Title
StoryMiner: An Automated and Scalable Framework for Story Analysis and Detection from Social Media

Author
Shahbazi, Behnam

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

StoryMiner: An Automated and Scalable Framework for Story Analysis and Detection from Social Media

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Behnam Shahbazi

2019

2019

ABSTRACT OF THE DISSERTATION

StoryMiner: An Automated and Scalable Framework for Story Analysis and Detection from Social Media

by

Behnam Shahbazi

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2019

Professor Vwani P. Roychowdhury, Co-Chair

Professor Douglas S. Parker, Co-Chair

The explosive growth of social media over the past decade, together with advancements in computational power, has paved the way for many large-scale sociological studies, which were not possible before. Social media sites are now the primary source of data for much of our insights into society, from trending topics to behavioral patterns of various groups such as online shoppers or political parties. One particular area of interest is the analysis of events and interactions through their descriptions in social media posts. Inferring and analyzing real-world events from social media in a large-scale automated way provides a platform for understanding real-world stories, which are not only influenced by but also heavily impact public opinion. Therefore, it is necessary to design computational and statistical tools to automatically extract social media stories. In this dissertation, we introduce StoryMiner, an automated and scalable machine learning framework rooted in a narrative theory that identifies and tracks multi-scale narrative structures from large-scale social media text.

Predicating our work on narrative theory, StoryMiner derives stories and narrative structures by automatically 1) extracting and co-referencing the actants (entities such as people and objects) and their relationships from the text by proposing an Open Information Extraction system, 2) assigning named-entity types and importance scores for entities and relationships using character-level neural language architectures and other traditional ma-

chine learning models, 3) making use of context-dependent word embeddings to aggregate actant-relationships and form contextual story graphs in which the nodes are the actants and the edges are their relationships, and 4) enriching the story graphs with additional layers of information such as sentiments or sequence orders of relationships.

StoryMiner allows academic and industry researchers to extract structured knowledge from unstructured text to inform practical decisions. To exhibit the benefits of our framework, among the many possible applications we showcase three major use cases: identification of differences in narrative structures between fake and real conspiracies, summarization of user product opinions from tweets, and reconstruction of plot summaries of famous novels from reader reviews on social reading sites such as Goodreads.

The dissertation of Behnam Shahbazi is approved.

Quanquan Gu

Ramin Ramezani

Timothy R. Tangherlini

Douglas S. Parker, Committee Co-Chair

Vwani P. Roychowdhury, Committee Co-Chair

University of California, Los Angeles

2019

To my parents, Nane Mahnaz and Baba Hamid . . .

To my brother, Ali Dadash . . .

To my love, Leili Joon . . .

To my beloved family, Shahbazi, Edalat-Nejad, and Tavabi . . .

To my many loving friends . . .

And to my favorite people who care for others and bring smiles on each others' faces . . .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am grateful to many people who have made my Ph.D. studies possible. I would like to thank my advisors, Professor Vwani Roychowdhury and Professor Stott Parker for their continuous guidance and mentorship. This journey would not have been possible without the thorough and fruitful discussions and guidance of Professor Roychowdhury and the continuous help, advice, and support of Professor Parker. I could not have imagined having better mentors for my studies at UCLA.

I would like to thank Professor Timothy Tangherlini, who brought invaluable perspective from Humanities to this work and whose insights, writings and feedback enriched this work. His knowledge and advice made major contributions to the direction of this research.

I would like to thank the rest of my doctoral committee: Professor Quanquan Gu, and Professor Ramin Ramezani. I am grateful for their support and efforts put into this work.

My sincerest gratitude to my fellow lab-mates for their contributions in developing this work: Shadi Shahsavari, Ehsan Ebrahimzadeh, Misagh Falahi, Arash Vahaabpour, and Shijun Lu. I acknowledge Kaicheng Wang for his contributions in developing the demo site of this work.

Furthermore, I am eternally indebted to my loving parents, the greatest brother, and the rest of our awesome family. Thank you for your immeasurable love and support. Words cannot express how grateful I am for what you have done for me.

I would also like to thank my friends: Leili Tavabi, Nazgol Tavabi, Amirhossein Reisizadeh, Farzan Oroumiyeh, Hamed Nilchi, Mehrdad Showkatbakhsh, Hossein Jahandideh, Babak Moatamed, Nazanin Farahpour, “14+3” group, ZiziGooloo team and many other colleagues, too many to list here, for all of their help. Thanks for making this road smoother for me.

Finally, my beloved committee member, Professor Mario Gerla, passed away a few months before my graduation. I am fortunate to have had the guidance and support of this kind, knowledgeable, and role model faculty member. Professor Gerla, your great breakthroughs, contributions, and kindness will always be remembered by the Computer Science community. Rest in peace.

VITA

2013	B.S. (Computer Engineering), Sharif University of Technology, Iran.
2014	Research Assistant, Embedded and Reconfigurable Systems Lab, Computer Science Department, UCLA.
2014	Teaching Assistant, Computer Science Department, UCLA. (Intro. C++ (CS31), Verilog Programming (CS152A), Logic Design (CSM51A))
2015	M.S. (Computer Science), Computer Science Department, University of California, Los Angeles.
2015	Data Scientist PhD Intern, PayPal Inc.
2015	Research Assistant, Big Data and Complex Network Group, Electrical and Computer Engineering Department, UCLA.
2016	Data Scientist PhD Intern, Adobe Systems Inc.
2017	Teaching Associate, Computer Science Department, UCLA (Adv. C++ (CS32), Verilog Programming (CS152A))
2018	Data Scientist PhD Intern, Factual Inc.
2018	Teaching Fellow, Computer Science Department, UCLA (Intro. C++ (CS31), Online Machine Learning and Big Data Course (ECE219))
2019	PhD Research Intern, Microsoft Research Lab.

CHAPTER 1

Introduction

> "The Knowledge Principle: If a program is to perform a complex task well, it must know a great deal about the world in which it operates."
>
> - Lenat & Feigenbaum

1.1 Overview

Social media has become a prominent part of our daily lives and a gigantic source of information keeping track of every-day real-world events. It is a comprehensive reflection of people's observations and opinions on concepts such as political views and product reviews.

The staggering amount of textual data produced daily offers an opportunity to understand and predict collective perceptions and real-world events.

Each social media post is driven by a narrative and can shed light on part of the story from a user's perspective. Synthesizing these posts reveals the holistic view of the underlying narrative and provides insights for decision-making procedures for various groups, from researchers to politicians, marketers or societal leaders. For example, because of the efficiencies of narrative to shape and influence communities and their ideological orientation, stories can be deployed as a means for undermining democratic institutions or for enticing people to join particular parties.

Detecting the emergence and circulation of stories in everyday life is complicated by the fragmentary nature of most storytelling. On social media people tend to shorten, interrupt, or refer to much larger, more elaborate narratives, making it difficult to extract actionable information. Consequently, policy makers and other institutions charged with the safe-guarding of civil society often miss the early emergence of narratives that influence people's real-world behavior. While stories are fragmentary when communicated, aggregating these fragments can provide a clear picture. Thus, it is necessary to develop computational and statistical tools to automatically transfer the unstructured observations to a unified and condensed narrative.

In this dissertation, we design and develop a set of machine learning models to process large datasets of fragmentary posts to reconstruct the narrative structures driving these conversations. We introduce StoryMiner, an automated and scalable story detection framework that processes textual data to form a *story graph*[1]. The story graphs summarize the information in the text and present the narratives as networks in which nodes represent the *actants* (characters) and edges represent their relationships.

This work provides researchers from different backgrounds a platform for automatically discovering narratives (in various granularity levels) in large-scale textual documents. The story graph can be enriched with additional layers of information depending on the nature of the input text and the research questions. For example, researchers can incorporate timestamps when they are interested in how a story evolves over time. Similarly, researchers interested in views prevalent among the general public can enrich the result of StoryMiner with sentiment analysis. Such additional analyses are further described throughout this dissertation within three applications: identifying the characteristics and structures of a fake story that distinguishes it from a real story, discovering consensus from product-centric tweets, and reconstructing plot summaries of famous novels from reader reviews on Goodreads.

[1] Italic terms are defined in the glossary.

1.2 Motivation

This dissertation has been established based on multiple motivations which can be categorized into two main themes. As described earlier, the main theme of this research is to introduce a narrative framework which is capable of identifying and representing narrative structures and contents from a large corpus of text. For example, our entity-relationship extraction components reveal the main characters and their interactions in sources such as news, online reviews, or even medical reports describing disease-medicine interactions in unstructured writings. The second motivation behind this work is to view social media posts and stories as the output of a generative model which expresses human thoughts in the form of writing, then, accordingly comes up with statistical models to estimate this hidden generative model by analyzing its visible textual outputs.

Understanding the inner process of human thought has historically been of great interest. Despite the extensive research conducted on this subject, there are still unresolved questions on how we think and how we express a thought. In this thesis, work done by humanities researchers motivates the formalization of our narrative model. It estimates the human thought process by predicting its outcome in forms of textual writings.

We model the conversion of a thought into a sentence through subjects (entities) and the relations between them. For instance, in figure 1.1 the sentences “Donald Trump wins the election, and is now the president of the U.S.” and “The election results are in and Donald Trump is the next president of the U.S.” both share a unique thought process where the entities are “Trump”, “election”, and “U.S.”, and the relationship between them is “wins” and “is president”. Social media with a rich source of textual information provides a plethora of sentences which are considered the “output” of our model. We use these outputs to construct the underlying story graph containing entities and relationships. Using the constructed graph, we make inferences about entities’ relations, which may evolve over time. Such variations allow us to infer certain structures and correlations among entities which could be useful in many applications and decision making processes. Question & Answering websites, semantic search engines, and a text summarization platforms are among our

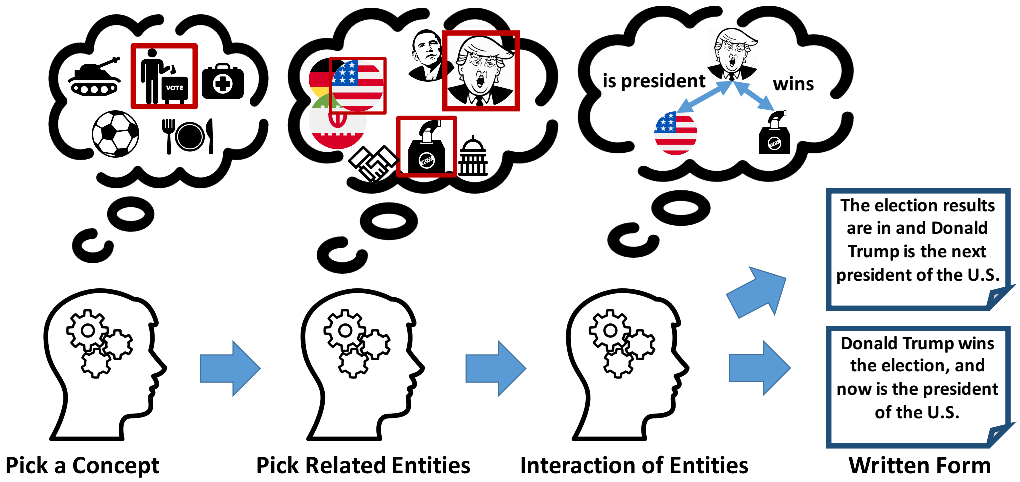

Figure 1.1: The thought process in this scenario is 1) Among the many concepts in mind, the election concept is considered here. 2) There are several participants and entities involved in this concept, such as alternative candidates, countries, etc. The entities involved in the scenario are selected. 3) Relations are shaped between entities and 4) Relationships among entities are verbalized using proper syntax. The result could be "Donald Trump wins the election, and now is the president of the U.S." or in one of many other forms such as "The election results are in and Donald Trump is the next president of the U.S.".

platform's applications. Therefore, we aim to provide this model in a formally representable design. To this end we need to overcome different challenges such as the fact that a single thought process can be verbalized in many different forms (Fig 1.1). In Chapter 3, we discuss the detailed modeling approaches we used to overcome such challenges and formally introduce our framework.

1.3 Summary of Contributions and Organization

In this thesis we develop StoryMiner, a machine learning framework for identifying narratives from social media text. StoryMiner consists of a set of novel approaches for relation extraction, actant discovery, and story graph construction. Namely, main contributions of this work are as follows:

- A sentence-level relation extraction system called **StoryMiner RelEx** that is particularized for story-specific relationships. It achieves comparable results with the state-of-the-art relation extraction systems in general domains (see Section 3.3). StoryMiner RelEx, however, is the only extraction system that is specifically designed to retrieve story-specific relationships and thus, offers an additional set of novel procedures compared to the common relation extraction methods. For example, StoryMiner RelEx a) simultaneously couples sentence-level relation extraction and paragraph-level co-reference resolution to resolve pronoun arguments to the nouns they refer to, b) uses argument headwords and dependency tree information to map arguments to actants - the nodes in story graphs, - and c) breaks down n-ary relationships into pairwise relationships to form edges in story graphs (see Sections 3.2.1, 4.5, 5.3.3.1).

