table 4.2: Projected Modularity Values for each dataset. Filtered Modularity values were found to be significant $p \leq 0.004$

The surprisingly low modularity values of signals like bio personal identifiers and location unigrams can be explained and accounted for by considering the salience of attributes. Personal identifiers and location unigrams are free-text attributes, naturally resulting in a less unified presentation of attributes and creates far more non-salient attributes than attributes like hashtags, which have a mechanism for seeing people and content using the same exact indicator as you. Thus, we consider the results on the filtered network, given in the fourth column of Table 4.2.

Now, personal identifiers have reasonably high modularity values, around 0.2 for all datasets except *Captain Marvel*. We also see large gains in the location unigrams attributes. A statistical test of the results was performed by performing the procedure on network generated from the configuration null-model 250 times. All results were found to be significant to $p \leq 0.004$. Overall, we see even stronger evidence that conversational communities differentiate themselves via multi-modal prototypes.

To understand the attributes that were filtered, the most and least salient (highest and lowest modularity vitality) attributes of each type for the *Election* dataset are given in Tables 4.3 and 4.4. Tables for the other datasets are given in the Appendix. In this dataset, personal pronouns were not salient overall. We also see that affiliation or support of different soccer teams was not salient. Neither were less-politically charged emojis like the heart emojis. Attributes associated with support for Donald Trump are salient, including personal identifiers like maga and patriot, hashtags like #maga, and the American flag emoji, . Politically liberal attributes like #resist, and #blacklives matter, are not often not salient. This could indicate that Trump-supporting users are isolated in a small number of communities while Biden-supporting or otherwise left-leaning users are dispersed in many conversational communities.

Personal ID		Mention		Hashtag		Emoji	
S	NS	S	NS	S	NS	S	NS
maga	she	@genflynn	@manutd	#maga	#blacklivesmatter		
patriot	her	@realdonaldtrump	@arsenal	#kag	#blm		
conservative	he	@potus	@bts_twt	#fbpe	#resist		
christian	him	@joebiden	@chelseafc	#trump2020	#bidenharris2020		
wife	writer	@kamalaharris	@lfc	#noafd	#fbr		

Table 4.3: The most salient (S) and least salient (NS) attributes of each attribute derived from user biographies within the *Election* Dataset

We observe that, paradoxically, #bidenharris2020 is one of the *most* salient hashtags when displayed within a name, but one of the *least* when displayed in a biography. This is counterintuitive but could be indicative of subtle usage differences across sub-communities. This outcome is consistent with a scenario where a hashtag is widely used in a biography, but only a specific community of users puts it in their name. Given that putting a hashtag in your name is a more prominent display than in your bio, it is plausible that these more extreme users would be more concentrated in communities and less dispersed throughout the network.

Hashtags in favor of Black Lives Matter or the Democratic Party are typically not

Name Hashtag		Location Unigram	
S	NS	S	NS
#fbpe	#blm	usa	england
#maga	#endsars	france	new
#bidenharris2020	#blacklivesmatter	india	ca
#stopthesteal	#biden	brasil	the
#trump2020	#biden2020	venezuela	united

Table 4.4: The most salient (S) and least salient (NS) attributes of each attribute *not* derived from user biographies within the *Election* Dataset

salient in the overall network. A possible explanation for this is that there are pro-Democrat conversational communities. Under this scenario, attributes like `#resist` will form many cross-cutting connections between these aligned communities. On the other hand, the overall salience of pro-Trump and pro-Republican attributes suggests that Trump supporters are concentrated in fewer conversational communities. We see that this is the case in the following section studying the prototypes of individual communities.

Lastly, the attribute-projection procedure illustrated in Figure 4.2 was carried out on the top 20 communities in each of the datasets, ranked by their contribution to modularity, Q_c^P . For the *Election* dataset, these communities contain 83.9% of the users. The coverage in other datasets is similar. After filtering the same 2% of attributes, the result was visualized in Figure 4.4. The resulting corresponding diagram for the unfiltered data is shown in the Appendix. We make four observations. First, the strong within-community edges (or loops) and the thin between-community edges illustrate the presence of prototypes for all datasets. Second, we see that prototype strength varies by community. In the *Reopen* dataset, for example, community 10 has a strong prototype, while community 14 does not. Third, some community prototypes are related, which can happen when two communities are related or sub-communities of a larger group. For example, communities 4 and 6 in the *Reopen* dataset share many attributes. Lastly, there are differences across datasets. Communities in the *COVID* and *Reopen* datasets have strong prototypes which are generally isolated, there is a cluster of communities with inter-related prototypes in the *Election* dataset, and communities in the *Captain Marvel* dataset tend to have some common attributes with many other communities. We will now explore the underlying attributes which lead to these effects.

4.3.2 The Construction of Prototypes

For a given community, c , the collection of attributes with highest values of $\overline{MV}_{c,j}$ in Equation 4.6 are taken to be the community’s prototype. Again, the top 20 communities are analyzed for each dataset, ranked by their contribution to modularity, $V_{Q_c^P}$. The prototypes for the top four communities are displayed in Figure 4.5, and while those of the remaining communities and remaining datasets are displayed in Appendix D. Results are shown with *all attributes*, including those filtered in the previous analysis. While an

attribute may not be salient in the overall network, it can still belong to a community’s prototype.

We observe that prototypes are coherent representations of communities’ multi-faceted identities which can be categorized into four dominant types: political, location or language, interest, and artificial. Political communities are those centered around specific politicians, political parties, or political ideologies. Examples of this include communities 1, 2, and 4 of the *Election* dataset, shown in Figure 4.5. Community 1 is made up of Trump supporters, which predominantly differentiates itself with  and #maga, but also shows support by mentioning General Flynn and Donald Trump directly. Community 2 is made up of Biden supporters which predominantly uses #resist and direct mentions of Joe Biden to differentiate itself, though it also uses hashtags like #bidenharris2020 and #blm. Its support of the Democratic Party is further shown with the use of , which indicates a “blue-wave,” or a large Democratic turnout in the election. While the use of  is consistent with previous work documenting its usage in the American left, that work found that  was a non-differentiator among pro and anti white-nationalist ideology [86]. However, previous work studying flag emojis specifically have found that the American flag is more popular among Republicans than Democrats [113]. Community 4 is made up of Black Lives Matter supporters, who often display she/her pronouns.

Using the *Election* prototypes in conjunction with the attribute-block diagram in Figure 4.4, it becomes clear that the cluster of related community prototypes (communities 1, 7, 9, 18, and 20), are all MAGA-related. While all are similar, each community tends to have a different focus. Community 7 is most defined by , community 9 focuses on support of General Flynn, community 18 on the restart-leader account (a pro-MAGA Iranian political group), and community 20 on QAnon. The strength and prevalence of pro-MAGA community prototypes are greater than that of communities supporting Joe Biden or the Democratic party, especially in datasets that are less-directly political (*Captain Marvel* and *COVID*). This suggests that pro-MAGA users tend to be more isolated into conversational communities of similar users, even if they have multiple sub-communities.

The second type of prototypes are those based on location and language. These prototypes are typically formed with country or city-based location unigrams, the flag of the country, and mentions of accounts which are related to the location. Two examples are communities 5 and 6 of the *Election* dataset, which are centered around India and France, respectively. In the case of Community 5, community members signal their Indian identity using “India” in their location, mentioning Prime Minister Modi’s Twitter handle, and using  in their biography. We observe that many of the communities with extremely strong prototypes, as visualized in Figure 4.4, are based on location or language. This could be because of the unified way of signaling identity (the names of countries and cities are agreed upon, unlike political hashtags), and because location-specific topics of discussion may be generating the communities. Communities in the *COVID* dataset are mostly of this type, explaining its strong and well-separated communities in Figure 4.4.

Third, many prototypes based on shared interests. The most common of such prototypes are K-pop fan groups, as seen in *Election* community 16, *Captain Marvel* communities 3 and 4, *COVID* communities 5 and 12, and *Reopen* community 17. Other interest-based groups include gamers, soccer fans, and TV show fans. The previously observed strong

association between communities 4 and 6 in the *Reopen* dataset can now be understood. Community 4 is an interest-based community supporting the Indian actor, Vijay. Members of this community also tend to be Indian, creating attribute overlap with members of Community 6, a more general Indian-location-based community.

Lastly, there are “artificial” communities, wherein users signal their intention to create a community that can inflate the popularity and reach of its members. These may also be referred to as “follow-back” communities, since the users signal that if a member of the community follows them, they will reciprocate. The #fbpe (follow-back Pro-EU) community is present in all datasets (Community 3 in *Election*, Community 10 in *Captain Marvel*, Community 2 in *COVID*, and Community 5 in *Reopen*). Members of the community use hashtags in their username. This makes it easier for members to identifier each other, because name-hashtags can be seen without clicking into a users’ biography. Generally, these communities do not have much in common beyond the community-signal itself. The recurrence of this community in all four datasets along with the size of the communities signal the prevalence of artificial communities in large Twitter discussions.

4.3.3 Relationship Between Prototypicality and Status

Over the course of the week following George Floyd’s murder in the *COVID* dataset, we observe 3.1 million data points indicating a user’s current and previous prototypicality and status. Note that the dataset contains repeated observations of users when users are active in the discussion for multiple days.

We first ask: who is prototypical? Social theory suggests that prototypical users are more likely to have high status or become leaders. To test this, we perform an independent t-test comparing prototypical users with non-prototypical users to see if their status differs. For both types of status, followers and community-degree, we find that prototypical users are of higher status, in agreement with the theory. Prototypical users, on average have 21% more followers and 39% more communication connections with their community, $p < 0.001$ in both cases.

With this in mind, we turn to *changes* in prototypicality, studying how users update their profiles. Changes to one’s Twitter profile are rare, but in 2.05% of the data points, users update their profile. This gives 64,701 data points to study how users adjust their personal identity signals. Now we ask: who changes their identity signals in the first place? Social theory would suggest that non-prototypical users would be more likely to change their identity signal, in hopes of becoming more prototypical and thereby gaining status and self-esteem. We compare prototypical to non-prototypical users and use a chi-square test to confirm whether or not they change their biography at a different rate. The prototypical users are 10.8% more likely to make profile changes than non-prototypical users, $p < 0.001$, going against our expectations. A possible explanation for this difference is based on the savviness of prototypical users. This explanation states that prototypical users are those who recognize prototypicality and its importance, since they have already signaled it and achieved high status. Because of this recognition, they are more likely to update their identity signals in the future. It is not possible to assess the meaning behind this relationship or the validity of this possible explanation without doing qualitative user

studies, which are beyond the scope of this work. The point is that because we only observe identity signals deliberately made by users, it is not too surprising that results differ from theory based on the latent user identity.

Lastly, we consider *how* users updated their profiles, were they making themselves more or less prototypical? Clearly, theory suggests that users should make themselves more prototypical. We see that 63.1% of profile changes resulted in higher user prototypicality. We would also expect that non-prototypical users would be the ones most likely to make positive changes. To test this, we perform another chi-squared test on the frequency of positive versus negative profile updates for prototypical and non-prototypical users. Surprisingly, we see that prototypical users were more likely to increase their prototypicality, $p < 0.001$, where 68.0% of updates among prototypical users increased their prototypicality compared to 57.0% among non-prototypical users. We see that both groups are increasing their prototypicality more often than not. The higher rate of positive moves the the prototypical group gives further evidence to the savviness hypothesis, though again verifying this theory is beyond the scope of our work.

4.4 Discussion

The main finding is that communication communities on Twitter do differentiate themselves via prototypes, as evidenced by the high levels of bipartite projected modularity in the multi-view attribute network. Further, we observe that these prototypes are multimodal. That is, they are constructed using multiple types of attributes, including hashtags, mentions, and emojis in their biography, hashtags in their name, and unigrams in their location. It has been known that these types of attributes are used to signal users' social identity [54, 79, 94, 130, 169, 188, 200], but these findings indicate that this identity signaling is part of a larger group process which plays out within discourse communities.

This finding also strengthens the notion that automatically extracted clusters of users within Twitter communication networks can in fact be communities. While network clustering algorithms are often referred to as tools for community detection, a cluster of users which interact with themselves more than others is not necessarily a community in the psychological sense. As Turner has argued, shared self-definition through social attributes is more important for group membership than the structure of the group's interactions [214]. While clustering algorithms extracts groups who have interactional cohesion, they might not have shared interests, beliefs, or identities. In the datasets we examined, members of communication clusters *do* signal beliefs, interests, and identities which help form a stronger basis of community. Recent work has suggested that the follower network can be partitioned into interest-based groups or "flocks" which can be used to understand public opinion [239]. Our findings suggest that this may also be done using communication clusters, which are more dynamic and can be collected on specific discussions.

Although the clusters within datasets that we examined have cohesive beliefs, interests and identities, it is likely that there are exceptions. With the methods and code that we develop in this work, the cases when this does and does not occur can be distinguished. Studies beyond the scope of political discussion are called for to understand the factors

which affect the exhibition or strength of community prototypes.

Our analysis also shows that the prototypes of individual communities shed light on their membership’s identities and beliefs. Because we are studying political datasets, it is natural that the prototypes are political in nature. However, the strength of prototypes and their starkly opposed political affiliations point at a form of political polarization not typically measured. Polarization is often studied using either the stance of users on specific issues [48, 83, 149], or the content that they retweet [68]. Here, we see polarization in terms of identity: the presence of political community prototypes indicates that the discussions between users who identify as MAGA Republicans and those who identify as the Democratic Resistance are largely separated. Future work using this framework to study interactions between users across polarized communities is of interest. More generally, using this framework to quantify a community member’s alignment with their prototype is of interest, given this alignment’s crucial role within the social identity perspective [207]. Studies in this direction could add granularity to the recent finding that identity cues have significant effects on users’ comment voting behavior on a social media site similar to Reddit [210]. Identity cues encoded in Twitter community prototypes are much stronger than those seen on sites like Reddit, and the ability to measure identity alignment could distinguish between different types of effects.

We further see that prototypicality is associated with status. Users who exhibit prototypical identity signals tend to have more followers and interact with their community more. They are also more likely to update their profile, and it is more likely that those updates result in increased prototypicality, relative to non-prototypical users. The later of these findings goes against existing theory, where we would expect non-prototypical users to increase their prototypicality more. A possible explanation for this difference is based on user savviness, where prototypical users are more likely to recognize the importance of aligning with their community and are more capable at doing so. Our data does not have any insight into the thought process of users making profile changes, so we limited to reporting the phenomenon.

An important limitation is the inability to attribute the causal mechanism behind these community prototypes. The effect of the follower network on these outcomes is also of interest for future work, since it naturally biases the interaction network. It has recently been shown that users who follow political elites on Twitter overwhelmingly follow those from only their ideological in-group [231]. Further, users may also choose to follow or unfollow users based on their displayed attributes; it has been shown that users may choose to unfollow, block, or mute users outside of their ideological in-group during times of polarization [27].

Beyond following networks, the interaction between user profiles and Twitter recommendations is an important factor which is challenging to study due to the proprietary nature of the platform’s design. It is possible that Twitter’s following recommendation algorithm leverages profile attributes to recommend that users with similar profiles follow each other. Such an algorithm could strengthen community prototypes and explain the higher status of prototypical individuals. Further, the utilization of user profile similarity in content recommendation could encourage users with similar profiles to engage with each other, which would also strengthen community prototypes. The deployment of such

algorithms could have large impacts on the structure and dynamics of online discourse communities. This includes the possibility of increasing levels of polarization in political discourse. This social process with algorithmic feedback could give a more specific mechanism driving the recently-named partisan sorting phenomena, where previously separate social divisions have become aligned on the basis of individual's social identity [212]. Investigations into the usage of such recommendation and their interplay with the group processes governing the creation and adoption of community prototypes is of interest for future work.

The remaining question regarding the identity context is whether or not community prototypes exist due to the *interactional* contexts studied in Chapters 2 and 3. In the following chapter, we consider the interplay between the interactional and personal context by constructing a joint contextual-network analysis pipeline and applying it to a specialized dataset.

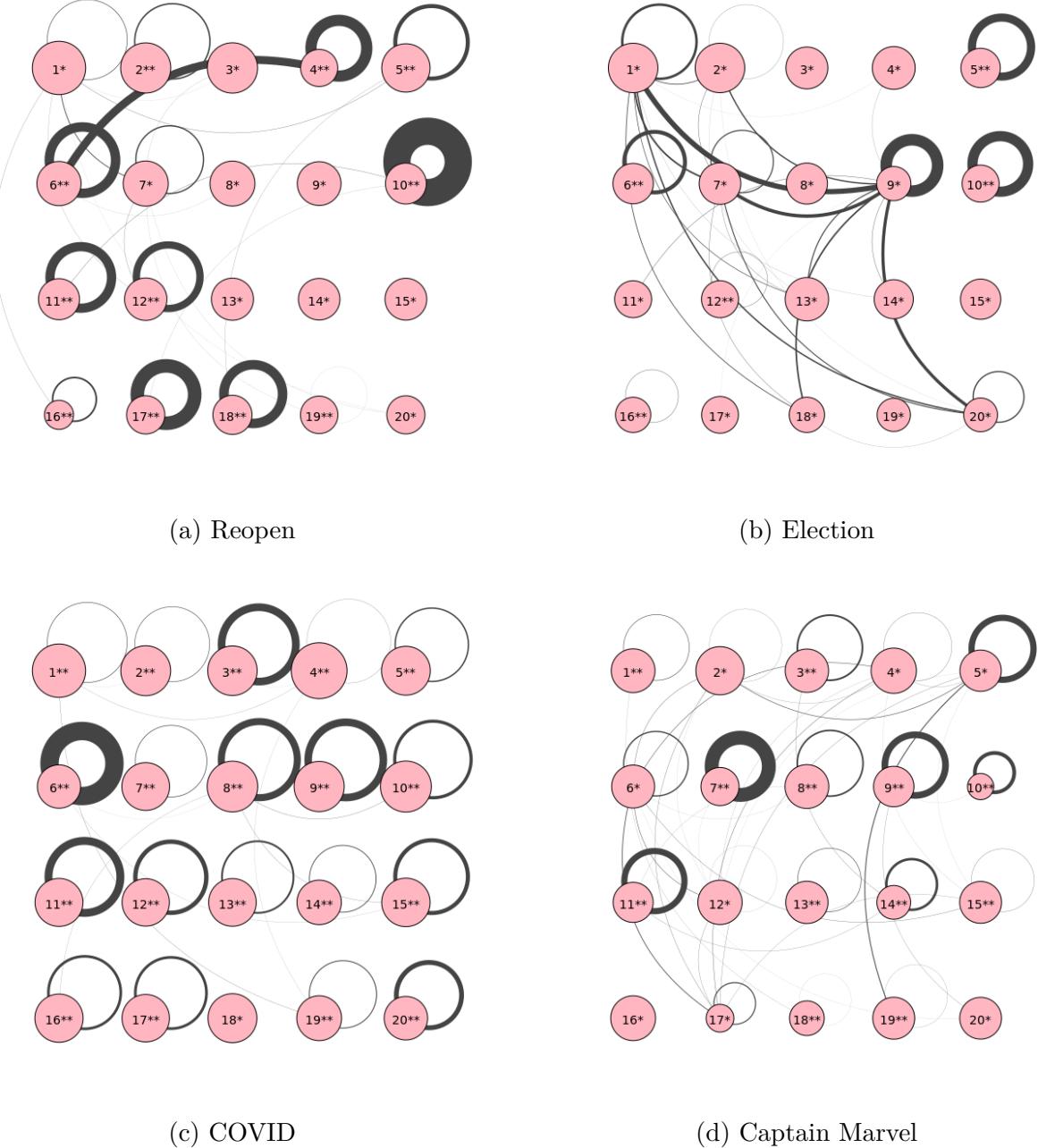


Figure 4.4: Following the process in Figure 4.2, observations for each dataset are shown. Each node is a community of users and the strength of connection between two nodes indicates the probability that users from those communities will share an attribute. If the probability of same-community members sharing attributes is higher than that of chance, the community is marked with a star. If the probability of a community member sharing attributes with non-community members is lower than that of chance, it is also marked with a star. If both are true, it is marked with two stars. Node size corresponds to the number of users in that community. For readability, a logarithmic scale is used, meaning that subtle node size differences correspond to drastic differences in community membership.

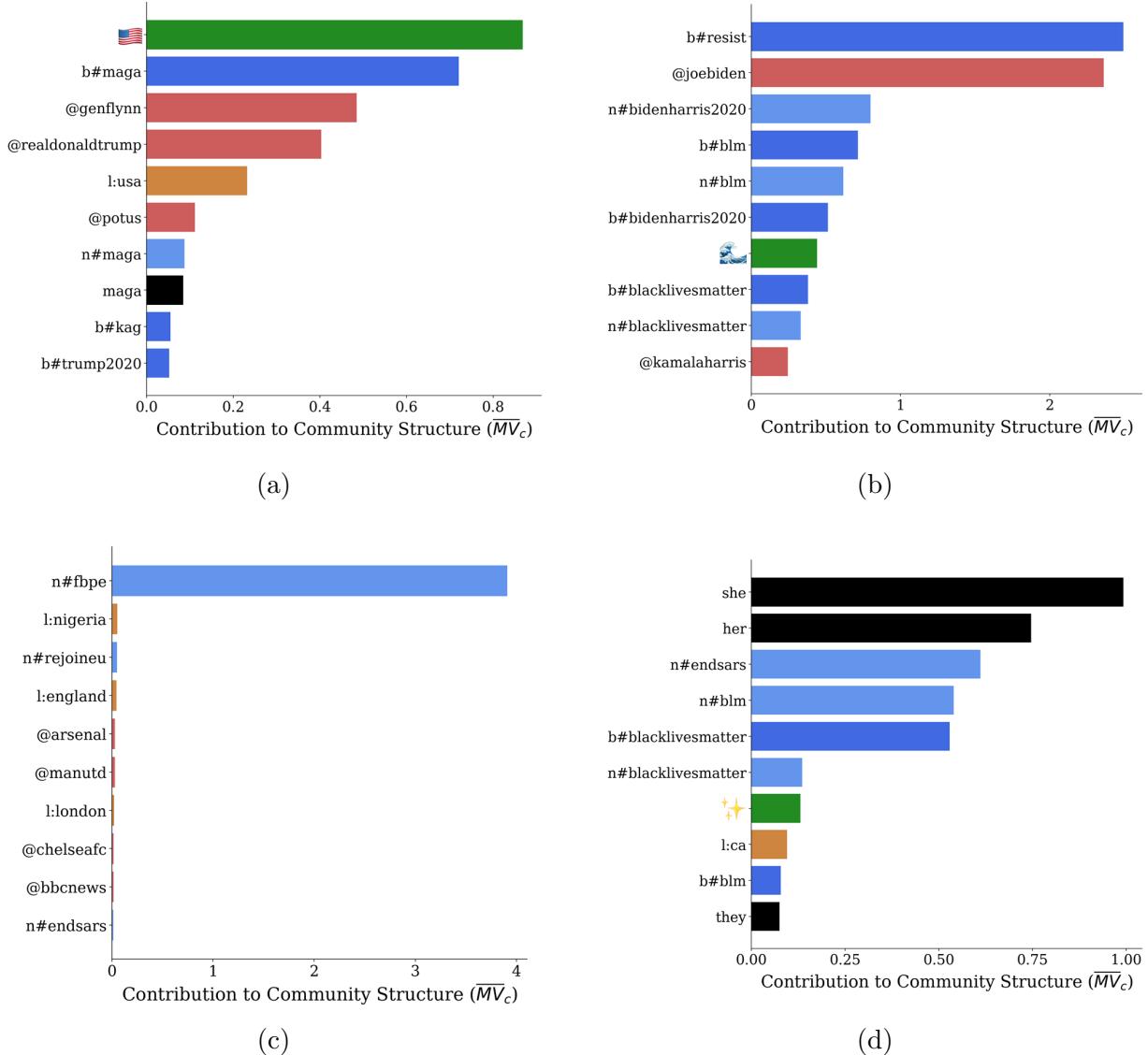


Figure 4.5: Prototypes of the top four communities in the *Election* dataset. Colors emphasize different modalities. Prefixes are representative as follows: b#: bio hashtag, n#: name hashtag, l: location unigram.

Chapter 5

Pipeline for Contextualized Conversation Dynamic Analysis

In this chapter, we aim to detail how the dynamical analyses developed in previous chapters fit together. The hope is that this chapter can act as a guide for performing dynamic contextualized network analysis on a new dataset. As such, we begin by detailing a data collection strategy that maximizes our ability to detect contexts. After following this strategy to collect the *News* dataset, we demonstrate the contextualized pipeline using it.

The only remaining research question we look to address in this chapter is as follows. It has now become clear that keyword-collected datasets contain many different conversational contexts. At the same time, we see that the communities within these datasets display community prototypes. Are community prototypes simply artifacts of mixed interactional contexts? We answer this question when demonstrating prototype analysis on contextualized networks.

5.1 Best Practices for Data Collection

5.1.1 Guiding Principles

There are two dominant factors which determine how well an online discussion dataset can be contextualized. The first factor is the presence of URLs. URLs provided a mechanism for connecting discussion across groups, while providing a signal that was just specific enough to label the context. This contrasts with hashtags, which also connect discussions but are too vague to help label discussions on their own. For example, almost the entirety of the *Reopen* dataset uses #Reopen, or could. So, a data collection strategy should ensure that many URLs will be captured, perhaps by directly querying them.

The second factor is the presence of conversational connections. These connections, seen as replies and quotes on Twitter, are the backbone of discussion, enabling us to label more of the dataset and to construct more accurate conversational networks. On Twitter, a keyword-based approach only inadvertently captures these connections. A key-word query on Twitter returns all of the Tweets matching the keywords. In the case of a quote tweet or

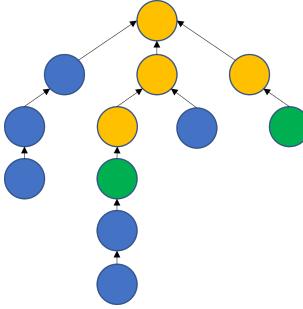


Figure 5.1: Example of a conversation Tree where each node is a Tweet connected to Tweets it replies to. The green nodes are those which match a key-word based query. The orange are those which could be obtained by crawling up from the keyword Tweets. The remaining Tweets, in blue, are those which can only be captured using the Conversation ID.

a retweet, the original tweet will also be contained. It will not the Tweets that a returned Tweet is replying to, nor will it return replies to it. There are two ways to rectify this.

The first method is useful for the V1 Twitter API, but does not capture all of the data. While a Tweet does not contain the full information for the Tweets that it references, it does contain their ID. For example, if a Tweet will tell you the ID of the Tweet it is replying to. These IDs can be queried directly with Twitter's API. If the newly collected Tweets are also replies, there will be a new set of IDs to query. This operation can be done recursively to crawl up the branches of the conversation that contain Tweets matching the original keyword query. This approach fills in much, but not all, of the conversational structure surrounding keyword queries.

The second approach, available using the V2 Twitter API, is to query the full conversational tree, using the conversation ID. Thus, for a keyword based dataset, the next step is to query all of the returned conversation IDs, which give the full trees. This approach gives all of the conversational structure surrounding the initial Tweets.

The differences in methods are visualized in Figure 5.1. The standard keyword approach only obtains nodes in green. The V1 crawling procedure can be used to obtain all the up-tree Tweets, which are given in orange. Clearly, much of the conversation is still missing. The rest of the conversation, shown in Blue, is can only be captured using the V2 conversation ID approach. The full conversational collection yields much more data than the other approaches, which may force analysts to consider shorter time periods or more targeted keywords.

Finally, it is important to consider the contexts that you expect to observe in your dataset. For small or targeted, research questions, it is sensible to be as strict as possible with the available data filters to obtain a dataset that has minimal contextual mixing before processing it. For larger dataset, such as those examined in this thesis, it might be helpful to deliberately capture more varied conversations, which can then be analyzed with the methods provided by this work.

5.1.2 Collection of the News Dataset

We now demonstrate a data collecting following these practices which we will call the *News* dataset. The purpose of this section is to give the details on how a high-quality dataset with varying contexts can be collected.

We are agnostic about the topic being collected on because we aim to collect various conversations. Content agnostic datasets are often collected on Twitter using their 1% random sample of Tweets¹. However, we have already stated the need for maximizing the presence of URLs in our data. Towards this end, we aim to collect the discussion around news articles. To make sure that our data contains discussions from different communities, we consider 6 major news sources of different types. Those are Reuters and AP (direct reporting), CNN and Fox (American with political bias), and RT and CGTN (state-sponsored).

To find the conversational trees discussing news from these sources, we query the Twitter API for all Tweets linking to one of the 6 websites. We further query the API on the timelines of the official Twitter accounts for these 6 sources. Lastly, the full conversations were collected for all of the Tweets obtained from the first 2 steps. Due to the large volume of data and the rate-limited API, data collection was limited to a 24 hour period, which could not be selected retroactively. This window was selected to be the entirety of October 28, 2022.

Note that this procedure does not obtain retweets, which are heavily restricted by the API. Because of this heavy restriction, researchers aiming to follow these guidelines may need to narrow the scope of their search even further to be able to fully collect their dataset in a reasonable amount of time. Retweets do not affect the conversational trees, nor do they affect their representation using Deep Tweet Infomax. Their main effect is in the conversational networks themselves, which are much bigger when retweets are considered. The presence of retweets also makes it slightly easier to assess user importance, though this is still possible only using replies and quotes. This is possible because replies and retweets are roughly proportional, though there are some differences in the factors that lead to both actions such as the type of account (e.g., news versus celebrity) and the sentiment of the Tweet's text [116]. The added benefit of obtaining retweets is inconsequential for our purposes, so we proceed without them.

The collection resulted in 5233570 Tweets from 1482034 Users. Tables 5.1 and 5.2 give the top 5 URLs from each collected source in terms of their number of instances in the dataset. From this table we see that there are various conversations taking place, though there is clearly a large focus on the attack of Nancy Pelosi's husband, and on the closing of Elon Musk's acquisition of Twitter, both of which occurred during collection. The counts provided in the tables give insights into the number of Tweets that each URL is able to directly label, and are a limited quantification of popularity. This measure does not account for how many followers the users who posted the URLs have, or the number of replies they generated. Still we see that the state-sponsored sources get significantly less traction than those of the other sources.

¹<https://developer.twitter.com/en/docs/twitter-api/tweets/volume-streams/introduction>

Source	URL	Count
Reuters	https://www.reuters.com/world/us/us-house-speaker-pelosi-husband-violently-assaulted-pelosi-statement-2022-10-28/	216
	https://www.reuters.com/markets/deals/elon-musk-completes-44-bln-acquisition-twitter-2022-10-28/	109
	https://www.reuters.com/business/energy/exxons-record-smashing-q3-profit-nearly-matches-apples-2022-10-28/	68
	https://www.reuters.com/business/energy/exxons-record-smashing-q3-profit-nearly-matches-apples-2022-10-28/	60
	https://www.reuters.com/business/energy/exxons-record-smashing-q3-profit-nearly-matches-apples-2022-10-28/	56
AP	https://apnews.com/article/paul-pelosi-assaulted-156ec-e77186eb11b97260af3c5122f67	141
	https://apnews.com/article/california-donald-trump-san-francisco-47c103cfe696df9faf0e57e1c7dd4f10	103
	https://apnews.com/article/fact-check-texas-identification-kits-104242791947	70
	https://apnews.com/article/virus-outbreak-race-and-ethnicity-suburbs-health-racial-injustice-7edf9027af1878283f3818d96c54f748	43
	https://apnews.com/article/156ece77186eb11b97260af3c5122f67	22
CNN	https://www.cnn.com/2022/10/28/politics/pelosi-attack-suspect-conspiracy-theories-invs/index.html	400
	https://www.cnn.com/2022/10/28/politics/paul-pelosi-attack/index.html	344
	https://www.cnn.com/politics/live-news/nancy-pelosi-husband-paul-attack/index.html	198
	https://www.cnn.com/2022/10/27/politics/kfile-tudor-dixon-conspiracy-democrats-topple-america/index.html	95
	https://www.cnn.com/2022/10/27/tech/elon-musk-twitter/index.html	88

Table 5.1: Top URLs from the first 3 targeted news sources in the *News* dataset.

Source	URL	Count
Fox	https://www.foxnews.com/politics/nancy-pelosi-husband-paul-assaulted-home-invasion-spokesman-says	218
	https://www.foxnews.com/video/6314469432112	179
	https://www.foxnews.com/politics/nancy-pelosi-husband-paul-pelosi-assaulted-san-francisco-suspect-david-depape-police-say	111
	https://foxnews.com/video/6314469432112	94
	https://www.foxnews.com/politics/republicans-demand-answers-biden-officials-report-china-opened-police-arm-ny	77
RT	https://www.rt.com/russia/565476-putin-valdai-club-takeaways/	33
	https://www.rt.com/russia/565460-west-sit-out-crises-caused-putin/	24
	https://www.rt.com/russia/565472-russia-enemy-west-putin/	23
	https://www.rt.com/news/565561-china-washington-nuclear-blackmail/	18
	https://www.rt.com/russia/565466-putin-values-tensions/	17
CGTN	https://news.cgtn.com/news/2022-10-28/U-S-is-fast-running-out-of-diesel-and-that-s-disastrous-1etnHBoeWL6/index.html	5
	https://news.cgtn.com/news/2022-10-28/German-Chancellor-Scholz-to-visit-China-1ev8NuF5ELm/index.html	2
	https://newseu.cgtn.com/news/2022-10-23/Protests-across-Europe-as-anger-builds-over-cost-of-living-crisis--1lvzxYsPmw/index.html	2
	https://news.cgtn.com/news/2022-10-28/Diplomatic-efforts-should-be-made-to-ease-Russia-Ukraine-tension-1ev38kZF6P6/index.html	2
	https://arabic.cgtn.com/news/2022-10-28/1585815981036662786/index.html	1

Table 5.2: Top URLs from the second 3 targeted news sources in the *News* dataset.

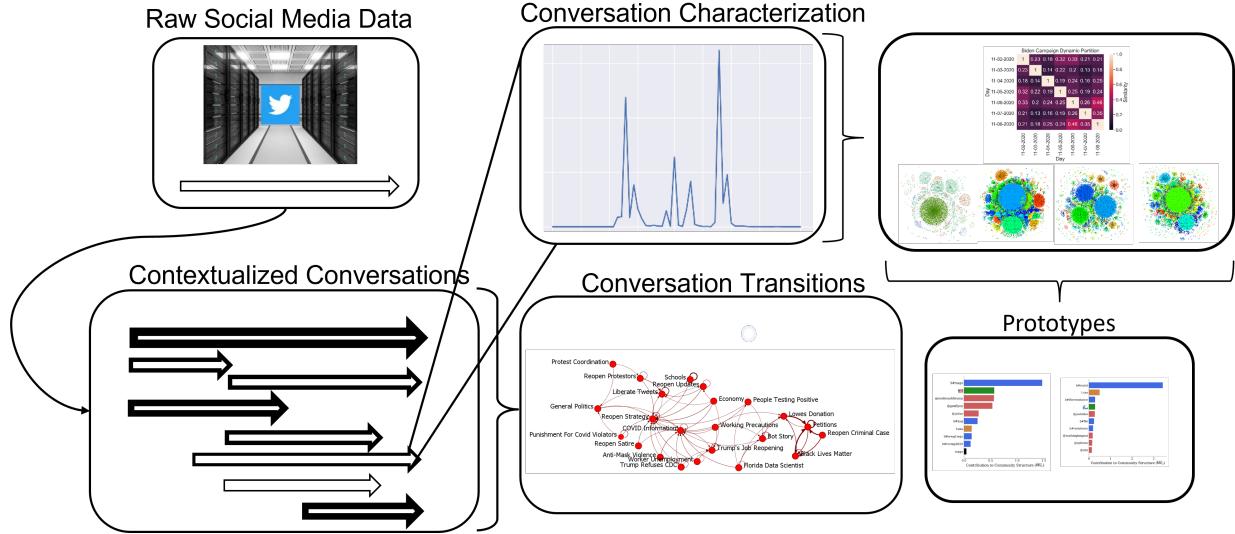


Figure 5.2: Full Contextualized Analysis Pipeline

5.2 Full Contextualized Analysis

A visual representation of the contextualized analysis pipeline is given in Figure 5.2. The flow of the pipeline closely follows the order in which these processes were developed in this dissertation. The steps are as follows:

1. Contextualization (Label-Based or Deep Tweet Max) - Chapter 2
2. Characterization of Contexts - Chapter 3
 - (a) Activity Plot Analysis
 - (b) Conversational Transition Analysis
3. Contextualized Dynamic Community Detection - Chapter 3
4. Contextualized Prototype Analysis - Chapter 4

The only step of this pipeline that has not been previously demonstrated is the use of prototypes on contextualized networks. Previously, we analyzed the prototypes of online communities derived from a non-contextualized approach. Now, we perform the analysis on a *contextualized* network broken down into snapshots where community structure is stable in time using the dynamic community detection method developed in Chapter 3. We will now walk through this pipeline on the *News* dataset.

5.2.1 Identifying Interactional Contexts

The unsupervised representation model was trained on the *News* dataset, using directed Tweet edges, Graph Attention aggregation, and mean summarization as that model configuration was determined to perform best in Chapter 2. To determine the appropriate number of clusters, the elbow method was applied with DB-SCAN, as shown in Figure 5.3. This method entails clustering the dataset multiple times, increasing the number of

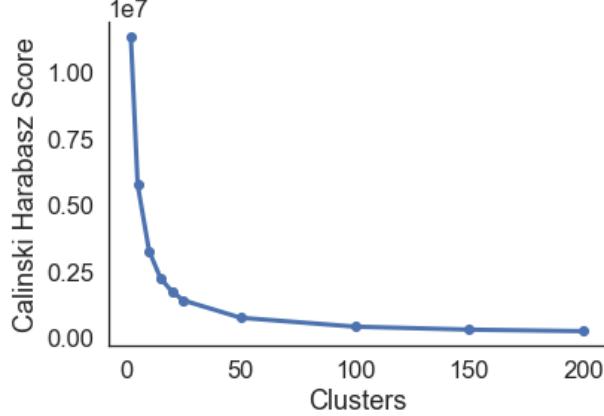


Figure 5.3: Cluster quality analysis to select the number of clusters in the *News* dataset.

clusters each time. Specifically, we test values of 2, 3, 4, 5, 10, 15, 20, 25, 50, 100, 150, and 200. Larger numbers were not tested due to the observed diminishing returns. Then, the cluster evaluation metric is plotted against the number of clusters. Here, we evaluate the clusters with the Calinski-Harabasz Index [33], where higher is better. This selection process recognizes that a model with more clusters should have a large increase in cluster quality to justify the complexity that a higher number of clusters brings. Evaluating Figure 5.3, see that the elbow method is actually not needed, because cluster quality decreases after increasing the number of clusters beyond 2. It seems that the data is largely split into two conversations, and breaking these out further introduces unnecessary complexity.

Now we move to label the two identified clusters. After applying the initial 3-gram approach developed in Chapter 2, it becomes clear that the dataset contains a notable amount of spam. The downside to collecting the full conversation trees is that there is no filtering mechanism for removing Tweets from spammers. The spam is evident from the top 3-grams, which include “uniswap exploited dude.” Further examination of this 3-gram show that it occurs due to spammers replying “I Wish I discovered this earlier. Uniswap is being exploited by this dude. More than \$200k so far (redacted URL) leaked in alpha group” over and over again to different conversations. To remove this noise, two changes were made. First, instances of 3-grams were weighted based on the number of likes they received, as spam is unlikely to receive likes. Specifically, they were weighted according to the fourth root of 1 plus their number of likes, such that Tweets with 1000 likes were approximately 5.6 times more important than those with none. The fourth root was used so that popular Tweets were weighted more without letting the 3-grams be entirely washed out by the few extremely viral Tweets in each cluster. Next, full text that was exactly replicated was only counted once, to remove the effect of direct copy-and-paste quotes.

The 3-gram with highest relative frequency in the first cluster is “let door hit.” Investigation into the raw tweets shows that this 3-gram stems from Elon Musk supporters using the phrase “don’t let the door hit you on your way out” and its variants to users saying they plan on leaving Twitter due to Musk’s purchase of it. The top Tweets in terms of likes in this cluster are from Elon Musk talking about the deal closing and changes he

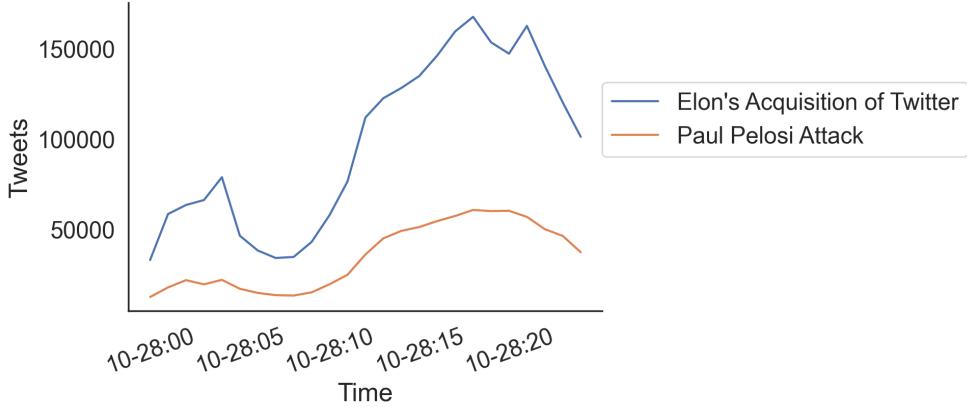


Figure 5.4: Activity curves in the *News* dataset.

plans on implementing. In the second cluster, the 3-gram with highest relative frequency is “speaker Nancy Pelosi,” which indicates that this cluster is dominated by discussion about the attack on Nancy Pelosi’s husband, Paul Pelosi. The top Tweets in terms of likes in this cluster are often talking about the attack, though there is noise. While there is noise in the dataset, we proceed referring to the clusters as *Elon’s Acquisition of Twitter* and *Paul Pelosi Attack*.

5.2.2 Characterization of Contexts

Activity Curves

The activity curves for the two contexts are displayed in Figure 5.4. We see that both curves follow the same pattern, though there is less overall activity in the conversation surrounding Paul Pelosi. There is a lull in activity, measured in Tweets per hour, in the early morning hours, when much of the country is asleep. As the news breaks on both accounts, the number of Tweets rapidly rises. In both cases, the number of Tweets per hour doubles in under 5 hours. From there, the discussion around Twitter’s acquisition continues to rise in popularity, while the Paul Pelosi discussion holds a steady peak. Both discussions hold near-peak activity going into the late hours of the 28th, though the activity does begin to taper off.

From these curves we can draw two conclusions. First, that both of these events were unanticipated, but important. While the Twitter acquisition was discussed previously, it was not known that it would go through or that there would be any updates of its status on the 28th. Paul Pelosi’s attack was completely unanticipated. Although the dynamics are collapsed to a single day within the *News* dataset, the activity curves are still useful to characterize the conversations within it.

Second, we note that there is a key difference in the *News* dataset activity curves compared to those seen earlier for the *Reopen* and *Election* datasets. Specifically, the transition from daily to hourly data introduces temporal cycles related to the sleeping patterns of the users in the dataset. Thus, dataset with long enough timelines to use day-

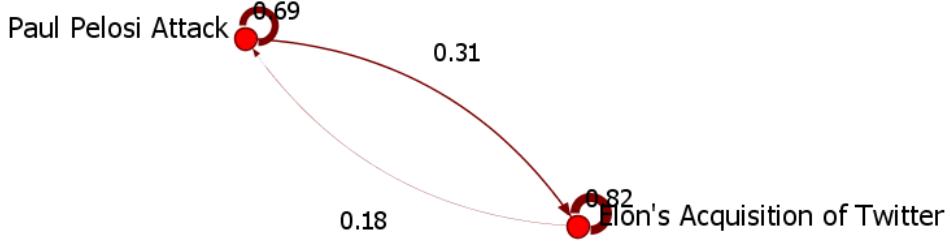


Figure 5.5: The contextual transition network of the *News* dataset.

level aggregation have the advantage of being easier to directly interpret. For hourly curves, the affect of discussions sparked in early morning hours may be hard to distinguish that from naturally increasing or decreasing activity due to users sleeping patterns. Luckily, this is not an issue for our current dataset, but is important to keep in mind.

Conversational Transitions

The analysis of conversational transitions is most interesting when there are many states in which users flow between. Still, for demonstration purposes, we proceed by showing the transition matrix between the two states in our dataset. The transition network is given in Figure 5.5.

Even with only two contexts, we can perform basic analysis. We see that both discussions are “sticky” in that people who are in the discussion are fairly unlikely to leave it. Users active in the Twitter acquisition conversation had an 82% chance of staying in that discussion if they were to tweet again, while those in the Paul Pelosi conversation had a 69% chance. We also see that users are more likely to transition from the Paul Pelosi discussion to that of the Twitter acquisition. This differential in transition probability combined with the higher level of overall activity seen in Figure 5.4 suggests that users have more interest in the acquisition, though both were important conversations. This is an intuitive result, as the change in ownership could have a direct effect on users of the platform.

5.2.3 Contextualized Dynamic Network Analysis

Now that the contexts have been identified and analyzed, we turn to their dynamic network analysis. Because the dataset has been collected within a 24 hour window, we select snapshot lengths of 4 hours. Shorter snapshots could be used, however they will not carry much network information in the early hours, as seen in Figure 5.4. The dynamic partitioning algorithm was run for both contexts, with the results shown in Figure 5.6.

We see that community structure is weekly correlated among snapshots, but it is still worth combining some snapshots into larger periods. Specifically, we the hours of 4am-12pm in the Pelosi context should be considered as one network, while the hours of the 8am-8pm should be considered one network in the Twitter discussion. Due to the size of

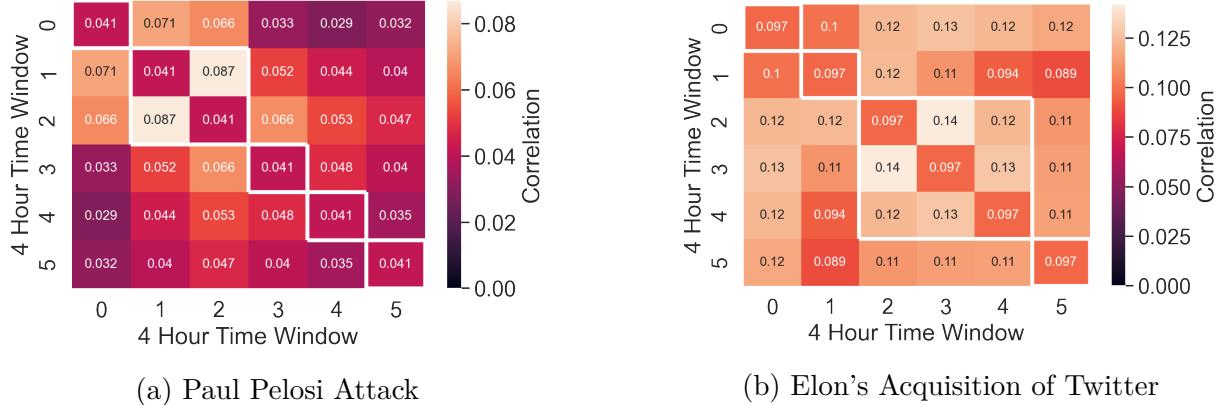


Figure 5.6: The dynamic partitioning algorithm results are given for the two contexts in the *News* dataset. The partitions are given as a white border in the similarity matrices. The diagonals are filled with the mean correlation for visualization purposes.

the networks, they are not visualized here. Instead, the network statistics are given in Tables 5.3 and 5.4.

Start	End	Nodes	Edges	Nodes in LC	Communities	Modularity
0	4	44970	66031	29733	5755	0.95
4	12	86558	131347	63132	9494	0.94
12	16	103754	173166	77191	10963	0.92
16	20	123609	220896	94574	12046	0.90
20	24	117353	197149	84871	13089	0.91

Table 5.3: Network snapshot statistics for snapshots within the *Paul Pelosi Attack* conversation. LC stands for largest component. Start and end indicate the hours of October 28th included for the snapshot.

We observe that the number of nodes and edges differs drastically between snapshots. In the Pelosi discussion, the community structure is strong and stable over time, whereas in the Twitter discussion the community structure is weaker and fades as time goes on. The extremely high values in the Pelosi discussion are inflated due to the many isolates and small components in the discussion. This is a side effect of using the text and hashtags to cluster data into conversations, Tweets which may not be directly interacting with the largest component of users can still be participating in the same discussion.

These contextualized snapshots have now controlled for variation in both interactional context and network dynamics, so they are appropriate for use by analysts looking to answer questions about this discourse. Along those lines, these snapshots could be used in a social cybersecurity pipeline [218].

Start	End	Nodes	Edges	Nodes in LC	Communities	Modularity
0	4	146344	238050	142826	1450	0.75
4	8	134916	214764	129428	2561	0.75
8	20	632684	1449869	626216	2896	0.64
20	24	306098	609431	296209	4433	0.67

Table 5.4: Network snapshot statistics for snapshots within the *Elon’s Acquisition of Twitter* conversation. LC stands for largest component. Start and end indicate the hours of October 28th included for the snapshot.

5.2.4 Contextualized Prototype Analysis

The last step of the pipeline is the perform prototype analysis on the contextualized dynamic networks. The application of this step enables us to answer one final research question: are community prototypes just an artifact of conversational context? In Chapter 2, we saw that Twitter datasets have many different discussions going on within them. In Chapter 4, ignoring these contexts, we saw that communities were well-separated by their identity attributes. Is this separation simply due to the different conversations in the dataset? For example, if a dataset has a discussion about politics and a discussion about news mixed together, it is possible that community detection differentiates a political community and a sports community, each of which use distinct identity signals. This would undermine the conclusions we made in Chapter 4. The other possibility in this example is that within the political discussion there are separate communities who describe themselves differently, and the same goes for the sports discussion. This would confirm, if not strengthen our previous conclusions.

To answer this question we apply prototype analysis to the contextualized networks. We show prototype analysis for the *News* dataset for the purposes of pipeline demonstration. To relate to our findings to those of Chapter 4 the contextualized prototype analysis is also shown for the *Reopen* dataset, focusing only on the measurement of the presence of the prototypes.

News Dataset

For each snapshot in each context, we show the measurements of prototypes in Tables 5.5 - 5.8. Unlike previous analyses, projected modularity values are not high for all modalities. For the Pelosi discussion, values are only high for biography mentions, while values are high for biography mentions and location unigrams for the Acquisition discussion. Still, among these modalities we see evidence for prototypes.

From here, the prototypes for each community within each snapshot can be observed, much like they were in Chapter 2. For demonstration purposes, we show two such prototypes in Figure 5.7, one from each context. These were selected to demonstrate the presence of highly similar prototypes across contexts and across time. These prototype similarities can be used to understand how large communities relate to different discussions without the need for matching their exact membership.

	1	2	3	4	5
B-Identifiers	0.0308	0.0253	0.0172	0.0132	0.0130
B-Mentions	0.6082	0.5731	0.4706	0.4186	0.4220
B-Hashtags	0.0268	0.0349	0.0249	0.0162	0.0139
B-Emojis	0.0068	0.0071	0.0062	0.0042	0.0039
N-Hashtags	0.0068	0.0071	0.0062	0.0042	0.0039
L-Unigrams	0.0040	0.0085	0.0051	0.0034	0.0026

Table 5.5: Projected modularity values for different identity attribute modalities and for snapshots in the *Attack of Paul Pelosi* contexts within the *News* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

	1	2	3	4	5
B-Identifiers	0.2750	0.2138	0.1389	0.1211	0.1282
B-Mentions	0.9944	0.9950	0.9162	0.9127	0.9126
B-Hashtags	0.1512	0.1656	0.1044	0.1093	0.0847
B-Emojis	0.0199	0.0215	0.0156	0.0126	0.0104
N-Hashtags	0.0199	0.0215	0.0156	0.0126	0.0104
L-Unigrams	0.0203	0.0468	0.0256	0.0222	0.0128

Table 5.6: 2% Filtered projected modularity values for different identity attribute modalities and for snapshots in the *Attack of Paul Pelosi* contexts within the *News* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

As a last note, we recognize that an identity modality does not have to be very salient overall for it to be used in a particular community. Both of the `#resist` communities use the hashtag in their biography and their name, despite that modality not being a particularly popular mechanism for distinguishing communities in these contexts. The biography mentions were more often used, which distinguished people's preferred news outlets, among other things. The location unigrams were also used often in the *Acquisition* context, which was more of a global discussion.

Results on the Reopen Dataset

To further confirm that the presence of prototypes is not solely a function of varying interactional contexts, we perform prototype analysis within 5 of the major contexts in the *Reopen* dataset. Within each context, the conversational networks were constructed and communities were extracted using Leiden clustering. From there, the projected modularity values are calculated for the 6 previously examined identity attribute modalities and are given in Table 5.9. The 2% filtered values are also given in Table 5.10.

Although the strength of the non-filtered values are lower within contexts than we observed for the entire dataset, there is still clear evidence for prototypes within contexts.

	1	2	3	4
B-Identifiers	0.0330	0.0253	0.0334	0.0247
B-Mentions	0.4337	0.3680	0.2800	0.3080
B-Hashtags	0.0761	0.0715	0.1088	0.0653
B-Emojis	0.0225	-0.0002	0.0377	0.0213
N-Hashtags	0.0225	-0.0002	0.0377	0.0213
L-Unigrams	0.0358	0.0278	0.0549	0.0337

Table 5.7: Projected modularity values for different identity attribute modalities and for snapshots in the *Elon’s Acquisition of Twitter* contexts within the *News* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

	1	2	3	4
B-Identifiers	0.0934	0.1224	0.0743	0.0773
B-Mentions	0.7354	0.7961	0.7181	0.7429
B-Hashtags	0.1169	0.1632	0.1806	0.1049
B-Emojis	0.0514	0.0182	0.0813	0.0544
N-Hashtags	0.0514	0.0182	0.0813	0.0544
L-Unigrams	0.2472	0.2252	0.3423	0.3751

Table 5.8: 2% Filtered projected modularity values for different identity attribute modalities and for snapshots in the *Elon’s Acquisition of Twitter* contexts within the *News* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

tualized networks. This is especially clear when we consider the filtered values, where all contexts have at least two modalities with modularity values above 0.2. In the case of the “Liberate Tweets” and “COVID Information” contexts, prototypes are constructed using 5 and 4 modalities, respectively.

Considering the results on the *Reopen* dataset in conjunction with those on the *News* dataset, we can conclude that community prototypes exist even when interactional context is accounted for. Further, we can still conclude that these prototypes are developed using multiple modalities. This is not to say that context does not affect community prototypes. It is still expected that certain attributes will only be salient within certain contexts.

5.3 Discussion

By considering the set of analyses developed in this dissertation as a pipeline, we have demonstrated how they fit together. The key takeaway, other than the procedure itself, is that the information gained from each step of the contextualized dynamic analysis helps us in the latter stages. That is to say that the methods developed, the inter- and intraactivity and network dynamic analyses as well as prototype analysis, are interrelated and

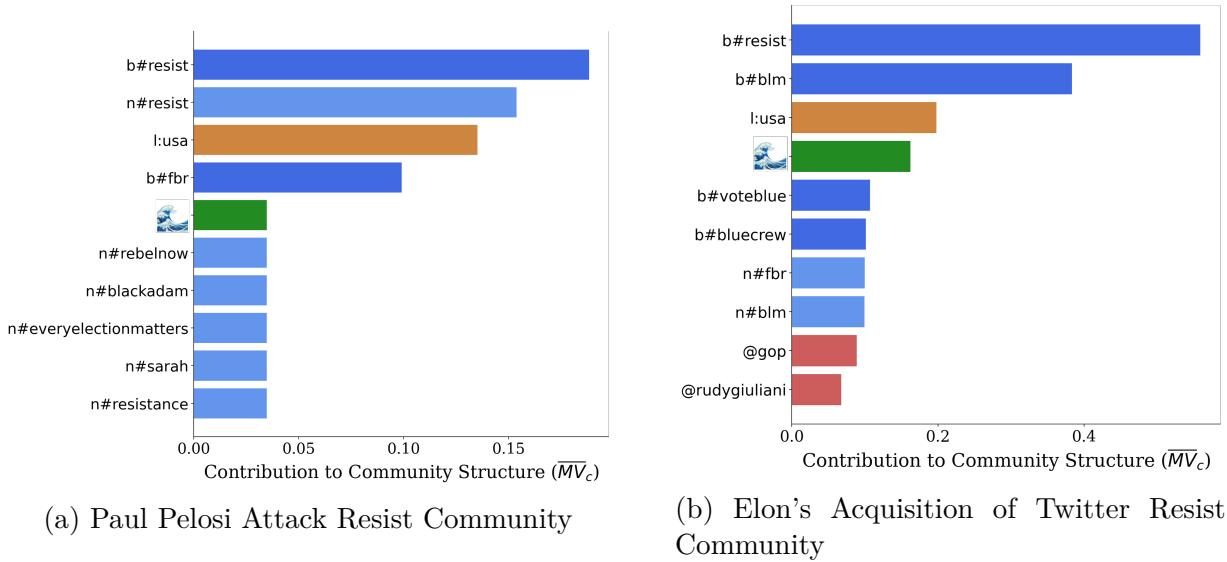


Figure 5.7: Prototypes of `#resist` communities from each of the contexts in the *News* dataset. Both instances occurred in the 2nd snapshot of their respective dynamic networks.

	Liberate Tweets	Reopen Strategy	BLM	COVID Info	Reopen Protesters
B-Identifiers	0.1633	0.0859	0.0090	0.0893	0.1455
B-Mentions	0.3572	0.2420	0.3103	0.3660	0.0929
B-Hashtags	0.2986	0.1639	0.0517	0.1918	0.0722
B-Emojis	0.1512	0.0968	0.0179	0.0988	0.0478
N-Hashtags	0.2320	0.1901	0.0322	0.1869	0.1264
L-Unigrams	0.0270	0.0356	0.0218	0.0266	0.0114

Table 5.9: Projected modularity values for different identity attribute modalities and for different contexts in the *Reopen* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

work together to unveil much more information than could be seen from the classic non-contextualized approach.

Naturally, the pipeline works best when good data is provided. The best-practices detailed in terms of data collection yield fuller conversation trees that can be better represented with the approaches developed in Chapter 2. We see that this is not without problems, however. The depths of conversation trees on Twitter, especially those started by major celebrities like Elon Musk, are polluted with spam. There is an inherent trade-off between collecting the whole discussion and collecting a clean discussion. We have shown that the contextualized network pipeline can work through this issue, however analysts performing network analysis on the data would still need to perform some sort of filtering, perhaps with bot and spam detection.

Along the way, the application of prototype analysis to dynamic contextualized networks has strengthened our findings in Chapter 4. While Chapter 4 found strong evidence

	Liberate Tweets	Reopen Strategy	BLM	COVID Info	Reopen Protesters
B-Identifiers	0.4376	0.1833	0.0785	0.2735	0.3434
B-Mentions	0.5148	0.3084	0.7322	0.4987	0.2478
B-Hashtags	0.4732	0.1927	0.3275	0.3672	0.1335
B-Emojis	0.2092	0.1136	0.0344	0.1434	0.0923
N-Hashtags	0.4449	0.4069	0.0718	0.3559	0.3063
L-Unigrams	0.0936	0.1009	0.0885	0.1019	0.0299

Table 5.10: 2% Filtered projected modularity values for different identity attribute modalities and for different contexts in the *Reopen* dataset. The prepended characters, B, N, and L represent bio, name, and location attributes, respectively. Values above 0.2 emboldened.

that Twitter communities differentiate themselves with multi-modal collections of identity-related attributes, it did so for non-contextualized data. By re-doing the analysis, as part of the pipeline, to contextualized data, we showed that this finding was not an artifact of interactional context mixing. This is not to say that community prototypes and interactional contexts are unrelated. Overall, location-based identity signals were weaker in the contextualized setting. Based on the prototype analysis of Chapter 4, location attributes were used to distinguish communities in specific locations, often ones that were outside of the United states. These communities also often had prototypes suggesting they spoke languages other than English. It seems that contextualization separated these communities into their own conversations. Within the US-centric contexts that were analyzed further, these communities, and thus these attributes, were not as important. In conclusion, some communities detected using non-contextualized data may inflate the presence of community prototypes, however we find that when this is controlled for prototypes are still present.

Chapter 6

Thoughts and Conclusions

6.1 Overview of Contributions

The first major contribution of this dissertation is the two methods developed in Chapter 2 for accounting for the interactional context of Twitter conversations. These methods enable us to move from the analysis of mixed context networks to contextualized networks, which in turn give a more reliable view of network structure. The analyses provided in that chapter demonstrated that contextualized networks can have radically different nodesets and different centrality rankings of the nodes they have in common. This is to that contextual mixing harms the validity of Twitter network analyses, and these methods provide a way of undoing that harm.

While Chapter 2 provided ways to perform existing network analyses with better accuracy, Chapter 3 showed that analysis of conversational contexts can help us answer new research questions. Specifically, we demonstrated that the 4 types of contextual dynamics (intra- vs. inter- context, and network vs. activity dynamics), work together to paint a rich portrait of a large online conversation. These analyses can be used to better understand the conversations that are present in a dataset at a high level, and are useful for better understanding the relationship between users. Specifically, the dynamic intra-context network analysis enabled us to further break down conversational networks by time periods of stable structure so that we can see how the network evolves over time. The method of dynamic community detection was demonstrated to be useful beyond conversational networks, where a change in the structure of Ukrainian legislator's co-voting network was identified. Turning to the dynamics between contexts, we were able to provide a method of detecting groups of suspicious users who may be coordinating to manipulate a discussion by moving from conversation to conversation in a synchronized way.

Next, various form of modularity vitality were developed, which links two of the largest and most active areas of Network Science Research: centrality and community detection. Specifically, modularity vitality was developed for unipartite, bipartite, and projected networks. Previous works considered network vitalities only for functions of a network. At the same time, there were efforts to develop community-aware centrality measures which also accounted for a network's partition. Modularity vitality joined these two efforts by taking

a vitality of modularity, which is a function of both a network and it's partition. This expansion in scope of network vitalities provided a community-aware centrality measure that was better tied to community detection theory than previous works.

In Chapter 4 we used this new network-based methodology to test the applicability self-categorization theory, a long-standing theory of offline social networks, to large-scale online networks for the first time. There, we found strong evidence that community prototypes exist, and that they are multi-modal. This means that Twitter users signal their online community membership using signals like hashtags, mentions, and emojis in their biography, while also using hashtags in their name and unigrams in their location field. Previous work had investigated how users signal their social identity with these modalities, and now we can tie this to a larger social process playing out over communities.

When considering how people change their identity signals, some results aligned with social theory of offline networks while others highlighted differences in the online setting. Specifically, we found that users who aligned with their community's identity had higher status, both within their community and on Twitter as a whole. This agreed with existing theory stating that prototypical users are more accepted by their group, and thus have higher potential for leadership. Theory also suggests that non-prototypical users should be more likely to update their identity signals to conform with their group's identity, since they should want to gain acceptance within their community. We demonstrated the opposite happens on Twitter: users who already fit in with their community are more likely to update their identity signals and are more likely to have those changes align with their group. This difference between online behavior and offline theory could be a result of social media savviness. Under this theory, there may be a difference in social media user's desires and their actions due to their understanding of their community and how to be successful on the platform.

Next, the contextualized network analysis pipeline developed in Chapter 5 enabled us to answer one last research question. Specifically, we demonstrated that the prototypes uncovered in Chapter 4 were not just an artifact of the interactional contexts studied in Chapters 3 and 4. After controlling for these contexts, community prototypes still existed.

Lastly, the contextualized pipeline was applied to a new dataset, starting all the way from data collection. This procedure serves as an example workflow that researchers can follow to apply these methods to answer their own research questions or to build off of the methods developed in this work.

6.2 Limitations and Future Directions

Perhaps the biggest limitation of this work is that it has only directly considered Twitter data. Twitter is a very important social media platform due to its widespread use among politicians, celebrities, and journalists for first reporting. However it is only the 7th largest social media platform in the United States, and is not growing as fast as smaller platforms like Reddit¹.

¹<https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/>

Twitter was selected for analysis because it met two criteria. First, it had both interactional context and personal context. Second, we were able to access the data from both. Because both of these criteria were met, we were able to fully explore contextualized analysis of conversations, and study the interaction between personal and interactional context. This would not have been possible on other platforms. Reddit, for example, does not have customizable biography attributes, meaning that there is no personal context on the platform to consider. Other platforms like Facebook and Instagram due have this feature, but researchers do not have access to the data to study it.

So, while Twitter was chosen out of necessity for our purposes, we cannot draw conclusions from our work beyond the scope of that platform. With that said, the tools developed, like that for the detection of prototypes and contextualization of Tweets are usable outside of Twitter, so long as the data is available. The application of these tools to understand differences in online behavior between other social platforms is of interest for future work.

Another major limitation of this work stems from Twitter's recommendation algorithms. These algorithms dictate the order in which content is shown to users, influencing their behavior. However, at time time of writing the specifics of these algorithms are unknown. Because of this, we cannot draw causal claims from our analysis, which is particularly important for that of community prototypes. Self-categorization theory seeks to make a causal claim: individual's social desires cause them to construct community prototypes and signal their prototypicality to the other members of their community. Our work shows that the outcome of this behavior does occur on social media: community prototypes do exist. However, the presence of Twitter's recommendation algorithms offer an alternative explanation for this observation that we cannot rule out. It is possible, even likely, that the algorithms use the similarity of user's biographies to rank content. This choice could also lead to the formation of prototypes, even in the absence of the social motivations described under the social identity perspective.

We cannot rule out these scenarios, and so we do not make any causal claims. However, some of our results provides evidence in favor of the social theory explanation of prototypes. When considering instances that users change their profile information, they are more likely to make changes that conform to their prototype. These decisions are totally up to the individual. That is they are not guided by any recommendation. Because users act in accordance to social theory in this scenario, it is possible that the theory holds even under the influence of recommendation. Future work may investigate this by modeling the affects of recommendation algorithms. As there is more pressure to provide transparency in social recommendations, hopefully this line of work should have more realistic assumptions to work with.

The last major limitation of the work is that contexts have been treated as discrete states. In the case of conversational contexts, each conversational network was modeled separately. In the case of personal contexts, each identity signal was treated individually. This discretization is extremely useful, without this we would be unable to apply any of the classical network analysis tools to our data. However, we know that conversations and identity signals are not truly discrete, they are related. Currently, a pair of very related conversations, say the *Reopen Protests* and the *Reopen Strategy*, are treated the same as an unrelated pair, like *COVID Information* and *Black Lives Matter*. We could see

the relationship between conversations through network similarity or the user transitions between the two, though this is a distinct type of similarity. We began to investigate alternative approaches for contextualized network analysis that account for conversation similarity in Appendix B with the development of vector contextualized networks. This approach, where each edge is represented in a vector space, is a powerful model that could be used to understand more complex social media platforms like TikTok, where interactions take place through multi-media content.

The problem of discrete states in identity signaling is more complex. While we know intuitively that `#blacklivesmatter` and `#blm` are related, their could be subtle differences in connotation or usage between different communities. Thus, building systems which can account for apparent similarities, perhaps through co-usage, while appropriately dealing with the complexities of social identity signaling is very difficult. This is a worthwhile challenge to undertake, as progress in this area could lead to an even better understanding of online social identity.

6.3 Beyond Twitter

The limitation that this work relies heavily on Twitter data is compounded by the fact that Twitter has updated its API prices, making academic research of the platform prohibitively expensive ². While there are large archives of old Twitter data available for researchers to examine, those interested in studying fresh data must turn to new sources. Although Twitter's change is a great loss for the academic community, it poses a great opportunity for researchers to broaden the scope of their study to online conversation beyond the platform.

Any website featuring a comment section is ripe for exploration using some of the methods detailed in this dissertation. Large platforms like YouTube, Reddit, and Tiktok, all have comment sections, but many other sites do too. A particularly interesting opportunity presents itself on news websites. Many of the major news sources have active comment sections below all of their articles.

The primary challenge for network scientists in branching out to these new platforms is modeling both the content of the comments and the content being discussed. For example, a network of comments on a collection of videos should capture the fact that both the videos and comments are related. Hopefully, the framework of vector contextualized network will provide a way forward in this new line of work.

6.4 Concluding Thoughts

The interactions that we have with each other online are complex. This complexity must be accounted for if we hope to get accurate answers to our research questions about social media and its impact on our society. We've taken preliminary steps towards this end, providing general frameworks to account for the personal and interactional context of these interactions. However, there is much work to be done. The social media landscape

²<https://www.wired.com/story/twitter-data-api-prices-out-nearly-everyone/>

is continuously evolving. While there are plenty of people engaging in online discussion in comment threads, alternative forms of online social connection are rapidly growing in popularity. Every day, more and more people are interacting with each other online by sharing videos and images, joining live streams, and entering virtual environments. Studying each of these new modalities will require some specialized research methods. We hope that this work can serve as a guide, and to some extent a unifying approach, for these future endeavors.