- A **hierarchical actant model** to partition entities into hierarchical groups with similar contextual roles based on context-dependent word embeddings. In our group, we originally proposed an embedding approach based on explicit factorization of suitably generated entity-relation matrices along with a new exterior point method to solve the factorization problem. Our approach demonstrated superior clustering performance over embeddings obtained by the optimal matrix completion approach based on SVD (see [1] for more information). However, over the course of this dissertation, we further utilize our work with the state-of-the-art context-dependent word embeddings such as BERT and Flair [2, 3]. We further propose models and algorithms to learn actants hierarchy by clustering the context-dependent embeddings. The hierarchical actant model offers a novel way to identify narratives in various granularity levels, ranging from a broad story to a more specific one (see Sections 3.2.1, 4.5).

- A **Story Model** to represent narratives in the form of networks (aka story graphs). Story graphs reveal narratives, narrative structure, the sequence orders of relationships, and other fundamental aspects of a narrative. Performing graph theory techniques on story graphs provides meaningful interpretations and results. For instance, ego networks or networks comprising a target set of actants reveal sub-stories surrounding certain actants, and graph connectivity distinguishes between a fake and a real news (see Chapters 3 and 4).

- A **fake news detection and summarization system**. Story Model captures useful information about real news, and fake news is often characteristically different in this model. Specifically, in real news the concepts in the narrative are more connected, whereas in fake news - because of the way people construct it - the concepts tend to be less connected. People cook up stories in parts, glue them together, and align otherwise unrelated domains of human interaction. Chapter 4 discusses our experimental results on Pizzagate and Bridgegate, a fake and a real news. We not only discover what was mentioned in summaries published by online newspapers, but we additionally identify the distinction in their narrative structures. (see Chapter 4).

- A **machine learning framework (aka StoryMiner) for story narrative detection** from text. This framework will be publicly available on GitHub repositories as well as on a demo webpage. StoryMiner achieves empirically powerful results in detecting stories from fragmentary posts. For instance, it automatically retrieves 93% and 97% of story plots from two famous novels - Of Mice and Men and To Kill a Mockingbird, respectively - from online reader reviews. The accuracy and effectiveness of StoryMiner have been verified via a set of computational experiments. Depending on the nature of the input text and the research questions, StoryMiner offers additional analysis techniques. For example in Chapter 5, StoryMiner summarizes user experiences with contact-less payment methods from tweets. Thus, it develops classification models to detect the type of an entity and a relationship and performs sentiment analysis to monitor views prevalent among the general public opinion (see Chapters 4 to 7).

This dissertation is organized as follows: Chapter 1 describes an overview of this thesis, along with its motivations. Chapter 2 discusses the terminology used in this work as well as related work. In Chapter 3, we present our framework in a formal definition and explain its underlying models and methodologies. Chapter 4 introduces our framework in detail and shows its powerful discoveries from analyzing the narrative structural differences between fake and real conspiracies. Chapter 5 demonstrates how our framework can couple with sentiment analysis to reveal facts and opinions about user experiences with contact-less payment methods from a corpus of transactional tweets. In Chapter 6, we show the success of our system in the plot reconstruction of four best-selling fictional novels from Goodreads reviews. Lastly, in Chapter 7, we conclude with a discussion of our demo page and the future directions this research could take.

CHAPTER 2

Background

> "The whole is more than the sum of the parts."
>
> \- Aristotle

2.1 Terminology

In this section we introduce some terminology to assist with understanding commonly used terms and concepts described in this thesis.

- **Entity**: An entity is a real-world object such as a person or a place. Entities can also be an abstract concept such as an email or a fictional character in a novel. In the context of this dissertation, we are interested in those entities that are part of a story narrative.
- **Named Entity**: A named entity is an entity, such as a person, location, organization, etc., which can be denoted with a proper name and can be classified into a set of predefined entity classes (persons, locations, etc). Table 2.1 shows common named entity classes used in this dissertation along with an example for each class.
- **Entity Mention**: An entity mention, aka mention or reference, is a token span surfaced in the text that refers to an entity. In the sentence "Obama was the president of the United States", the word "Obama" is a mention of the real-world named entity "Barack Obama". Entity mentions are likely to appear in certain syntactic forms, such as the subject or object of the sentences in text corpora. We study the structure and

Named Entity Type	Definition	Example
PERSON	People, including fictional	Chris Christie
ORG	Companies, agencies, institutions	Port Authority
GPE	Countries, cities or states	Fort Lee
FAC	Facilities such as buildings, airports, bridges	GW Bridge
NORP	Nationalities or religious or political groups	Republican party
LOC	Non GPE locations such as bodies of water	Hudson River
EVENT	Named storms, wars, sports events	Hurricane Sandy
OTHER	Miscellaneous	Email

Table 2.1: Named Entity Types/Classes, their definition and an example for each type

parsing of sentences to retrieve such mentions. A challenging task is to map mentions to entities, and in this work unless we specify, we assume that mentions with the same surface word refer to the same entity. This assumption may not be true in certain cases where a single surface word maps to multiple entities. For example, the surface word "Obama" in a context could refer to either "Barack Obama" or "Michelle Obama". In each application chapter, we further discuss how we disambiguate such cases.

- **Binary Relationship**: A binary relationship, aka relationship, pairwise relationship, or relation tuple, refers to the interaction between two entities (or mentions). In this dissertation "relationship" mainly refers to the actual wordings that expresses the connection between two entities. For example, the verb phrase "studies at" in the sentence "Behnam studies at UCLA" which relates the subject "Behnam" to the object "UCLA". We represent this relationship in this dissertation as triples: e.g. (Behnam, studies at, UCLA).

- **Multi-way Relationship**: A multi-way relationship, aka n-ary relationships, or multi-actant hyper-edges, expresses a connection between more than two entities. For example, in the sentence "parents use religions to get exemptions" there is a multi-way relationship of "XX using YY to get ZZ" between parents, religions, and exemption.

According to narrative theory, multi-way relationships can be broken down into pairwise relationships. For instance, our example could be broken down into (parents, use, religions), (parents, get, exemption), (religions, are used for getting, exemption). In this work, we use syntax structures to break down multi-way relationships into pairwise relationships that are fundamental components of our proposed story model (refer to Chapter 3). The pairwise relationships can be further link together to recover the possible loss of information in this breaking down process. In Chapter 6 we discuss a possible solution to link relationships and consequently infer their sequencing. This is, however, an on-going future direction of this work.

- **Argument**: An argument, aka arg1 or arg2, refers to either of the two entity mentions of a relationship. For instance, in the relationship (parents, use, religious), "parent" is the first argument (or arg1), "religious" is the second argument (or arg2), and "use" is the relationship (relationship phrase, rel, or verb) connecting them.

- **Head-word**: A head-word is the main word (token) in an argument or relationship phrase. For example in the extraction (John, had, a severe accident), the headwords are "John", "had", and "accident" for arg1, rel, and arg2, respectively. We annotate headwords by surrounding them with braces. Thus, with headword annotations, our example relationship becomes ({John}, {had}, a severe {accident}).

- **Attribute**: An attribute is an observed property of an individual mention, for example the words that co-occur with a mention in the same argument. For example, in the argument "President Obama", Obama is the headword and president is its attribute. These attributes can be used as complementary information about entities or relationships and are often useful for disambiguation. For instance, the headword mention of "Obama" in the arguments "Barack Obama" and "Michelle Obama" can be resolved given the additional attributes.

- **Actant**: An actant refers to an entity (e.g. single character) or a group of entities (e.g. collection of characters) that serve the same or similar role in the setting we study and have close ties between themselves. In this work, actants allow us to retrieve multi-scale narratives with various granularity levels of abstraction. For example when analyzing the Pizzagate scandal, different people might have equivalent roles in certain contexts. Thus, although Tony Podesta and John Podesta are two different entities, in certain cases, they play similar roles in our analysis and therefore they could be grouped under the same actant - “Podestas”. In this dissertation, actants could be defined as pre-defined groups or they could be automatically learned from the data. In case of automated actant discovery, we define two concepts: Superactants (or supernodes) and Subactants (or subnodes) which are described as follow.

- **Superactant/Subactant**: A superactant, aka supernode, refers to an automated grouping of highly related arguments which are centered on a single actant. For example, arguments such as “Hillary”, “Hillary Clinton Foundation”, “Hillary Supporters” are such arguments centered around Hillary and they form a superactant. They include different attributes of Hillary Clinton such as her supporters or foundation, which we refer to as subactants. In Chapter 3, we further discuss how we use contextualized embeddings to automatically break down a supernode to its set of semantically contextualized subactants (or subnodes).

- **Story Graph**: A story graph, aka story narrative or summarization network, refers to a contextual network in which nodes represent the main actants (or superactants) and edges represent the relationships between them. Story graphs explain and reveal the story surrounding its actants in a simple/sparse network. Figure 2.1 is an example story graph, revealing the story behind the Bridgegate scandal which is described in Chapter 4. Various visualization techniques and representation models are offered in this dissertation to demonstrate these networks in easily interpretable layouts.

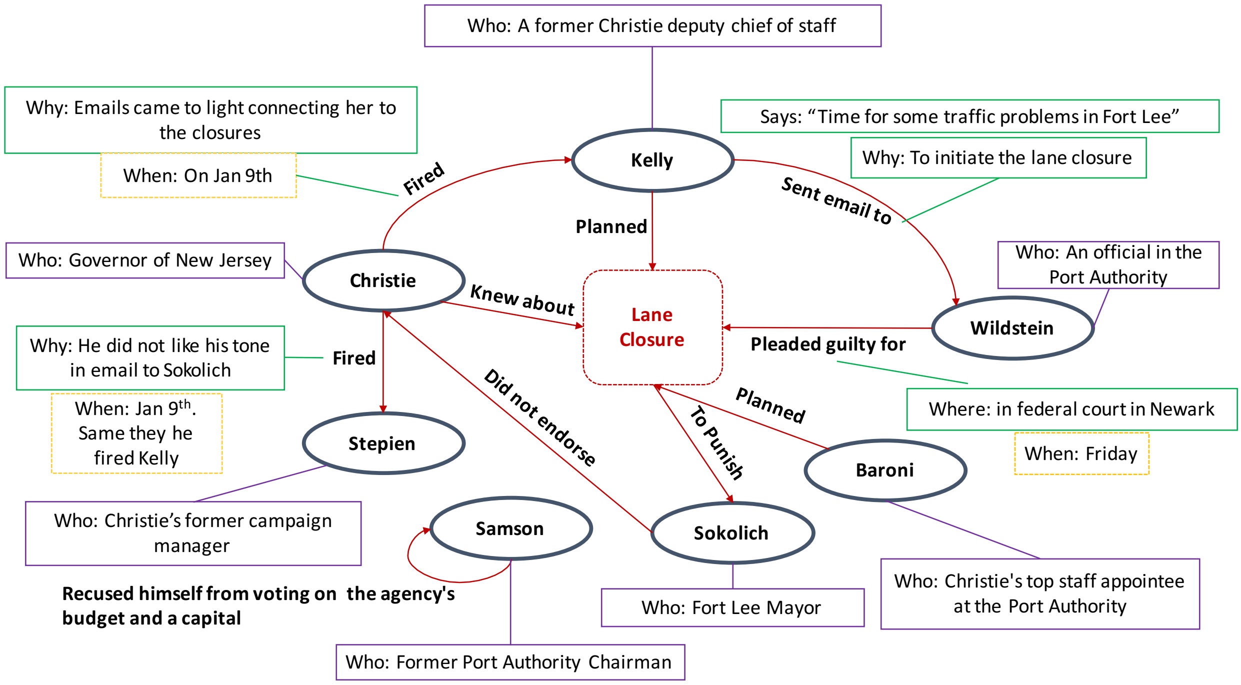

Figure 2.1: An example story graph, retrieved from news articles about the Bridgegate conspiracy. The Bridgegate conspiracy was discovered by investigative reporters to be a political payback operation launched by the inner circle of New Jersey Governor Chris Christie taking aim at the Democratic mayor of Fort Lee, New Jersey, Mark Sokolich, who had refused to endorse the governor in his reelection bid. Christie's assistants conspired with members of the Port Authority to close several toll lanes to the George Washington bridge, thereby causing catastrophic traffic jams that lasted for a week in early September 2013. When asked, these assistants said that the lane closures were part of a traffic study. A formal investigation into the decision to close the lanes was launched in 2014 and, during the ensuing five years, the overall contours of the conspiracy were revealed and various actors were indicted, tried and sentenced to prison.

2.2 StoryMiner: Pipeline and Literature Review

In this section, we discuss the StoryMiner's pipeline to provide a high-level picture of how we transfer unstructured text into structured summarization networks, aka story graphs. We further discuss the related works to StoryMiner.

The pipeline of StoryMiner is demonstrated in Figure 2.2. Rooted in narrative theory, StoryMiner derives story graphs by:

- Extracting and co-referencing the actants and their relationships from the text by proposing an Open Information Extraction system.
- Assigning named-entity types and importance scores for entities/actants and relationships using character-level neural language architectures and other traditional machine learning models.
- Making use of context-dependent word embeddings to aggregate actant-relationships and generate story graphs.

Depending on the nature of the input data as well as the research questions, story graphs can be enriched with additional layers of information, such as sentiment or sequence ordering of their relationships, or analysis of their network structures.