Appendix A

Manual Interactional Context Annotation Details

A.1 Reopen

A.1.1 Liberate Tweets

Trump Tweeted in support of the reopen protests, writing tweets like “LIBERATE MICHIGAN!” Seed nodes include these Tweets of support and articles discussing their implications.

A.1.2 Trump’s Job Reopening

A more general discussion formed around tweets and editorials talking about the pros and cons of Trump’s strategy of reopening the country.

A.1.3 Trump Refuses CDC Guidance

The Trump administration announced it would not implement the Centers for Disease Control and Prevention’s 17-page draft recommendation for reopening America.

A.1.4 Fauci Critique

Sparked largely from President Trump’s discussion began questioning Fauci’s qualifications, trustworthiness, and the job that he had done advising the country on COVID safety.

A.1.5 Recall Whitmer

A push to recall the Governor of Michigan, Gretchen Whitmer, due to her perceived failures in handling the pandemic. The concerns were primarily economic, though some claimed without evidence that her lockdown policies caused more to get sick.

A.1.6 Florida Data Scientist

A Florida data scientist accused state officials of covering up the extent of the pandemic and was subsequently fired.

A.1.7 Arizona Scientists

Similar to the Florida case, the Governor of Arizona fired a scientist speaking out against the Governor's plans to reopen the state.

A.1.8 Reopen Updates

Many articles were posted that various places or businesses were reopening, not reopening, closing down, or announcing an extended lockdown.

A.1.9 Reopen Commentary

A general discussion about the affects of lockdown, reopening, and the experience.

A.1.10 Reopen Strategy

Opinion pieces and Tweets from major influencers detailing what they think the reopen strategy should be. The most prominent of which was an editorial from Joe Biden.

A.1.11 Economy

A general discussion about the economy including the state of business and the stock market.

A.1.12 Worker Unemployment

A more specific discussion about how worker unemployment was related to the lockdowns, mostly political and separate from the discussion about economic indicators.

A.1.13 Mask Orders

Discussion about masks orders, specifically, which were discussed mostly separately from lockdowns and things reopening.

A.1.14 Schools

Updates and discussion about school closures, the safety surrounding in-person education and the alternatives.

A.1.15 Reopen Satire

Satire about the state of the country and the efforts to reopen.

A.1.16 COVID Information

Information about the latest COVID statistics and safety instructions. Most discussion centered around URLs to dashboards.

A.1.17 Worker Precautions

Discussion about what workers need, and in some cases were not getting, to safely return to work.

A.1.18 Lockdown Hypocrisy

Frustration about proponents of COVID-restrictions, particularly politicians, breaking the rules or guidelines.

A.1.19 Vaccine

Discussion about vaccine development.

A.1.20 Anti-Vaccination

Both discussion about anti-vaccination messages, and meta-discussions about the groups pushing them.

A.1.21 People Testing Positive

Updates about famous people testing positive for COVID.

A.1.22 Anti-Mask Violence

Stories of people who are anti-mask committing acts of violence because of mask rules.

A.1.23 Punishment for Lockdown Violators

Ethical discussion about consequences for lockdown violators in the form of fines, jail time, or loss of employment.

A.1.24 Reopen Protesters

Discussion about the many reopen protests across the country, but primarily about Operation Gridlock in Michigan.

A.1.25 Protest Coordination

Discussion about how the reopen protests appear to be coordinated, and how things like Trump's Tweets may have helped in that coordination.

A.1.26 Black Lives Matter

The resurgence of the Black Lives Matter movement following the murder of George Floyd.

A.1.27 Bot Story

An interview with Kathleen Carley was released implying that a large number of the accounts tweeting about the Reopen protests were bots.

A.1.28 Reopen Criminal Cases

Calls to reopen various criminal investigations. Those that could be linked to BLM were, so these are separate discussions of events.

A.1.29 Petitions

Miscellaneous petitions that were not directly linked to BLM.

A.1.30 Lowes Donation

Lowes announced it was donating 25 million dollars in grant money for minority-owned businesses trying to reopen amid the COVID-19 coronavirus disease pandemic.

A.1.31 Healthcare Workers

Discussion about healthcare workers' efforts throughout the pandemic, largely on World Health Day.

A.1.32 Hurricane Support

A joint support effort was announced from the 5 living former presidents.

A.1.33 General Flynn

Discussion surrounding the opening of a legal fund for General Michael Flynn to support him through the investigations into his actions in the 2016 presidential campaign.

A.1.34 General Politics

Discussions about politics that were not tied to a specific event and were distinct from the other contexts.

A.1.35 Boycott China

Calls to boycott Chinese goods and services.

A.1.36 Entertainment

Various news and discussion about art, music, and popular culture.

A.1.37 Memes

Many of the top tweets were memes about the pandemic or the protests but were too general to be categorized into those contexts.

A.1.38 Oregon Burning Aborted Babies

An energy plant in Oregon was reported to be burning medical waste from Canada to provide power. Aborted fetuses were included in the tissue, sparking outrage. Though the primary link discussed was from a questionable pro-life news source, lifenews.com, the story was later verified on Snopes.

A.1.39 Miscellaneous

A final context for those that did not fit in the others, sometimes personal anecdotes or jokes.

A.2 Election

A.2.1 Claims of Fraud

A number of false claims that the election was being stolen or that fraud was being committed were spread during the vote counting period of the election.

A.2.2 Spam

Various promotional URLs that were mass-replied to popular tweets.

A.2.3 Biden Campaign

Campaign videos and messages supporting the Biden and Harris ticket.

A.2.4 Trump Campaign

Campaign videos and messages supporting the Trump and Pence ticket.

A.2.5 Election Updates

Official updates and live-view voting maps as the votes were being tallied.

A.2.6 Biden Won

Discussion that Biden had won the election starting before the bulk of the mail-in votes were cast in key states like Pennsylvania, as Biden supporters anticipated the bulk of these votes would be democratic based on previous partisan differences in mail-in voting.

A.2.7 Trump Won

Discussion that Trump had won the election starting the night of the election when he lead in early counts prior to the inclusion of mail-in votes. Discussion that Trump won continued as supporters believed the false claims of election fraud.

A.2.8 Vote Information

Information on how to vote, including the application to register, the location of polls, and instructions for casting mail-in ballots.

A.2.9 Vote Counting

Chatter about the votes being counted, including calls to count all the votes and to stop the count.

A.2.10 Trump Has COVID

Tweets about Trump catching COVID in early October of 2020. Discussion stems from people reflecting on those events in light of the election.

A.2.11 Democrat Comedy

Comedy videos supporting the democrats.

A.2.12 Biden Racism

Discussion centered around videos of racist comments from Joe Biden when he was a senator.

A.2.13 Antifa

Unsubstantiated claims about what Antifa was doing during the election.

A.2.14 Democratic Fundraising

Links to raise money for democratic campaigns.

A.2.15 A\$AP Rocky

Discussion about Trump's role in the release of rapper A\$AP Rocky from jail in Sweden.

A.2.16 Biden's Bus

Video and discussion about a Biden campaign bus which was surrounded on the highway by a caravan of pickup trucks displaying Trump flags. The trucks slowed the bus down until the cops were called and they intervened.

A.2.17 Hunter's Laptop

Discussion of Hunter Biden's laptop, which was abandoned at a Delaware computer shop in 2019. The laptop sparked controversy when Trump supporters claimed it contained evidence of corruption.

A.2.18 Attacks on Voting Officials

Instances of attacks or attempted attacks on voting officials, which were carried out under the belief that election officials were committing fraud.

A.2.19 Kamala Equity

Kamala Harris Tweeted a video explaining the concept of equity, which lead to conservatives to claim that she was endorsing communism.

A.2.20 Covid and Trump

Discussion of how Trump handled COVID.

A.2.21 Project Veritas

Discussion of content produced by Project Veritas, an American far-right activist group founded by James O'Keefe in 2010. The group produces deceptively edited videos of its undercover operations, which use secret recordings in an effort to discredit mainstream media organizations and progressive groups

A.2.22 Voter Purge

Discussion of the removal of voters from the public registry.

A.2.23 Election Memes

Various memes discussing the election in a way that does not fit better into one of the other contexts.

A.2.24 USPS

Discussion about USPS, efforts to defund it, and its role in the election by delivering mail-in ballots.

A.2.25 Trump to Declare Early

Early warnings that Trump was likely to declare victory on election night despite knowing that the results would not be finalized and were likely to move against his favor as mail-in votes were counted.

A.2.26 Biden's Health

Speculation about Biden's health, with many saying he was unfit for office.

A.2.27 Trump Motivation

A collection of motivational videos edited by fans of Trump using his speeches as the narration.

A.2.28 Suing Trump

Speculation about the lawsuits that Trump would face should he lose the election.

A.2.29 Black Lives Matter

Discussion of the Black Lives Matter movement.

A.2.30 Deported Veteran

Alex Murillo's story about being deported to Mexico after serving in the United States Navy.

A.2.31 Meadows Gets COVID

White House chief of staff Mark Meadows tested positive for COVID.

A.2.32 Alex Trebek

The death of Alex Trebek, beloved host of Jeopardy! the game show.

A.2.33 Anti-QAnon

Discussion of the dangers of QAnon, the far-right conspiracy theory and political movement.

A.2.34 Federal Workers

Discussion of Executive Order 13957, which created a new class of federal employees within the civil service making it easier to hire and fire civil service employees.

A.2.35 NBA White House

Discussion of the tension between NBA players and Trump, specifically surrounding the traditional invitation of the NBA champions to the White House.

A.2.36 Miscellaneous

A final context for those that did not fit in the others, sometimes personal anecdotes or jokes.

Appendix B

Vector-Contextualized Networks

B.1 Development of Vector Contextualized Networks

In Chapter 2, Deep Tweet Infomax was developed to obtain vector-representations of Tweets, and thereby of interactions. This approach recognized that context can be represented as a vector, which enables us to compute similarity. Thus, we can use the vectors to determine interactions that are similar to one another. From there, we clustered the data into discrete contexts, which can be thought of as conversations. The transition from a continuous representation of contexts to a discrete one enabled us to apply network and sequence analyses to the data.

However, the discretization of contexts loses information. How similar are two discussions? After clustering and considering discussions as discrete, we lose the ability to answer this question directly. In this appendix, we demonstrate a framework for network analysis on vector-contextualized data that does not require the discretization step.

Vector-contextualized data is that which encodes interactions between entities in a vector space. This type of data is shown in Figure B.1. A simple example of vector-contextualized data is a human contact network, where the location of physical interactions is recorded. Thus, each edge of the network is given a location in a 2-dimensional vector space. Note the difference from spatial networks, which assign nodes a fixed location in vector space [60]. Here, nodes are free to move around.

Sticking with the interaction network example, we can think about how to construct a network. Consider that we want to study the interactions that occur on a college campus. In this scenario, a boundary in the vector space is drawn, and edges within that boundary are included while those outside the boundary are excluded. From there, we could tally up the edges to form a network. We could also imagine cases where the distance of an edge to a reference point, say the center of the campus, could indicate its importance.

With these intuitions in place, we define vector contextualized networks. The following are the requirements for vector contextualized networks:

1. Network data (interactions between discrete entities) in a vector space.
2. An *inclusion function* set by the researcher. This function determines whether or not edges should be included in the network

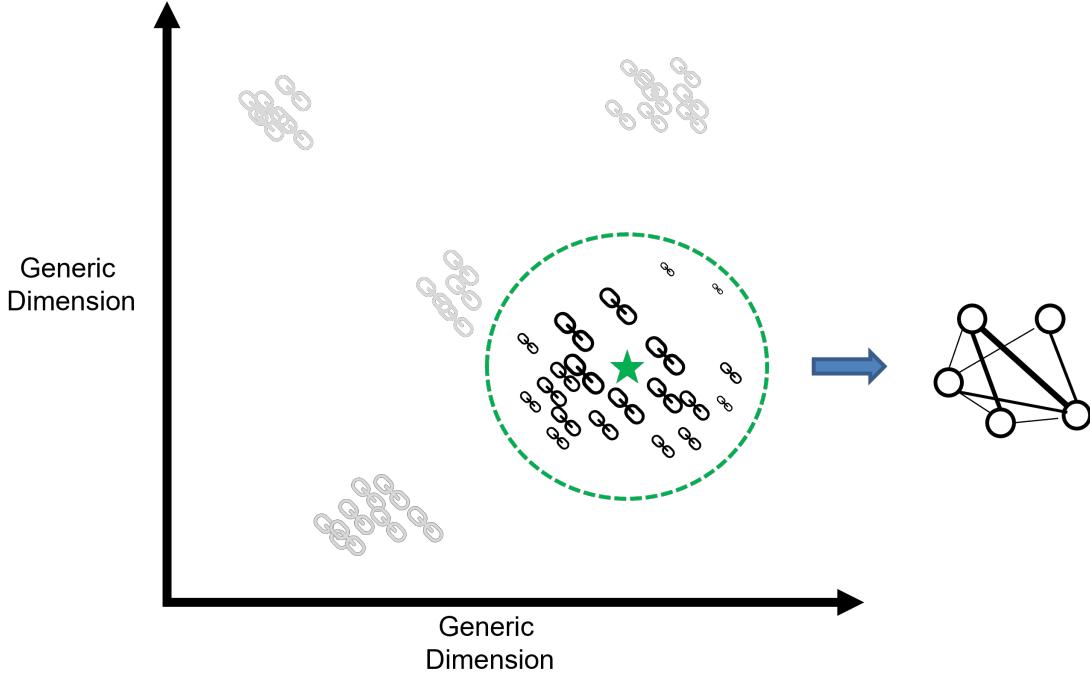


Figure B.1: Schematic of vector-contextualized networks.

3. Optional: a weighting function based on the distance from an edge to a fixed reference point, set by the researcher. The weighting function determines how much the edge instance increases the edge weight in the vector contextualized network.

From there, a vector-contextualized network is constructed as follows. The edges determined irrelevant by the inclusion function are discarded. If an edge weight function is supplied, it is used to determine the weights of the edges. Finally, the edge weights are summed over to obtain a single vector-contextualized network. Note that interactions between entities can re-occur in multiple locations. The weights of these instances are summed over to give a single weight between the two entities.

This process is illustrated in Figure B.1. A reference point is defined in the space. From there, the weighting function dictates that the further an edge is from the reference point, the lower its weight. Eventually the weights are so low, that they are not considered, as indicated by the radius surrounding the reference point. Thus, the inclusion function indicates that only those within the radius are to be included. Lastly, the edge-weights are summed over to obtain a network.

We can see that as the reference point is moved, we obtain a different network specifically centered around that point. This means that different networks can be constructed for different points in the space. If the reference points are near to each other, than we expect their networks to be correlated. Thus, the similarity originally encoded in the vector space will be translated into our contextualized networks.

B.2 Case Study on the Election Dataset

We now demonstrate a vector-contextualized network analysis on the *Election* dataset, where vectors are obtained from Deep Tweet Infomax. The first question, is how do we set our reference points in vector space? To reference points that are both objective and interpretable, we use hashtags. Specifically, we pick out important hashtags within the dataset and construct a vector-contextualized network around each point. To start, the reference hashtags and their similarity in the vector space are shown in B.2.

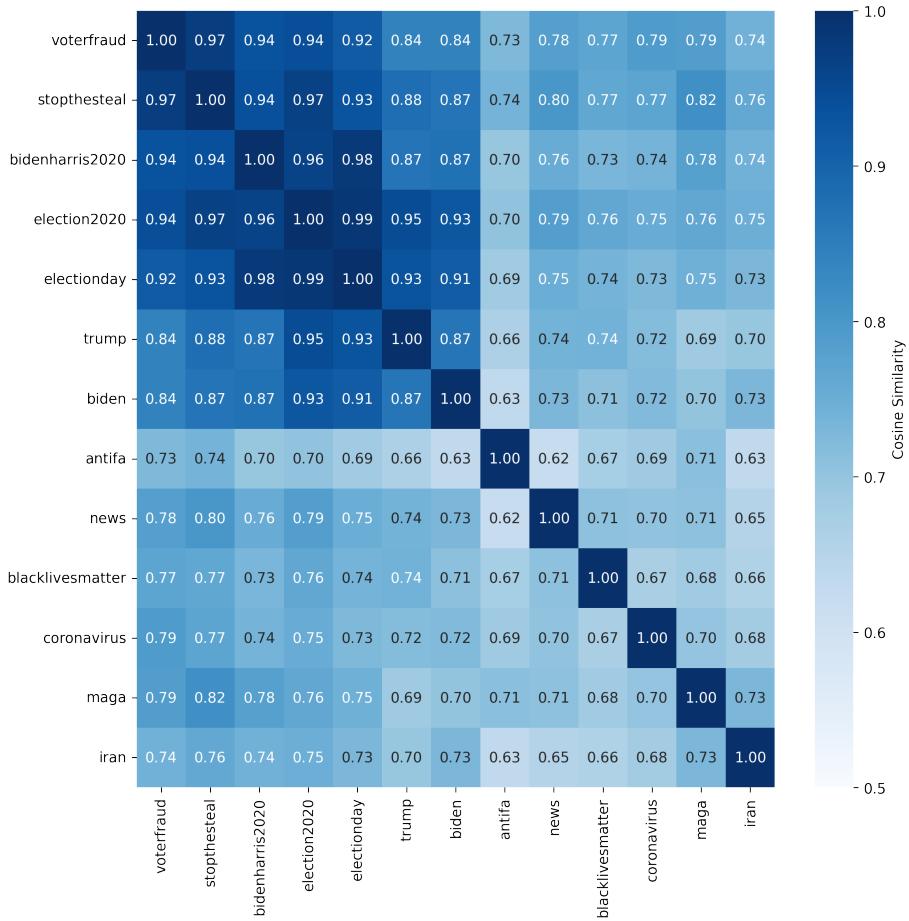


Figure B.2: Vector similarity between hashtags in the *Election* dataset.

Now that the reference points are in place, we move to define an inclusion function. We define this function to include all edges that have cosine similarity greater than 0.65 with the reference point in question. The threshold was somewhat arbitrarily selected, though runs with alternative thresholds lead to similar results. Finally, an edge-weighting function was selected. Here, we decide to evenly weight all edges that are included in the network.

To summarize, we build a network surrounding a hashtag. We consider all Tweets which are similar to that hashtag (cosine similarity above 0.65). For those included tweets, we construct a weighted interaction network between users where the weights indicate the

number of interactions between them.

Applying this to the *Election* dataset, our first question is how do the contextualized network compare in their most basic properties? To answer this, we show the overlap in each pair of vector-contextualized network’s nodesets and edgesets in Figure B.3a. We see that for related hashtags, or those that are similar in the vector space, we obtain similar networks. In our data, this corresponds to a very similar set of networks for the most prominent hashtags in the data, as shown in the top-left corners of the plots. These networks contain nearly all the same nodes and edges.

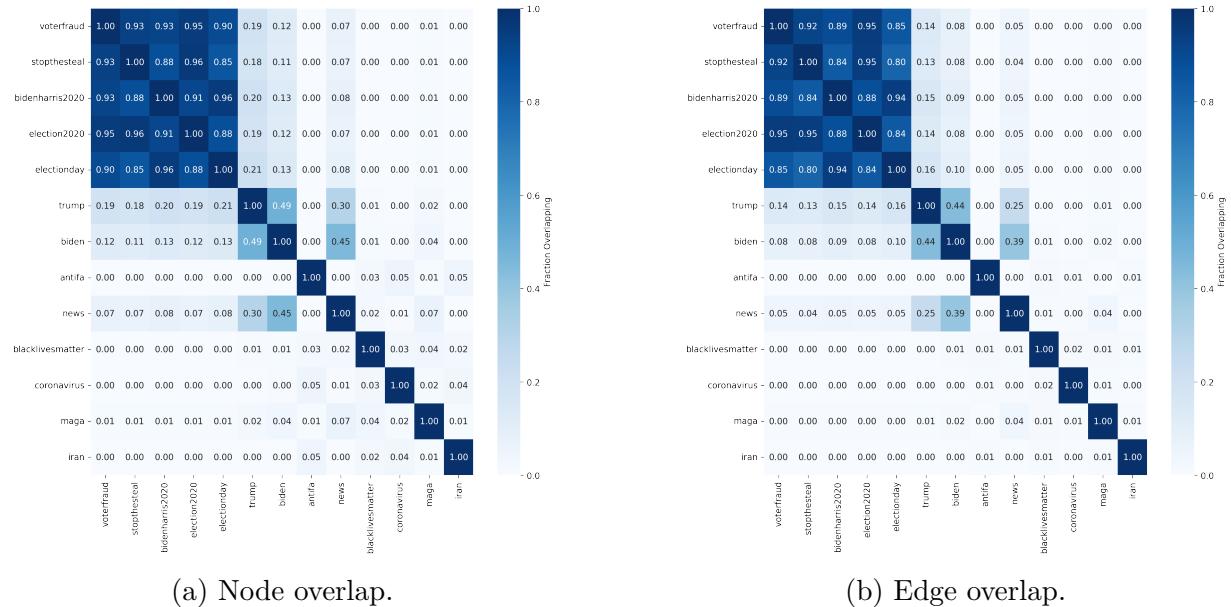


Figure B.3: Overlap of nodes and edges in the vector-contextualized networks of the *Election* dataset centered around different hashtags.

In contrast, the more independent hashtags have more independent networks. These hashtags, like #news, are not similar to the other hashtags because they are talked about in a different way. Intuitively, we can understand these as different conversations. That is, the discussion of #news was very different from the discussion of #blacklivesmatter. As a result, the networks, too, are very different. Specifically, we see that they have very little overlap in both their nodes and their edges.

Mirroring our analysis in Chapter 2, we also compare the centrality rankings between contexts. Again, we control for the differences in nodesets, and only rank the nodes based on their overlapping subgraphs. The nodes are ranked according to their Pagerank, and the Kendal-Tau correlation is shown between contexts in Figure B.4. Again, we see that similar contexts have very similar centrality rankings. However, this time, we see that less similar contexts, such as #trump and #biden, still have high correlation of their centrality rankings. Part of this result is due to the fact that the same influential people (those with many followers) are active in both discussions. In other cases, such as the comparison between #antifa and #coronavirus, the effect is due to the small number of users active in both discussions.

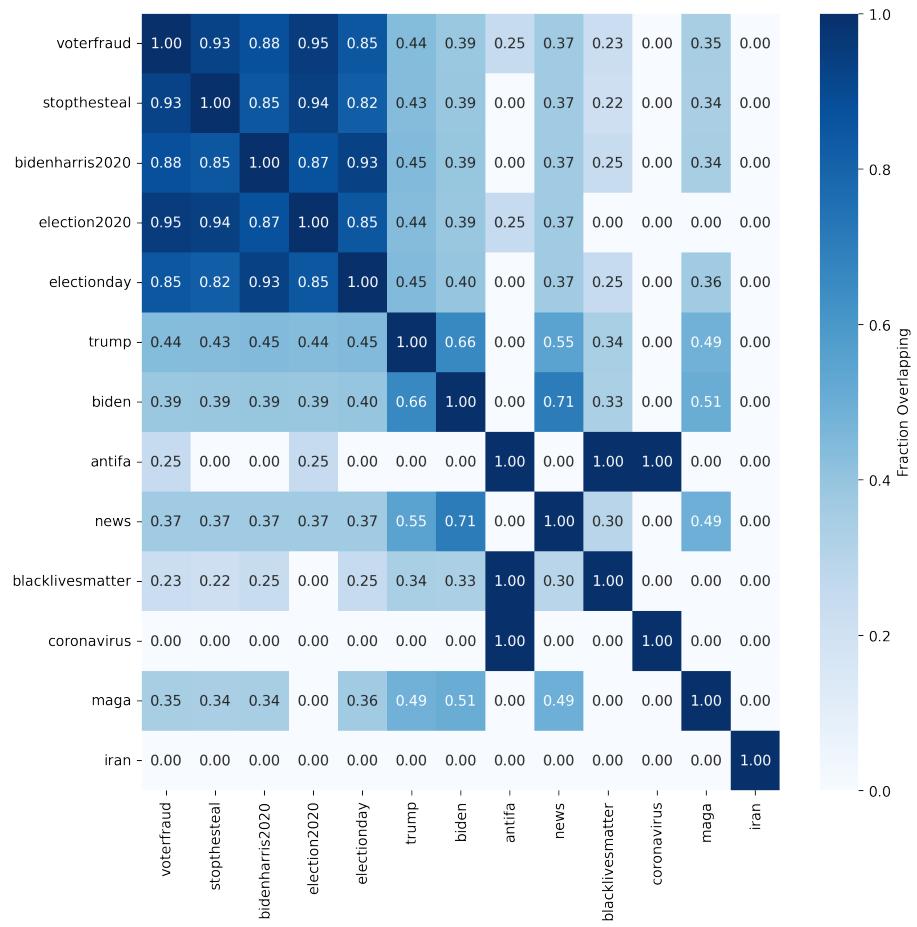


Figure B.4: Correlation of centrality ranking of nodes in different vector-contextualized networks within the *Election* dataset.

In summary, this case study has demonstrated that the vector-contextualized network approach can leverage the tools of standard network analysis, while still incorporating the vectorized contextual information. Thus, when consider similar contexts, we get similar networks. The approach is very general, and could be applied to any situation where vectors can be used to capture the similarity between a set of interactions.

Appendix C

Modularity Vitality

C.1 Introduction

Modular structure is a key phenomenon in the study of real-world networks. Networks from a wide array of disciplines exhibit modular structure, meaning that nodes tend to be found in well-connected groups[75]. Discovery of these clusters have been repeatedly shown to be meaningful within their context though empirical studies [90, 121, 167]. Further, a “No Free Lunch” theorem has been proved for community detection, stating that no algorithm can uniquely solve community detection, and implying that multiple valid community definitions can exist for a single network [170].

Another fundamental question in Network Science is that of centrality. Put simply, how important is each node in a network? Many centrality measures have been defined over the years, each measuring “importance” in a different way. Classically, centrality measures are defined to be a graph invariant. However, network communities have been shown to be pervasive in nature, and it has been shown that networks can have multiple meaningful definitions of communities. So, it is natural to ask the question: how important is each node in a network *given some definition of groups?* When group structure is considered, the relative importance of nodes may change. For example, a fairly average node in classical terms may be a hub within a small community, boosting its importance within this context. In this work we refer to centrality measures accounting for community structure as “community-aware centrality measures.” The question of community-aware centrality lies at the intersection of the fundamental areas of centrality and community structure. As such, applications to community-aware centrality are far-ranging. Here, we show applications to immunization strategies for infectious disease, robustness testing for large infrastructure networks, and privacy-based data filtering strategies.

Most of the existing community-aware centrality measures extend classic centralities by considering within-community links and between-community links separately, before applying a weighted sum to get a single score [72, 73, 85]. This approach acknowledges the difference between links which fall within communities and those which fall between them, but ultimately gives no insight into what role a node is playing; hub-nodes and bridge-nodes can receive similarly high values without a way to distinguish them. Further,

the weighting schemes to date have been hand-crafted, rather than derived from existing community theory, making them somewhat subjective. Cherifi et al. have acknowledged that there is room for improvement on this front [45].

When discussing the modularity matrix, Newman introduced “community-centrality,” which measures a node’s potential to contribute to group structure [157]. Since the measure was of *potential* contribution, community-centrality is a classical centrality-measure, independent of any defined partition. To obtain a community-aware centrality from a similar line of reasoning, we propose to measure a node’s actual contribution to the group structure encoded in a specific partition. By doing so, we obtain a community-aware centrality grounded in community detection theory and free from hand-crafted weighting schemes.

For the measure of *actual* node contributions, we turn to vitalities [119]. In their work, Koschützki et al. define vitality as the difference between the value of an arbitrary real function, f , applied to the graph G and the same function’s value when applied to the graph G with the vertex of interest removed. By doing this, a single node’s contribution can be measured and the observed value can be positive or negative. This is closely related to the key-player problem, which roughly asks to what extent a network is relying on a node’s presence to remain cohesive [25].

If the graph index is chosen to be a cluster quality metric, the vitality, then, measures a node’s contribution towards group structure. There are many such cluster quality metrics in the literature to choose from [128]. Vitalities have previously only been applied to classical centrality measures, thus, they have been defined as functions that only take graphs as arguments. Since we are interested in community-aware centralities and vitalities, we will define vitality as a function that takes a graph and its partition as arguments.

Nodes can contribute positively or negatively to community structure. This difference is encoded in the vitality’s sign, allowing us to distinguish nodes based on their role. Nodes which have negative cluster quality vitality are detrimental to group structure, meaning that they are connecting groups, making them a community bridge. Similarly, positive cluster quality vitality nodes are community hubs.

The focus of this work is on a specific cluster quality vitality - modularity vitality. Newman’s modularity is used as the objective function for many popular community detection algorithms, making it a natural choice to measure cluster quality [22, 29, 46, 158, 213]. Thus, this measure has stronger grounding in community theory than those prior, with no need for a hand-crafted weighting function. We show that manipulation of the original modularity function leads to a scalable method of calculating modularity vitality, where the calculation for all nodes scales as $O(M + NC)$ time, where M is the number of links, N the number of nodes, and C the number of communities.

Modularity vitality was tested on generated modular networks and on two real-world networks: the Pennsylvania Road Network, and a large Twitter network collected from the discussion of the Canadian Election of 2019. In our experiments, modularity vitality out-performs existing community-aware centralities showing potential applications for immunization strategies, control of diffusion over networks, and for robustness testing.

While other studies have demonstrated the fragility of infrastructure networks, in our first case study, we show that the road network is over 8 times more fragile than could be seen with measures only weakly tied to community detection theory [51]. By targeting only

1.6% of nodes with lowest modularity vitality, the PA road network’s largest component can be reduced to less than 1% of its original size, effectively destroying the network.

In the second case-study, the social media communication network was extremely robust, as demonstrated through the ineffectiveness of all community-aware centrality attacks on the network. The robustness of Twitter networks has serious implications for Social Cybersecurity [39, 164]. One of the core areas in this emerging discipline is developing counter-measures for the mitigation of fake or misleading news on social media. The problem of “Fake News” has gotten more attention recently, though many basic questions in the space are left open [124]. It is often suggested that network metrics can be used to identify points for stopping the spread of misinformation [197]. However, our results suggest that this is not the case. The robustness of Twitter networks suggest that even well-targeted interventions at the user level are unable to hamper the ability of information to spread. This result is aligned with the observed phenomena that misinformation continually resurfaces on social media [196].

Lastly, we show that modularity vitality can be used to perform greedy attacks to decrease modularity. This gives an alternate approach to the community-deception problem, which seeks to obscure communities from detection algorithms by altering network links in order to preserve privacy. Modularity vitality was used to perform community-deception on a large twitter network. The method decreased modularity by 41%, however this decrease comes at the cost of 2% of nodes and 45% of edges. While a removal of 2% of nodes leads to a sizable decrease in modularity, this process has diminishing returns. This suggests that a scalable and effective strategy for community deception is to obscure which popular accounts a user follows. This differs from the typical strategy, which rewrites edges instead of deleting them.

C.2 Prior Work

C.2.1 Preliminaries

Before describing the prior work, we begin with the notations and definitions that we will rely on for the remainder of the work.

Definition C.2.1 (Graphs). A graph is a pair $G = (V, E)$ where V is a set of nodes or vertices, and E is a set of edges or links. Let us denote $N = |V|$ as the total number of nodes and $M = |E|$ as total the number of edges. Let $v_i \in V$ denote a node i and $e_{i,j} = (v_i, v_j, w_{i,j}) \in E$ denote an edge between nodes i and j with weight $w_{i,j} > 0$. Finally, the adjacency matrix A is an $N \times N$ matrix with $A_{i,j} = w_{i,j}$ if $e_{i,j} \in E$ and $A_{i,j} = 0$ otherwise. For this work we only consider undirected graphs, that is $A_{i,j} = A_{j,i}$.

Definition C.2.2 (Partitions). A partition of graph G is $\mathfrak{C} = \{\gamma_1, \gamma_2, \dots, \gamma_C\}$ where γ_i is the set of nodes within community i s.t. $\gamma_i \cap \gamma_j = \emptyset, i \neq j, \forall i, j \in \{1, \dots, C\}$, and $\gamma_1 \cup \gamma_2 \cup \dots \cup \gamma_C = V$. We denote $C = |\mathfrak{C}|$ as the total number of communities. For convenience, we define a community vector, $\mathbf{c} = [c_1, c_2, \dots, c_N]$, where c_i indicates the community of node i .

Definition C.2.3 (Total and Community Degrees). The total degree of a node v_i is equal

to the sum of its edges. Let us denote this by $k_i = \sum_j A_{i,j}$. Next, define the community-degree of node v_i as the sum of edges towards nodes belonging to community c . We denote this as

$$k_i^c = \sum_{j=1}^N A_{i,j} \delta(c_j, c)$$

where the $\delta(a, b)$ is an indicator function s.t. $\delta(a, b) = 1$ if $a = b$, 0 otherwise.

Definition C.2.4 (Internal and External Degrees). The internal degree of node v_i is the sum of edges connected to v_i within its community. That is $k_i^{\text{internal}} = k_i^{c_i}$. The external degree of node v_i is the sum of edges connected to v_i and communities not equal to that of v_i . That is

$$k_i^{\text{external}} = \sum_{j=1}^N A_{i,j} (1 - \delta(c_i, c_j)) = k_i - k_i^{\text{internal}}.$$

The number of internal links in the graph G is given by $M^{\text{internal}} = \frac{1}{2} \sum_{i,j} A_{i,j} \delta(c_i, c_j)$.

Definition C.2.5 (Group-Fraction). Let G be a graph and \mathfrak{C} be a partition of the graph G . The group-fraction of community c is given by

$$\mu_c = \sum_{v_i \in \gamma_c} \frac{k_i^{\text{internal}}}{k_i} = \sum_{v_i \in \gamma_c} \frac{k_i^c}{k_i}.$$

Note that this is not equal to the fraction of edges within a community.

C.2.2 Modularity and Grouping

There is no “best” way to evaluate cluster quality and as such, many cluster quality metrics have been defined [128]. While vitality measures on any of these cluster quality functions could be an interesting and unique contribution, we focus our work on modularity. We have chosen modularity for several reasons. First, some of the earliest discussions of community-aware centrality are given by Newman when exploring modularity [157]. Next, many of the popular community detection algorithms attempt to maximize modularity. Thus, studying the vitality of the quantity used to obtain the communities in the first place keeps measures consistent. Lastly, we will show that modularity vitality in particular has a non-trivial vitality function which can be calculated efficiently.

The most common definition of modularity is that given by Newman, which is the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random [158]. The definition of Newman modularity is as follows.

Definition C.2.6. Given graph G and partition \mathfrak{C} , let us define modularity as the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random [158]. We can write modularity Q of the graph G as:

$$Q(G, \mathfrak{C}) = \frac{1}{2M} \sum_{i,j} \left(A_{i,j} - \frac{k_i k_j}{2M} \right) \delta(c_i, c_j), \quad (\text{C.1})$$

Modularity in this form has been studied extensively, and the most commonly used community detection algorithms seek to maximize this quantity [29]. Because it is an NP-hard problem, many different methods have been proposed to varying degrees of success [22, 46, 213]. The Louvain method has prevailed for years, and has repeatedly been shown to give meaningful communities in empirical studies [22].

However, recently, Traag, Waltman, van Eck have shown a flaw in the Louvain method [213]. Because of its update step, Louvain does not guarantee that its communities are internally connected. It was shown that, in fact, many communities are often not connected when using the method on real-world datasets. To fix this, Traag, Waltman, and van Eck have proposed Leiden grouping, which is slightly faster than Louvain, guarantees well-connected communities, and often achieves higher modularity. As such, we proceed using Leiden grouping.

C.2.3 Network Centrality Measures

Newman began the discussion of centrality based on community structure when studying the modularity matrix [157]. He defined “community-centrality” based on the eigenvectors of the modularity matrix. Despite its name, this is a classical centrality measure. Instead of measuring the actual contribution of a node, community-centrality measures a node’s *potential* to impact modularity. The derivation from the modularity matrix give community-centrality a strong theoretical link to communities, but has some drawbacks.

First, potential impact can be very different from actual impact. A related second point is that methods which only use graph structure are unable to adapt to different graph partitions, which is significant given that networks can have multiple meaningful definitions of communities. Lastly, there are some practical issues. The modularity matrix is dense, making it memory inefficient. Additionally, approximations are typically needed for computation on large graphs.

Masuda takes an eigenvalue approach to achieve a community-aware centrality, though not one derived from modularity [140]. Instead, he builds off of the idea of dynamical importance as defined by Restrepo et al [184]. The largest eigenvalue of a graph’s adjacency matrix is related to the ease of diffusion over the graph. Based on this fact, dynamical importance orders nodes based on the change in largest eigenvalue from the node’s removal. To leverage group structure, Masuda applied this strategy to the group-to-group network, calling it the “mod-strategy.” This method is computationally efficient since only the largest eigenvalue is needed, and because it is calculated on the group network, which is far smaller than the actual network. Formally, nodes were ordered based on the following equation:

$$\text{Mas}_i = (2\tilde{u}_{c_i} - x) \sum_{c \neq c_i} \tilde{u}_c k_i^c \quad (\text{C.2})$$

$$x = \frac{1}{\tilde{\lambda}} \sum_{c \neq c_i} \tilde{u}_c k_i^c, \quad (\text{C.3})$$

where $\tilde{\lambda}$ is the group network’s largest eigenvalue, and $\tilde{\mathbf{u}}$ is its corresponding eigenvector. Intuitively, the value of the eigenvector corresponds to the importance of that group.

Thus, Masuda's method gives importance to nodes based on the group it belongs to, and its connectivity to other important groups. The more connections to important groups, the higher the score, meaning that nodes bridging communities will be ranked highly.

More recently, degree-based measures have taken favor, due to their interpretable form and their scalability. To get at the relationship within and between communities, these measures use internal degree and external degree.

One of the earlier examples is “commn-centrality,” CC , proposed by Gupta et al [85]. This centrality is defined as follows:

$$CC_i = \left(1 - \frac{\mu_{c_i}}{|\gamma_{c_i}|}\right) \frac{k_i^{\text{internal}}}{\max_{v_j \in \gamma_{c_i}} k_j^{\text{internal}}} \times R_{c_i} + \left(1 + \frac{\mu_{c_i}}{|\gamma_{c_i}|}\right) \left(\frac{k_i^{\text{external}}}{\max_{v_j \in \gamma_{c_i}} k_j^{\text{external}}} \times R_{c_i} \right)^2 \quad (\text{C.4})$$

where R_{c_i} is user-defined, but is commonly chosen as $R_{c_i} = \max_{v_j \in \gamma_{c_i}} k_j^{\text{internal}}$. The group fraction μ is used so that internal degree takes precedence for weak groups, and out degree takes precedence for strong groups. One issue with commn-centrality, however, arises when a community is disconnected from the rest of the graph. In such a case, $\max_{v_j \in \gamma_{c_i}} k_j^{\text{external}} = 0$, so commn-centrality is undefined. This commonly occurs, especially during network robustness testing, so we do not consider commn-centrality in our experiments.

Afterward, Ghalmene et al. have proposed a number of alternatives which are well defined for community components [72, 73]. The simplest of which is the number of neighboring communities centrality, which just counts the number of communities in a node's immediate neighborhood; we will call it b_i . Expanding on this, the community hub-bridge centrality, CHB was defined as:

$$\text{CHB}_i = |\gamma_{c_i}| k_i^{\text{internal}} + b_i k_i^{\text{external}} \quad (\text{C.5})$$

where, again, b_i is the number of communities neighboring node i . [73]. The number of neighboring communities centrality was out-performed by the more sophisticated community-hub-bridge centrality, so we omit it from our results to preserve readability.

Generalizing this approach beyond just degree, Ghalmene et al. introduce “modular-centrality”. They note that a graph G can be decomposed into G^{internal} and G^{external} , where only the internal or external links are retained, respectively. Then, internal centrality can be calculated as: $\Gamma^{\text{internal}}(G) = \Gamma(G^{\text{internal}})$, where Γ is a classical centrality measure. The same logic can be used to obtain external centrality. It can be seen that when Γ is selected to be the degree, we get the same internal and external degree as we have previously defined. Modular centrality is a two-dimensional vector encoding internal and external centrality. Ghalmene et al. note that there are many ways that this vector can be used to obtain a single number, as is needed for ranking tasks. One of their proposed methods is the weighted modular centrality, WMC, which takes a weighted sum of the components:

$$\text{WMC}_i = \mu_{c_i} \Gamma_i^{\text{internal}} + (1 - \mu_{c_i}) \Gamma_i^{\text{external}} \quad (\text{C.6})$$

where μ_{c_i} is, again, the group fraction for community c_i . Note that this is the opposite weighting scheme as Gupta's; when communities are strong, modular-centrality places preference to internal degrees. We also see Masuda's weighting giving preference to bridges. To cover the full spectrum of these previous community-aware centralities, we also consider an adjusted version of modular-centrality, AMC, where the weighting scheme favors bridges:

$$\text{AMC}_i = (1 - \mu_{c_i})\Gamma_i^{\text{internal}} + \mu_{c_i}\Gamma_i^{\text{external}} \quad (\text{C.7})$$

Note that this has also been previously defined as as “Weighted Community Hub-Bridge” centrality [73]. Due to the similarity of the measures, we will continue using the name “Adjusted Modular Centrality.” We also note that Ghalmane’s work has been extended to overlapping communities, however this work only considers non-overlapping community structure [71].

With the exception of Masuda’s work, these methods all rely on a weighting scheme of internal and external centrality. The weightings are not derived from network-theoretic principles, but are based on observations seen in network studies. Ideally, a centrality would be derived from established theory, and would eliminate the need for comparison of subjective weighting. While Masuda’s measure is derived from network theory, it is based on network connectivity, rather than community detection.

For this work, we will look to vitalities, which measure a node’s contribution to some global property [119].

Definition C.2.7 (Network Vitality). For an arbitrary real function $f : \mathcal{G} \rightarrow \mathbb{R}$ defined on graph space \mathcal{G} we write the associated vitality V_f as:

$$V_f(G, i) = f(G) - f(G - \{i\}),$$

for any $G(V, E) \in \mathcal{G}$ and $i \in V$. Where $G - \{i\}$ denotes the graph G after the removal of node i .

To the best of our knowledge, vitality measures of cluster-quality functions have yet to be studied. When cluster-quality functions are considered, the graph index must also be a function of the network partition, \mathfrak{C} . Here, we select f to be modularity, giving modularity vitality.

Definition C.2.8 (Community-Aware Vitality). Extending the Definition C.2.7 we can write the community-aware vitality as:

$$V_f(G, \mathfrak{C}, i) = f(G, \mathfrak{C}) - f(G - \{i\}, \mathfrak{C} - \{i\})$$

Through manipulation of the modularity equation, we show the calculation of modularity vitality for all nodes has time complexity of $O(M + NC)$, providing the scalability of measures like commn and modular, while maintaining the theoretic link to community detection. At the same time, our modularity-derived measure is signed. Negative values indicate nodes are detracting from group structure, and are thus acting like community bridges. Positive valued nodes are then more hub-like. Thus, unlike other measures, modularity vitality shows both how central a node is *and* what way the node is central.

C.2.4 Evaluation: SIR Models and Network Robustness

Evaluation of centrality measures can be subjective, since different measures may be useful for different tasks. However, many of the prior community-aware centrality measures have been evaluated from an immunology perspective [72, 73, 85, 140]. In this scenario, a disease is spreading over a network. The centrality measure in question is used to determine which nodes are given immunity. Then, the “best” centrality measure is that which leads to the smallest outbreak. The fundamental assumption is that central nodes will be spreaders, so immunizing them should result in smaller outbreaks.

Typically, the most basic epidemic model is used: the SIR model [115]. In this model, each node is either susceptible, infected, or recovered. After an initial node is infected, it infects its neighbors with probability p . At the same time, the infected nodes can recover with probability r . Recovered nodes are no longer susceptible, and can no longer spread the disease. The simulation is iterated on until there are no infected nodes remaining. The number of nodes that were ever infected is called the epidemic size. By immunizing nodes, the epidemic size can be decreased. It is the goal, then, to pick an immunization strategy that leads to the smallest epidemic size.

Simulations of this type are closely related to the sub-field of Network Robustness [34, 195]. Network Robustness refers to how a network responds to attacks. Understanding how networks react with missing nodes or edges has important implications in many fields, including but not limited to biology and ecology. Attacks typically take the form of removal of edges or removal of nodes. We will focus on removal of nodes.

One method of evaluating an attack’s effectiveness is through network fragmentation [51]. Fragmentation σ can be defined as the size of the remaining largest component N_ρ relative to the initial size of the graph, N , where ρ is the fraction of nodes removed. Fragmentation can then be given as $\sigma(\rho) = \frac{N_\rho}{N}$. This is a useful measure because networks often rely on connectivity to function properly. Disconnected components in biological, communication, or power-grid networks are in serious danger of failing completely.

Now, we can see that immunization strategies are effectively network attacks. By immunizing a node, it and its links are removed from the network. Immunizing many nodes fragments the network, slowing diffusion. In fact, the fragmentation, σ , is the worst-case scenario for an SIR model. Given the right parameterization, the disease in an SIR model will infect all nodes in the component the disease initialized in. This behavior is guaranteed with $p = 1$, and $r = 0$, indicating full infection with no possibility of recovery. The same effect can be achieved with other parameters depending on how the simulated interactions play out. If the initial node is in the largest component, the worse-case scenario is that all nodes in the largest component get infected. Thus, σ can be used to measure the effectiveness of an immunization strategy without the need for expensive SIR simulations.

Replacing simulated network flow with network connectivity also results in a more fair comparison between network metrics. Centrality measures often make assumptions about how flow occurs in a network, and are thus favorable when simulated flow matches those assumptions, and less favorable when they do not [24]. Thus, a fragmentation approach does not bias the results towards centrality measures which are best aligned with the assumptions of the simulation.

From a network robustness perspective, different types of attacks have been developed. In general, a centrality measure is calculated for each of the nodes, and the node with the highest centrality is removed, or immunized. Early studies looked at node attacks based on degree [34]. Later, Holme generalized this idea along with two styles of attacks: initial and recomputed [103]. In the initial case, centralities are calculated once and the top-k nodes are removed. In the recomputed case, centralities are recomputed each time a node is removed. This makes the attack more expensive to compute, but more effective.

In this framework, attacks are defined by two characteristics, the centrality measure and the style. Common choices of centrality measure are degree and betweenness. Betweenness has been shown to be much more damaging to a network, but is far more expensive to compute [51, 103]. The shorthand for these methods are based on the acronym of the centrality and style; IB means an attack using initial calculation of betweenness centrality, while RD is recomputed degree.

A connection between the modular structure in networks and their robustness has been illustrated by da Cunha et al. in [51]. The authors developed a more complex attack strategy which is able to fragment real-world networks far more quickly than the simple methods previously described. They achieve this by ensuring that nodes are attacked only when they are in the largest component and when they are connecting groups. This strategy is called a Module-Based-Attack, MBA.

Though effective, attacks using betweenness centrality do not scale to the size of networks commonly seen on social media. For weighted networks, a single calculation of betweenness scales as $O(NM + N^2 \log N)$, making RB scale as $O(N^2 M + N^3 \log N)$ [28]. This makes RB intractable for medium-sized networks, which is why da Cunha et al. use IB as the base for their attack method [51]. However, even IB is intractable for very large networks. Additionally, the computation of largest component at every step adds to the method’s complexity. The most scalable methods are those that use an “initial” strategy with a local measure.

Based on this, we use fragmentation to evaluate our method in comparison to the following measures: Masuda (Mas), Community-Hub-Bridge (CHB), Modular-Centrality-Degree (WMC-D), Adjusted-Modular-Centrality-Degree (AMC-D), and Degree (Deg). Evaluation is performed in three steps. In the first, networks are generated to measure how different community-aware centralities perform under varying attack strategies. In this step “initial,” “repeated,” and “module-based” attacks are performed. Second, the Pennsylvania road network is studied. This is a large highly modular network, which exemplifies the power of community-aware centrality measures. Finally, a large Twitter communication network is studied from the Canadian Elections of 2019. Here, the robustness of social media networks is demonstrated. In the second and third steps, only “initial” strategies are taken due to the size of the networks.

C.2.5 Community Deception

Community Deception has recently been formalized by Fionda and Pirro [64]. They argue community detection is a very powerful tool, and could potentially be too powerful for privacy-sensitive applications. In order to protect sensitive data that is easily identifiable,

community structure should be obscured. The goal, then, is to edit a network to prevent a specific community’s detection. The most relevant framing they provided to the present work is Modularity-Based Deception. In this framing, the goal is to re-wire edges such that modularity of a community is minimized. This approach is based on the modularity equation, similarly to the present work, and is scalable. In a similar line of work, Chen et al. propose a genetic algorithm to perform a “Q-Attack,” which edits the network to minimize the modularity of the entire network’s partition, not just that of a single community [42]. Due to the combinatorial nature of genetic algorithms this approach did not scale and was only tested on nodes with approximately 100 nodes.

Waniek et al. also consider the single-community case [227]. In this work, a modularity-inspired measure was used to determine how well a community is concealed. The authors then randomly rewire a specified number of internal edges as external edges. This approach demonstrated that social network users had the power to conceal their community from detection. However, the method is non-deterministic, so its effectiveness varies depending on which edges were selected in each round of simulation. The lack of distinction between the best edges to add or remove also makes it difficult for users to best select actions to conceal their community.

Lastly, Nagaraja takes a different view of the problem wherein an adversary is attempting to uncover the community structure of the entire network with a surveillance strategy [154]. The work proposes several counter-strategies to conceal communities with edge alterations. Nagaraja concludes that these strategies work based on how they impact the network’s modularity, without explicitly maximizing for impact on modularity. The present enables this to be explicitly maximized.

Here, we show that Modularity Vitality can be used to efficiently perform community deception on the entire network rather than a specific community. Rather than rewiring edges, we remove all edges attached to nodes with the highest modularity vitality. This has the benefit of keeping maintaining network accuracy for links that are present, but ultimately does change the degree distribution. In a social media setting, this amounts to hiding which popular accounts a user follows, rather than re-wiring individual following relationships. We demonstrate the power of this approach by performing community deception on a social media communication network with 7.5 million nodes, and 130 million edges.

C.3 Calculating Modularity Vitality

Newman’s community centrality measured a node’s *potential* to contribute to modularity. To calculate the *actual* contribution, we can calculate the modularity vitality: the difference between the modularity of the original partition, and the modularity of the partition after the removal of a specified node. Given that community-aware centralities are commonly evaluated using the effect of node removals, modularity vitality seems to be a natural approach. Note that once a node is removed, the network could be re-grouped, and the group structure could potentially be quite different. Once regrouping is considered, there is no closed-form solution to what the new modularity would be, since the maximization

procedure would need to be re-run. Thus, regrouping is typically not considered, and we do not consider it here [140].

Modularity vitality is defined as:

$$V_Q(G, \mathfrak{C}, i) = Q(G, \mathfrak{C}) - Q(G - \{i\}, \mathfrak{C} - \{i\}). \quad (\text{C.8})$$

A naive computation of this expression is quite expensive. Modularity itself has time complexity $O(M)$. Thus, naively recalculating this in order to calculate the modularity vitality for all nodes has complexity $O(MN)$. However, there is an efficient way of updating modularities after the removal of a node.

The modularity after the removal of node i can instead be calculated using the following expression:

$$\begin{aligned} Q(G - \{i\}, \mathfrak{C} - \{i\}) = \\ \frac{M^{\text{internal}} - k_i^{\text{internal}}}{M - k_i} - \frac{1}{4(M - k_i)^2} \sum_{c \in \mathfrak{C}} (d_c - h_{i,c})^2 \end{aligned} \quad (\text{C.9})$$

$$h_{i,c} = k_i^c + k_i \delta(c, c_i). \quad (\text{C.10})$$

We will now derive this equation.

Theorem C.3.1. *If we remove node i from the graph G then the new modularity of the new graph $G - \{i\}$ can be written as:*

$$\begin{aligned} Q(G - \{i\}, \mathfrak{C} - \{i\}) = \\ \frac{M^{\text{internal}} - k_i^{\text{internal}}}{M - k_i} - \frac{1}{4(M - k_i)^2} \sum_{c \in \mathfrak{C}} (d_c - h_{i,c})^2 \end{aligned} \quad (\text{C.11})$$

$$h_{i,c} = k_i^c + k_i \delta(c, c_i). \quad (\text{C.12})$$

The value $h_{i,c}$ measures the number of edges a node has to that community, and if the node is a member of said community, its degree is added. The degree must be added because d_c double-counts the number of internal links in a community.

Proof. The removal of node i from graph G results in a new graph denoted by $G - \{i\}$. The same applies to the community vector, which is denoted by $\mathfrak{C} - \{i\}$.

First, we re-write Modularity as given in Equation C.1:

$$\begin{aligned}
Q(G, \mathfrak{C}) &= \frac{1}{2M} \sum_{i,j=1}^N \left(A_{i,j} - \frac{1}{2M} k_i k_j \right) \delta(c_i, c_j) \\
&= \frac{1}{2M} \sum_{\gamma \in \mathfrak{C}} \sum_{v_i, v_j \in \gamma} \left(A_{i,j} - \frac{1}{2M} k_i k_j \right) \\
&= \frac{1}{2M} \sum_{\gamma \in \mathfrak{C}} \underbrace{\sum_{v_i, v_j \in \gamma} A_{i,j}}_{2M^{\text{internal}}} - \frac{1}{4M^2} \sum_{\gamma \in \mathfrak{C}} \sum_{v_i, v_j \in \gamma} k_i k_j \\
&= \frac{M^{\text{internal}}}{M} - \frac{1}{4M^2} \sum_{\gamma \in \mathfrak{C}} \sum_{v_i, v_j \in \gamma} k_i k_j \\
&= \frac{M^{\text{internal}}}{M} - \frac{1}{4M^2} \sum_{\gamma \in \mathfrak{C}} \sum_{v_i \in \gamma} k_i \sum_{v_j \in \gamma} k_j
\end{aligned}$$

Let

$$d_c = \sum_{v_i \in \gamma_c} k_i = \sum_{v_j \in \gamma_c} k_j$$

Now can express modularity in terms of number of links and total degrees of nodes:

$$Q(G, \mathfrak{C}) = \frac{M^{\text{internal}}}{M} - \frac{1}{4M^2} \sum_{\gamma_c \in \mathfrak{C}} d_c^2 \quad (\text{C.13})$$

This form is easier to derive the new modularities from.

Equation C.13 can then be applied to graph on graph $G - \{i\}$ to find the new modularity:

$$\begin{aligned}
Q(G - \{i\}, \mathfrak{C} - \{i\}) &= \\
&\frac{M^{\text{internal}} - k_i^{\text{internal}}}{M - k_i} - \frac{1}{4(M - k_i)^2} \sum_{\gamma_c \in \mathfrak{C}} \tilde{d}_{i,c}^2
\end{aligned}$$

Now to calculate $\tilde{d}_{i,c}$ we can break this down in two cases:

Case 1. If $c \neq c_i$ we have :

$$\tilde{d}_{i,c} = \sum_{v_j \in \gamma_c} k_j - k_i^c$$

Case 2. If $c = c_i$ we have :

$$\tilde{d}_{i,c} = \sum_{v_j \in \gamma_c} k_j - k_i^c - k_i$$

Let :

$$h_{i,c} = k_i^c + k_i \delta(c, c_i)$$

then finally we have:

$$\tilde{d}_{i,c} = d_c - h_{i,c}$$

Giving us the final expression for the modularity once node i is removed:

$$Q(G - \{i\}, \mathfrak{C} - \{i\}) = \frac{M^{\text{internal}} - k_i^{\text{internal}}}{M - k_i} - \frac{1}{4(M - k_i)^2} \sum_{c \in \mathfrak{C}} (d_c - h_{i,c})^2$$

■

By looking at Equation C.9, we observe that the only new information needed to update modularity after removing a node is contained in the node's immediate neighborhood and the vector of community degrees. The worst-case scenario would be to calculate updated modularity for the center node of a star-graph, which has degree M . When Equation C.9 is used, the time complexity of calculating the new modularity becomes $O(M + C)$. While this seems to not be an improvement, the worst-case scenario is far worse than the average case, since most node degrees are far less than M . In fact, the calculation of Equation C.9 for *all nodes* in a network has time complexity of only $O(M + NC)$. Given that typically $C \ll N$, this is a major improvement over the naive implementation's complexity of $O(MN)$. This improvement allows for analysis of very large graphs for which $O(MN)$ operations could be prohibitively expensive if not infeasible.

By studying modularity vitality, rather than just the simple new modularity after a node is removed, it is easy to identify which nodes are increasing modularity and which are decreasing it. As Newman noted, "it is entirely possible for individual vertices to simultaneously make both large positive and negative contributions to modularity" [157]. A simplistic approach would be to add the absolute value of the two, but Equation C.8 balances them to see which contribution prevails for each node. Since nodes with positive modularity vitality are contributing positively towards community structure, they can be thought of as hubs within their community. Their removal decreases the strength of their communities, thus decreasing modularity. Conversely, nodes negatively contributing to group structure will have negative modularity vitality. Negative contributions to group structure are facilitated through connections between groups, so nodes with highly negative modularity vitality are community bridges. Removing these community bridges increases modularity. A measure which does not have the issue of large positive and negative contributions balancing out is explored in Section C.5.4, though it does not perform as well as modularity vitality.

Like many previous measures, modularity vitality is correlated with degree. This correlation is intuitive: nodes with many connections have the most potential to impact group structure, either positively or negatively. It can be seen in the new modularity equation:

as node degree increases, the denominator decreases, leading to increase in the magnitude of modularity vitality. However, modularity vitality is more complex, since it takes into account which groups a node is connected to. Nodes connected to larger groups have a bigger impact than those connected to smaller groups. This mirrors Masuda’s measure, where a node’s importance is based on the importance of its group and the group(s) it is connected to. The difference here is that modularity vitality measures a group’s importance with the number of internal links, while Masuda’s uses the eigenvector centrality with the group to group network.

C.4 Methodology

C.4.1 Fragmentation-Based Evaluation

As discussed in Section C.2.4, evaluation based on network fragmentation is similar to the SIR evaluation used in other studies, like [72, 73], however is less expensive computationally and is easier to interpret. Module-based attacks (MBA’s) were tested in such a framework, where they were shown to effectively fragment networks [51]. Again, fragmentation σ is the size of the largest component after the attack, divided by the original largest component. Fragmentation is measured as a function of ρ , the fraction of nodes removed in the attack: $\sigma(\rho) = \frac{N_\rho}{N}$. Similarly, fragmentation can be looked at as a function of the fraction of edges removed, η . Note that here we are only targeting nodes, not edges, but the fraction of remaining edges is still an interesting quantity to study, as we see in Section C.5.3.

An immunization or fragmentation strategy’s effectiveness depends on how many nodes are removed, as seen by the notation $\sigma(\rho)$. To measure the overall effectiveness, the fragmentation function can be integrated. The lower the integral, the more effective the strategy, so we will call this the cost function that we are trying to minimize: $C_\rho = \int_\rho \sigma(\rho) d\rho$. For comparison, the cost with respect to edges can be of interest, though it is not directly being optimized: $C_\eta = \int_\eta \sigma(\eta) d\eta$.

Thus, we will evaluate all of the attack strategies in Section C.4.2, using C . We will do so in three parts: generated networks, the PA road network, and a Twitter network obtained from user to user conversations surrounding the Canadian Election of 2019. Each part highlights different aspects of the proposed method.

C.4.2 Attack Strategies

Attack strategies are the rules that govern which nodes are to be immunized, or removed from the network. Generally these strategies are independent of centrality measure, so can be paired with any measure of a researcher’s choosing. Holme outlined two strategies: initial and repeated [103]. In the initial attack, a centrality measure is calculated for each of the nodes. Then, the top-k nodes are selected to be attacked. The procedure is outlined in Algorithm 2.

Perhaps the biggest issue with the initial attack strategy is its redundancy. After the first node is removed, the centralities of the following nodes change. However, these

changes go un-detected in the initial attack model, leading to the selection of nodes that are no longer in central positions. This, to some extent, can happen due to random effects of node and edge removal in a network. The extent to which random removals impact centrality values and rankings has been previously studied by Borgatti and others, who find that the accuracy of centrality measures drops off smoothly as the number of random changes to the graph increases, though this effect is dependent on the network's topology [26, 66]. Perhaps more importantly, there are non-random effects at play. It is known that certain central nodes are responsible for the centrality of other nodes, and that this can be measured with exogenous centrality [59].

The redundancy issue of the initial attack strategy is resolved in the recomputed attack strategy wherein the centralities are recomputed after each node removal. The full algorithm is shown in Algorithm 3. Though effective, the recompute step adds scalability issues. For a centrality measure that takes $O(M)$ time, the attack takes $O(NM)$ time. This means for expensive calculations like betweenness, the recompute strategy will be intractable, $O(N^3 \log N)$ for weighted networks [28].

A more sophisticated strategy is given by da Cuhna et al, called Module-Based-Attack (MBA) [51]. The authors find that use of group structure leads to effective fragmentation. Group-based structure is incorporated by only attacking nodes which bridge communities. Further, only nodes in the current largest component are attacked. While largest component is recomputed, the centrality measures are not. The full procedure is given in Algorithm 4, where \oplus denotes append operation. While not as complex as the recompute method, the update of the largest component and node bridges makes the method significantly more computationally expensive when compared to the simple initial attack.

Algorithm 2: Initial Attack

Result: List of removed nodes \mathcal{L}

```

1  $\mathcal{L} \leftarrow \emptyset;$ 
2  $k \leftarrow$  the number of nodes to remove;
3  $\mathcal{S} \leftarrow$  List of all nodes sorted by a centrality measure (function);
4 while  $|\mathcal{L}| < k$  do
5    $\tau \leftarrow$  top node in  $\mathcal{S}$ ;
6    $\mathcal{L} \leftarrow \mathcal{L} \cup \tau;$ 
7    $\mathcal{S} \leftarrow \mathcal{S} \setminus \tau;$ 
8 end

```

Thus, for small generated networks we take I, R, and MBA. For the PA-Road Network and Twitter networks, however, only the “initial” attack strategy is computed, as it is the most scalable. These are combined with degree as well as the previously discussed local community-aware centrality measures: Masuda (Mas), Community-Hub-Bridge (CHB), Modular-Centrality-Degree (WMC-D), Adjusted-Modular-Centrality-Degree (AMC-D), and Degree (Deg). We compare these existing approaches to modularity vitality in two forms. First, we take the original modularity vitality (MV), attacking from negative to positive, in order to target community-bridges. Second, we consider the absolute value of the mod-

Algorithm 3: Repeated or Recomputed Attack

Result: List of removed nodes \mathcal{L}

```
1  $\mathcal{L} \leftarrow \emptyset;$ 
2  $k \leftarrow$  the number of nodes to remove;
3  $G \leftarrow$  the initial graph;
4 while  $|\mathcal{L}| < k$  do
5    $S \leftarrow$  List of all nodes in  $G$  sorted by a centrality measure
      (function);
6    $\tau \leftarrow$  top node in  $S$ ;
7    $\mathcal{L} \leftarrow \mathcal{L} \cup \tau;$ 
8    $G \leftarrow G \setminus \tau;$ 
9 end
```

ularity vitality (AMV), which targets nodes based on their overall contribution to group structure, positive or negative. A third form was considered, where nodes were attacked from positive to negative modularity vitality. This hub-first strategy did performed poorly, and is omitted from result tables to preserve readability.

C.5 Network Fragmentation

C.5.1 Generated Networks

First, we compared community-aware centralities using generated networks. By using generated networks we can repeat tests many times. We constructed modular networks using the cellular network model, similar to that of Masuda [140]. In this model, “cells” are random sized Erdős-Rényi networks with high density, simulating clusters. Then, the cell-to-cell network is also modeled as an Erdős-Rényi network. When two cells are linked in the group-to-group network, random nodes from each are selected and a link is drawn between them. For this study, cellular networks were created using the parameters shown in Table C.1. This results in an unweighted, undirected random network with community structure.

The eight previously discussed community-aware centrality measures were paired with the three possible attack schemes, initial, recomputed, and MBA, to give 24 attacks. Each time a network was generated all 24 attacks were performed on the network and the corresponding cost functions C_ρ and C_η were recorded. The average cost of the 24 attacks for 100 generated networks is given in Table C.2. The average modularity for these 100 networks when grouped with Leiden grouping was 0.91. The modularity vitality attack consistently outperforms all other attacks both in terms of node cost and edge cost, suggesting that it is the best community-aware centrality measure for this type of synthetic network. The fact that attacking nodes with negative modularity vitality is more effective than nodes that are high in modularity vitality magnitude suggests that community bridge nodes are more important than community hub nodes in cellular networks. The success of

Algorithm 4: Module-Based Attack (MBA)

Result: List of removed nodes \mathcal{L}

```

1  $\mathcal{L} \leftarrow \emptyset;$ 
2  $G \leftarrow$  the initial graph;
3  $B \leftarrow$  the set of nodes bridging communities in  $G$ ;
4  $\mathcal{S} \leftarrow$  List of all nodes in  $G$  sorted by a centrality measure
   (function);
5  $LC \leftarrow$  the set of nodes in the largest component of  $G$ ;
6 while  $|B \cap LC| > 0$  do
7    $\tau \leftarrow$  top node in  $S$ ;
8   if  $\tau \in B$  and  $\tau \in LC$  then
9      $G \leftarrow G \setminus \tau;$ 
10     $LC \leftarrow$  the set of nodes in the largest component of  $G$ ;
11     $B \leftarrow$  the set of nodes bridging communities in  $G$ ;
12     $\mathcal{L} \leftarrow \mathcal{L} \cup \tau;$ 
13     $\mathcal{S} \leftarrow \mathcal{S} \setminus \tau;$ 
14   else
15     if  $\tau \notin B$  then
16        $\mathcal{S} \leftarrow \mathcal{S} \setminus \tau;$ 
17     else
18        $\mathcal{S} \leftarrow \mathcal{S} \oplus \tau$ 
19     end
20   end
21 end

```

Table C.1: Cellular Network Parameters. $\mathcal{U}(a, b)$ denotes the uniform random distribution between numbers a and b ; $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2 .

Variable	Description
$N = 1000$	Number of nodes
$N_c = \mathcal{U}(10, 20)$	Number of cells
$n_i = \mathcal{N}(N/N_c, N_c/5),$	Number of nodes per cell
$p_i = \mathcal{U}(0.1, 0.25)$	Density of internal cell relationships
$p_o = \mathcal{U}(0, 0.5)$	Density of the cell-to-cell network

the “adjusted” modular-degree centrality provides further evidence of this, since it places

Table C.2: Results for attacks on the generated cellular networks. The values shown are the average over 100 simulations. Methods introduced in this work are on the left of the double column. The best results are emboldened.

Method	MV	AMV	AMC-D	Mas	CHB	WMC-D	Deg
Initial C_ρ	0.165	0.211	0.169	0.250	0.383	0.381	0.347
Initial C_η	0.247	0.308	0.268	0.371	0.576	0.599	0.578
MBA C_ρ	0.086	0.087	0.088	0.134	0.101	0.103	0.100
MBA C_η	0.157	0.162	0.173	0.235	0.211	0.219	0.216
Recomputed C_ρ	0.107	0.126	0.130	0.162	0.331	0.337	0.309
Recomputed C_η	0.188	0.205	0.221	0.266	0.608	0.616	0.586

greater importance on community bridges, while the original modular-degree focuses on hubs and does not score as well.

C.5.2 PA-Road Network

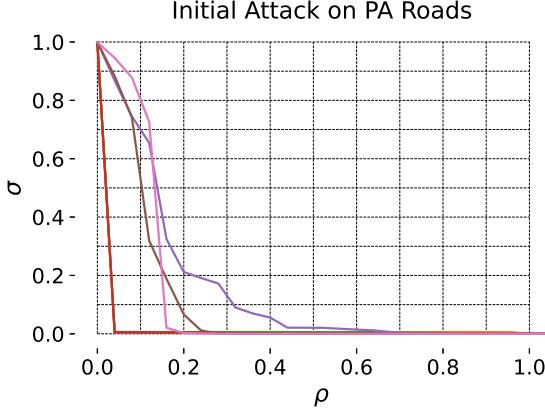
One particularly well-suited application for community-aware centrality measures is the analysis of large infrastructure networks. These networks typically have two properties: very high modularity and low maximum degree. High modularity makes group-based approaches appropriate. Low maximum degree often means that simple degree-based attacks will be ineffective. Additionally, their large size make effective approaches like MBA intractable, or at least very costly. Instead, we show that initial-attacks with community-aware centrality measures are very effective, and that our modularity-based methods are the most effective by far.

As an example, we use the Pennsylvania Road Network [127]. Roads are represented by edges, while intersections are represented by nodes. This network has 1,088,092 nodes, and 1,541,898 edges. When grouped with Leiden grouping maximizing modularity, 499 clusters are obtained with a modularity of 0.990. Its maximum degree is 18. The extremely high modularity and low maximum degree make it an ideal candidate for community-aware centrality measures.