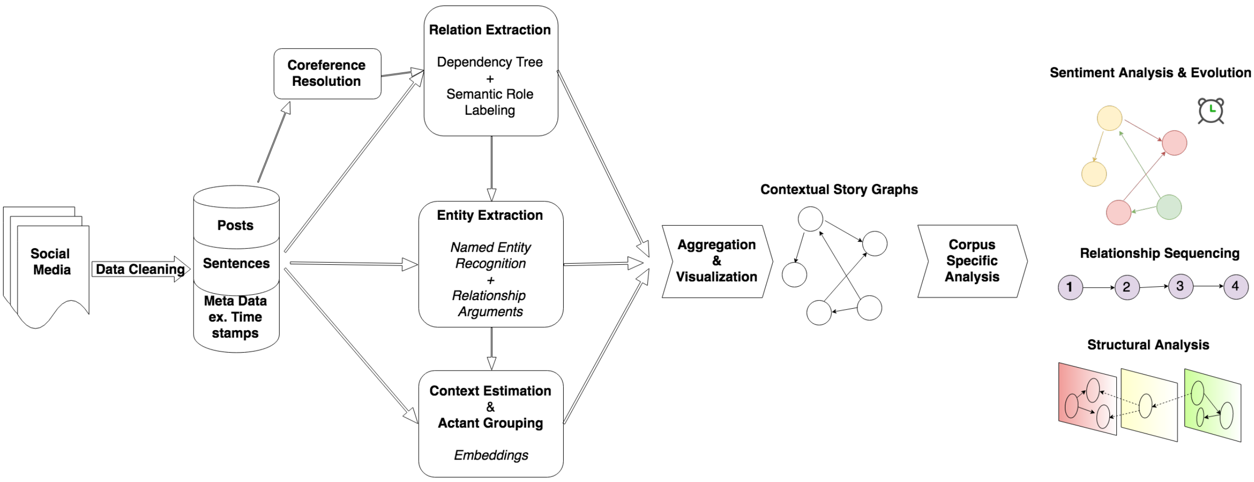

Figure 2.2: StoryMiner's pipeline. After data preprocessing and co-reference resolution, entities and relationships are extracted and grouped together to form contextual story graphs. The story graphs can be enriched with additional layers of information, such as sentiment or sequence ordering of their relationships, or analysis of their network structures.

The proposed pipeline consists of several research problems in different NLP domains such as Open Information Extraction, Topic Modeling, and Knowledge Base Construction. We first review the relevant literature before describing in detail how our framework is different from other works in Chapters 4 - 6.

- **Open Information Extraction:** Open Information Extraction (Open IE), well-studied within Natural Language Processing, turns the unstructured information expressed in natural language text into a structured representation by extracting entities and their relationships [4, 5, 6]. The common approaches are based on syntactic structures [5], dependency tree structures [4, 7, 8], or Semantic Role Labeling [9]. Some researchers use knowledge bases like Freebase [10], DPpedia [11], or Yago [12] to disambiguate entities and relationships [13]. While extracting entities and relationships is an integral part of our framework, the main focus of this thesis is on how to leverage Open IE to automatically extract stories' narrative structures.

- **Topic Modeling:** Topic Modeling refers to statistical modeling of latent topics that occur in a collection of documents. It is applied to various tasks such as Opinion and Aspect Mining [14, 15, 16] and Event Detection [17, 18, 19, 20]. Most attempts at summarizing large text corpora using topic modeling algorithms, including the well-known Latent Dirichlet Allocation (LDA) model, essentially extract clusters of words that are highly likely to occur together [21]. They are powerful in clustering words into contextual topics. However, they lack modeling of the interactions between entities. In this dissertation, we leverage topic model approaches to discover topics, domains, and even in early stage clustering of entities, and combine them with Open IE to identify the underlying stories in retrieved topics.

- **Knowledge Base Construction:** Knowledge Bases (KB) provide a structured representation of human knowledge helpful to various NLP domains such as search engines, question-answering tools, and recommender systems. Popular examples are DBPedia [11], Yago [12], FreeBase [10], DeepDive [22], NELL [23], and ConceptNet [24, 25]. Those graphs are often built from semi-structured knowledge, such as Wikipedia, or extracted from the web with a combination of statistical models, embedding techniques, and linguistic methods [26]. Most Knowledge graphs emphasize well-known real-wold entities (e.g. Barack Obama or Michelle Obama) with certain relationships (e.g. has_spouse) among them. A comprehensive review of knowledge graphs is available in [27].

Note that this dissertation is not an attempt to redefine and build a semantic network (in the usual sense), or an entity relationship network, such as the mentioned knowledge garphs. In such networks, the actant attributes and relationships are usually fixed and are chosen from a set of pre-defined attribute lists. Thus "Apple" is a corporate entity (hence a node) and "ApplePay" is a mobile phone product entity (another node), and they are connected by the relationship of "owned or operated by", which again is a pre-specified category. Similarly, a movie actress is connected by edges to movies she has acted in, and each such edge could be qualified by a number of attributes, such as the type of character role she has played in that part. StoryMiner primarily aims at capturing actants and their relationships that emerge under specific circumstances and situations, and are driven by the need to deal with such exigencies. That is, StoryMiner is particularly suited for representing story and narrative dynamics, where the overarching structure does not vary much, but the specific instances of the actants, their roles, and their relationships vary significantly based on the circumstances.

CHAPTER 3

StoryMiner: Models and Methods

"You are what you read."

- Esko Valtaoja

3.1 A Graphical Narrative Model for Generation of Posts and Story Fragments

We propose a generative network model, in which **actants** (people, places and objects) are the nodes, and **the relationships between pairs and groups of actants** are the edges or hyper-edges. These edges/hyper-edges are labeled with the nature of the observed relationships (e.g., based on actions, or attributes), the context of the relationships, and their likelihoods. We note that certain situations are better captured when they are represented as hyper-edges, involving multiple actants. Consider for example, the verb/action "Used" in the following sentence: "Podesta used the restaurant, Comet Pizza, to hide a ring for trafficking in children." In the Semantic Role Labeling (SRL)[28][29] parlance, the verb "Used" has at least three arguments or semantic slots: "X = Podesta = Who Uses" "Y= Comet Pizza = What is Used by X", and"Z= Hiding a ring for trafficking in children = What X Uses Y for." Thus a hyper-edge, connecting the actant nodes "Podesta," "Comet Pizza," and "Ring for trafficking in children" via the coupled semantic roles would be a sufficient representation. Such a hyper-edge and the information that it carries, however, can be expressed and represented by a set of three pairwise relationships that are coupled: 1) ("Podesta", Used, "the restaurant, Comet Pizza"); 2) ("Podesta", Hid, "ring for trafficking in children"); and 3) ("Comet Pizza", Hosted, "ring for trafficking in children"). In fact, different versions of such coupled pairwise relationships/edges can be used to represent the same hyper-edge. For the rest of this paper we will assume that the networks only have pairwise edges, and any multi-actant hyper-edge has been decomposed into a constituent set of coupled pairwise edges.

We also note that *this is **not** an attempt to redefine and build a semantic network* (in the usual sense), or an entity relationship network, such as Google's Knowledge Graph. In such networks, the actant categories or types are usually predefined, such as persons, organizations, and places. Similarly different attributes and relationships among the actants are usually chosen from a set of predefined attribute lists. For example, such semi-automated databases will have a node entry for "Hillary Clinton", along with several relationship edges with other nodes. For example, "(Lives in), (America)", "(Is a), (Politician)", and "(a member of), (the Democratic Party)", where the first argument is a relationship label and the second argument is another actant node; here, for example, "America", "Politician", and "The Democratic Party" are other nodes in the network. We make use of such databases to recognize named entities and their attributes using publicly available software platforms, such as *Flair* [3], which helps us determine the various category or knowledge domains to which the actants belong.

Our graphical models, on the other hand, are primarily aimed at capturing actants and the interactant relationships which emerge under specific circumstances and situations, and that are driven by an underlying narrative framework. They are particularly suited for representing story and narrative dynamics where the overarching structure does not vary much, but the specific instances of the actants, their roles, and their relationships vary significantly based on the circumstances. For example, a "(arg1, relationship, arg2)" of the kind "(Hillary Clinton) (runs) (a covert child trafficking ring)" will not be included in any usual semantic network *a priori*. It might get incorporated at a much later date, once the narrative has already played out. Yet it is a common narrative trope (whether true or not) to report a public figure's "skeletons in the closet." Indeed, politicians and other public figures are constantly monitored by the press and other societal institutions who are keenly interested in discovering instances of abuse and other criminal activities. As such, the "corrupt politician" is a well-known archetype. Given the domain of politics, what varies are the identities of the actants, the nature of the crimes committed, and the motivations for committing those crimes or covering up the evidence.

The specifics of a "corrupt politician" narrative need to be pieced together in real time as pieces of information (whether credible or not) are revealed.

Our computational approach to modeling story dynamics is to assume that the stories (and partial stories) are generated by an underlying domain-dependent structured model, where we use observed data to fill in the parameters of the model.

Formally, a particular narrative dynamics is characterized by an underlying set of r relationships, $R = \{R_1, R_2, \cdots, R_r\}$, and k contexts, $C = \{C_1, C_2, \cdots, C_k\}$. *These are model parameters that are either given a priori, or estimated from the data.* A context C_i is a hidden parameter, or, to a borrow a physics concept, the 'phase' of the underlying system, which defines the particular environment in which actants operate. It expresses itself in the distributions of the relationships among the actants, and is captured by a labeled and weighted network $G_{C_i}(V_{C_i}, E_{C_i})$. Here, $V_{C_i} = \{A_1, A_2, \cdots, A_n\}$, where each A_j is an actant, and has associated with it a context specific probability or weight $p_{C_i}(A_j)$ that determines the actant's likelihood of participating in the given context. The edge set E_{C_i} consists of m_{C_i} ordered pairs $e_{(C_i,j)} = (A_{j_1}, A_{j_2})$, where each such pair is labeled with a distribution over the relationship set R, $D_{(C_i,j)}(R)$.

Note that relationships are represented by categories of words (most often verbs) grouped together, where each category is comprised of verbs that imply a similar relationship. Therefore, the *(Abuse)* relationship is realized in sentences by a set of domain-specific verbs, including *(abuse, molest, rape, trafficking in)*, which connect the actants "John Podesta", "Hillary Clinton" and "James Alefentis" with the actant "Children" in the Pizzagate conspiracy theory.

Given such a model, the process of generating a social media post or a story fragment is shown in Figure 3.2. A person (user) first picks a context C_i and then samples the network $G_{C_i}(V_{C_i}, E_{C_i})$. That is, the user draws a set of actants according to the node distributions,

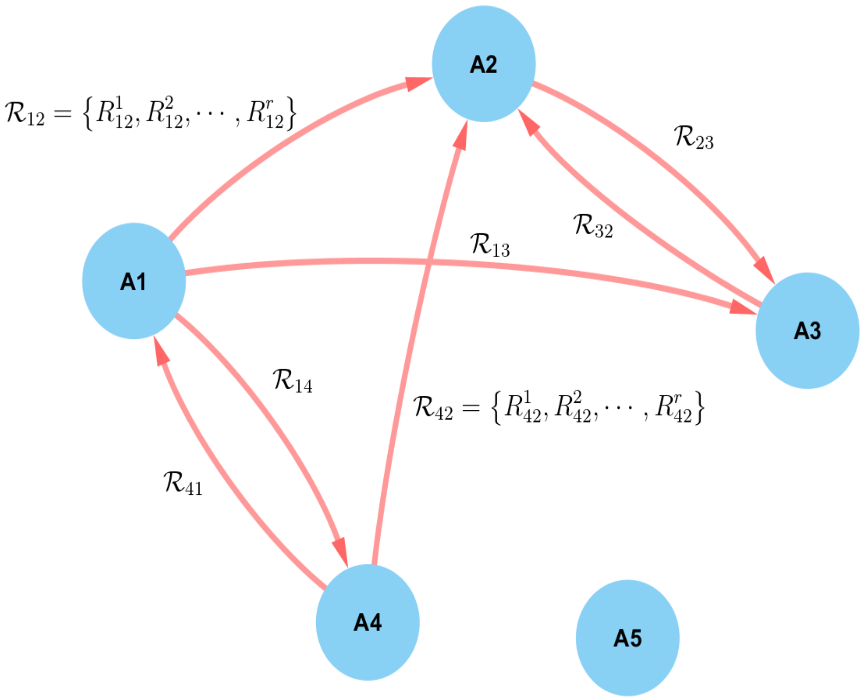

Figure 3.1: **A graphical Model of Narratives:** For a story revolving around a set of actants $A_1, ..., A_n$, the narratives can be divided into a set of contexts. In each context, the story is summarized as a set of interactions (relationships) between actants as shown in the figure. Therefore, an edge between actants A1 and A2, for example, carries a set of relationships $R_{12} = \{R_{12}^1, R_{12}^2, \cdots, R_{12}^r\}$ that exist between the two actants, and the significance of each relationship in this context. It is important to note that relationships come from not only verbs, but also other syntactic structures in the text that imply relationships.