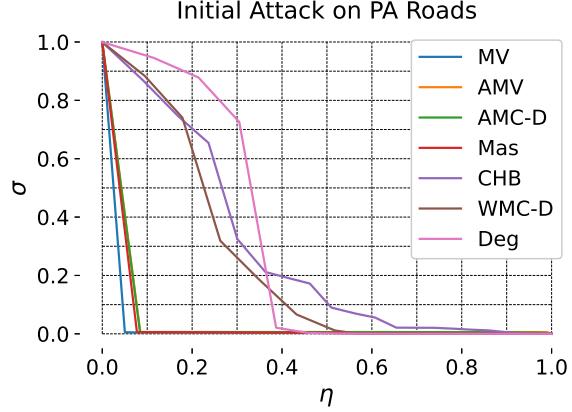
In Figure C.1, we see the fragmentation as a function of nodes and edges removed for each strategy. Here, we see the largest component can be effectively brought to zero by removing 1.6% of nodes with the highest modularity vitality values. Removing only positive-valued nodes and removing nodes based on the absolute value of their modularity vitality value give very similar results. The quantitative results, as measured by C_ρ and C_η are given in Table C.3.

Additionally, we show the plot as a function of edges removed, for the same strategies. The edge plot shows that while modularity vitality fragments the networks best given a number of nodes, it is also most efficient in terms of edges.

With just a degree-based attack, it would appear that the Pennsylvania road network is robust. In fact, the community-aware centrality methods show that it is quite fragile. Using an I-MV attack, the network can be almost completely fragmented by targeting only 1.6% percent of nodes, bringing the largest component down to less than 1% of its original



(a) Fragmentation by nodes removed.



(b) Fragmentation by edges removed.

Figure C.1: Fragmentation of the PA-Road Network. Results for modularity-vitality, absolute modularity-vitality, adjusted modular-centrality, and Masuda are extremely similar, so overlap on both figures. The legend in (b) also applies to the plot in (a), as well as both plots in Figure C.2.

size. This improves over the previous best measure, modular-degree, by a factor of over 8.5.

C.5.3 Canadian Election Twitter Network

Another relevant application of community-aware centrality is social media networks. Since social media has become so embedded in everyday life, scalable tools to understand it are essential. Given the increasing polarization of online discussion, as described in concepts like filter bubbles, it is not enough to know what actors are important in general. Instead, it is necessary to understand what actors are important within and between key online communities. Community-aware centralities make this a measurable problem.

To study the effectiveness of our community-aware centrality measures we again use network fragmentation, due to its connection with diffusion. Diffusion on social media is an important phenomena to understand as a way to combat misinformation, among other things. Users who fragment the network when removed are those who have the most power to spread information.

For this study, we use the network created from Twitter data collected during 2019 Canadian federal election. The goal was to obtain a user-to-user communication network where users were active in political discussion. First, we used a keyword search of Twitter's API to collect tweets related to the Canadian Election during the month of October. From here, the unique users were recorded, giving a list of users active in political discussion. While a user to user network could be constructed with this data, many links would be missing, since only tweets with our keywords can be used. To construct a more complete network, Twitter's API was used to scrape the timelines of all users in our list. This new collection was then truncated to the week of the election. Finally, the all-communication

graph was computed from this dataset, where link weights are the sum of the mentions, retweets, and quotes. The “Election Week” network, has 7,523,125 nodes, and 130,086,491 links. When grouped with Leiden grouping, 557 communities were discovered, with a modularity of 0.691.

Figure C.2 shows the fragmentation results on the election week network. Again, the quantitative results are given in Table C.3. The Adjusted-Modular-Degree measure and the classical degree measure effectively tie for node-based efficiency.

The structure and properties between the PA Roads network and the Election Week network are very different. This difference is reflected in Figure C.2. Perhaps most striking is how poorly the modularity vitality method performs in terms of ρ . While other methods fragment the network removing 10-30% of nodes, the positive modularity vitality method does not fragment the network until nearly all nodes are removed.

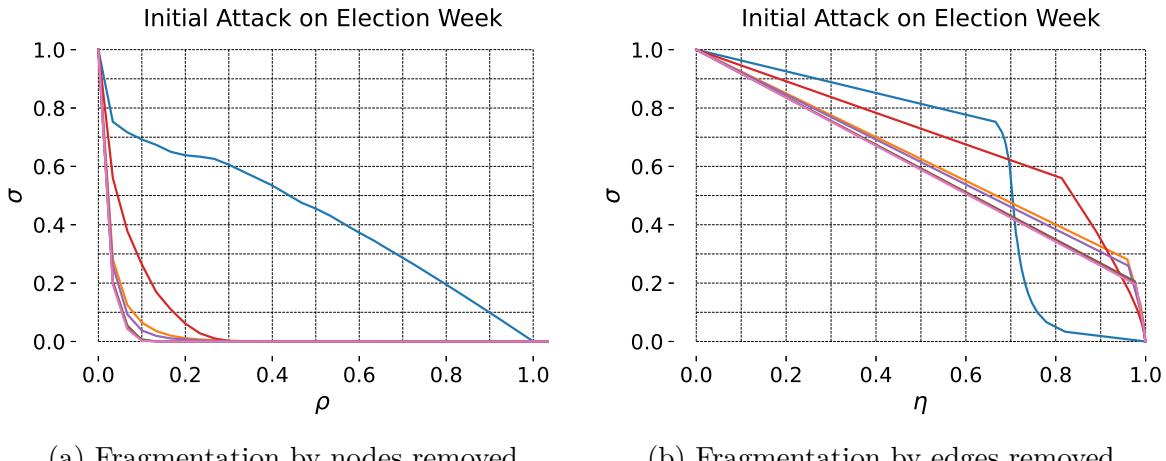


Figure C.2: Fragmentation of the Election-Week Network. The legend can be found in Figure C.1(b).

At first this seems like a failure of the modularity vitality method. However, inspection of Figure C.2 (b), shows otherwise. In terms of links, the modularity vitality method is actually the most efficient attack strategy. This counter-intuitive result occurs because *none* of the methods are very effective at fragmenting the network. The largest component is small when removing 10-30% of nodes using the other methods, but those nodes account for over 95% of the networks links. The difference in bridge-first Modularity-Vitality attacks and all others, however, does highlight the fact that networks with extremely high-degree nodes will require mixed or hub-first approaches to be efficiently fragmented. Even accounting for this aspect of the network, the election week network exhibits extreme robustness to these types of attacks.

Table C.3: Results for initial attacks on the PA-Road Network and the Canadian-Election Twitter Network. Methods introduced in this work are on the left of the double column. The best results are emboldened.

Network	MV	AMV	AMC-D	Mas	CHB	WMC-D	Deg
PA-Roads C_ρ	0.013	0.016	0.015	0.014	0.162	0.120	0.122
PA-Roads C_η	0.021	0.026	0.026	0.027	0.281	0.262	0.305
Election C_ρ	0.430	0.032	0.022	0.070	0.029	0.023	0.022
Election C_η	0.635	0.673	0.656	0.694	0.667	0.654	0.651

C.5.4 Additional Experiments

Community-Degree

Though the signed aspect of modularity vitality is quite useful, it is possible that a node has high positive and negative components of modularity in Equation C.9, resulting in a modularity vitality near zero. These nodes may be particularly important for networks with low modularity. We can adjust Equation C.9 to obtain a measure which credits nodes for hub *and* bridge behavior. By changing the subtraction of $h_{i,c}$ to addition, this effect is achieved. After this adjustment, there is no need for a separate internal term, making the final measure:

$$CD_i = \frac{1}{4(M - k_i)^2} \sum_{c \in \mathcal{C}} (d_c + h_{i,c})^2 \quad (\text{C.14})$$

Again, attachment to large groups is favored over attachment to small groups. Since this is just weighting the degree, we will call it Community-Degree (CD). The previous results including this measure are shown in Tables C.4-C.7, and in Figures C.3 and C.4.

Community-Degree is highly correlated with degree, and so performs similarly. Based on these results, it seems that the signed centrality is more effective while also conveying more information.

Results on Other Generated Networks

For completeness, networks lacking strong group structure were generated. Scale-free networks were generated using the Barabási-Albert model using parameters $n = 1000$, $m = 8$, and $\gamma = 1.5$. Over the 100 iterations tested the average modularity from Leiden grouping was 0.196. The results are given in Table C.6.

Erdős-Rényi networks with parameters $n = 1000$, $p = 0.015$, were also created. These parameters were chosen to give similar density to the cellular networks previously studied. Networks were generated until a connected network was reached. Over the 100 connected networks, the average modularity from Leiden grouping was 0.240. The results are given in Table C.7.

The results across network types are similar. First, none of the attacks are very effective. Both the node and edge cost are higher than that seen for the Election network, which was robust. With that said, the degree and modular-degree attacks were consistently the most efficient in terms of nodes. This is intuitive; without more meaningful structure,

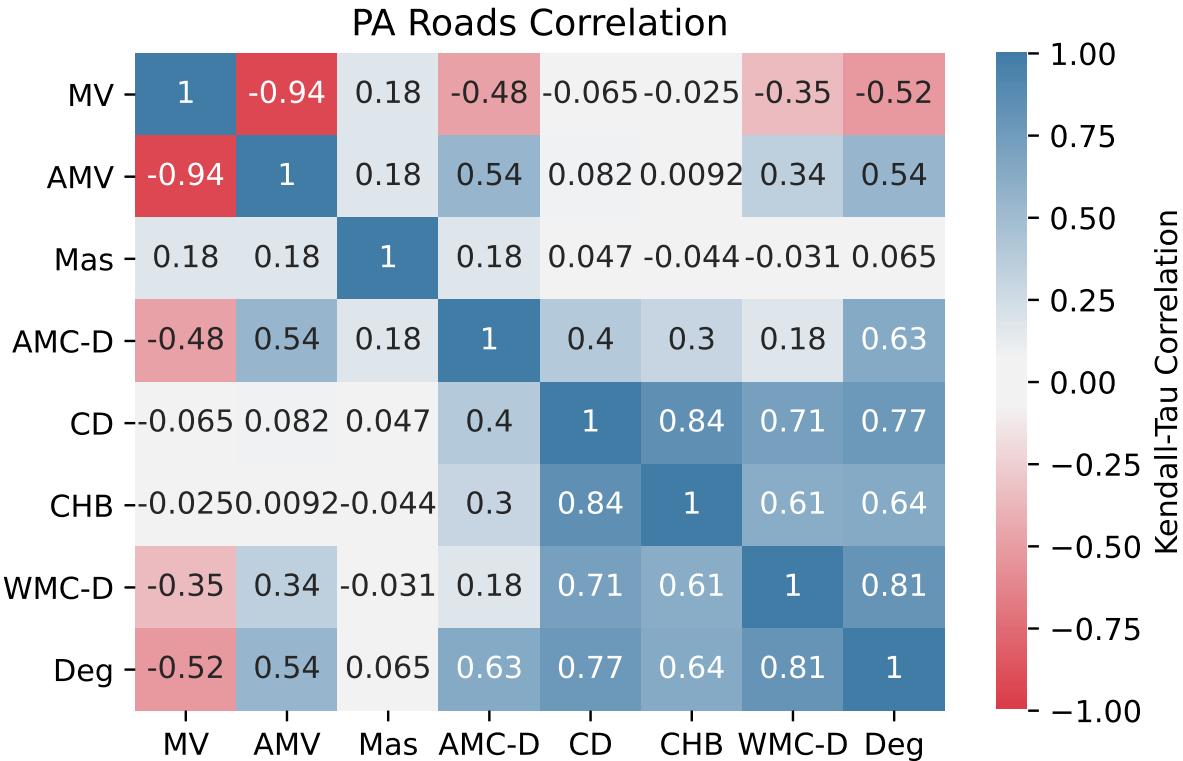


Figure C.3: Extended version of Figure C.5 to include Community-Degree. Kendall-Tau Correlation of the “initial” attack strategies on the PA Roads Network.

the most effective strategy is to look at the node with the most edges. This results in a high edge-based cost, however. So we see that modularity vitality actually performs best in terms of edge-cost. Lastly, we see that the adjusted-modular degree that we proposed performs similarly to the original. Adjusted measure performs much better on highly modular networks, while performing similarly on less modular networks.

Table C.4: Extended version of Table C.2 to include Community-Degree. Results for attacks on the generated cellular networks. The values shown are the average over 100 simulations. Methods introduced in this work are on the left of the double column. The best results are emboldened.

Method	MV	AMV	CD	AMC-D	Mas	CHB	WMC-D	Deg
Initial C_ρ	0.165	0.211	0.361	0.169	0.250	0.383	0.381	0.347
Initial C_η	0.247	0.308	0.560	0.268	0.371	0.576	0.599	0.578
MBA C_ρ	0.086	0.087	0.099	0.088	0.134	0.101	0.103	0.100
MBA C_η	0.157	0.162	0.210	0.173	0.235	0.211	0.219	0.216
Recomputed C_ρ	0.107	0.126	0.320	0.130	0.162	0.331	0.337	0.309
Recomputed C_η	0.188	0.205	0.599	0.221	0.266	0.608	0.616	0.586

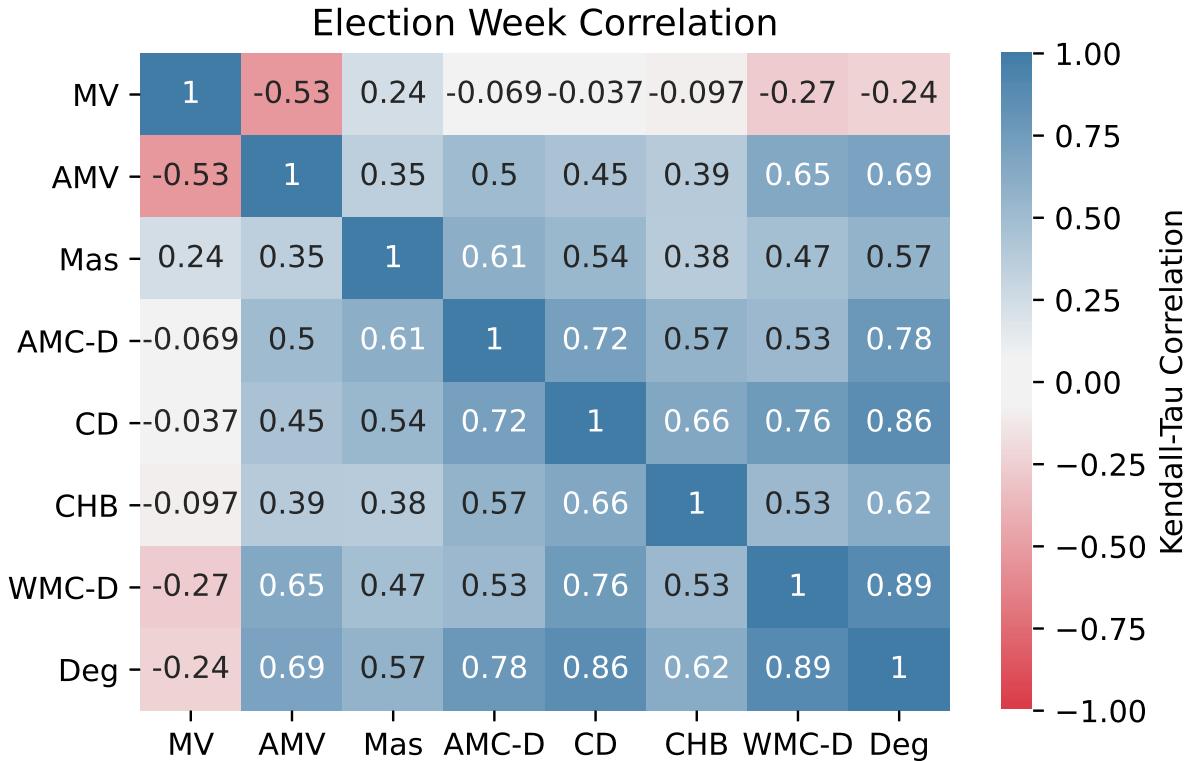


Figure C.4: Extended version of Figure C.6 to include Community-Degree. Kendall-Tau Correlation of the “initial” attack strategies on the PA Roads Network.

Table C.5: Extended version of Table C.3 to include Community-Degree. Results for initial attacks on the PA-Road Network and the Canadian-Election Twitter Network. Methods introduced in this work are on the left of the double column. The best results are emboldened.

Network	MV	AMV	CD	AMC-D	Mas	CHB	WMC-D	Deg
PA-Roads C_ρ	0.013	0.016	0.126	0.015	0.014	0.162	0.120	0.122
PA-Roads C_η	0.021	0.026	0.264	0.026	0.027	0.281	0.262	0.305
Election C_ρ	0.430	0.032	0.023	0.022	0.070	0.029	0.023	0.022
Election C_η	0.636	0.673	0.661	0.656	0.694	0.667	0.654	0.651

C.5.5 Discussion

We see that modularity-based methods were very effective in all three studies. The modularity vitality method shows that the PA Road network is over 8.5 times as fragile as could be seen with degree, weighted modular centrality, and community-hub-bridge centrality, giving similar results to Masuda and the adjusted modular-centrality.. While the standard modularity vitality attack was effective on the PA-Road network, it was not on the Election week network. However, using the absolute-value of modularity vitality resolves the issue. This implies that attacking community-bridges is not enough. By taking the absolute value, both community-bridges and community-hubs are attacked, leading to a

Table C.6: Results for attacks on the generated scale free networks. The values shown are the average over 100 simulations. Methods introduced in this work are on the left of the double column. The best results by method are emboldened. The best results overall are marked with a star.

Method	MV	AMV	CD	AMC-D	Mas	CHB	MC-D	Deg
Initial C_ρ	0.483	0.424	0.263	0.256	0.344	0.361	0.254	0.243
Initial C_η	0.834*	0.856	0.882	0.881	0.884	0.879	0.881	0.880
MBA C_ρ	0.430	0.364	0.243	0.239	0.277	0.292	0.242	0.235
MBA C_η	0.839	0.859	0.880	0.880	0.881	0.877	0.880	0.880
Recomputed C_ρ	0.296	0.305	0.224	0.227	0.256	0.258	0.223*	0.223*
Recomputed C_η	0.878	0.878	0.880	0.880	0.880	0.881	0.880	0.880

Table C.7: Results for attacks on the generated Erdős-Rényi networks. The values shown are the average over 100 simulations. Methods introduced in this work are on the left of the double column. The best results by method are emboldened. The best results overall are marked with a star.

Method	MV	AMV	CD	AMC-D	Mas	CHB	WMC-D	Deg
Initial C_ρ	0.493	0.491	0.479	0.475	0.486	0.492	0.473	0.472
Initial C_η	0.683	0.681	0.715	0.723	0.706	0.675*	0.724	0.728
MBA C_ρ	0.483	0.484	0.469	0.466	0.474	0.485	0.464	0.462
MBA C_η	0.683	0.681	0.714	0.722	0.706	0.675*	0.724	0.727
Recomputed C_ρ	0.461	0.482	0.429*	0.454	0.451	0.446	0.430	0.430
Recomputed C_η	0.700	0.681	0.739	0.739	0.729	0.718	0.738	0.740

method that is more robust across networks, even if it might not be the top-performer for specific networks.

As much as the values of a centrality are important, often the ranking of node centralities takes precedence. This is the case with network attacks studied in this work. So to go beyond the fragmentation results, the Kendall correlation of each method was calculated to compare the resulting node-rankings [115]. Figures C.5 and C.6 show the correlations for the Road network and the Election network, respectively. These correlations allow us to see the similarity of centrality ranking, regardless of the effectiveness of said ranking. Though more clearly in Figure C.5, we see that the existing degree-based metrics are highly correlated. This is intuitive, as they are all alterations on a weighted degree. While connecting certain groups might give a node a higher or lower score depending on the metric, a low degree usually leads to a low score.

Absolute modularity vitality has moderate correlation to the existing methods. Most notably, it has strongest connections to the modular-degree centrality. However, the standard modularity vitality has lower correlation. The combination of these observations show that modularity vitality is leveraging similar information to modular-centrality applied to degree, while giving those values a sign indicating the type of central role they are playing:

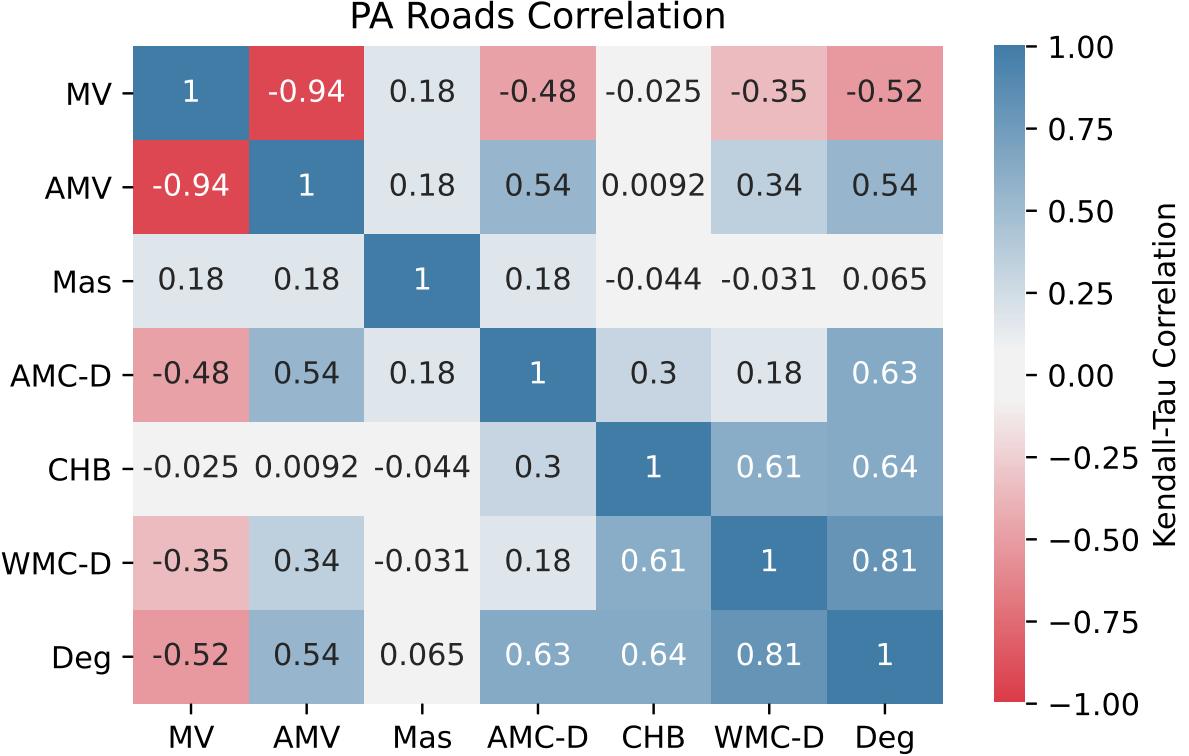


Figure C.5: Kendall-Tau Correlation of the “initial” attack strategies on the PA Roads Network.

hub or bridge. The correlation between modularity vitality and its absolute value give further information about a network’s structure. In the road network, the strong negative correlation (-0.94) indicates that most nodes are community hubs, not bridges. The same is seen with the election week network though to a lesser extent since the correlation is -0.53. This result is consistent with the networks’ high modularities, and that the road network’s modularity is much higher than election week’s. This added information is a key contribution of the work, and will be of use for deep dives into network data.

Lastly, we see that our adjusted version of the modular-degree centrality gives improvements over the original modular-centrality, and that it has stronger correlations to the modularity-based methods. Based on these results, it is possible that the generalized modular-centrality should also be adjusted to favor bridges. In general, it seems that bridge-favoring methods have performed best in our experiments. This is intuitive from a diffusion perspective. If a network is highly modular, the groups themselves can act as silos to contain what is being diffused if the community-bridge nodes are removed. For the road network, modular-centrality points to areas that need extra redundancy to create a more robust transportation network.

From the social network, we see that targeting bridges is not always enough. In the presence of community bridges and large community hubs, an approach that attacks both is necessary. The absolute modularity vitality method attacks both, but the election network

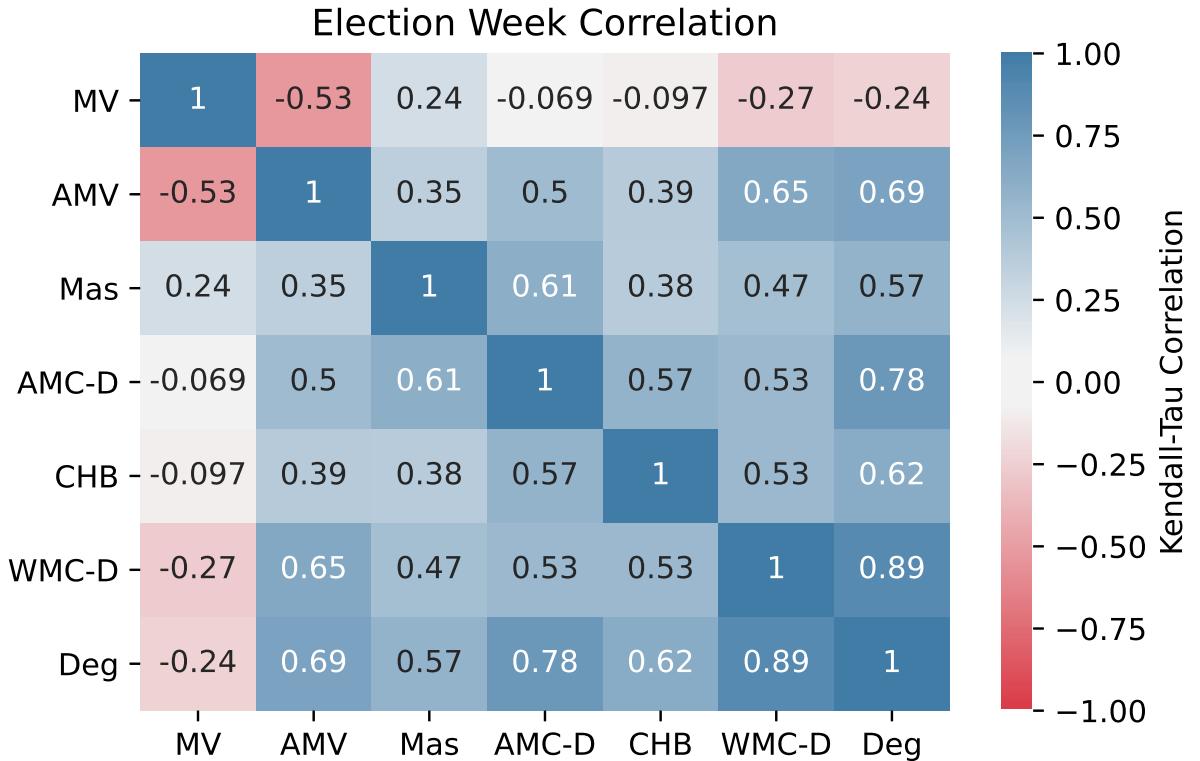


Figure C.6: Kendall-Tau Correlation of the “initial” attack strategies on the Election Week Network.

was robust to even this attack.

In the context of misinformation on social media, both users acting as hubs within fringe communities and users attempting to bridge communities play key roles. Further, network robustness is both a strength and a weakness in this context. A robust communication network means many users have the power to spread information. This allows for distributed power of information but also means that user-based interventions to hamper the spread of misinformation will be ineffective. It is commonly stated that network metrics may identify key points where misinformation diffusion can be stopped [196]. However, we find that not to be the case. The networks are too robust to have a number of points that control diffusion. This may explain why disinformation tends to repeatedly resurface [196]. While identifying key users spreading misinformation is useful for characterizing efforts to share fake news, we must look beyond user-based interventions to actually fight its spread.

C.6 Community Deception

The goal of community deception is to hide a community from detection algorithms [42, 64]. The motivation behind this is typically to protect privacy. Sensitive user data is often over-mined, and network community information is one of the ways in which identifiable

information can be discovered. The idea, then, is to alter the network such that community information is harmed, as measured through modularity of the original grouping on the altered network.

Previously, modularity vitality attacks were used to maximize fragmentation. However, fragmentation is only a by-product of the modularity vitality attack. The attack's true objective is to maximize modularity. As shown in Figure C.7, the same attack used to fragment the Election Week network increases its modularity. In fact, all of the fragmentation methods increase modularity. By attacking nodes which bridge communities, the communities become more separated and modularity increases. The figure shows that the different attacks give similar change in modularity, though the vitality approach is most efficient, since it explicitly increases modularity.

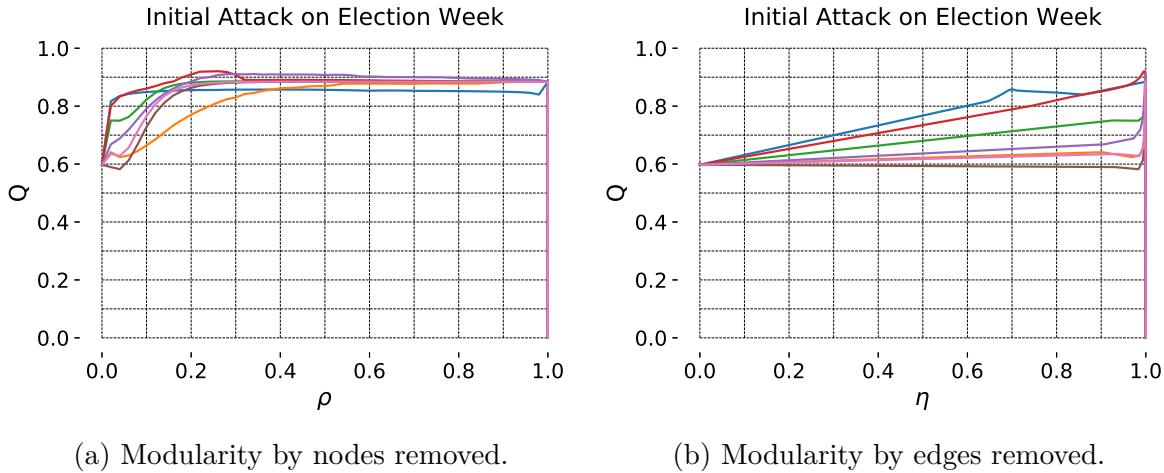


Figure C.7: Changes in modularity due to the fragmentation attacks, all using the initial strategy.

For community deception, the power of the modularity vitality method is the ability to select community hubs instead of bridges. Since community deception seeks to *minimize* modularity, the attack can simply be reversed by selecting the node with the highest *positive* modularity vitality. Thus, a greedy solution to the node-based community deception problem is a recomputed, reversed, modularity vitality attack. A faster approximation to this is the initial, reversed, modularity vitality attack.

Previous methods considered edge rewirings. In practice, this may be difficult or problematic, since links in the altered network may or may not truly exist. An alternate approach is to remove a small subset of the nodes. While the altered network will have less links than the original, all links in the altered network are links in the original network. By leveraging the modularity equation itself, we can select the nodes guaranteed to minimize modularity in a scalable way. As a demonstration of this, community-deception was performed on the Canadian Election network, and the results are given in Figure C.8, for both the fast approximation of the greedy approach. For networks of this scale, even the greedy approach is very expensive. Using the initial attack strategy, modularity can be dropped from approximately 0.7 to just over 0.4 by removing less than 2% of nodes, as

shown in Figure C.8 (a). However, Figure C.8 (b) shows that this comes at a cost of 45% of the nodes edges. Modularity can be decreased further, though with diminishing returns. Modularity levels out when about 8% of nodes and 50% of edges are removed, resulting in a final modularity of 0.36, which is a 49% decrease.

We know from the modularity vitality equation that this strategy is attacking hubs, and this is seen by the fact that the first 2% of nodes targeted are accounting for 45% of links. Intuitively, this suggests that a user's connections to Twitter accounts that are popular within a community reveal that user's identity as a community member. If this identity is to be protected, then hiding these key hubs, as measured through modularity vitality, is the most effective strategy.

This presents a dilemma to social media users wishing to conceal their online community: the most effective strategy is to un-friend or un-follow the community's leaders, which would undoubtedly harm the community itself. The extent of this harm is dependent on the platform. On Twitter, for example, users may interact without a following relationship. On other platforms, like Facebook, the extent of these interactions is more limited. This leaves it up to the social media companies to protect their users by allowing them to hide their affiliations to other accounts, or at least to community leaders.

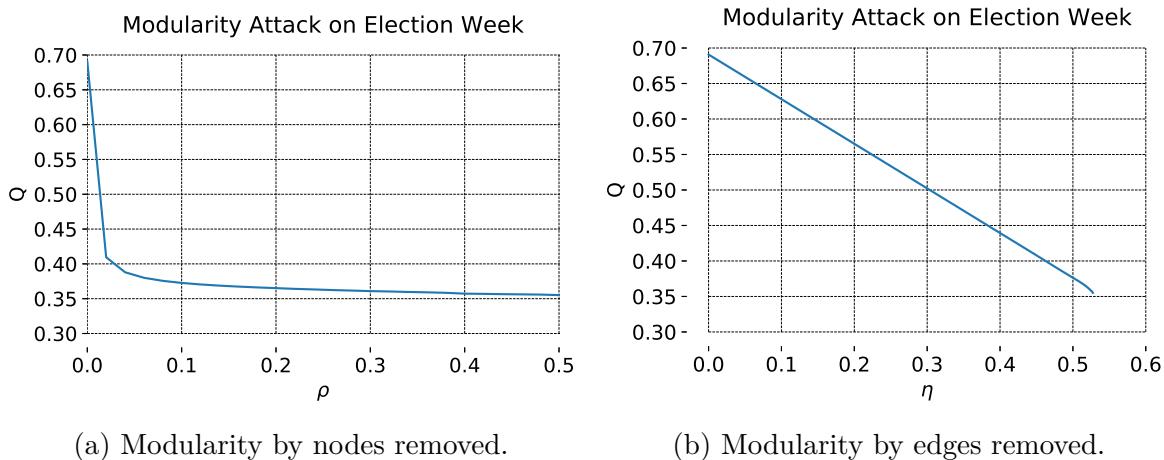


Figure C.8: Community deception on the Election-Week Network using the initial attack strategy.

A choice must be made when performing community-deception: At what point is does the cost of deleting network data outweigh the benefit of decreased modularity? For this case, if only nodes are of interest, there is only a very small price to pay to decrease modularity by 41%. If edges are important to consider, the cost is higher. This is only an approximation of the greedy approach. The greedy approach, recomputed reversed modularity vitality attack, will likely achieve even better results. A study of this comparison on smaller networks along with non-greedy alternatives is left for future work. Additionally, this type of attack could be combined with the previously studied edge re-wiring attacks to give even obscure communities even more effectively. Lastly, explicit modularity *maximization* through node removal could have interesting applications, such as node filtering

to obtain more interpretable groups. This, too, is left for future work.