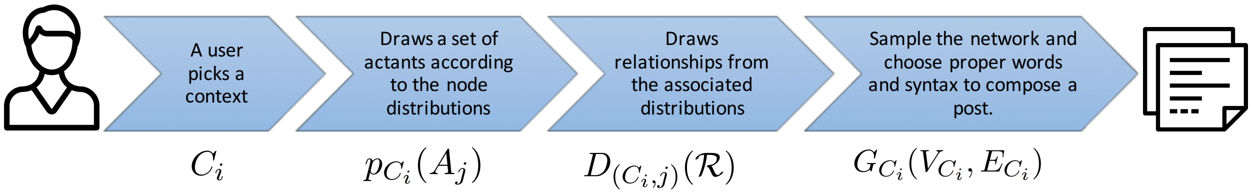

Figure 3.2: Modeling the steps a user takes to generate a social media post or a story fragment.

$p_{C_i}(A_j)$. Then the user draws for relationships from among the associated distributions $D_{(C_i,j)}(R)$. *The user then composes the post according to these outcomes, by choosing the proper words and syntax, in particular nouns for the actants, and the associated verb phrases (or other syntactical constructs) for the relationships.*

3.2 Learning Narrative Structure from Large Scale and Unstructured Text Data

Our methodology is predicated on the underlying structure of the narrative framework (as illustrated in Fig. 3.1) that captures how a narration instance emerges via a collective negotiation process. As illustrated in Fig. 3.2, each post describes relationships among only a subset of actants (which are yet not known to our automated algorithms). Others join in and share their own information, in terms of new or existing actants, that either (i) affirm or contradict information in other posts, or (ii) bring in new information revealed either through a judicial or journalistic investigation process, or via the collective intent of an endemic population who "see evidence and connections." The overall narrative is thus distributed across a series of social media posts and reports in different publications.

From a machine learning perspective, given such a generative process, we need to estimate all the hidden parameters of the model, such as actants, the set of relationships, and the edges and their labels. In other words, we have to jointly estimate all the parameters of the different layers of the model, as described in the following.

3.2.1 Joint Estimation: Actants, Contexts, and Relationships

We assume that the given corpus is a sample syntactic output of our graphical generative model. The underlying sets of actants, their semantic relationships and the contexts that determine different groups of relationships among the same actants are unknown. Thus, we need a formal data-driven function/measure to characterize these sets, so that they can be estimated from the text corpus. A functional model for actants used in this paper can be described as follows: *An Actant is a set of Noun Phrases (e.g., named entities and head words in a parse tree) that play very similar semantic roles in the corpus.* The semantic role of a noun phrase is measured by the semantic similarity of the words and phrases around it in the parsing tree. For example, (i) Phrases such as "Clinton" "Hillary" "Hillary Clinton" form one actant category because of their high frequency, both as individual "head" words, and as co-occurring words in noun-phrases. As per our intuitive definition of an actant, because they are part of the same arguments in syntactic relationships they have similar semantic roles; (ii) Phrases such as "Supporter of Clinton," "Clinton follower" and "Clinton insiders" form a distinct semantic context because of the close semantic similarity of the words, Supporter, Follower, and Insider. (iii) Phrases such as "Clinton Foundation", "Clinton Foundation Fund raising", "Clinton Donor" and "Clinton Foundation Contributions" form yet another distinct actant context, because of the semantic similarities of the words, Foundation, Fund Raising, Donor, and Contributions. These examples guide not only how to automate the determination of actants, but also that the actants themselves have a hierarchical structure, based on the different semantic and contextual roles they play. The phrases in (i) (dealing with the different contexts for the actant Hillary Clinton) can be considered as a super-actant or a **supernode**, and the phrases in (ii) and (iii) (dealing with different facets and distinct roles that are associated with the actant, Hillary Clinton) as sub-actants or **subnodes**. The subnodes are specific semantic contexts that directly relate to the supernode, and are expected to have relationships that are semantically homogeneous with the rest of the actant groups. As described later in this chapter, subnodes are automatically retrieved via clustering techniques. Furthermore, cluster pruning methods are used to delete and merge subnodes and handle cases where a hierarchy is not guaranteed for a supernode.

Semantic and functional similarity of words have been historically difficult to compute, and were manually cataloged in dictionaries, thesauruses, and manually created databases, such as WordNet and VerbNet. Recent advances in data-driven methods of embedding words and phrases into a multidimensional vector space [30][31] such that their Euclidean distances have correlations with their semantic similarity have opened up opportunities to assign a quantitative measure to the similarity metric. The embeddings of syntactic argument phrases can be clustered, with each cluster representing a separate actant. As we demonstrate in our results, this procedure of clustering embeddings of relationship phrases nearly automates the process of jointly estimating the actants and their hierarchy.

Figure 3.3 provides a flowchart of the computational steps executed in our end-to-end pipeline. The salient computational steps are described in the following:

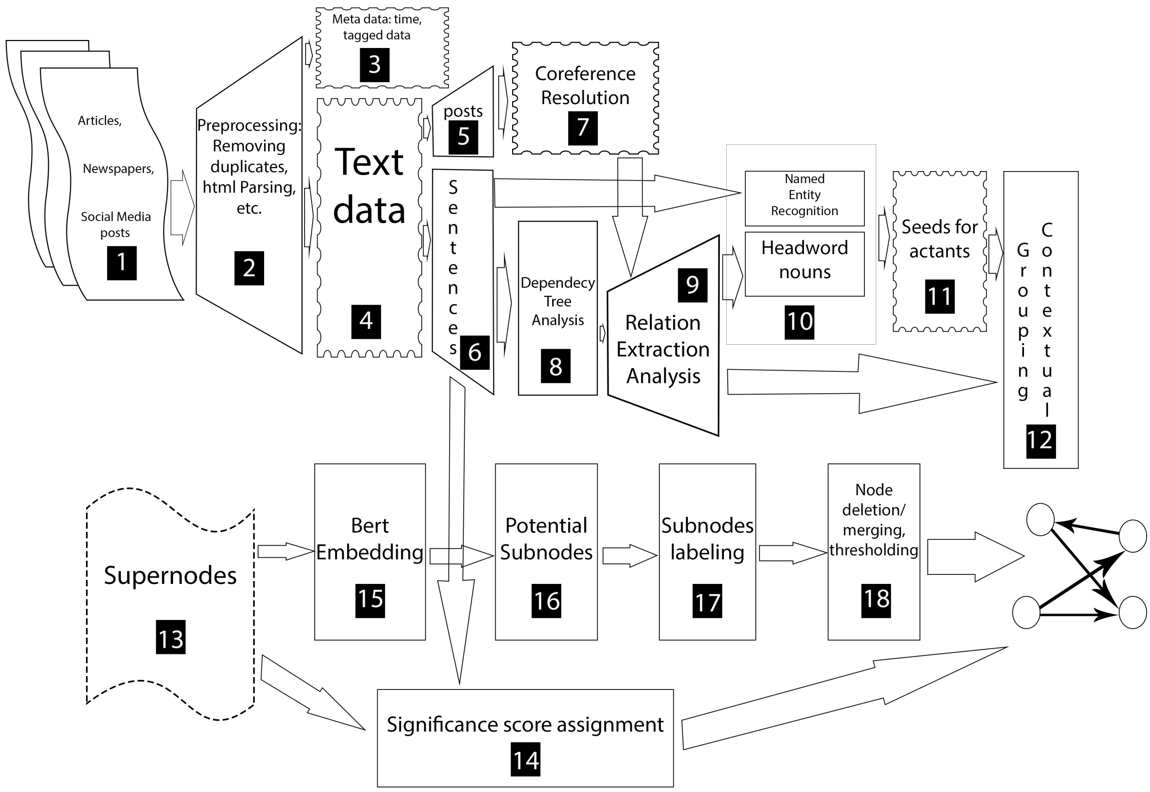

Figure 3.3: Representation of the narrative framework discovery pipeline.

Syntax-Based Relationship Extractions (Blocks 7 and 8 in Fig. 3.3): Each sentence in the cleaned text corpus is processed to extract specific patterns of syntax relationship tuples in the form of (arg_1, rel, arg_2) where 'arg1' and 'arg2' are noun phrases, and 'rel' is a verb or other types of phrase. Our relation extraction combines dependency tree and Semantic Role Labelings (SRL) [28][29]. We first design a set of patterns (such as Subject-Verb-Object (SVO) and Subject-Verb-Preposition (SVP)) to mine extractions from dependency trees by using Stanford Dependency Parser and various extensions [28, 32, 33, 34, 35, 36, 37, 38]. The patterns are extensions of two prior works: Open Language Learning for Information Extraction (OLLIE) [4] and ClauseIE [8]. Second, we form extractions from SENNAs Semantic Role Labeling (SRL) model. We combine dependency-based extraction techniques together with SRL to increase recall of our system. Then we apply cleaning and de-duplication techniques to select unique and high precision extractions. A list of all the syntax relationship patterns, their definitions, and some examples are further described in Chapter 4 (See Table 4.1).

This task of sentence-level syntax relationship extraction has been studied in work on Natural Language Processing [4, 5, 6, 28, 37, 38]. While extracting syntax relationships is an integral part of our framework, our work differs from such methods in one key aspect. *We take a holistic approach to extracting actants and their relationships*: we aggregate sentence-level extractions to form corpus-level actants and their pairwise relationships. As described previously, and in more detail in the following, the noun phrases arg1 and arg2 are aggregated across the whole corpus to group them into semantic categories or actants. This aggregation process (based on the generative model of narratives) also takes into account contextual differences, where the relationships between actants change in different situations. Such corpus-level structure cannot be inferred by simply extracting relationships seen in sentences. Furthermore, syntax-based relationships, such as SVO (subject, verb, object), were then tuned to capture story-specific syntactic forms of expressions. For example, to fit our generative model, we break up three-way relationships into multiple pairwise relationships: a sentence, such as "Bilbo steals the ring from Gollum in the Misty Mountains," would be broken up into three pairwise relationships: : (Bilbo, steals, the ring); (Bilbo, steals the ring from, Gollum); and (Bilbo, steals the ring in , Misty Mountains). Also, arg_1 and arg_2 could

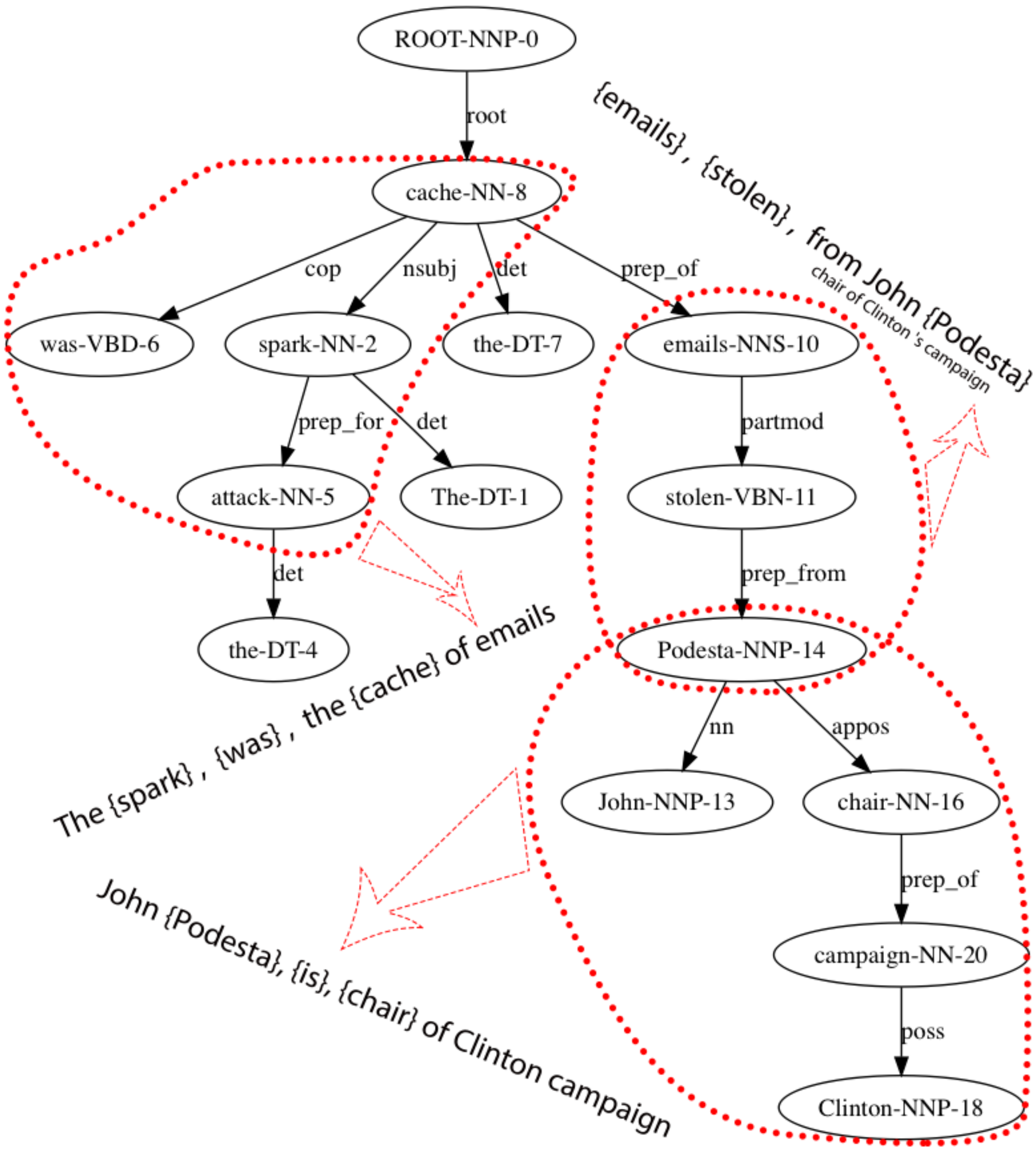

Figure 3.4: **An Example of Syntax-based Relationship extraction patterns:** The sentence, *"The spark for the attack was the cache of e-mails stolen from John Podesta, chair of Clinton's campaign"* is analyzed to extract three relationship tuples. These relationships are then aggregated across the entire corpus to create the final narrative network.

be pronouns, and for our actant determination procedure it is important that we determine to which nouns or noun-phrases they refer. Since pronouns often refer to nouns in preceding sentences we use groups of sentences belonging to the same post as input to a co-reference tool. We apply the output maps (pronouns resolved to nouns) to replace the resolved pronouns in the noun phrases, arg1 and arg2, with their corresponding nouns. Therefore, a major fraction of the pronouns are replaced after syntactic extraction. This corresponds to block number 7. The input to this block is posts and the output is used in block 9.