C.7 Conclusion

Both centrality measures and community detection are core research areas in Network Science. At the intersection of these areas, community-aware centrality measures attempt to quantify how central nodes are given a network partition. Though the areas are closely related, the current community-aware centralities are not strongly tied to community theory. Here, we examine modularity vitality, which measures the change in modularity of a network and its partition if a node were to be removed. Thus, modularity vitality measures each node’s individual contribution to group structure. This measure is directly derived from the modularity equation, giving the measure a strong link to community detection theory. We derive a scalable method of calculating modularity vitality, which improves over the naive method usually by a factor of N , allowing for the analysis of massive networks.

Unlike existing measures, however, modularity vitality not only quantifies how important a node is, but in which way it is important. Once groups are introduced, nodes can take on two central roles: hubs within their community, and bridges between communities. The role is encoded in the sign of modularity vitality; nodes with negative values are bridges, while positive-valued nodes are hubs.

Modularity vitality was tested in three settings: generated cellular networks, the Pennsylvania Road Network, and a Twitter network capturing conversation around the Canadian Election of 2019. In these tests, we saw that modularity-based methods outperformed existing community-aware centralities as measured through network fragmentation. Our results show that the Pennsylvania Road network is over 8.5 times more fragile than measure only weakly tied to community detection theory would have concluded, and that community bridges play a more important role than community-hubs.

Further, we saw that the social media conversation network is very robust, and that both community-hubs and community-bridges play important roles in that robustness. Additionally, the presence of extremely high-degree nodes lead to bridge-first methods performing worst, since high-degree nodes are typically well-grouped. Robust communication is aligned with Social Media’s business interests, since they give many users the potential to “go-viral,” encouraging engagement. The specific source of this robustness remains an area of future research, though the balance of nodes with positive and negative modularity vitality nodes suggests that the presence of many community bridges may be a factor. This theory is in agreement with the results on the PA network, which has very few bridges and is extremely fragile. A robust communication network suggests that user-based interventions are not an effective strategy to fight the spread of misinformation, since an extreme intervention like user-removal only has a small impact on potential diffusion.

Many prior community-aware centralities give preference to community-bridges over community-hubs. Using modularity-vitality without taking the absolute value also targets bridges instead of hubs. Based on this, we include a modified version of Ghalmale’s generalized community-aware centrality measure where bridges are favored instead of hubs. This alternate version of their community-aware centrality when applied with degree performed

better in our experiments. Further studies could explore if this change is an improvement when combined with classical centrality measures other than degree.

Lastly, we recognize that modularity vitality can be used as a greedy solution to the community-deception problem. Community-deception seeks to remove nodes or edges to maximally reduce modularity, which could be important for privacy protection in data distribution. While previous work uses a genetic algorithm to select nodes or edges which may reduce modularity, modularity vitality can be used to select the node that will maximally decrease modularity. Recomputing modularity vitality at each removal provides a greedy solution to the community-deception problem, but we use the faster approximation: only calculating modularity vitality once. While the genetic algorithm could scale to networks with two hundred nodes, the approximation of the greedy method scales to networks with millions of nodes and hundreds of millions of links, as demonstrated on the election week network. Through this demonstration we see that modularity can be decreased by 41% while only removing less than 2% of nodes, but this comes at a cost of 45% of the edges. Still, community-deception is a combinatorial optimization problem, so there are almost definitely better solutions. Going forward, the greedy approach using modularity vitality may be a useful baseline.

The findings suggest that the most effective strategy currently available to users attempting to protect their community identity is to remove their connections to community leaders. This strategy clearly will negatively impact the community itself, leaving users with little options to protect their privacy. It is up to social media companies to protect this privacy by allowing users to hide their connections.

We have demonstrated that modularity vitality is a powerful method of finding nodes that bridge communities or are hubs within their communities at scale. Modularity is but one of many cluster evaluation functions. Exploration of vitalities of these other functions could give an alternative view of nodal contributions to community structure. Community quality vitalities, and community-aware centralities more generally have many applications to areas such as infrastructure robustness, traffic improvement, immunization, and social media. Deeper dives into these application areas using the techniques proposed here could be fruitful areas of future research.

Appendix D

Additional Prototype Results

D.1 Extended Results Diagrams and Tables

Here we display the tables and Figures that could not be fit in the main article body.

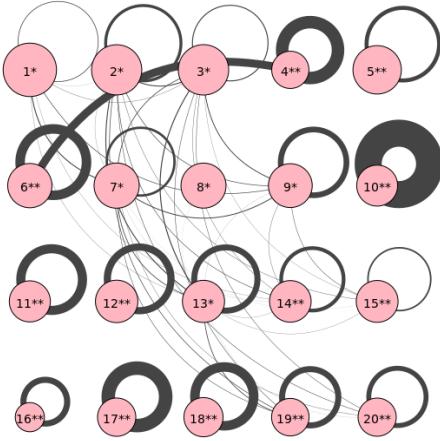
D.1.1 Community Diagram on Unfiltered Data

D.1.2 Salient Attributes

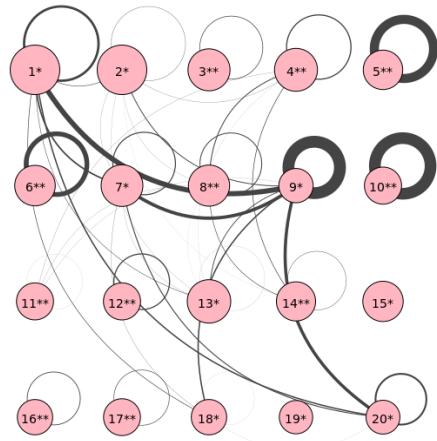
Tables of the most and least salient *biography* attributes for the *Reopen*, *COVID*, and *Captain Marvel* datasets are given in Tables D.1, D.3, and D.5, respectively. Accordingly, the tables for *non-biography* attributes are given in Tables D.2, D.4, and D.6.

Personal ID		Mention		Hashtag		Emoji	
S	NS	S	NS	S	NS	S	NS
she	writer	@genflynn	@manutd	#maga	#blacklivesmatter	🇺🇸	❤️
her	he	@actorvijay	@nytimes	#kag	#blm	✨	💙
maga	him	@realdonaldtrump	@lfc	#wwg1wga	#resist	🌊	💖
they	husband	@potus	@arsenal	#trump2020	#resistance	🏳️	💕
black lives matter	wife	@salesforce	@chelseafc	#followbackhongkong	#fbr	+	🌈

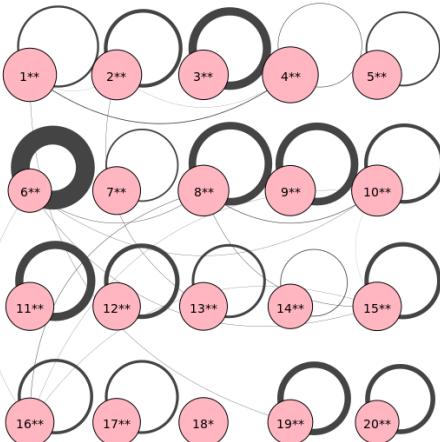
Table D.1: The most salient (S) and least salient (NS) attributes of each attribute derived from user biographies within the *Reopen* Dataset



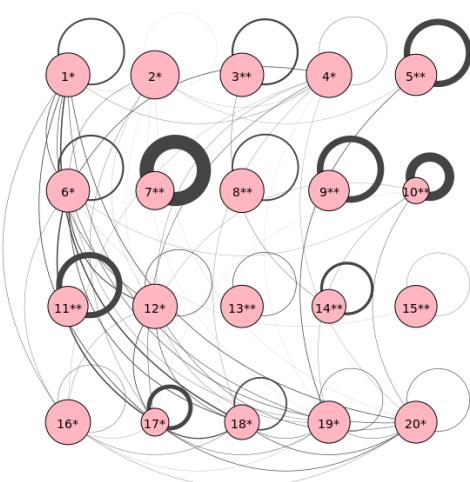
(a) Reopen



(b) Election



(c) COVID



(d) Captain Marvel

Figure D.1: The community-to-community shared-attribute relationships are shown just as in Figure 4.4, using the *unfiltered* data.

Name Hashtag		Location Unigram	
S	NS	S	NS
#blm	#blacklivesmatter	england	usa
#fbpe	#acab	india	new
#maga	#stayhome	london	united
#wwg1wga	#bim	uk	ca
#kag	#junkterrorbill	africa	the

Table D.2: The most salient (S) and least salient (NS) attributes of each attribute *not* derived from user biographies within the *Reopen* Dataset

Personal ID		Mention		Hashtag		Emoji	
S	NS	S	NS	S	NS	S	NS
she	ig	@flamengo	@bts_twt	#maga	#blacklivesmatter	🇺🇸	❤️
her	instagram	@vascodagama	@manutd	#kag	#mufc	✨	🤍
they	writer	@fluminensefc	@lfc	#resist	#ynwa	🇫🇷	💙
maga	music	@realdonaldtrump	@realmadrid	#trump2020	#bernie2020	🇮🇳	⚽
he	fan account	@narendramodi	@fcbarcelona	#wwg1wga	#bts	🇮🇹	💛

Table D.3: The most salient (S) and least salient (NS) attributes of each attribute derived from user biographies within the *COVID* Dataset

Name Hashtag		Location Unigram	
S	NS	S	NS
#oustduterte	#loona1stwin	argentina	usa
#yoapruebo	#fbpe	france	de
#apruебо	#	brasil	new
#facciamorete	#bernie2020	españa	the
#maga	# flattenthecurve	india	ca

Table D.4: The most salient (S) and least salient (NS) attributes of each attribute *not* derived from user biographies within the *COVID* Dataset

Personal ID		Mention		Hashtag		Emoji	
S	NS	S	NS	S	NS	S	NS
she	gamer	@weareoneexo	@bts_twt	#maga	#resist	🇺🇸	❤️
her	writer	@actorvijay	@twitch	#kag	#blacklivesmatter	🏳️	🤍
fan account	music	@genflynn	@manutd	#2a	#marvel	😎	💙
fub free	ig	@b_hundred_hyun	@marvel	#trump2020	#blm	🐰	✨
maga	artist	@iamsrk	@lfc	#nra	#mufc	🇲🇾	💕

Table D.5: The most salient (S) and least salient (NS) attributes of each attribute derived from user biographies within the *Captain Marvel* Dataset

Name Hashtag		Location Unigram	
S	NS	S	NS
#saveodaat	#twoofus	malaysia	usa
#releasethesnydercut	#fightforwynonna	brasil	the
#maga	#renewodaat	france	ca
#fbpe	#saveshadowhunters	thailand	new
#peoplesvote	#savedaredevil	indonesia	england

Table D.6: The most salient (S) and least salient (NS) attributes of each attribute *not* derived from user biographies within the *Captain Marvel* Dataset

D.1.3 Prototypical Attributes

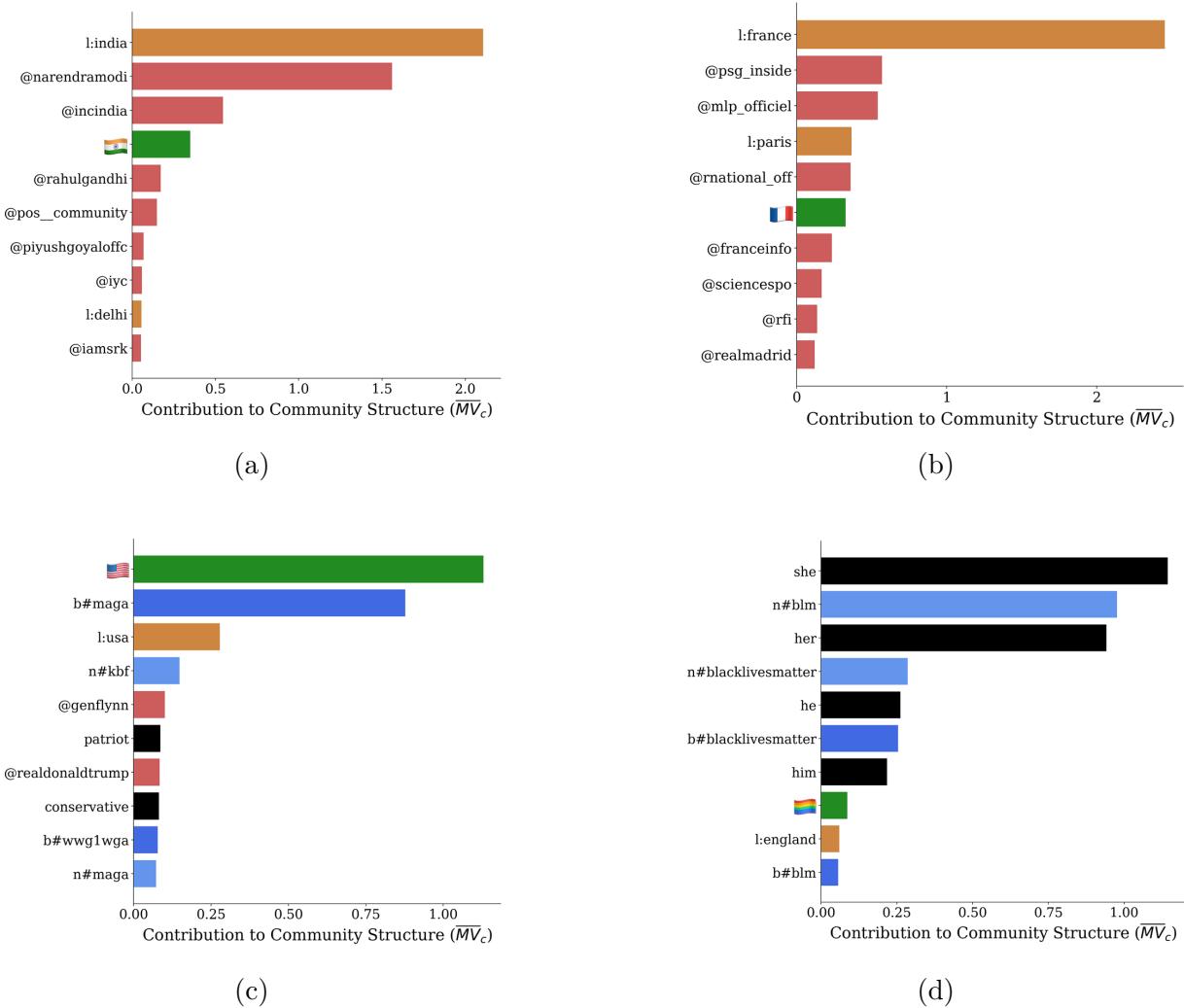


Figure D.2: Prototypes of the communities 5-8 in the *Election* dataset.

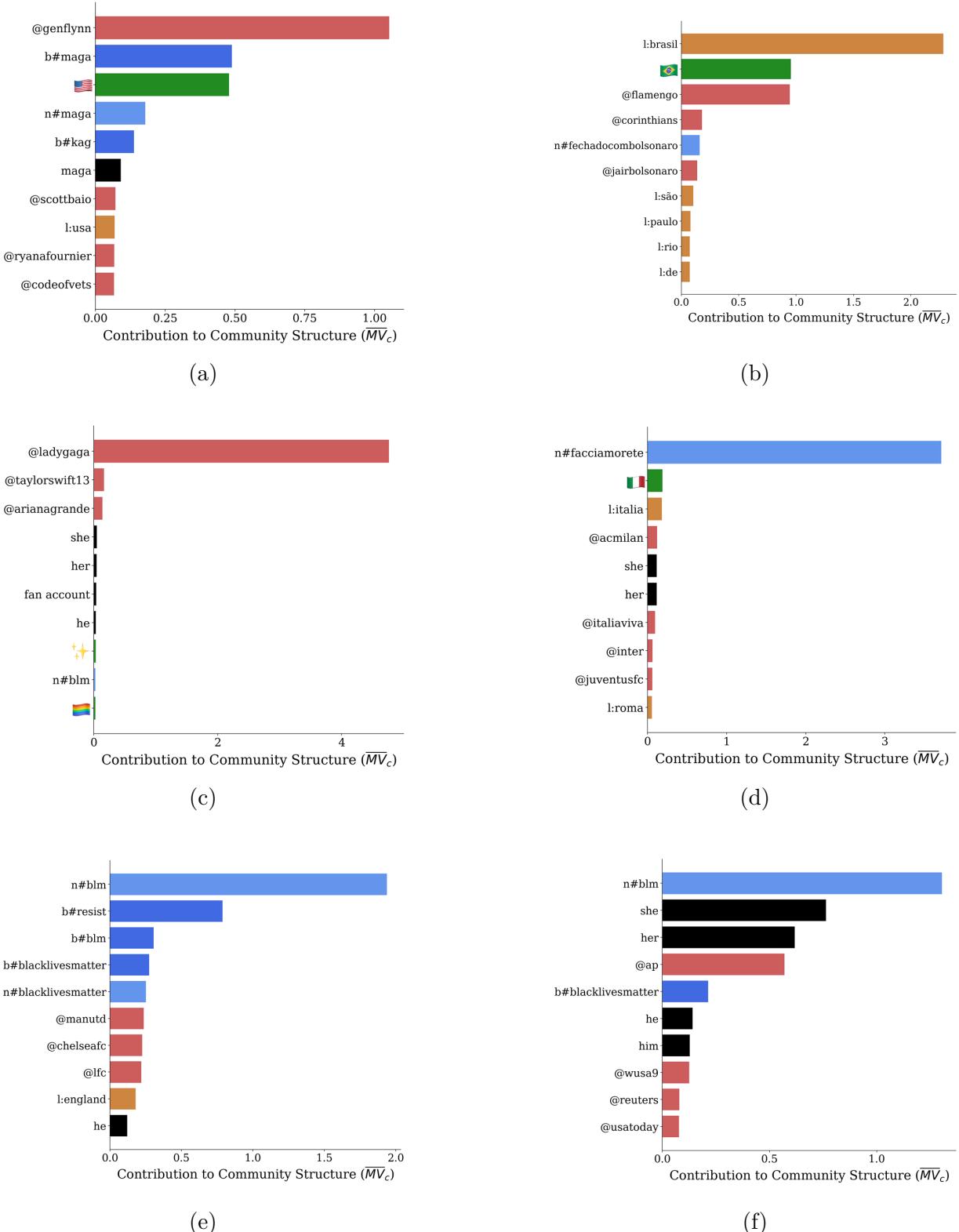


Figure D.3: Prototypes of the communities 9-14 in the *Election* dataset.

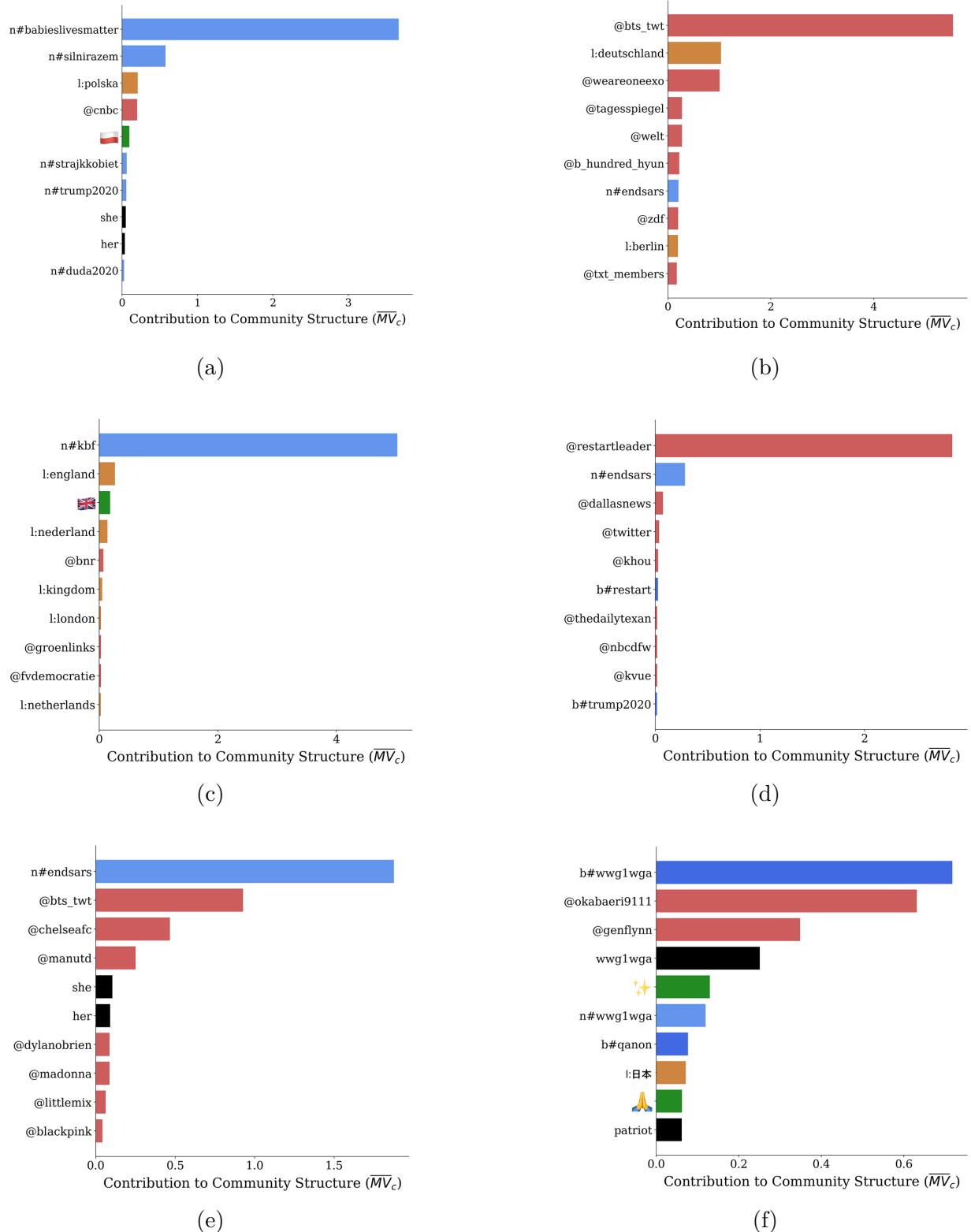


Figure D.4: Prototypes of the communities 15-20 in the *Election* dataset.

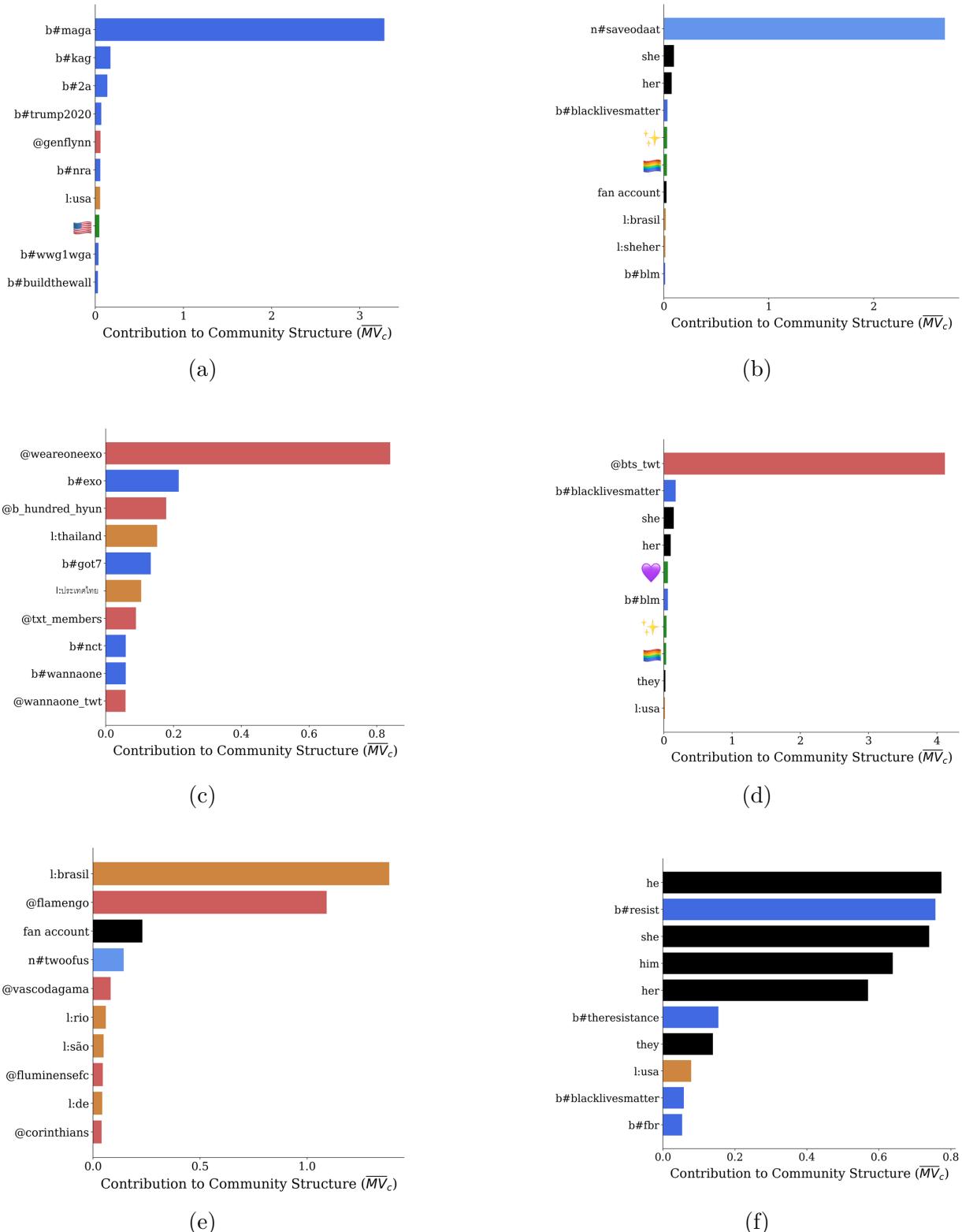


Figure D.5: Prototypes of the communities 1-6 in the *Captain Marvel* dataset.

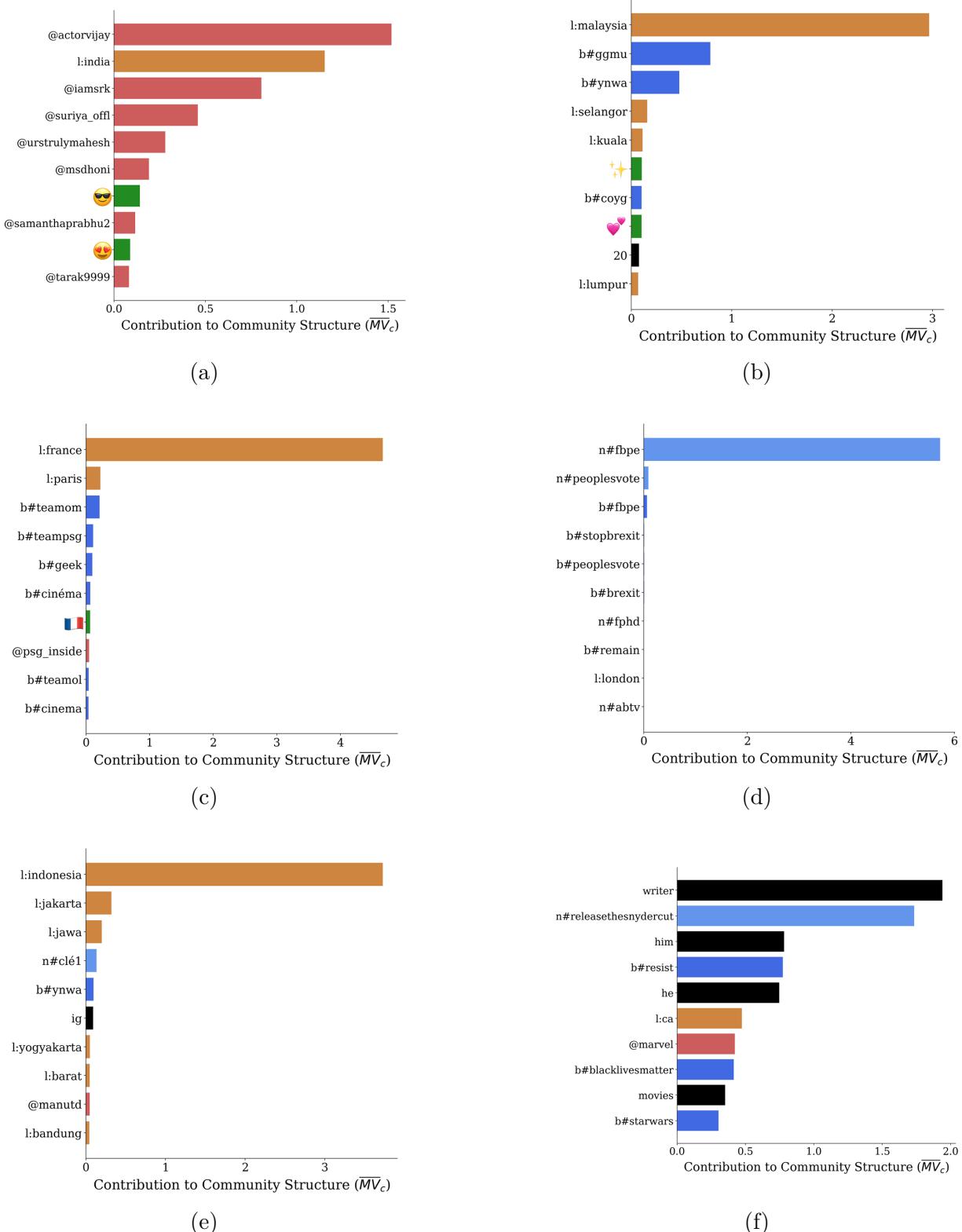


Figure D.6: Prototypes of the communities 7-12 in the *Captain Marvel* dataset.

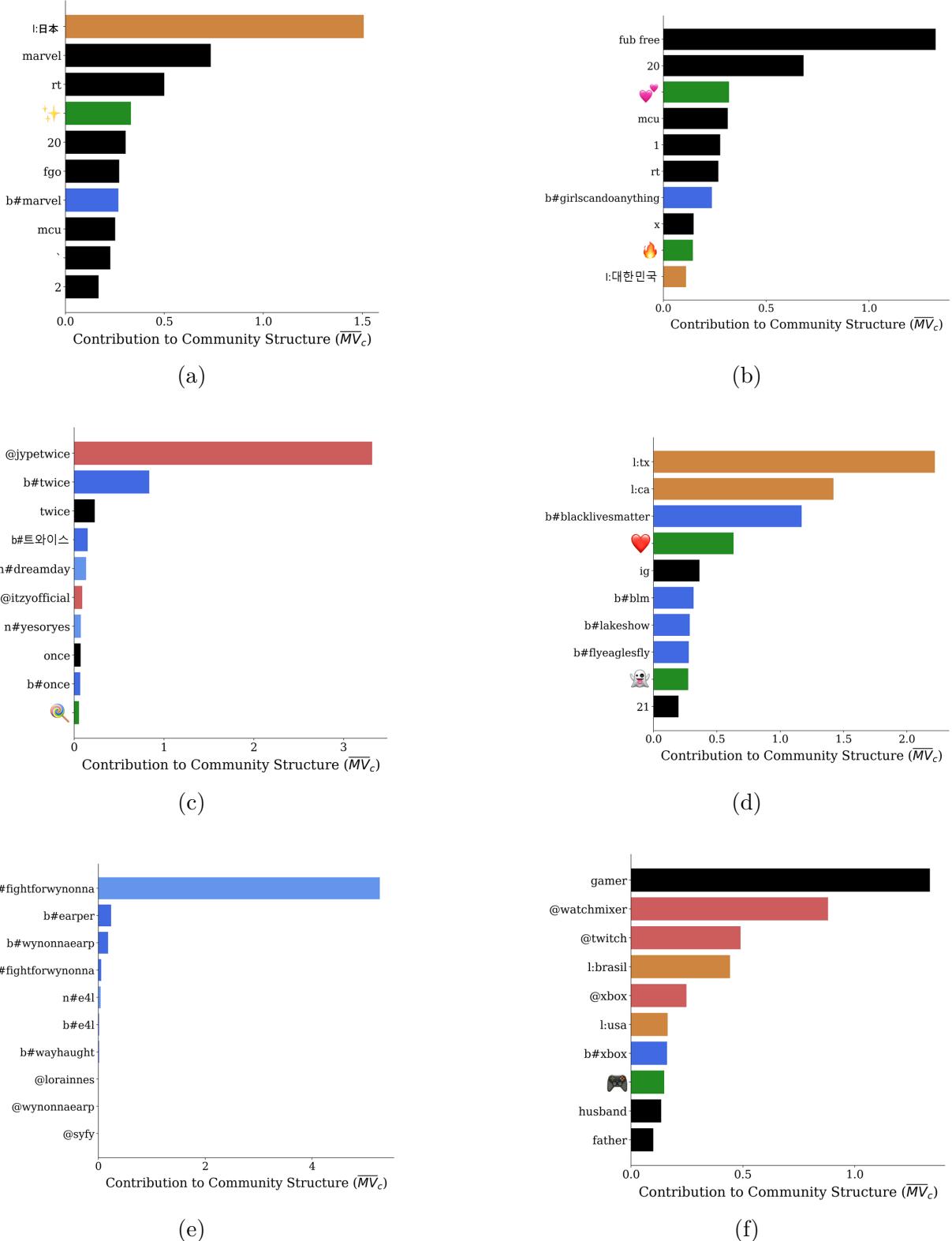


Figure D.7: Prototypes of the communities 13-18 in the *Captain Marvel* dataset.