Actants Discovery (Blocks 10 through 18 in Fig. 3.3)

Formally, let P be the set of all relationship and noun phrases, (i.e. all phrases, arg1, arg2, and rel occurring in any syntactic extraction (arg1, rel, arg2)), we define an embedding mapping $E : P \rightarrow R^n$, that maps the set of phrases to a real vector of dimension n. Given any phrase, $P_i \in P$, $E(P_i) = \mathbf{y_i} \in R^n$ (and without loss of generality we assume $\|\mathbf{y_i}\| = 1$). Moreover, the mapping E is such that if P_i and P_j are semantically close phrases, i.e., they mean almost the same thing even if they do not use the exact same words, then their corresponding embeddings must satisfy $\|\mathbf{y_i} - \mathbf{y_j}\| \approx 0$. This requirement enables an unsupervised approach to actant determination: One can cluster the embedding vectors to obtain semantically close actant groups.

One approach would be, for example, to take all the noun phrases (i.e., arg1 and arg2) and get their embeddings using Word2Vec or Glove [30][31], or their more contextual version, BERT [2], and cluster them using k-means clustering to obtain actant candidates. These clusters can then be further processed to merge very similar clusters to form a combined larger actant group, or to deleted clusters that are not meaningful enough or too heterogeneous (for example, as measured by the entropy of the word distributions in the phrases clustered together). This direct approach, however, suffers from two major drawbacks: (i) The noun phrases, even after resolving pronouns/co-references, are dominated by high frequency pronouns, such as “they”, “I”, and “she”, or not so meaningful online terminologies, such as “URL”. This results in large clusters comprised of high-frequency but irrelevant actant groups, while more relevant actant groups get merged together to form heterogeneous clusters. (ii) The current embedding techniques tend to be flat, i.e., there is no inherent hierarchy in the vector space in which the words and phrases are embedded. Thus, the example of the “Hillary Clinton” supernode and the subnodes related to “Clinton Foundation” and “Clinton Campaign” cannot be easily replicated by using such a flat embedding.

The above observations motivated us to adopt a two-step process: (i) *Contextual Grouping of high frequency entities and concepts to create* ***Supernodes***: We first create a ranked list of named entities and concepts. Then we define a supernode as a context consisting of all the argument phrases that have a limited but unique and highly-correlated subset of the entities/concepts as substrings. Thus, for example in the Pizzagate dataset, we find all phrases with any of the following words {Clinton, Hillary, Hillary Clinton} as one single supernode. Similarly, we find {Pizza, Commet Pizza, Ping, Pong} as the seed words for another supernode. Thus a supernode defines a general context, which can be further divided into subactants or subnodes as described next. (ii) *Use embedding vectors to cluster arguments in a supernode to create* ***subnodes***: Now that we have defined meaningful contexts, we cluster the embeddings of the phrases belonging to a supernode to create subnodes.

Determining Superactants or Supernodes: (corresponding to blocks 10 through 13) After retrieving syntax extractions from the corpus sentences, we generated and ranked a list of entities, which was then used to form the seeds for potential actants. The ranking is based on the frequency of occurrences of the entities in the noun phrases arg1 and arg2. This ranking consists of both named entities as well as concepts such as "closures" and "email". For Named Entity Recognition (NER), we used the Flair framework [3], a character-level neural language model for contextualized string embeddings. We used the Flair pre-trained model and limited the candidate actants to eight main types (see Table 2.1). For concept discovery, we created a ranking of the frequent headwords in the noun phrases, arg1 and arg2. This method provides a second ranking of headwords including non-named entities. We then combined the two rankings, and ranked each entity according to the summation of its frequency in the two lists. The list could be truncated to delete all nodes below a certain frequency threshold. The truncated list is the original list of all entities/concepts to be considered for creating supernodes.

The subset of entities/concepts that define a supernode is computed in a hierarchical fashion:

- (**Step-0:**) Let's define "original list" as the ranked list of entities/concepts and "current list" as the remaining list of entities/concepts that are potential candidates to be picked as a seed in the supernode construction. The current entity/concept list is set equal to the original list. The maximum number of seed nodes in a supernode is set to k.

- (**Step-I:**) If the current list is empty then terminate (supernode construction is completed). Otherwise, select the highest ranked entity/concept in the current list (in the first iteration, the entire original list is the current list). Let this entity be E_1. Add E_1 to the list of seed nodes for the new supernode, S. Remove E_1 from the current list. Set the seed-node list size, $|S| = 1$.

- (**Step-II:**) Find all phrases/arguments where any of the seed nodes in $|S|$ (i.e. the supernode being constructed) appears as a sub-string, and let this be called P.

- (**Step-III:**) Compute the most frequent entity/concept in the original list (other than the seed nodes already extracted) in P. Let this be E.

- (**Step-IV:**) If E has been processed before, i.e., it is no longer in the current list, then jump to Step-VI.

- (**Step-V:**) If E is in the current list, then add it to the list of seed nodes, S. Remove it from the current list of entities/concepts. Increase the size count, $|S| = |S| + 1$. If $|S| = k$ (where k is the maximum size of the supernode seed list S), then go to Step-VI. Otherwise jump to Step-II.

- (**Step-VI:**) The current list of seed nodes, S, is the new supernode. Return to Step-I to start creating a new supernode.

Subnode creation and labeling: (corresponding to blocks 15 through 18) Each supernode represents a meaningful context, and is defined by its set of argument phrases. For each phrase we compute a BERT embedding [2] and cluster the embeddings of the phrases via k-means clustering. Supernodes have varying sizes, i.e., different supernodes have larger or smaller number of argument phrases, and it is a computationally involved task to optimize k, the number of clusters for each supernode. It is often computationally inefficient to optimize k for every supernode. Thus, we avoid such customization and we fix a single value of k (for both Pizzagate and Bridgegate, we empirically picked $k = 20$) for all supernodes and then delete insignificant clusters or merge two very similar clusters as described in the following: (i) **Deletion of small size clusters**: For each supernode, we plot the size distribution of the k clusters, and we find that a certain percentage always has significantly smaller size than the average. Therefore, we define a threshold based on the ratio of the size of a cluster and the average size of the clusters for that supernode; all clusters with a ratio below this threshold are deleted. The rest of the larger clusters are processed as potential subnodes. (ii) **Merging of very similar clusters:** For each cluster, we generate a ranked list of the words that appear in the phrases that define the cluster. The ranking is based on a TF*IDF score; where TF is the frequency of the word/term in the phrases of the subnode, and IDF is the inverse of the number of paragraphs/posts that the word has appeared in the entire dataset. A list of n (corpus dependent, e.g. $n = 2$ for Bridgegate and $n = 5$ for Pizzagate) top significant words from this list is then used to create a *label* for the cluster. For the particular implementation in this paper, we start with the first word in the ranked list, and then add the next word only if its score is greater than $\alpha *$(score of its predecessor) for some corpus dependent $\alpha < 1$ (e.g. for Pizzagate we used $\alpha = 0.5$ and for Bridgegate $\alpha = 0.7$); if the next word is not significant then we stop. We also stop if we reach n top words in this list of significant words. Thus, for each cluster we determine a label comprising of at most n representative words. Next we consider all the k clusters and merge all clusters with identical labels. *Each such merged cluster is now a subnode.*

Contexts and context-dependent relationships: For computational purposes, we defined a particular context as the set of sentences where two actant categories as determined by noun phrases belonging to the same supernodes, for example, appear together in the same sentence. A context is characterized by the set of relationship phrases that have already been computed from these sentences. To further distill this set of relationship phrases and create a ranked order among them, we consider only the verbs in the relationship phrases. Verbs are known to capture binary relationships in large-scale corpora [5]. The contexts defined by verbs have discriminative power since they capture the different roles played by the same actants in different contexts.

Computing significance scores for relationships (Block 14 in Fig. 3.3): In order to establish the significance of a relationship, as summarized by their verb phrases, for a particular pair of actants (i.e., a context), we employ the notion of conditional probability: A verb is contextually significant if:

$$P_{pair} = Prob[\text{verb}|\text{the sentence has both actants}]$$

$$>>$$

$$P_{corpus} = Prob[\text{verb in any sentence in the corpus}].$$

Such a measure attenuates the effect of commonly occurring verbs such as "has", "is", and "are" (for which $P_{pair} \approx P_{corpus}$), while accentuating topical verbs that describe meaningful relationships between actants. Since there are many verbs involved in any context, we rank the relative significance of the different verbs via a scoring/weighting function $f(P_{pair}, P_{corpus})$, and then select the top ones as the verb set to characterize the context. We empirically tested various scoring functions, including TF-IDF style scoring functions and the Kullback-Leibler (KL) divergence metric, to prioritize for the verbs that are more significant to a pair. After testing various scoring functions we decided to implement the above idea computationally as follows: we lemmatized the verbs and then stemmed them. For every stemmed verb, N_v, we computed

$$P_{corpus}(v) = \frac{N_v}{N}$$

where N_v is the number of times verb v occurred in the corpus, and N is the sum of the frequencies of all the verbs in the corpus. Then, for any given context, defined as the set of all sentences where the two actants co-occur, we computed

$$P_{pair}(v) = \frac{N_v(C)}{N(C)}$$

where $N_v(C)$is the number of times verb v occurred in the given context, and $N(C)$ is the sum of the frequencies of all the verbs in the context. Then we computed

$$ln \frac{P_{pair}(v)}{P_{corpus}(v)}$$

for all verbs v, and ranked them in decreasing order to obtain the set of top verbs that characterized the given context.

Multi-Scale Narrative Network Generation: The network defined by all the subnodes and their relationship edges, which are labeled by the most significant relationship phrases/verbs, is the final narrative network for a particular corpus. This network will tend to have a relatively large number of nodes (e.g. for Pizzagate 89 and Bridgegate 114) and high edge density (e.g. number of edges for Pizzagate is 344, and for Bridgegate is 778). The subnodes and supernodes however have different roles and importance, and meaningful sub-networks can be extracted for projecting different facets of the narrative network. For example, power-relationship networks, ego networks, super-node level networks, and networks comprising a target set of entities or actants. These various meaningful subnetworks and their interpretations are discussed in Section 4.5.

Structural Centrality of Nodes and Edges: Various measures of centrality and importance can be computed for each of the nodes and edges in the network. Eigen-centrality or Page Rank for nodes, and Betweenness for edges are example measures. A set of central nodes in a narrative network can be defined as a set of minimal size whose removal breaks up the network into disjoint connected components. For example, as illustrated in Figure 4.11, the removal of the node labeled as Wikileaks in the Pizzagate narrative network breaks it up into disjoint connected components that define different domains that the actants inhabit. For the Bridgegate narrative network, such a small size set of central nodes does not exist. The network of connections among the various actants were already in place prior to the conspiracy event centered around the closure of lanes.

In Chapter 4, we further discuss and justify the above architecture by providing in-depth evidence and results from analysis of Pizzagate and Bridgegate scandals. We demonstrate how our methodology identifies fake and real stories and compares their narrative structures.

3.3 Results

StoryMiner's models and methods form a machine learning software that is capable of identifying story narratives from large-scale text. Detailed applications and results of StoryMiner on various datasets are described throughout the following chapters. StoryMiner assists with discovering fake news stories, their structure, and how they differ from actual stories. StoryMiner can also build consensus models from user reviews on various topics such as their experiences with products. The major contributions of StoryMiner discussed in later chapters.

The underlying Open Information Extraction (Open IE) system of StoryMiner is called **StoryMiner RelEx**. It is a sentence-level relation extraction system particularized for story-specific relationships which achieves comparable results to the state-of-the-art Open IE systems. More importantly, it extracts and breaks down multi-way relationships that are essential components within the stories, including the relationships that are not necessarily covered by the other Open IE systems. Table 3.5 follows the evaluation criteria proposed by [39] and shows how StoryMiner ranks among the current state-of-the-art Open IE systems. In this table systems are ranked based on F1-scores on a manually-labeled set of sentences. For an in-depth explanation of the criteria used, see [39].

	Extractions	*Matches*	*Exact matches*	*Prec. of matches*	*Recall of matches*	*Prec.*	*Recall*	*F1*
MinelE (Gashteovski et al., 2017)	252	134	10	0.75	0.83	0.4	**0.323**	**0.358**
ClauseIE (Del Corro and Gemulla, 2013)	223	121	**24**	0.74	0.84	0.401	0.298	0.342
StoryMiner RelEx	180	111	4	0.7	0.78	0.433	0.251	0.318
OpenIE 4 (Mausam, 2016)	101	74	5	0.68	0.84	0.501	0.182	0.267
Ollie (Mausam et al., 2012)	145	74	8	0.73	0.81	0.347	0.175	0.239
ReVerb (Fader et al., 2011)	79	54	13	0.83	0.77	**0.569**	0.121	0.2
Stanford (Angeli et al., 2015)	371	99	2	0.79	0.65	0.21	0.188	0.198
PropS (Stanovsky et al., 2016)	184	69	0	0.59	0.8	0.222	0.162	0.187

Figure 3.5: Comparing StoryMiner RelEx with the state-of-the-art Open Information Extraction Systems.

Furthermore, StoryMiner offers i) a hierarchical actant model to partition entities into hierarchical groups with similar contextual roles ii) a Story Model to represent narratives in the form of networks that reveals stories, narrative structures, relationship sequencing, and iii) a demo webpage and a set of GitHub repositories for public use.

CHAPTER 4

StoryMiner for Fake News Structure and Threat Assessment

> "The best way to predict the future is to invent it."
>
> - Alan Kay

Although a great deal of attention has been paid to how conspiracy theories circulate on social media, and the deleterious effect they have on political institutions, there has been little work done on understanding their narrative structure. Predicating our work on narrative theory, we present an automated pipeline for the discovery and description of the narrative frameworks of conspiracy theories that circulate on social media and other forums. We apply the same approach to an actual conspiracy reported in the news media, and highlight the structural differences between the two narrative frameworks. We base this work on two separate comprehensive repositories of blog posts and news articles describing the well-known conspiracy theory Pizzagate from 2016, and the New Jersey political conspiracy Bridgegate from 2013. To derive the narrative frameworks, we automatically extract and aggregate the actants (people, places, objects) and their relationships from the posts and articles in their respective repositories, making use of named-entity detection and word embeddings for domain discovery, NLP tools for data cleaning as well as actant-relationship extraction, and context-dependent word embeddings (BERT) for actant-relationship aggregation. Through this process, we derive the underlying narrative framework, represented as an actant-relationship network, for each of these events. We show how the Pizzagate framework relies on the interpretation of hidden knowledge to link otherwise unlinked domains of human interaction, and hypothesize that this multi-domain focus is an important feature of conspiracy theories. We contrast this to the single domain focus of an actual conspiracy. While Pizzagate relies on the alignment of multiple domains through links created with hidden knowledge, Bridgegate remains firmly rooted in the single domain of New Jersey politics. We propose that, while the narrative framework of a conspiracy theory stabilizes quickly, the narrative framework of an actual conspiracy develops more slowly as revelations come to light. [1]

4.1 Introduction

Conspiracy theories and their factual counterpart, conspiracies, have long been studied by scholars from a broad range of disciplines, including political science [40, 41, 42], philosophy [43], psychology[44, 45, 46, 47, 48, 49], law[50], sociology [51, 52], folklore[53, 54] and history [55, 56]. The recent amplification of conspiracy theories on social media and internet forums has led to an increase in attention paid to how these stories circulate[57, 58, 59], and engendered discussions of the impact these stories may have on decision making[60, 61, 62]. Rosenblum and Muirhead suggest that the corrosive nature of conspiracism intrinsic to these stories, their impact on Democracy writ large and, more narrowly, on democratic institutions such as a free, independent press, warrant significant study[40].

Despite the attention that conspiracy theories have drawn, little attention has been paid to their narrative structure, beyond the recognition that conspiracy theories rest on a strong narrative foundation[50] or that there may be typologies useful for classifying them according to certain narrative features[63].

Part of the challenge of studying conspiracy theories as narrative can be traced to their inaccessibility and the fragmentary manner in which they are often discussed. Although the rise of social media has provided a convenient arena for studying the emergence of conspiracy theories, a form of what narratologists have labeled narrative complexes, the fleeting nature

[1]In addition to my advisors, I would like to acknowledge the following people for their contributions to this Chapter: Prof. Tim Tangherlini, Shadi Shahsavary, Ehsan Ebrahimzadeh, Peter Broadwell.

of communications on the forums where conspiracy theories grow and circulate makes them difficult to identify, track and study. Posts to forums where conspiracy theories take shape are often incomplete or allude to offline conversations or inaccessible websites. In addition, conspiracy theorists frequently refer to events, places, and people with coded or otherwise hard to decipher language. Consequently, determining the underlying narrative framework of the conspiracy theory–its cast of characters, the relationships between those characters, the contexts in which those relationships arise, and the previously hidden events the interpretation of which comprise the conspiracy theory's action–is difficult. Yet understanding this underlying narrative framework, which is often the work of multiple people negotiating the boundaries of the conspiracy theory through repeated, albeit brief, interactions, can provide significant insight into the various sources of threat imagined by the conspiracy theorists, identify the sources of the conspiracy theorists alleged hidden or special knowledge, reveal the interpretive strategies applied to those sources, and detail the strategies developed by the storytellers to counteract the threats encoded in the conspiracy theory [64, 65]. The strategies that adherents to the conspiracy theory develop to counteract what they perceive as threats can have real world consequences, as evidenced by the case of Edgar Welch, who opened fire with a rifle in a Washington DC area family restaurant while investigating the claims of the Pizzagate conspiracy theory[66].

In the following work, we differentiate between conspiracy theories, which are largely fictional accounts that comprise, "scenarios made up of many beliefs and narratives which are accepted on faith and used to link and give meaning to stressful events"[67], and actual conspiracies, which are factual events comprised of malign actors working covertly, often in an extralegal manner, to effect some sort of outcome beneficial to those actors[41]. While conspiracies actually exist and are sometimes uncovered, conspiracy theories do not need to have any basis in truth. They are by their very nature always uncovered, since they only exist in narrative. A motivating question for our work is whether the narrative frameworks of conspiracy theories differ in any consistent and identifiable manner from those of actual conspiracies.

To answer this question, we developed a pipeline of interlocking computational methods to determine the narrative framework undergirding a knowledge domain or connecting several knowledge domains, be it an actual conspiracy or an expansive conspiracy theory. We base the concept of knowledge domain on George Boole's notion of discourse, and his key observation that, "In every discourse, whether of the mind conversing with its own thoughts, or of the individual in his intercourse with others, there is an assumed or expressed limit within which the subjects of its operation are confined"[68]. Extending earlier work on anti-vaccination blog posts and the legend/rumor genre in general[64, 69, 70, 71], we refine an actant-relationship model based on Greimas's actantial model[72, 73]. For Greimas, the model consists of three main components: actants (people, places, things), relationships between actants, and a sequencing of these relationships[74, 75]. Operationalizing this approach allows us to determine an actant-relationship graph which describes the narrative framework for a particular domain[64, 70], and aligns well with narrative theory that proposes that in any narrative domain there are limits on the admissible actants and the relationships between them[76]. Any storytelling event, such as a blog post, activates a subset of actants (nodes) and relationships (edges) from the narrative framework. The more often an actant-relationship is activated, the more likely it is to be activated in future tellings, with additions and deletions becoming less and less common, aligning with Anderson's law of self-correction[77]. As more people contribute stories or parts of stories, the narrative framework stabilizes since the nodes and edges become more heavily weighted each time they are activated. Even though the story may never be told in full, the members of the community circulating these stories and story fragments collectively recognize the immanent narrative that provides a framework for understanding and the creation of additional stories[78].

Recognizing that people rarely tell complete stories[79], and that random sampling from an internet forum would potentially miss important actants and their relationships, we present an automated pipeline for aggregating actants and relationships from as comprehensive a collection of posts or articles within a particular knowledge domain as possible to discover the underlying, stable narrative framework behind a complex conspiracy or conspiracy theory. The actants are combined into supernodes consisting of subnodes that represent the context dependent relationships of that actant. The relationships constitute the edges that connect nodes, which are in turn ranked based on their significance. The resulting network graph comprises the narrative framework. The pipeline we have developed is domain independent, and enables the automatic discovery of the underlying narrative framework or frameworks for any domain or group of domains.

In this work, we use the pipeline to discover the narrative framework for the Pizzagate conspiracy theory, and contrast it with the narrative framework of the Bridgegate conspiracy. We hypothesize that the Pizzagate framework, despite relying on multiple domains of knowledge, reaches a stable state relatively quickly and then becomes resistant to additions or deletions, except in certain circumstances when it expands quickly by aligning nodes and relationships from additional domains to those already contributing to the conspiracy theory. These alignments occur through the interpretation of hidden knowledge accessible, at least initially, only to the conspiracy theorists. These interpretations manifest as heretofore unknown relationships (edges) between actants that cross domains, or the identification of an actant (node or supernode) from one domain in a separate domain where they were not otherwise known to be active. For example, in the Pizzagate conspiracy theory, Hillary Clinton is discovered to be an actant not only in the domain of politics, but also in the domain of Satanic pedophilia.

By way of contrast, Bridgegate, while broad in its scope with a large number of actants and interactant relationships, is confined, as most actual conspiracies may be, to a single domain, in this case New Jersey politics. Despite the limited single domain purview of the conspiracy, the narrative framework was still in flux nearly six years after the initial conspiracy was uncovered, and the number of actants and relationships discovered were far greater than those discovered for Pizzagate.

4.2 Data

Data for this study were derived from two online repositories. Both were open for research use, and represented comprehensive repositories for study. For the Pizzagate conspiracy theory, we made use of an archive of posts and articles created by active members of the Pizzagate community. As with many other conspiracy theories, the community discussing and negotiating the boundaries of Pizzagate archived their own discussions, creating a useful resource for studying the emergence and the eventual stabilization of the conspiracy theory. The initial archive was available on a mediawiki specifically created to archive the far ranging conversations of the group. For Bridgegate, we relied on an archive of news reports developed by the UCLA library from a series of sources focusing on the northern part of New Jersey. The seed articles for the initial collection were either tagged or otherwise directly categorized as being about the closure of the lanes on the George Washington bridge, and then additional articles were indexed based on that initial seeding.

In its broadest outline, the Pizzagate conspiracy theory was uncovered through the Wikileaks dump of emails from John Podesta, the campaign manager for Hillary Clinton's unsuccessful run for the presidency in 2016. Through fanciful interpretations of these emails, conspiracy theorists revealed that they had discovered Hillary Clinton's alleged involvement, through John Podesta and his brother, in a pedophilic sex trafficking ring being run out of the basement of a pizza parlor in Washington DC. The conspiracy theory took root with a series of tweets in early November 2016, with the first appearance of the #Pizzagate Twitter hashtag on November 6, the day before the US presidential election[57]. Discussions of the conspiracy theory tapered off, as measured by activity on Twitter, in December 2016, around the time that Welch was apprehended with his gun outside of the restaurant after surrendering to police[57]. Pizzagate has experienced a bit of rebirth as part of the much larger QAnon conspiracy theory that began to develop in late October 2017, as well as the highly derivative Donutgate conspiracy theory, which centers on Voodoo Doughnuts, a chain of donut stores from the Pacific Northwest.

By way of contrast, the Bridgegate conspiracy was discovered by investigative reporters to be a political payback operation launched by the inner circle of New Jersey Governor Chris Christie taking aim at the Democratic mayor of Fort Lee, New Jersey, Mark Sokolich, who had refused to endorse the governor in his reelection bid. Christie's assistants conspired with members of the Port Authority to close several toll lanes to the George Washington bridge, thereby causing catastrophic traffic jams that lasted for a week in early September 2013. When asked, these assistants said that the lane closures were part of a traffic study. A formal investigation into the decision to close the lanes was launched in 2014 and, during the ensuing five years, the overall contours of the conspiracy were revealed and various actors were indicted, tried and sentenced to prison.

For Pizzagate, our data set consisted of 17,498 posts comprising 42,979 sentences, with an end date of February 2018. We used a similar end date for Bridgegate, and consequently worked with an archive of 385 news reports comprising 20,433 sentences. Because of this end date, we missed the events of April and May 2019 based on the revelations of one of the main conspirators, Bridget Ann Kelley, subsequent to her sentencing for her role in the conspiracy. These revelations highlighted the role of an otherwise seemingly unimportant actant, Walter Timpone, and added several new relationship edges to the Bridgegate narrative framework. The fact that additional information related to an actual conspiracy emerges over a prolonged period of time (here, five and a half years) might be one of the tell-tale signs distinguishing a conspiracy from a conspiracy theory. In our study of Pizzagate, despite the three year scope of this study, the number of actants in the narrative remained stable.