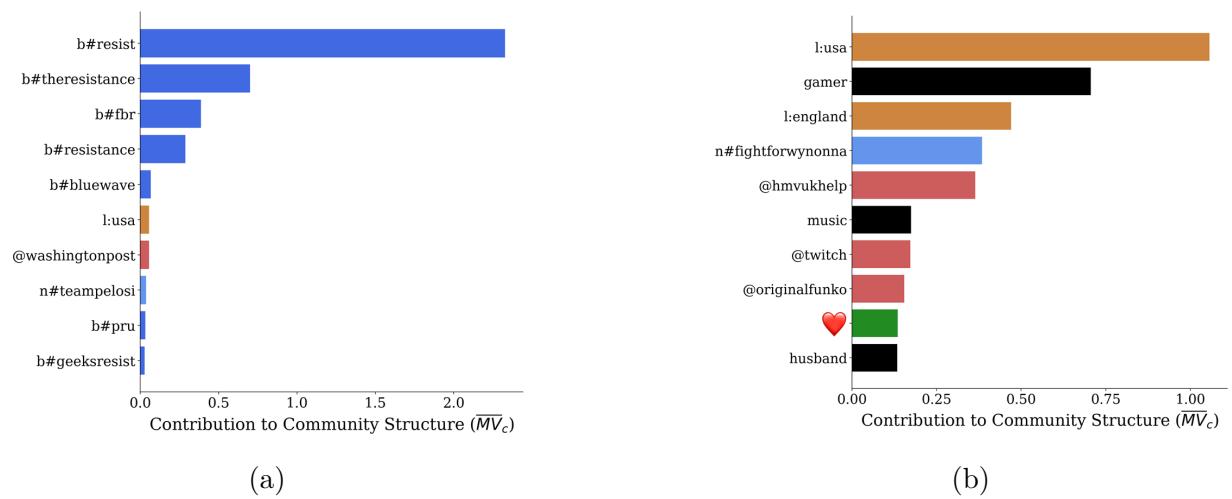


Figure D.8: Prototypes of the communities 19-20 in the *Captain Marvel* dataset.

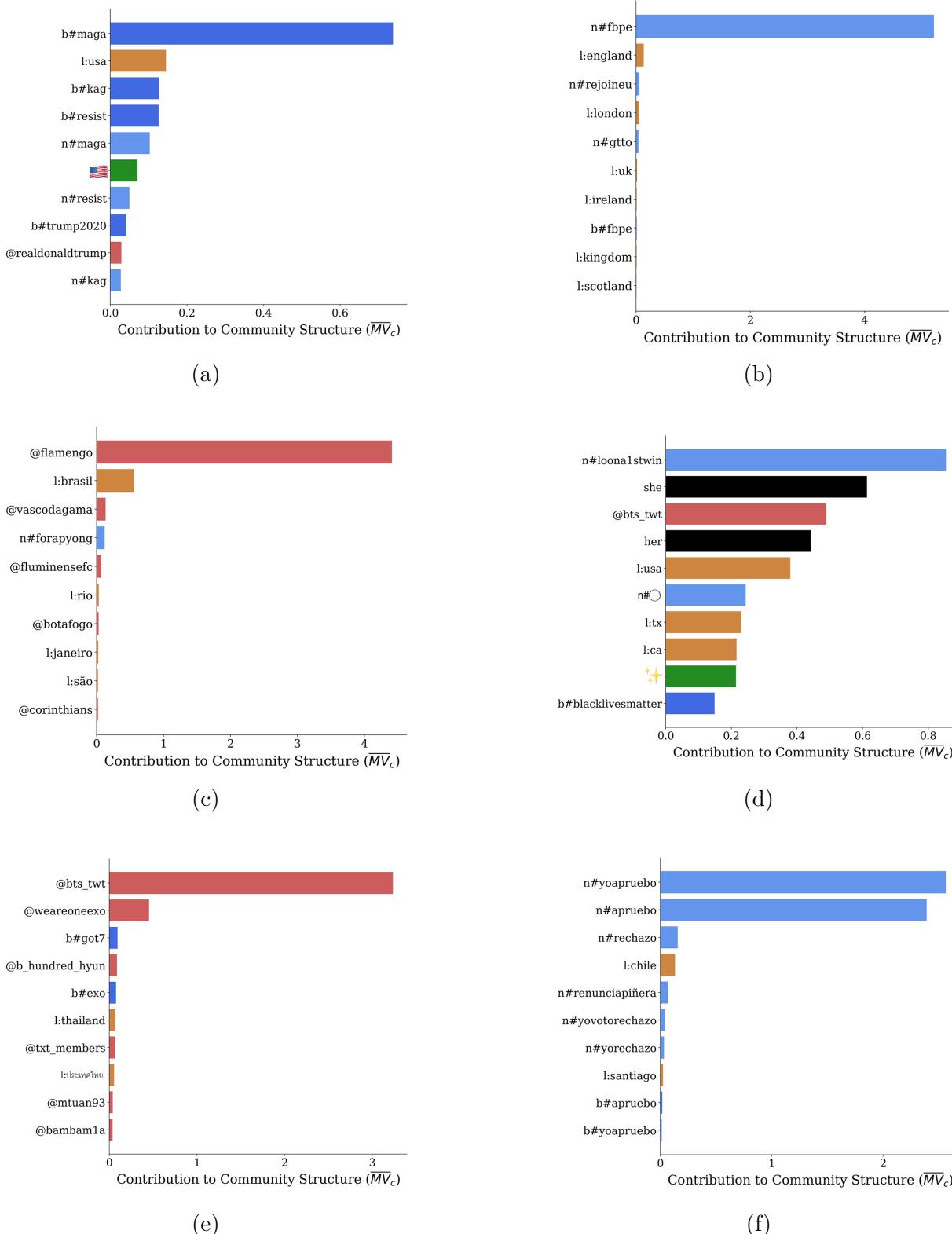


Figure D.9: Prototypes of the communities 1-6 in the *COVID* dataset.

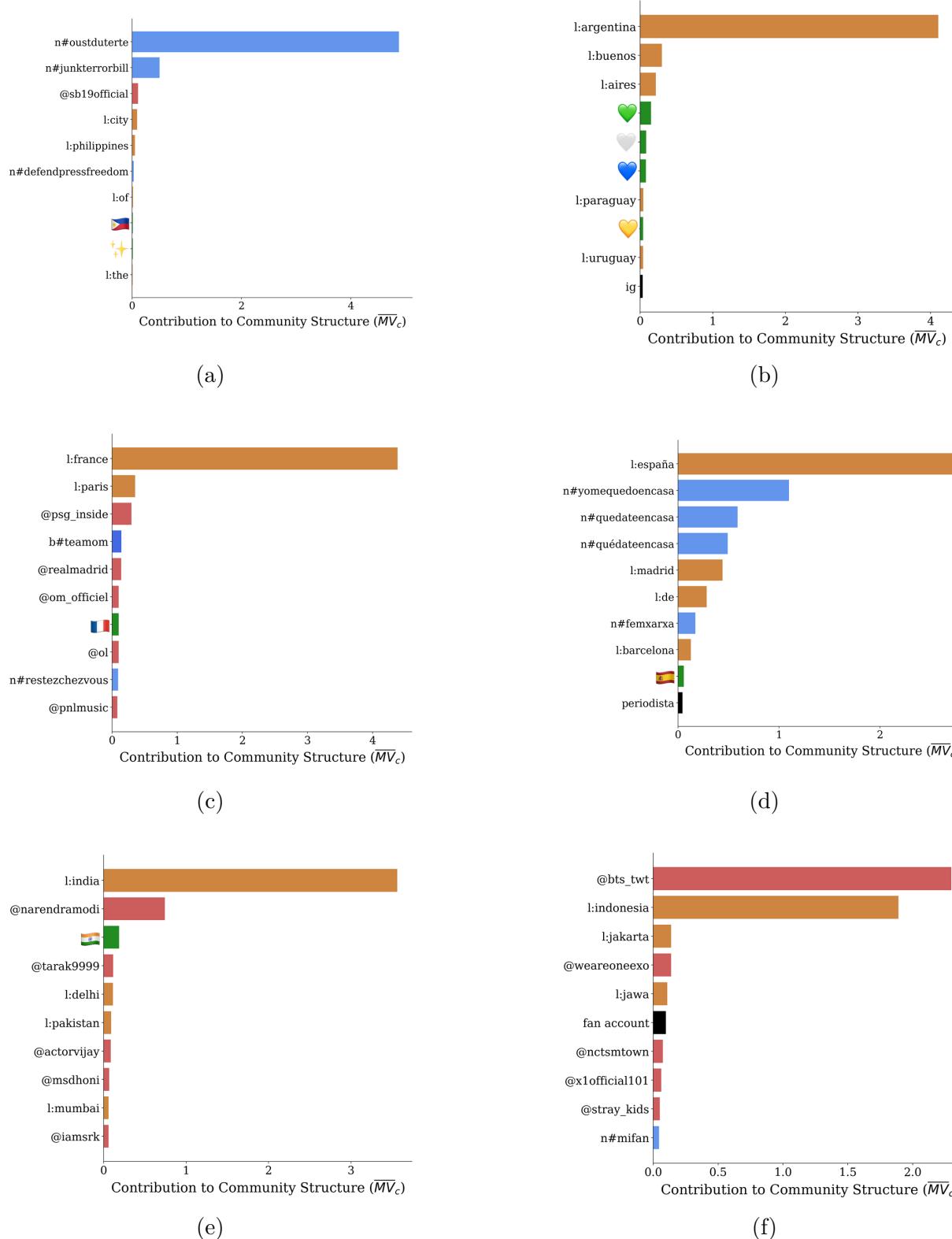


Figure D.10: Prototypes of the communities 7-12 in the *COVID* dataset.

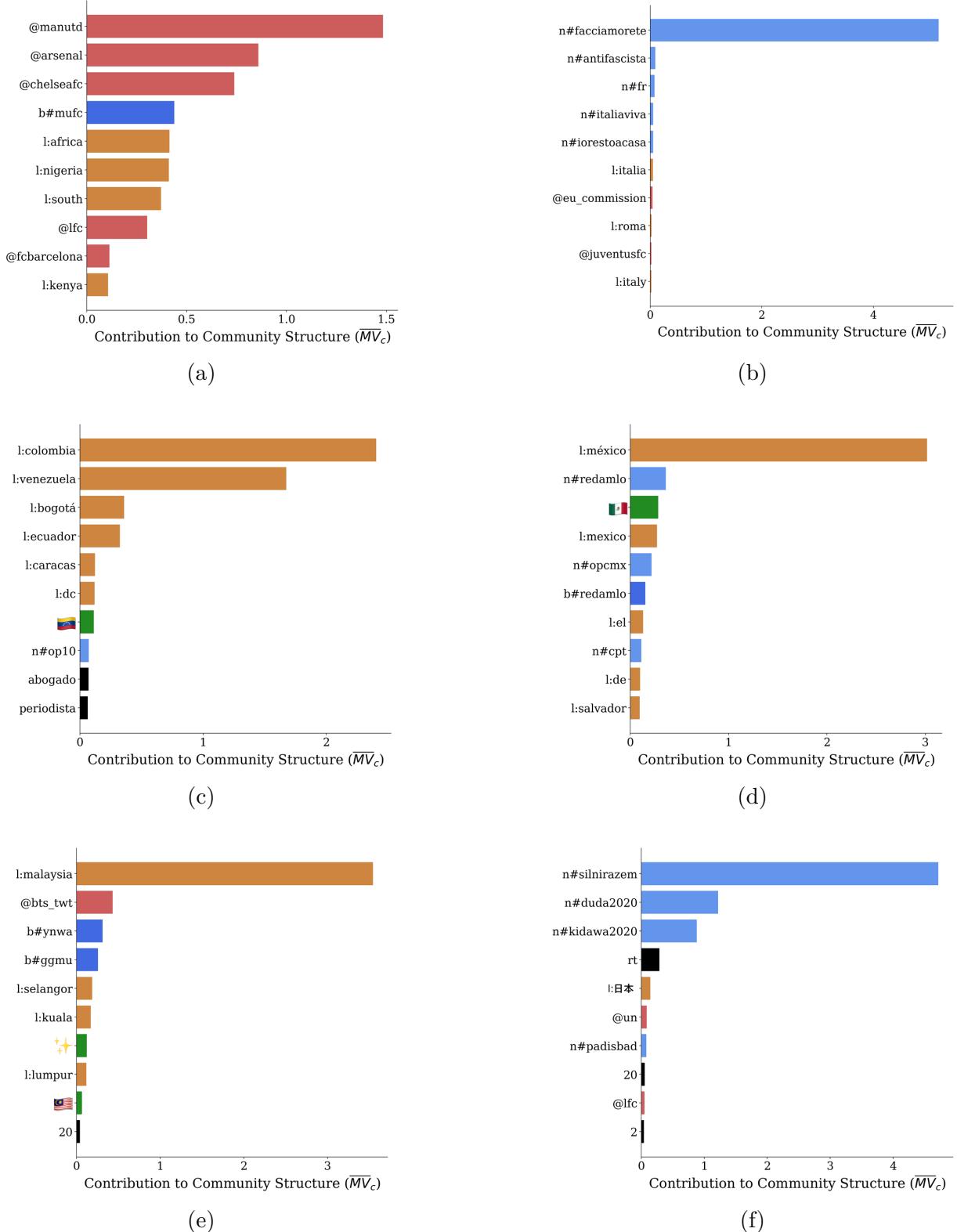


Figure D.11: Prototypes of the communities 13-18 in the *COVID* dataset.

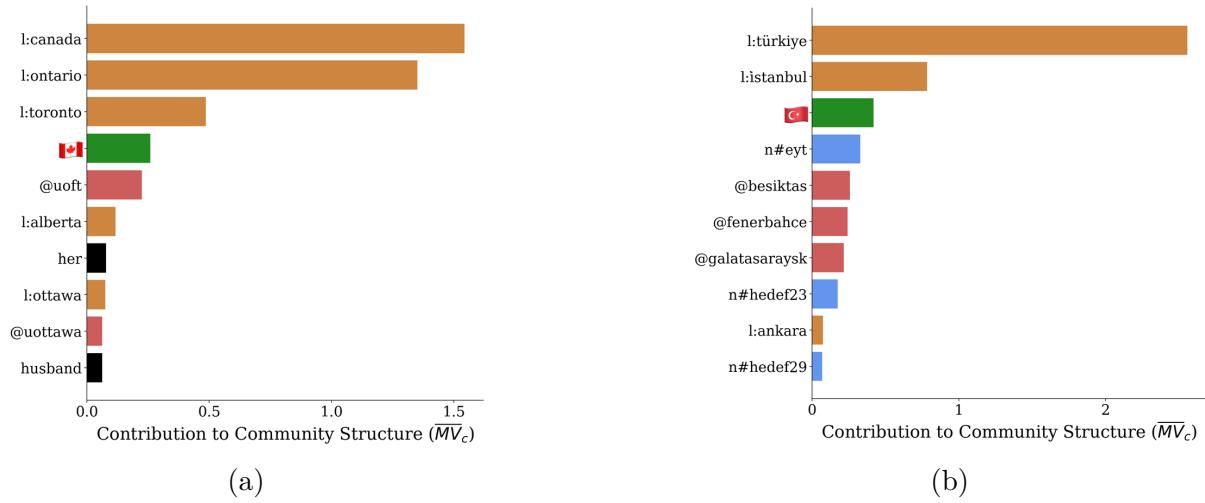


Figure D.12: Prototypes of the communities 19-20 in the *COVID* dataset.

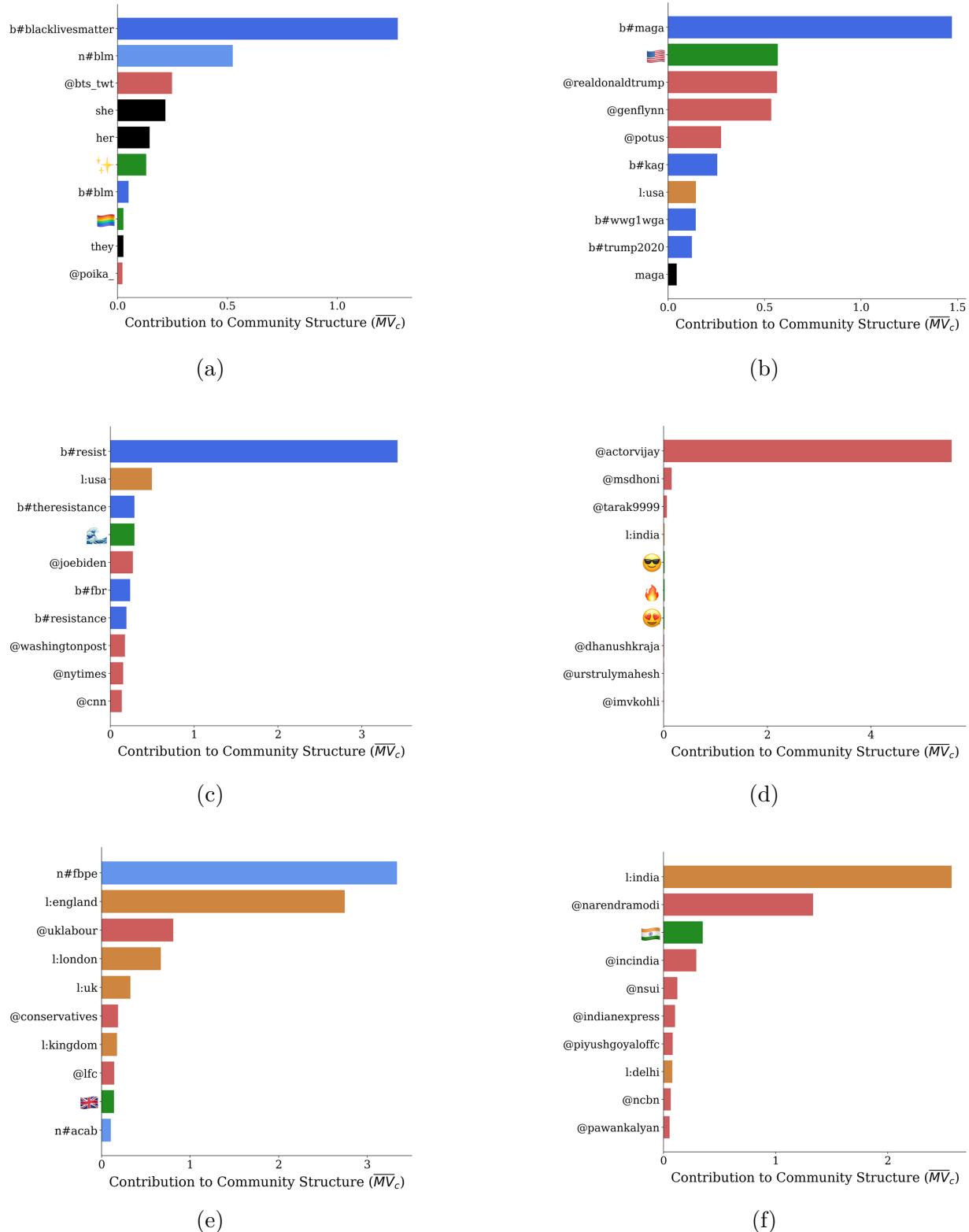


Figure D.13: Prototypes of the communities 1-6 in the Reopen dataset.

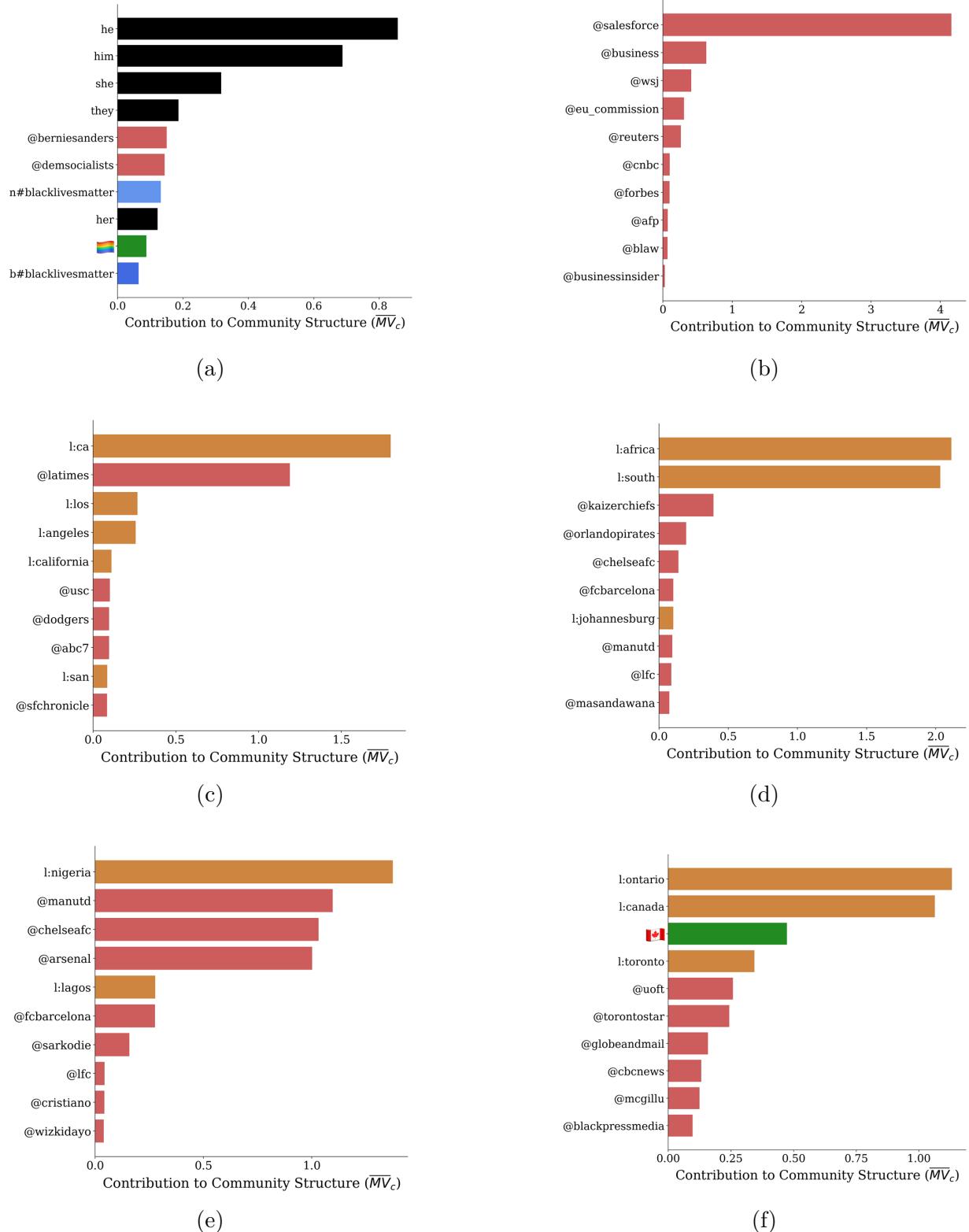


Figure D.14: Prototypes of the communities 7-12 in the Reopen dataset.

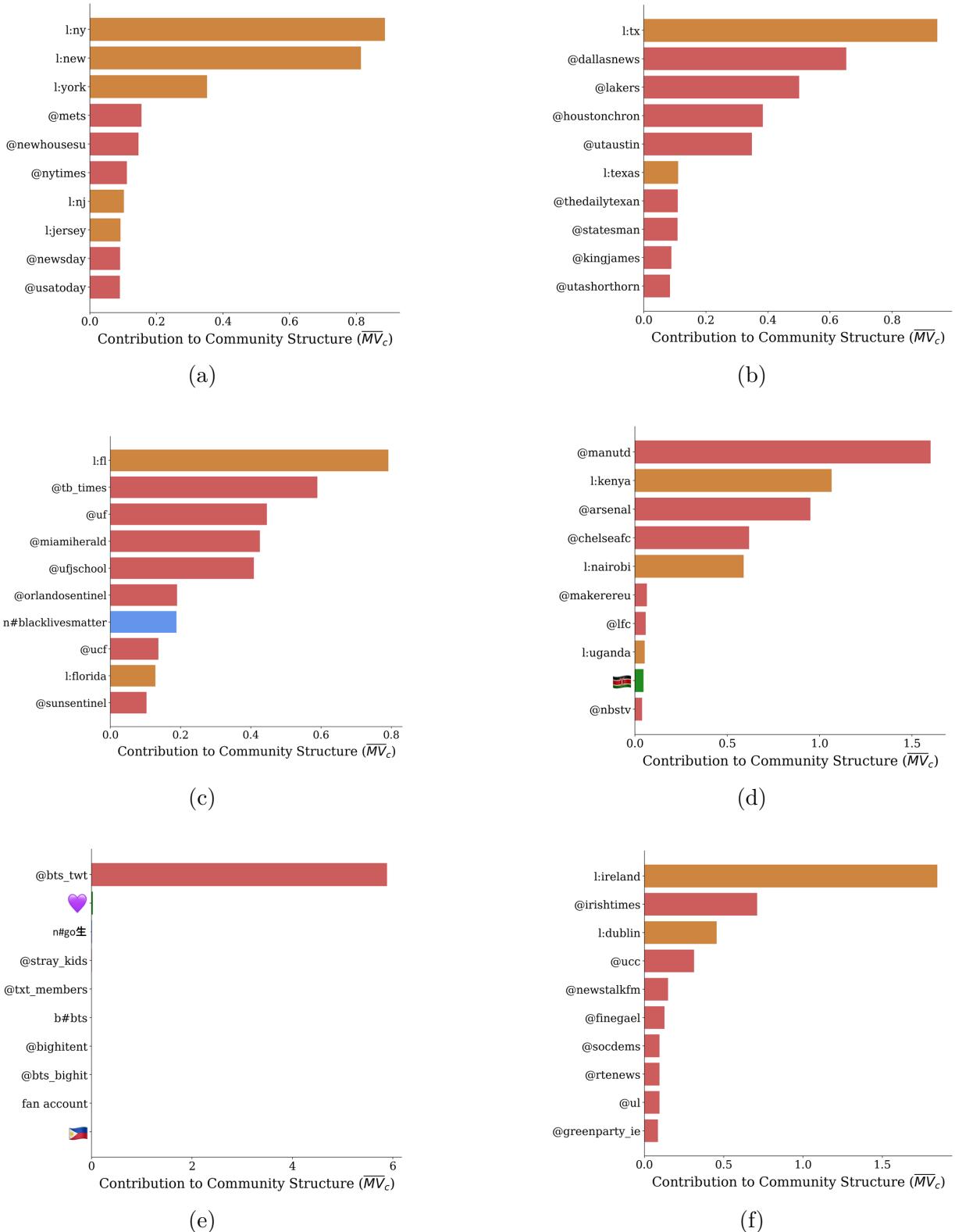


Figure D.15: Prototypes of the communities 13-18 in the Reopen dataset.

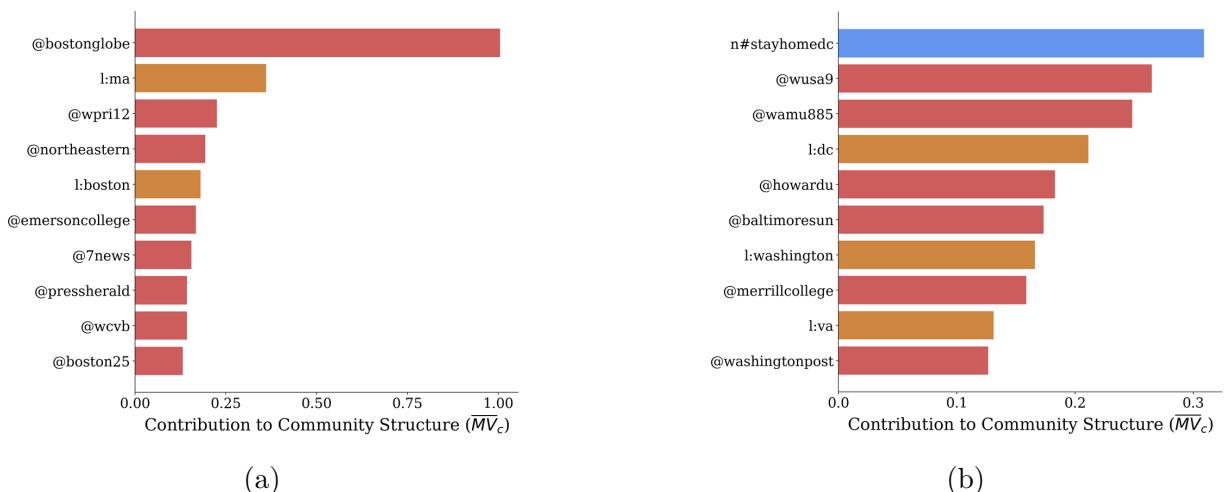


Figure D.16: Prototypes of the communities 19-20 in the Reopen dataset.

Bibliography

- [1] Andrew Abbott. Sequence analysis: New methods for old ideas. *Annual review of sociology*, 21(1):93–113, 1995. [Cited on page 47.]
- [2] Dominic Abrams and Michael A Hogg. Comments on the motivational status of self-esteem in social identity and intergroup discrimination. *European journal of social psychology*, 18(4):317–334, 1988. [Cited on page 78.]
- [3] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, 47(1):10, 2014. [Cited on page 54.]
- [4] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *nature*, 466(7307):761–764, 2010. [Cited on page 6.]
- [5] Dina Al Raffie. Social identity theory for investigating islamic extremism in the diaspora. *Journal of Strategic Security*, 6(4):67–91, 2013. [Cited on page 77.]
- [6] Alberto Aleta and Yamir Moreno. Multilayer networks in a nutshell. *Annual Review of Condensed Matter Physics*, 10:45–62, 2019. [Cited on page 5.]
- [7] Thayer Alshaabi, Jane L Adams, Michael V Arnold, Joshua R Minot, David R Dewhurst, Andrew J Reagan, Christopher M Danforth, and Peter Sheridan Dodds. Storywrangler: A massive exploratorium for sociolinguistic, cultural, socioeconomic, and political timelines using twitter. *Science advances*, 7(29):eabe6534, 2021. [Cited on pages 7 and 17.]
- [8] Thayer Alshaabi, Michael V Arnold, Joshua R Minot, Jane Lydia Adams, David Rushing Dewhurst, Andrew J Reagan, Roby Muhamad, Christopher M Danforth, and Peter Sheridan Dodds. How the world’s collective attention is being paid to a pandemic: Covid-19 related n-gram time series for 24 languages on twitter. *Plos one*, 16(1):e0244476, 2021. [Cited on pages 7 and 17.]
- [9] David Alvarez-Melis and Martin Saveski. Topic modeling in twitter: Aggregating tweets by conversations. In *Tenth international AAAI conference on web and social media*, 2016. [Cited on pages 7, 16, and 23.]
- [10] Rudy Arthur. Modularity and projection of bipartite networks. *Physica A: Statistical Mechanics and its Applications*, page 124341, 2020. [Cited on page 82.]
- [11] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In *Modeling and optimization in mobile, ad hoc and wireless networks (WiOpt), 2010 proceedings of the 8th international symposium on*, pages

- 513–519. IEEE, 2010. [Cited on page 46.]
- [12] Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, and Qinna Wang. Communities in evolving networks: definitions, detection, and analysis techniques. In *Dynamics On and Of Complex Networks, Volume 2*, pages 159–200. Springer, 2013. [Cited on pages 46 and 54.]
- [13] Matthew Babcock and Kathleen M Carley. Operation gridlock: opposite sides, opposite strategies. *Journal of Computational Social Science*, pages 1–25, 2021. [Cited on page 9.]
- [14] Matthew Babcock, Ramon Villa-Cox, and Kathleen M Carley. Pretending positive, pushing false: Comparing captain marvel misinformation campaigns. In *Disinformation, misinformation, and fake news in social media*, pages 83–94. Springer, 2020. [Cited on page 11.]
- [15] Michael J Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102, 2007. [Cited on page 82.]
- [16] Mihovil Bartulović. *On Trail Comparison, Clustering, and Prediction: Building a Framework for Working with Sequential Network Data*. PhD thesis, Carnegie Mellon University, 2021. [Cited on pages 7, 47, and 71.]
- [17] Matthew C Benigni, Kenneth Joseph, and Kathleen M Carley. Online extremism and the communities that sustain it: Detecting the isis supporting community on twitter. *PloS one*, 12(12):e0181405, 2017. [Cited on page 71.]
- [18] Parantapa Bhattacharya, Muhammad Bilal Zafar, Niloy Ganguly, Saptarshi Ghosh, and Krishna P Gummadi. Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 357–360, 2014. [Cited on pages 8 and 79.]
- [19] Francesco C Billari. Sequence analysis in demographic research. *Canadian Studies in Population [ARCHIVES]*, pages 439–458, 2001. [Cited on page 47.]
- [20] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. [Cited on pages 7 and 16.]
- [21] Christopher Blöcker, Juan Carlos Nieves, and Martin Rosvall. Map equation centrality: community-aware centrality based on the map equation. *Applied Network Science*, 7(1):1–24, 2022. [Cited on page 84.]
- [22] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. [Cited on pages 5, 46, 60, 134, and 137.]
- [23] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X. [Cited on pages 8 and 22.]
- [24] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005. [Cited on pages 9 and 140.]

- [25] Stephen P Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006. [Cited on pages 9 and 134.]
- [26] Stephen P Borgatti, Kathleen M Carley, and David Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2):124–136, 2006. [Cited on pages 5, 13, 14, and 147.]
- [27] Cigdem Bozdag. Managing diverse online networks in the context of polarization: Understanding how we grow apart on and through social media. *Social Media + Society*, 6(4):2056305120975713, 2020. [Cited on page 94.]
- [28] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001. [Cited on pages 141 and 147.]
- [29] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2007. [Cited on pages 134 and 137.]
- [30] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. [Cited on pages 8 and 27.]
- [31] Christian Brzinsky-Fay and Ulrich Kohler. New developments in sequence analysis, 2010. [Cited on page 47.]
- [32] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284, 2000. [Cited on page 72.]
- [33] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. [Cited on page 104.]
- [34] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000. [Cited on pages 140 and 141.]
- [35] Gian Maria Campedelli, Mihovil Bartulovic, and Kathleen M Carley. Pairwise similarity of jihadist groups in target and weapon transitions. *Journal of Computational Social Science*, 2:245–270, 2019. [Cited on page 47.]
- [36] Gian Maria Campedelli, Mihovil Bartulovic, and Kathleen M Carley. Learning future terrorist targets through temporal meta-graphs. *Scientific reports*, 11(1):1–15, 2021. [Cited on page 7.]
- [37] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep Neural Networks for Learning Graph Representations. In *AAAI*, 2016. [Cited on pages 8 and 17.]
- [38] Kathleen M Carley. Social cybersecurity: an emerging science. *Computational and mathematical organization theory*, 26(4):365–381, 2020. [Cited on pages 4 and 15.]
- [39] Kathleen M Carley, Guido Cervone, Nitin Agarwal, and Huan Liu. Social cybersecurity. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 389–394. Springer, 2018. [Cited on pages 4, 15, and 135.]

- [40] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020. [Cited on page 24.]
- [41] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. HARP: Hierarchical Representation Learning for Networks. *arXiv:1706.07845 [cs]*, June 2017. URL <http://arxiv.org/abs/1706.07845>. arXiv: 1706.07845. [Cited on pages 8 and 18.]
- [42] Jinyin Chen, Lihong Chen, Yixian Chen, Minghao Zhao, Shanqing Yu, Qi Xuan, and Xiaoniu Yang. Ga-based q-attack on community detection. *IEEE Transactions on Computational Social Systems*, 6(3):491–503, 2019. [Cited on pages 142 and 158.]
- [43] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *OpenReview Preprint*, 2019. [Cited on page 42.]
- [44] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12): 2928–2941, 2014. [Cited on pages 7 and 16.]
- [45] Hocine Cherifi, Gergely Palla, Boleslaw K Szymanski, and Xiaoyan Lu. On community structure in complex networks: challenges and opportunities. *Applied Network Science*, 4(1):1–35, 2019. [Cited on page 134.]
- [46] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004. [Cited on pages 134 and 137.]
- [47] Christophe Combet, Christophe Blanchet, Christophe Geourjon, and Gilbert Deleage. Nps@: network protein sequence analysis. *Trends in biochemical sciences*, 25 (3):147–150, 2000. [Cited on page 47.]
- [48] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Fifth international AAAI conference on weblogs and social media*, 2011. [Cited on pages 5, 79, and 94.]
- [49] Iain Cruickshank. *Multi-view Clustering of Social-based Data*. PhD thesis, Carnegie Mellon University, Oct 2020. [Cited on page 5.]
- [50] Iain J Cruickshank and Kathleen M Carley. Characterizing communities of hashtag usage on twitter during the 2020 covid-19 pandemic by multi-view clustering. *Applied Network Science*, 5(1):1–40, 2020. [Cited on page 86.]
- [51] Bruno Requiao da Cunha, Juan Carlos González-Avella, and Sebastián Gonçalves. Fast fragmentation of networks using module-based attacks. *PloS one*, 10(11): e0142824, 2015. [Cited on pages 134, 140, 141, 146, and 147.]
- [52] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. The spreading of

misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016. [Cited on page 77.]

- [53] 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. [Cited on page 8.]
- [54] Kitsy Dixon. Feminist online identity: Analyzing the presence of hashtag feminism. *Journal of Arts and Humanities*, 3(7):34–40, 2014. [Cited on pages 78 and 93.]
- [55] Tien Huu Do, Duc Minh Nguyen, Evangelia Tsiliogianni, Bruno Cornelis, and Nikos Deligiannis. Twitter user geolocation using deep multiview learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6304–6308. IEEE, 2018. [Cited on page 17.]
- [56] Peter Sheridan Dodds, Joshua R Minot, Michael V Arnold, Thayer Alshaabi, Jane Lydia Adams, Andrew J Reagan, and Christopher M Danforth. Computational timeline reconstruction of the stories surrounding trump: Story turbulence, narrative control, and collective chronopathy. *PloS one*, 16(12):e0260592, 2021. [Cited on pages 7, 17, and 45.]
- [57] Bertjan Doosje, Fathali M Moghaddam, Arie W Kruglanski, Arjan De Wolf, Liesbeth Mann, and Allard R Feddes. Terrorism, radicalization and de-radicalization. *Current Opinion in Psychology*, 11:79–84, 2016. [Cited on page 77.]
- [58] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 226–231. AAAI Press, 1996. [Cited on pages 5, 28, and 72.]
- [59] Martin G Everett and Stephen P Borgatti. Induced, endogenous and exogenous centrality. *Social Networks*, 32(4):339–344, 2010. [Cited on pages 83 and 147.]
- [60] Paul Expert, Tim S Evans, Vincent D Blondel, and Renaud Lambiotte. Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*, 108(19):7663–7668, 2011. [Cited on pages 5 and 127.]
- [61] Wei Feng, Chao Zhang, Wei Zhang, Jiawei Han, Jianyong Wang, Charu Aggarwal, and Jianbin Huang. Streamcube: Hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *2015 IEEE 31st international conference on data engineering*, pages 1561–1572. IEEE, 2015. [Cited on pages 7, 17, 21, and 45.]
- [62] Emilio Ferrara. Twitter spam and false accounts prevalence, detection and characterization: A survey. *arXiv preprint arXiv:2211.05913*, 2022. [Cited on page 29.]
- [63] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. [Cited on page 27.]
- [64] Valeria Fionda and Giuseppe Pirro. Community deception or: How to stop fearing community detection algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 30(4):660–673, 2017. [Cited on pages 141 and 158.]
- [65] Bailey K Fosdick, Daniel B Larremore, Joel Nishimura, and Johan Ugander. Con-

figuring random graph models with fixed degree sequences. *Siam Review*, 60(2):315–355, 2018. [Cited on page 82.]

- [66] Terrill L Frantz, Marcelo Cataldo, and Kathleen M Carley. Robustness of centrality measures under uncertainty: Examining the role of network topology. *Computational and Mathematical Organization Theory*, 15(4):303, 2009. [Cited on pages 13, 14, and 147.]
- [67] Junhao Gan and Yufei Tao. Dbscan revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 519–530, 2015. [Cited on pages 5 and 28.]
- [68] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27, 2018. [Cited on page 94.]
- [69] Venkata Rama Kiran Garimella and Ingmar Weber. A long-term analysis of polarization on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017. [Cited on page 5.]
- [70] Jing Ge. Emoji sequence use in enacting personal identity. In *Companion proceedings of the 2019 world wide web conference*, pages 426–438, 2019. [Cited on page 78.]
- [71] Zakariya Ghalmane, Chantal Cherifi, Hocine Cherifi, and Mohammed El Hassouni. Centrality in complex networks with overlapping community structure. *Scientific reports*, 9(1):1–29, 2019. [Cited on page 139.]
- [72] Zakariya Ghalmane, Mohammed El Hassouni, Chantal Cherifi, and Hocine Cherifi. Centrality in modular networks. *EPJ Data Science*, 8(1):1–27, December 2019. ISSN 2193-1127. doi: 10.1140/epjds/s13688-019-0195-7. URL <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-019-0195-7>. Number: 1 Publisher: SpringerOpen. [Cited on pages 9, 87, 133, 138, 140, and 146.]
- [73] Zakariya Ghalmane, Mohammed El Hassouni, and Hocine Cherifi. Immunization of networks with non-overlapping community structure. *Social Network Analysis and Mining*, 9(1):45, 2019. [Cited on pages 9, 79, 87, 133, 138, 139, 140, and 146.]
- [74] Arnab Kumar Ghoshal, Nabanita Das, and Soham Das. Influence of community structure on misinformation containment in online social networks. *Knowledge-Based Systems*, 213:106693, 2021. [Cited on page 77.]
- [75] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002. [Cited on page 133.]
- [76] Jennifer Golbeck, Summer Ash, and Nicole Cabrera. Hashtags as online communities with social support: A study of anti-sexism-in-science hashtag movements. *First Monday*, 2017. [Cited on page 78.]
- [77] Mark K Goldberg, Malik Magdon-Ismail, Srinivas Nambirajan, and James Thompson. Tracking and predicting evolution of social communities. In *SocialCom/PASAT*, pages 780–783. Citeseer, 2011. [Cited on page 46.]

- [78] Benjamin H Good, Yves-Alexandre De Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical review E*, 81(4):046106, 2010. [Cited on pages 6 and 79.]
- [79] Eduardo Graells-Garrido, Ricardo Baeza-Yates, and Mounia Lalmas. Every colour you are: Stance prediction and turnaround in controversial issues. In *12th ACM Conference on Web Science*, pages 174–183, 2020. [Cited on pages 78 and 93.]
- [80] Derek Greene, Donal Doyle, and Padraig Cunningham. Tracking the evolution of communities in dynamic social networks. In *Advances in social networks analysis and mining (ASONAM), 2010 international conference on*, pages 176–183. IEEE, 2010. [Cited on page 54.]
- [81] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363 (6425):374–378, 2019. [Cited on pages 4 and 15.]
- [82] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 855–864, San Francisco, California, USA, 2016. ACM Press. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939754. URL <http://dl.acm.org/citation.cfm?doid=2939672.2939754>. [Cited on pages 8 and 17.]
- [83] Pedro Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, pages 215–224, 2013. [Cited on pages 79 and 94.]
- [84] Roger Guimera and Luís A Nunes Amaral. Functional cartography of complex metabolic networks. *nature*, 433(7028):895–900, 2005. [Cited on page 79.]
- [85] Naveen Gupta, Anurag Singh, and Hocine Cherifi. Centrality measures for networks with community structure. *Physica A: Statistical Mechanics and its Applications*, 452:46–59, 2016. [Cited on pages 9, 133, 138, and 140.]
- [86] Loni Hagen, Mary Falling, Oleksandr Lisnichenko, AbdelRahim A Elmadany, Pankti Mehta, Muhammad Abdul-Mageed, Justin Costakis, and Thomas E Keller. Emoji use in twitter white nationalism communication. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, pages 201–205, 2019. [Cited on pages 78 and 91.]
- [87] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017. [Cited on pages 8, 17, and 26.]
- [88] Jiyoung Han and Christopher M Federico. Conflict-framed news, self-categorization, and partisan polarization. *Mass Communication and Society*, 20(4):455–480, 2017. [Cited on page 77.]

- [89] Claire Hardaker and Mark McGlashan. “real men don’t hate women”: Twitter rape threats and group identity. *Journal of Pragmatics*, 91:80–93, 2016. [Cited on page 8.]
- [90] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature*, 402(6761):C47–C52, 1999. [Cited on page 133.]
- [91] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. [Cited on page 27.]
- [92] Josephine Hennessy and Michael A West. Intergroup behavior in organizations: A field test of social identity theory. *Small group research*, 30(3):361–382, 1999. [Cited on page 77.]
- [93] AmaÇ HerdaGdelen, Wenyun Zuo, Alexander Gard-Murray, and Yaneer Bar-Yam. An exploration of social identity: The geography and politics of news-sharing communities in twitter. *Complexity*, 19(2):10–20, 2013. [Cited on page 8.]
- [94] Andrea P Herrera. Theorizing the lesbian hashtag: Identity, community, and the technological imperative to name the sexual self. *Journal of lesbian studies*, 22(3):313–328, 2018. [Cited on pages 78 and 93.]
- [95] Jerry R Hobbs. Topic drift. *Conversational organization and its development*, 38:3–22, 1990. [Cited on page 19.]
- [96] Michael A Hogg. Intragroup processes, group structure and social identity. *Social groups and identities: Developing the legacy of Henri Tajfel*, 65:93, 1996. [Cited on page 78.]
- [97] Michael A Hogg. Social identity theory. In *Understanding peace and conflict through social identity theory*, pages 3–17. Springer, 2016. [Cited on page 8.]
- [98] Michael A Hogg. Social identity, self-categorization, and the small group. In *Understanding group behavior*, pages 227–253. Psychology Press, 2018. [Cited on page 8.]
- [99] Michael A Hogg, John C Turner, and Barbara Davidson. Polarized norms and social frames of reference: A test of the self-categorization theory of group polarization. *Basic and Applied Social Psychology*, 11(1):77–100, 1990. [Cited on pages 8 and 77.]
- [100] Michael A Hogg, Dominic Abrams, Sabine Otten, and Steve Hinkle. The social identity perspective: Intergroup relations, self-conception, and small groups. *Small group research*, 35(3):246–276, 2004. [Cited on page 77.]
- [101] Petter Holme. Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(9):234, September 2015. ISSN 1434-6028, 1434-6036. doi: 10.1140/epjb/e2015-60657-4. URL <http://arxiv.org/abs/1508.01303>. arXiv: 1508.01303. [Cited on page 7.]
- [102] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012. [Cited on page 7.]
- [103] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulner-

- ability of complex networks. *Physical review E*, 65(5):056109, 2002. [Cited on pages 85, 141, and 146.]
- [104] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010. [Cited on pages 7 and 16.]
 - [105] Matthew J Hornsey. Social identity theory and self-categorization theory: A historical review. *Social and personality psychology compass*, 2(1):204–222, 2008. [Cited on pages 8 and 77.]
 - [106] Xinyu Huang, Dongming Chen, Tao Ren, and Dongqi Wang. A survey of community detection methods in multilayer networks. *Data Mining and Knowledge Discovery*, 35(1):1–45, 2021. [Cited on page 5.]
 - [107] R Robert Huckfeldt. Social contexts, social networks, and urban neighborhoods: Environmental constraints on friendship choice. *American journal of Sociology*, 89(3):651–669, 1983. [Cited on page 13.]
 - [108] Daniel J Isenberg. Group polarization: A critical review and meta-analysis. *Journal of personality and social psychology*, 50(6):1141, 1986. [Cited on page 77.]
 - [109] Shanto Iyengar, Gaurav Sood, and Yphtach Lelkes. Affect, not ideologya social identity perspective on polarization. *Public opinion quarterly*, 76(3):405–431, 2012. [Cited on page 77.]
 - [110] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, 2007. [Cited on page 79.]
 - [111] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. [Cited on pages 8 and 22.]
 - [112] Duncan MacRae Jr. *Dimensions of Congressional Voting: A Statistical Study of the House of Representatives in the Eighty-First Congress*. University of California Press, 1958. [Cited on page 60.]
 - [113] Ankit Kariyaa, Simon Rundé, Hendrik Heuer, Andreas Jungherr, and Johannes Schöning. The role of flag emoji in online political communication. *Social Science Computer Review*, 40(2):367–387, 2022. [Cited on pages 78 and 91.]
 - [114] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. [Cited on page 40.]
 - [115] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927. [Cited on pages 140 and 156.]
 - [116] Jihie Kim and Jaebong Yoo. Role of sentiment in message propagation: Reply vs.

- retweet behavior in political communication. In *2012 international conference on social informatics*, pages 131–136. IEEE, 2012. [Cited on page 100.]
- [117] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [Cited on page 28.]
 - [118] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. [Cited on pages 8 and 17.]
 - [119] Dirk Koschützki, Katharina Anna Lehmann, Leon Peeters, Stefan Richter, Dagmar Tenfelde-Podehl, and Oliver Zlotowski. Centrality indices. In *Network analysis*, pages 16–61. Springer, 2005. [Cited on pages 83, 134, and 139.]
 - [120] David Krackhardt. Cognitive social structures. *Social networks*, 9(2):109–134, 1987. [Cited on page 55.]
 - [121] Ann E Krause, Kenneth A Frank, Doran M Mason, Robert E Ulanowicz, and William W Taylor. Compartments revealed in food-web structure. *Nature*, 426(6964):282–285, 2003. [Cited on page 133.]
 - [122] Sumeet Kumar, Ramon Villa Cox, Matthew Babcock, and Kathleen M Carley. A weakly supervised approach for classifying stance in twitter replies. *arXiv preprint arXiv:2103.07098*, 2021. [Cited on page 5.]
 - [123] Andrea Lancichinetti and Santo Fortunato. Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122, 2011. [Cited on page 79.]
 - [124] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018. [Cited on pages 4, 15, and 135.]
 - [125] Janette Lehmann, Bruno Gonçalves, José J Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 251–260, 2012. [Cited on pages 46, 49, and 50.]
 - [126] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-bigraph: A large scale graph embedding system. *Proceedings of Machine Learning and Systems*, 1:120–131, 2019. [Cited on page 17.]
 - [127] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009. [Cited on pages 6 and 150.]
 - [128] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640, 2010. [Cited on pages 134 and 136.]
 - [129] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019. [Cited on page 24.]

- [130] Jinhang Li, Giorgos Longinos, Steven Wilson, and Walid Magdy. Emoji and self-identity in twitter bios. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 199–211, 2020. [Cited on pages 78, 79, and 93.]
- [131] Kwan Hui Lim and Amitava Datta. Following the follower: Detecting communities with common interests on twitter. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 317–318, 2012. [Cited on page 79.]
- [132] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 685–694. ACM, 2008. [Cited on page 46.]
- [133] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(2):8, 2009. [Cited on page 46.]
- [134] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. [Cited on page 28.]
- [135] Thomas Magelinski and Kathleen M Carley. Community-based time segmentation from network snapshots. *Applied Network Science*, 4(1):25, 2019. [Cited on page 16.]
- [136] Thomas Magelinski, Mihovil Bartulovic, and Kathleen M Carley. Canadian federal election and hashtags that do not belong. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 161–170. Springer, 2020. [Cited on pages 7, 17, 21, and 85.]
- [137] Thomas Magelinski, Mihovil Bartulovic, and Kathleen M Carley. Measuring node contribution to community structure with modularity vitality. *IEEE Transactions on Network Science and Engineering*, 8(1):707–723, 2021. [Cited on pages 9, 84, and 85.]
- [138] Thomas Magelinski, Lynnette Hui Xian Ng, and Kathleen M Carley. A synchronized action framework for responsible detection of coordination on social media. *arXiv preprint arXiv:2105.07454*, 2021. [Cited on page 71.]
- [139] José M Marques and Dario Paez. The ‘black sheep effect’: Social categorization, rejection of ingroup deviates, and perception of group variability. *European review of social psychology*, 5(1):37–68, 1994. [Cited on page 79.]
- [140] Naoki Masuda. Immunization of networks with community structure. *New Journal of Physics*, 11(12):123018, 2009. [Cited on pages 137, 140, 143, and 148.]
- [141] Naoki Masuda and Petter Holme. Detecting sequences of system states in temporal networks. *Scientific reports*, 9(1):1–11, 2019. [Cited on pages 7, 14, 16, and 47.]
- [142] Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1119–1141, 2017. [Cited on page 47.]

- [143] Clark McCauley and Sophia Moskalenko. Mechanisms of political radicalization: Pathways toward terrorism. *Terrorism and political violence*, 20(3):415–433, 2008. [Cited on page 77.]
- [144] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pages 33–42. IEEE, 2017. [Cited on pages 5 and 28.]
- [145] Leland McInnes, John Healy, and Steve Astels. hdbSCAN: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. [Cited on pages 5 and 28.]
- [146] Shahan Ali Memon and Kathleen M Carley. Characterizing covid-19 misinformation communities using a novel twitter dataset. *arXiv preprint arXiv:2008.00791*, 2020. [Cited on page 77.]
- [147] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. [Cited on page 8.]
- [148] Fergal Monaghan and David O’Sullivan. Leveraging ontologies, context and social networks to automate photo annotation. In *Semantic Multimedia: Second International Conference on Semantic and Digital Media Technologies, SAMT 2007, Genoa, Italy, December 5-7, 2007. Proceedings 2*, pages 252–255. Springer, 2007. [Cited on page 13.]
- [149] Alfredo Jose Morales, Javier Borondo, Juan Carlos Losada, and Rosa M Benito. Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033114, 2015. [Cited on pages 79 and 94.]
- [150] Peter J Mucha and Mason A Porter. Communities in multislice voting networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(4):041108, 2010. [Cited on page 5.]
- [151] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010. [Cited on pages 55 and 61.]
- [152] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010. [Cited on page 5.]
- [153] David G Myers and Helmut Lamm. The group polarization phenomenon. *Psychological bulletin*, 83(4):602, 1976. [Cited on page 77.]
- [154] Shishir Nagaraja. The impact of unlinkability on adversarial community detection: effects and countermeasures. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 253–272. Springer, 2010. [Cited on page 142.]
- [155] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning Distributed Repre-

- sentations of Graphs. *arXiv:1707.05005 [cs]*, July 2017. URL <http://arxiv.org/abs/1707.05005>. arXiv: 1707.05005. [Cited on page 8.]
- [156] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004. [Cited on page 46.]
 - [157] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006. [Cited on pages 134, 136, 137, and 145.]
 - [158] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006. [Cited on pages 5, 82, 134, and 136.]
 - [159] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004. [Cited on page 82.]
 - [160] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002. [Cited on page 46.]
 - [161] Tam T Nguyen, Karamjit Singh, Sangam Verma, Hardik Wadhwa, Siddharth Vimal, Lalasa Dheekollu, Sheng Jie Lui, Divyansh Gupta, Dong Yang Jin, and Zha Wei. Word and graph embeddings for covid-19 retweet prediction. In *CIKM AnalytiCup 2020*, pages 5–8, 2020. [Cited on page 17.]
 - [162] Nynke MD Niezink, Tom AB Snijders, and Marijtje AJ van Duijn. No longer discrete: Modeling the dynamics of social networks and continuous behavior. *Sociological Methodology*, 49(1):295–340, 2019. [Cited on page 7.]
 - [163] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. A tale of many cities: universal patterns in human urban mobility. *PloS one*, 7(5):e37027, 2012. [Cited on page 5.]
 - [164] National Academies of Sciences Engineering and Medicine, editors. *A Decadal Survey of the Social and Behavioral Sciences: A Research Agenda for Advancing Intelligence Analysis*. The National Academies Press, Washington DC, 2019. ISBN 978-0-309-48761-0. doi: 10.17226/25335. URL <https://www.nap.edu/catalog/25335/a-decadal-survey-of-the-social-and-behavioral-sciences-a>. [Cited on page 135.]
 - [165] Miles Osborne, Saša Petrovic, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Bieber no more: First story detection using twitter and wikipedia. In *Sigir 2012 workshop on time-aware information access*, pages 16–76. Citeseer, 2012. [Cited on pages 7 and 17.]
 - [166] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. [Cited on pages 9 and 40.]
 - [167] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814–818, 2005. [Cited on page 133.]

- [168] Shimei Pan and Tao Ding. Social media-based user embedding: A literature review. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6318–6324, 7 2019. doi: 10.24963/ijcai.2019/881. [Cited on page 17.]
- [169] Arjunil Pathak, Navid Madani, and Kenneth Joseph. A method to analyze multiple social identities in twitter bios. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–35, 2021. [Cited on pages 8, 9, 78, 80, and 93.]
- [170] Leto Peel, Daniel B Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science advances*, 3(5):e1602548, 2017. [Cited on pages 55, 59, and 133.]
- [171] Tiago P Peixoto. Descriptive vs. inferential community detection: pitfalls, myths and half-truths. *arXiv preprint arXiv:2112.00183*, 2021. [Cited on pages 6 and 79.]
- [172] Tiago P Peixoto and Laetitia Gauvin. Change points, memory and epidemic spreading in temporal networks. *Scientific reports*, 8(1):1–10, 2018. [Cited on pages 7 and 16.]
- [173] Tiago P Peixoto and Martin Rosvall. Modelling sequences and temporal networks with dynamic community structures. *Nature communications*, 8(1):1–12, 2017. [Cited on pages 7 and 16.]
- [174] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 701–710, New York, New York, USA, 2014. ACM Press. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623732. URL <http://dl.acm.org/citation.cfm?doid=2623330.2623732>. [Cited on pages 8 and 17.]
- [175] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 181–189, 2010. [Cited on page 17.]
- [176] Keith T. Poole. *Spatial Models of Parliamentary Voting*. Cambridge University Press, 2005. [Cited on page 60.]
- [177] Keith T. Poole and Howard Rosenthal. A spatial model for legislative roll call analysis. *American Journal of Political Science*, 29(2):357–384, 1985. [Cited on page 60.]
- [178] Stephany Rajeh, Marinette Savonnet, Eric Leclercq, and Hocine Cherifi. Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Scientific Reports*, 11(1):1–15, 2021. [Cited on page 9.]
- [179] Stephany Rajeh, Ali Yassin, Ali Jaber, and Hocine Cherifi. Analyzing community-aware centrality measures using the linear threshold model. In *International Conference on Complex Networks and Their Applications*, pages 342–353. Springer, 2021. [Cited on page 9.]
- [180] Stephany Rajeh, Marinette Savonnet, Eric Leclercq, and Hocine Cherifi. Compara-

- tive evaluation of community-aware centrality measures. *Quality & Quantity*, pages 1–30, 2022. [Cited on page 79.]
- [181] Stephany Rajeh, Marinette Savonnet, Eric Leclercq, and Hocine Cherifi. Modularity-based backbone extraction in weighted complex networks. In *International Conference on Network Science*, pages 67–79. Springer, 2022. [Cited on page 85.]
 - [182] Scott A Reid. A self-categorization explanation for the hostile media effect. *Journal of Communication*, 62(3):381–399, 2012. [Cited on page 77.]
 - [183] Yuxiang Ren, Bo Liu, Chao Huang, Peng Dai, Liefeng Bo, and Jiawei Zhang. Heterogeneous deep graph infomax. *arXiv preprint arXiv:1911.08538*, 2019. [Cited on pages 8 and 27.]
 - [184] Juan G Restrepo, Edward Ott, and Brian R Hunt. Weighted percolation on directed networks. *Physical review letters*, 100(5):058701, 2008. [Cited on page 137.]
 - [185] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*, 2018. [Cited on page 17.]
 - [186] Alexander Robertson, Walid Magdy, and Sharon Goldwater. Self-representation on twitter using emoji skin color modifiers. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018. [Cited on page 78.]
 - [187] Alexander M Robertson and Peter Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 1998. [Cited on page 36.]
 - [188] Nick Rogers and Jason J Jones. Using twitter bios to measure changes in self-identity: Are americans defining themselves more politically over time? *Journal of Social Computing*, 2(1):1–13, 2021. [Cited on pages 78 and 93.]
 - [189] Carla Anne Roos, Sonja Utz, Namkje Koudenburg, and Tom Postmes. Diplomacy online: A case of mistaking broadcasting for dialogue. *PsyArXiv preprint*, 2022. [Cited on page 29.]
 - [190] Giulio Rossetti and Rémy Cazabet. Community discovery in dynamic networks: a survey. *ACM Computing Surveys (CSUR)*, 51(2):35, 2018. [Cited on pages 46, 54, and 56.]
 - [191] Emanuele Rossi, Henry Kenlay, Maria I Gorinova, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael Bronstein. On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features. *arXiv preprint arXiv:2111.12128*, 2021. [Cited on page 23.]
 - [192] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009. [Cited on page 80.]
 - [193] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. [Cited on pages 5 and 28.]
 - [194] Anne Schulz, Werner Wirth, and Philipp Müller. We are the people and you are fake news: A social identity approach to populist citizens’ false consensus and hostile

- media perceptions. *Communication research*, 47(2):201–226, 2020. [Cited on page 77.]
- [195] Darren M Scott, David C Novak, Lisa Aultman-Hall, and Feng Guo. Network robustness index: A new method for identifying critical links and evaluating the performance of transportation networks. *Journal of Transport Geography*, 14(3):215–227, 2006. [Cited on page 140.]
- [196] Jieun Shin, Lian Jian, Kevin Driscoll, and François Bar. The diffusion of misinformation on social media: Temporal pattern, message, and source. *Computers in Human Behavior*, 83:278–287, 2018. [Cited on pages 135 and 158.]
- [197] Kai Shu, H Russell Bernard, and Huan Liu. Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pages 43–65. Springer, 2019. [Cited on page 135.]
- [198] Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736, Online, November 2020. Association for Computational Linguistics. [Cited on page 30.]
- [199] Grigori Sidorov, Francisco Velasquez, Efstatios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860, 2014. [Cited on page 36.]
- [200] Lauren Reichart Smith and Kenny D Smith. Identity in twitter’s hashtag culture: A sport-media-consumption case study. *International Journal of Sport Communication*, 5(4):539–557, 2012. [Cited on pages 78 and 93.]
- [201] Tom A. B. Snijders. The Statistical Evaluation of Social Network Dynamics. *Sociological Methodology*, 31(1):361–395, January 2001. ISSN 0081-1750, 1467-9531. doi: 10.1111/0081-1750.00099. URL <http://doi.wiley.com/10.1111/0081-1750.00099>. [Cited on page 7.]
- [202] Tom AB Snijders, Gerhard G Van de Bunt, and Christian EG Steglich. Introduction to stochastic actor-based models for network dynamics. *Social networks*, 32(1):44–60, 2010. [Cited on page 7.]
- [203] PK SriJith, Mark Hepple, Kalina Bontcheva, and Daniel Preotiuc-Pietro. Sub-story detection in twitter with hierarchical dirichlet processes. *Information Processing & Management*, 53(4):989–1003, 2017. [Cited on pages 7, 14, 17, 19, and 45.]
- [204] Asbjørn Steinskog, Jonas Therkelsen, and Björn Gambäck. Twitter topic modeling by tweet aggregation. In *Proceedings of the 21st nordic conference on computational linguistics*, pages 77–86, 2017. [Cited on page 23.]
- [205] Leo G Stewart, Ahmer Arif, and Kate Starbird. Examining trolls and polarization with a retweet network. In *Proc. ACM WSDM, workshop on misinformation and misbehavior mining on the web*, volume 70, 2018. [Cited on page 5.]

- [206] Jimeng Sun, Christos Faloutsos, Christos Faloutsos, Spiros Papadimitriou, and Philip S Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696. ACM, 2007. [Cited on page 58.]
- [207] Henri Tajfel. Social identity and intergroup behaviour. *Social science information*, 13(2):65–93, 1974. [Cited on pages 77 and 94.]
- [208] Lei Tang, Huan Liu, Jianping Zhang, and Zohreh Nazeri. Community evolution in dynamic multi-mode networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, page 677, Las Vegas, Nevada, USA, 2008. ACM Press. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401972. URL <http://dl.acm.org/citation.cfm?doid=1401890.1401972>. [Cited on page 7.]
- [209] Dane Taylor, Rajmonda S Caceres, and Peter J Mucha. Super-resolution community detection for layer-aggregated multilayer networks. *Physical Review X*, 7(3):031056, 2017. [Cited on page 47.]
- [210] Sean J Taylor, Lev Muchnik, Madhav Kumar, and Sinan Aral. Identity effects in social media. *Nature Human Behaviour*, pages 1–11, 2022. [Cited on page 94.]
- [211] Robert Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953. [Cited on page 28.]
- [212] Petter Törnberg. How digital media drive affective polarization through partisan sorting. *Proceedings of the National Academy of Sciences*, 119(42):e2207159119, 2022. [Cited on page 95.]
- [213] Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019. [Cited on pages 5, 79, 134, and 137.]
- [214] John C Turner. Towards a cognitive redefinition of the social group. In *Research Colloquium on Social Identity of the European Laboratory of Social Psychology, Dec, 1978, Université de Haute Bretagne, Rennes, France; This chapter is a revised version of a paper first presented at the aforementioned colloquium*. Psychology Press, 2010. [Cited on pages 6 and 93.]
- [215] John C Turner and Katherine J Reynolds. Self-categorization theory. *Handbook of theories in social psychology*, 2(1):399–417, 2011. [Cited on page 8.]
- [216] John C Turner, Michael A Hogg, Penelope J Oakes, Stephen D Reicher, and Margaret S Wetherell. *Rediscovering the social group: A self-categorization theory*. basil Blackwell, 1987. [Cited on page 8.]
- [217] John C Turner, Katherine J Reynolds, et al. A self-categorization theory. *Rediscovering the social group: A self-categorization theory*, 1987. [Cited on pages 77 and 80.]
- [218] Joshua Uyheng, Thomas Magelinski, Ramon Villa-Cox, Christine Sowa, and Kathleen M Carley. Interoperable pipelines for social cyber-security: assessing twitter

- information operations during nato trident juncture 2018. *Computational and Mathematical Organization Theory*, pages 1–19, 2019. [Cited on pages 5, 15, and 107.]
- [219] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. [Cited on page 30.]
- [220] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. [Cited on pages 8 and 27.]
- [221] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019. [Cited on pages 8, 17, and 27.]
- [222] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015. [Cited on page 27.]
- [223] Maximilian Walther and Michael Kaisser. Geo-spatial event detection in the twitter stream. In *European conference on information retrieval*, pages 356–367. Springer, 2013. [Cited on pages 7, 17, and 45.]
- [224] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural Deep Network Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 1225–1234, San Francisco, California, USA, 2016. ACM Press. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939753. URL <http://dl.acm.org/citation.cfm?doid=2939672.2939753>. [Cited on pages 8 and 18.]
- [225] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 697–702. IEEE, 2007. [Cited on page 36.]
- [226] Yuan Wang, Jie Liu, Jishi Qu, Yalou Huang, Jimeng Chen, and Xia Feng. Hashtag graph based topic model for tweet mining. In *2014 IEEE International Conference on Data Mining*, pages 1025–1030. IEEE, 2014. [Cited on pages 7, 17, and 21.]
- [227] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2):139–147, 2018. [Cited on page 142.]
- [228] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*. Cambridge university press, 1994. [Cited on pages 5 and 9.]
- [229] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *Proceedings of the international AAAI conference on web and social media*, volume 5, pages 401–408, 2011. [Cited on page 45.]
- [230] Cynthia Weston, Terry Gandell, Jacinthe Beauchamp, Lynn McAlpine, Carol Wiseman, and Cathy Beauchamp. Analyzing interview data: The development and evolution of a coding system. *Qualitative sociology*, 24:381–400, 2001. [Cited on page 18.]
- [231] Magdalena Wojcieszak, Andreu Casas, Xudong Yu, Jonathan Nagler, and Joshua A

- Tucker. Most users do not follow political elites on twitter; those who do show overwhelming preferences for ideological congruity. *Science Advances*, 8(39):eabn9418, 2022. [Cited on page 94.]
- [232] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):1–35, 2013. [Cited on page 6.]
- [233] Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020. [Cited on page 24.]
- [234] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a non-negative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596, 2013. [Cited on page 6.]
- [235] Jiang Yang and Scott Counts. Predicting the speed, scale, and range of information diffusion in twitter. In *fourth international AAAI conference on weblogs and social media*, 2010. [Cited on page 79.]
- [236] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what@ you# tag: does the dual role affect hashtag adoption? In *Proceedings of the 21st international conference on World Wide Web*, pages 261–270, 2012. [Cited on pages 21 and 78.]
- [237] Michael Miller Yoder, Qinlan Shen, Yansen Wang, Alex Coda, Yunseok Jang, Yale Song, Kapil Thadani, and Carolyn P Rosé. Phans, stans and cishets: Self-presentation effects on content propagation in tumblr. In *12th ACM Conference on Web Science*, pages 39–48, 2020. [Cited on pages 9 and 78.]
- [238] Erin York Cornwell and Rachel L Behler. Urbanism, neighborhood context, and social networks. *City & Community*, 14(3):311–335, 2015. [Cited on page 13.]
- [239] Yini Zhang, Fan Chen, and Karl Rohe. Social media public opinion as flocks in a murmuration: Conceptualizing and measuring opinion expression on social media. *Journal of Computer-Mediated Communication*, 27(1):zmab021, 2022. [Cited on pages 79 and 93.]
- [240] Yong Zhuang and Osman Yağan. Information propagation in clustered multilayer networks. *IEEE Transactions on Network Science and Engineering*, 3(4):211–224, 2016. [Cited on page 6.]
- [241] Yong Zhuang and Osman Yagan. A vector threshold model for the simultaneous spread of correlated influence. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019. [Cited on page 8.]
- [242] Yong Zhuang, Alex Arenas, and Osman Yağan. Clustering determines the dynamics of complex contagions in multiplex networks. *Physical Review E*, 95(1):012312, 2017. [Cited on page 6.]
- [243] Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages

2105–2114, 2016. [Cited on pages 7 and 16.]