Although Pizzagate was conveniently archived by the community members themselves, and the Bridgegate conspiracy was reported and archived by newspapers covering New Jersey politics, our approach does not require pre-established data sets. While having comprehensive data collections eliminates an initial step in the narrative framework discovery pipeline, earlier work demonstrates a method for determining active domains of discussion in any collection of internet resources[70]. The first step in the pipeline can be tuned to capture actants that may be of interest and the extent of a domain can be discovered from there. In that earlier work, we showed how a topic modeling approach based on a hierarchical method reveals broad topics of discussion in a large social media space that we identify as knowledge domains[70]. Posts, discussions and articles related to these knowledge domains can then be selected to constitute the study corpus. Cleaning the data would result in a machine actionable corpus similar to those we developed for Pizzagate and Bridgegate.

4.3 Visualization

Visualization of the narrative framework as subgraphs as well as a complete graph takes place in two steps, the first fully automatic, and the second with user supervision to present more easily read visualizations. The initial graph and subgraph visualizations make use of NetworkX, where actants are imported as nodes, and all relationships are imported as directed edges; labels on those edges are based on phrases with the highest significance scores.

A series of supervised visualizations allow for user input for layout and labeling. MuxViz is used to visualize the interconnections between domains for multi-domain narrative frameworks [80], while Oligrapher is used for visualizing narrative frameworks with substantive edge labeling[81]. These latter graphs are inspired by the hand-drawn graphs mapping power relations by the artist Mark Lombardi[82]. Transformation of the node and edgelists between the automated pipeline and the required file formats for MuxViz and Oligrapher is done through a collection of simple scripts. Additional visualizations are generated in other graph visualization packages such as Cytoscape[83] and Gephi[84]. Final parameterization of the visualizations are user determined in the individual application.

4.4 Limitations

There are limitations with the current methodology that we hope to address in future research. The data can be very noisy, as in the case of Pizzagate where social media posts were the primary source. This can create significant noise in relationship extraction if one is not careful: A missing punctuation mark, for example, can completely change the dependency tree structure and lead to erroneous extractions of both the arguments and the relationship phrases. Also, while pronoun resolution is needed and desirable to improve coverage (that is, to capture relationships amongst entities when they are expressed in terms of pronouns), it can also add noise, by resolving pronouns to the wrong nouns. We designed several automated rules to err on the side of caution: for example, if a relationship is extracted from a very long sentence, and the actants are far apart, we disregarded such extractions. Similarly, if pronoun resolution substituted a pronoun with a long noun phrase, we disregarded such resolutions. Both these rules decreased the potential coverage of relationships and identifying actants, but allowed us to develop a set of rules giving us high confidence that most of the relationships extracted were correct. Even with such stringent measures in place, we do get noisy syntactic relationships at the level of sentences. The steps of aggregation (we do not include relationships without a certain number of repetitions) and sorting of relationships via their significance scores considerably improve the accuracy of the summary relationships. The process of denoising both our syntactic and aggregate extractions is an ongoing research project.

As already noted, because of ambiguities in extractions and the noisiness of the data, actant aggregation is not always accurate; our methods err on the side of caution and tend not to resolve all duplicate entities so as to avoid incorrectly resolving distinct entities into a single node. Finding clear aggregations for relationships is equally challenging, although we expect that refinements to context aware embedding methods will help this process considerably. Also, assignment of supernodes to particular domains can lead to similar ambiguities. While the pipeline currently works only on English language materials, one can introduce NLP tools tuned to other languages into the work-flow. An eventual expansion to the pipeline would be the use of language detection and appropriate branches in the pipeline for the detected languages, thereby facilitating the use of multilingual corpora.

Because of the *ad hoc* nature of many of the online resources for studying conspiracy theories, it is difficult to extract consistent time data. This problem is exacerbated by two factors: inconsistent approaches to time stamping on the blogs and forums that include this information, and the common practice of participants re-posting or quoting from earlier posts. Dating and time stamping is not as significant a problem for newspaper articles, which are often the most readily available source for studying actual conspiracies, even though many newspapers publish articles directly from news services, thereby introducing duplication into the corpora.

Currently, some hand labeling of supernodes and relationships for clarity in visualizations is inevitable. The relationship labels, in particular, are neither as semantically rich nor grammatically correct as human generated labels. Nevertheless, the automatically generated relationship labels act as an informative starting point for human generated labels.

Finally, we have not addressed the sequencing of events in the stories. Consequently, our narrative frameworks provide a snapshot view of the domain(s) in question and do not include the third component of Greimas's model, namely the order in which inter-actant relationships are established.

4.5 Methodology and Results

The underlying models and methods of this work are described in Chapter 3. To avoid, re-iterating of those concepts, we refer you to read the explanations in Chapter 3. In this section, however, we recap our relation extraction component through some examples and focus on discussing our results.

After cleaning, the relation extraction provides us with a ranked list of candidate entities used to seed the discovery of subnodes and supernodes, and a series of inter-actant relationships. For each of the two corpora, we find a very large number of relationships of various types and patterns (Fig 4.1). A "relationship type" refers to a descriptive name for a collection of related patterns. For instance, SVO is a relationship type describing relationships between a subject and an object of a sentence. It is extracted by following a set of syntactic patterns in dependency trees such as (nsubj, verb, dobj) and (nsubjpass, verb, dobj)[2]. These dependency-based patterns, which are extensions of prior Open Information Extraction systems, Ollie [4] and ClauseIE [8], further combined with relationships extracted from SENNA's Semantic Role Labelings (SRL) to achieve high recall relationships. We next apply denoising techniques to select high precision extractions.

Table 4.1 describes our main relationship patterns with an example extraction for each of pattern. In addition to those, we have some extended patterns grouped under "Other"

[2]A complete definition of the dependency labels (e.g. nsubj, prep) are described in the Stanford's typed dependencies manual[85])

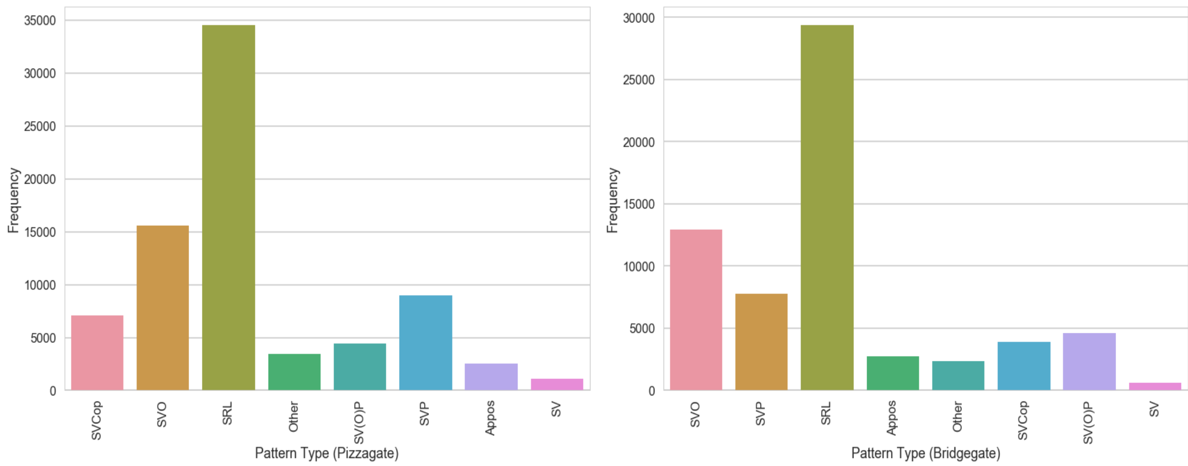

Figure 4.1: **Relationship extraction patterns.** Patterns by total number for A: Pizzagate (left) and for B: Bridgegate (right).

category shown in Figure 4.1. For example relationships extracted from "conjunct and" relationships (conj_and edges in dependency tree). A "conjunct and" is the relation between two elements connected by the "and" coordinating conjunction. For instance from a sentence "Prosecutors have charged Kelly and Baroni.", not only we extract (Prosecutors, have charged, Kelly) as a SVO extraction, but also we additionally retrieve (Prosecutors, have charged, Baroni) because the object (Kelly) is connected to another noun (Baroni) via a "conj_and" edge in the dependency tree. Figure

After tokenizing and lemmatizing the extracted headword lists, the resulting unsorted grouping provides a seed for the subnode lists and supernode lists (Table 4.2).

After taking the union of the arguments with each of these terms, and determining the BERT embedding [2] for each argument, k-means clustering (k=20) results in a series of subnodes. After pruning and merging, we determine the supernodes and their corresponding subnodes for each narrative framework (Table 4.3.

In all, we count a total of 24 supernodes, and 89 subnodes for Pizzagate, and a total of 114 supernodes, and 144 subnodes for Bridgegate.

[3]We take SV extractions only if subject does not come with an object or a complement, and the verb is among a set of predefined intransitive verbs such as die, or walk.

Table 4.1: Main relationship patterns along with examples

Type	Main Patterns	Example Sentence	Derived Extraction
SVO	(nsubj, verb, dobj)	Christie fired Kelly	(Christie, Fired, Kelly)
SVP	(nsubj, verb (no obj), prep)	Wildstein resigned on Dec. 6th	(Wildstein, resigned, on Dec 6th)
SVCop	(nsubj, verb, noun-cop)	The lane closures were retribution	(The lane closures, were, retribution)
SV(O)P	(nsubj, verb (with obj), prep)	The lanes were shut down for a traffic study	(The lanes, were shut down, for a traffic study)
SV^3	(nsubj, verb)	Biden died of heart attack	(Biden, died)
Appos	(word, appos, word)	Christie fired that aide, Bridget Anne Kelly, a deputy chief of staff.	(Bridget Anne Kelly, is, a deputy chief of staff)
SRL	(A0, Verb, A1) (A0, Verb, A2) (A1, Verb, A2)	Ring was uncovered by the leaked Podesta emails dumped by Wikileaks	(by wikileaks, dumped, the leaked podesta emails), (by the leaked Podesta emails dumped by wikileaks, uncovered, ring)

Table 4.2: A sampling of the Named Entity Recognition (NER) and the Headwords for Pizzagate and Bridgegate.

	Pizzagate	**Bridgegate**
Headwords	Children, Kids, Evidence, Ring	Governor, Closures, Email
NER	Alefantis, Clinton, Podesta	Christie, Wilstein, Kelly

Table 4.3: A sample of the top 5 supernodes and subnodes for Pizzagate and Bridgegate.

Pizzagate		**Bridgegate**	
Supernodes	*Subnodes sample*	*Supernodes*	*Subnodes sample*
[Podesta]	John Podesta, Tony Podesta, leaked Podesta email, Podestas, Podesta	['christie', 'christi', 'christies', 'governor', 'chris', 'former']	christie governor, chris new jersey governor, christie
['pizza', 'comet', 'ping', 'pong']	comet pizza, comet pizza story, ping pong comet, comet, ping pong review facebook	['authority', 'author', 'authorizing', 'authorities', 'authors', 'authorization', 'port', 'executive']	['authority port', 'report authority port', 'executive director', 'baroni executive director', 'report', 'authority transportation']
[alefantis]	James alefantis, alefantis, james alefantis instagram, owner james alefantis	['wildstein', 'david']	['wildstein', 'wildstein david', 'wildstein david executive former']
[traffick]	child sex trafficking, ring trafficking, ring trafficking, human pedophilia trafficking	['lee', 'fort', 'mayor', 'sokolich']	['sokolich', 'fort lee', 'sokolich mark mayor', ' mayor effort sokolich', 'lee fort lane traffic']
[child]	child, child porn, child trafficking	['bridges', 'bridge', 'george', 'washington', 'lane']	['scandal bridge bridgegate', 'closure lane', 'bridgegate', 'bridget kelly', 'closure gwb controversy lane', 'lane']

We create two different networks describing entity relationships for the subnodes. One network includes only named individuals and their professional or personal relationships. An example of an actant/relationship pair in this network for Pizzagate is: “John Podesta is Hillary Clinton’s campaign chief.” A second network is derived from contextual and interaction based relationships, such as the “Podestas had dinner.” Since each subnode represents a contextually sensitive use of the supernode category, we can visually represent these subnodes and relationships as subgraphs for each supernode. For example, the subgraph for the Podesta supernode reveals a series of attributes of Podesta (e.g. Clinton campaign manager), and a series of context dependent relationships (e.g. having dinner with his brother) (Fig 4.2).

For the subnodes of Pizzagate, we discover average degree of a node is approximately 36. By way of contrast, for Bridgegate, it is almost twice that number, approximately 72.

The relationships between supernodes can be discovered by collapsing the subnode subgraphs, and labeling the edges between supernodes with the relationship with the highest relevance score over the subgraph edges. These supernode and relationship graphs can also be visualized (Fig 4.3).

For Bridgegate, we generate the same types of subgraphs: supernodes and their constituent subnodes, as well as supernodes and their highest ranked relationships. By combining these graphs, we create the overall narrative framework of the conspiracy or conspiracy theory, which can be visualized in an automated fashion as described above (Fig 4.5 and Fig 4.6).

The supernode graphs for the two corpora have a limited number of actants, and visually present a clear overview of the main categories of actants for the narrative. The number of edges, comprised of the complete set of edges between subnode pairs, in contrast, means that the basis for the relationships between supernodes is occluded (Fig 4.4).

Visualizations of the complete subnode graphs for the two corpora, Pizzagate (89 subnodes and 438 relationships) and Bridgegate (144 subnodes and 928 relationships), make clearer the groups of relationships between subnodes, but at the expense of a proliferation of nodes (Fig 4.5 and Fig 4.6).

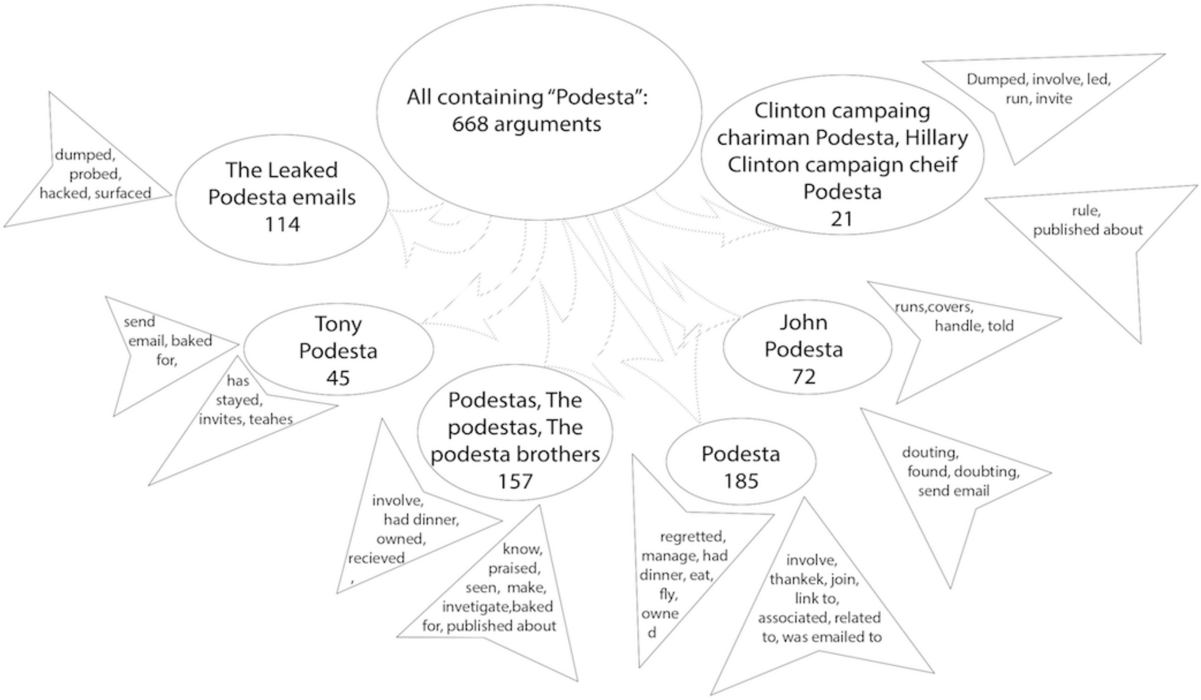

Figure 4.2: **Subgraph of the Podesta supernode.** The supernode consists of several subnodes, including those automatically labeled as *leaked emails*, *Tony Podesta*, *John Podesta*, *the Podesta brothers*, and *John Podesta as Hillary Clinton's campaign manager*. The most significant context dependent relationships for each of the subnodes are presented as labeled, directed edges. For example, the aggregated edge labels are due to relationships such as (where only the lead verbs have been picked as labels), (“Podesta emails”, “were dumped by”, “Wikileaks”), (“Podesta”, “are involved in” “pedo rings”), and (“John Podesta”, “runs”, “child trafficking rings”). See Fig. 4.8 for further examples of such edges where both ends of relationships are shown.

Visualizing the ego-networks for any of these subnodes results in a meaningful subgraph with clear labels on the edges (Fig 4.7).

In this example, the prominent relationships are between John Podesta and his brother, collectively as the Podestas, the Wikileaks emails, and pedophilia. A closeup of those nodes and relationships includes meaningfully labeled edges (Fig 4.8).

Similarly, in Bridgegate, the Bridget Anne Kelly subnode ego-network reveals a large set of relationships (Fig 4.9).

It is worth noting that the Bridgegate ego networks are far more regular, with pairs of directed edges between Kelly and her ego network nodes. A sub-selection of nodes with named individuals highlights the types of relationships between them (Fig 4.10). It reveals the specific relationships between her and other important (as determined both by their frequency and centrality in the narrative network) named individuals in the ego network subgraph. For each such named entity, we added a self-loop edge labeled with *automatically derived descriptions of the entity.* These relationships show the endemic nature of the Bridgegate conspiracy: all actants are local to New Jersey politics. Because the edges are labeled with only the lead verbs appearing in the relationship phrases, sometimes the edge labels may appear to be out of context. We, however, have the original phrases. For example, the relationship "pinned" from Chiristie to Kelly comes from sentences, such as: *Critchley said evidence to support that claim is contained in interview summaries that accompanied a report commissioned by Christie's office that* ***pinned the blame largely on*** *Kelly and Wildstein.* Similarly, the relationship "transmitted" from Kelly to Christie, comes from a sentence such as: "*If she* ***transmitted*** *it to a Christie staffer on their NON"*, where the pronoun "she" has been correctly resolved to "Kelly" and the phrase "Christie staffer" has been mapped to the supernode "Chris Christie". Another interesting part of the reporting was that Kelly and Baroni were treated almost as one single unit, with a preponderance of sentences such as, "***Kelly and Baroni*** *are accused of conspiring with David Wildstein, Baroni's former deputy at the Port Authority, to close two of three local access lanes ......*

Developing a complete understanding of the narrative framework proceeds in steps: the supernode visualization provides an overview of the main actants for which one can find the constitutive subnodes. Navigating across the subnode ego-networks, with their semantically rich edges between subnode pairs provides a comprehensive understanding of the actants and their relationships.

For Pizzagate, subsets of the supernodes define a series of otherwise unrelated or weakly related domains: Democratic politics, where actants such as Hillary Clinton and the Clinton foundation are dominant; the Podestas, with John and Tony Podesta as the major actants; casual dining, dominated by James Alefantis and Comet Ping Pong, with weak relationships to Democratic politics through fundraising, and with the Podestas through a shared love of pizza; Satanism and Pedophilia, where actions such as cannibalism and child trafficking, and actants such as children and hidden tunnels dominate. The Wikileaks domain, dominated by actants such as email, Wikileaks, and Instagram, provides the glue for the narrative framework, with relationships based on fanciful interpretations of words such as pizza and cheese. After eliminating the relationships generated by the Wikileaks subnodes, the connections between the various domains disappear, leaving the separate domains as a disjoint series of smaller connected components (Fig 4.11). This disjuncture only occurs when we eliminate the links generated by the Wikileaks subnodes, and not when we eliminate links generated by any of the other domains.

For Bridgegate, the domain is more limited to the world of New Jersey politics, and does not have a similar glue that holds the narrative framework together. Removing nodes with high connectivity does not result in the same disjoint series of smaller connected components as it does in Pizzagate; rather the main component remains connected. This level of connectedness makes sense given the single domain nature of the true conspiracy and the multiple connections between political actors in New Jersey politics.

Finally, a time analysis of the Bridgegate data reveals that, unlike the Pizzagate data in which all of the actants emerged over the course of approximately one month, the cast of actants associated with Bridgegate took nearly six years to be fully described, with several spikes in the emergence of new actants related to the discovery of new aspects of the conspiracy and various court cases (Fig 4.12).

4.6 Discussion

The New York Times, in its reporting of Pizzagate, generated a hand drawn graph of the actants and relationships that their reporters were able to discover in a purely manual manner as of December 16, 2016[86]. Although this cannot be considered a ground truth graph, it does provide a convenient external resource from a trusted news outlet against which to compare the results of our automated process. Our methods are able to discover not only the top level nodes and relationships proposed by the New York Times, but also additional nodes and relationships (Fig 4.13). Importantly, our method also reveals the central role played by Wikileaks, and the otherwise disconnected nature of the main domains of the narrative framework, supporting our hypothesis that conspiracy theories are built by aligning otherwise unrelated domains of human interaction through the interpretation of discovered or hidden knowledge to which the conspiracy theorists either have special access or for which they have a particularly astute interpretive ability.

It is worth noting that our extractions also include Bill Clinton and contributions to the Clinton campaign and foundation, which are missing in the NY Times graph, but clearly important in the discussions among Pizzagate conspiracy theorists. By way of contrast, because of the relative paucity of mentions of cannibalism, our thresholding drops that node, which otherwise appears in the NY Times graph. Exploring the relationships between actants in our graph reveals a richer set of connections between actants (226 edges with 206 different labels). Whereas there is no link between John Podesta and Satanism in the NY Times graph, our discovery reveals that, according to the conspiracy theorists, John Podesta follows Satanism. We also discover several other rich relationships between nodes, such as one that tells us that Tony Podesta owns weird art that uses coded language to promote pedophilia (Fig 4.14).

The New York Times also conveniently drew a hand-drawn graph of the actants and relationships as they were known on April 8, 2015 for the Bridgegate conspiracy (Fig 4.15)[87]. Although our method identifies significantly more nodes than presented in the NY Times illustration, we accurately identify all of the actants and the most important relationships present in that graph. Some of the actants identified by the NY Times have a relatively low ranking in our pipeline discovery; if we take the first twenty-eight actants in our ranked list (the number of actants in the NY Times graph), we miss ten actants and their various relationships present in the NY Times graph. These are replaced by actants such as Shawn Boburg, whose reporting broke the scandal, Randy Mastro, a former federal prosecutor whose report exonerated Christie, and Michael Critchlet, Bridget Anne Kelly's attorney. We capture all of the NY Times actants when we extend our list to take our top 122 actants (out of a ranked list of 609). At a threshold of the top fifty actants, we discover all but three of the NY Times identified actants (Paul Nunziato, Evan Ridley, and Lori Grifa).

As noted, all of the actants in the Bridgegate conspiracy come from a single domain, namely that of New Jersey politics. Consequently, the narrative framework is not created from the alignment of otherwise weakly connected domains, but rather is fully situated in a single domain. Similarly, there is no information source, such as Wikileaks, on which the framework depends to maintain its status a single connected component. Even the deletion of a fairly important actant, such as Bridget Kelley along with her relationships, does not lead to a series of disjoint graphs as was the case in Pizzagate when the Wikileaks associated nodes were deleted. Indeed, even if all of the Bridgegate actants conspiracy related relationships were deleted–as if the conspiracy had never happened–New Jersey politics (for better or worse) would continue to exist as a single connected component.

We believe that these three features: a single domain of interaction, a robustness to deletions of nodes and relationships, and a proliferation of peripheral actants and relationships may be key characteristics distinguishing an actual conspiracy from a conspiracy theory. Reporting on actual conspiracies introduces new actants and relationships as part of the process of validating what has actually happened. Consequently, this reporting feeds the core giant network with more evidence, resulting in a denser network over time. Conspiracy theories, by way of contrast, are formed rapidly. Since the only evidence to support any of the actants and relationships comes from the story tellers themselves, the network structure stabilizes quickly. This stabilization aligns well with folkloric theory, where an essentially constant and relatively small set of actants and relationships determines the boundaries of admissible stories (or story fragments) after the initial narrative burst finishes[54][88][77]. In short, a conspiracy theory is likely characterized by a comparatively small number of actants, multiple interconnected domains, and the fragility of the framework, which can easily be disconnected into a series of disjoint subgraphs.

A potentially useful aspect of our narrative framework discovery is its generative nature.

Once the narrative framework is established, one can generate admissible stories or story parts (e.g. forum posts) that conform to the overarching framework by selecting already established actants and relationships. Although such a capacity might be used to create and perpetuate conspiracy theories, it might just as easily be deployed to interrupt narrative frameworks that are fueling anti-democratic behaviors. At the very least, it allows for deep and powerful insight into story generation, and the underlying factors that allow people to participate in the creation and circulation of these narratives. Similarly, understanding the significant structural differences in narrative frameworks between folkloric genres such as rumors, legends and conspiracy theories on the one hand, and factually reported conspiracies on the other hand, will be useful for testing the validity of emerging narratives.

4.7 Concluding Remarks

The years of the Trump presidency including the 2016 presidential election, have been marred by what increasingly has come to be known as fake news. Lazer et al propose that fake news be understood as "fabricated information that mimics news media content in form but not in organizational process or intent"[89]. Discerning fact from fiction is difficult given the speed and intensity with which both factual and fictional accounts can spread through both recognized news channels and far more informal social media channels. Consequently, there is a pressing need for methods to understand not only how stories circulate on and across these media, but also the narrative frameworks on which these stories rest. Recognizing that a series of stories or story fragments align with a narrative framework that has the hallmarks of a fictional conspiracy theory might help counteract the degree to which people come to believe in–and subsequently act on–conspiracy theories.

Conspiracy theories have in the past been disregarded as the fanciful fantasies of fringe members of society, not worthy of serious concern. An increasing awareness that people are making real-world decisions based on informal stories that circulate on and across their social networks, and that conspiracy theories are a significant part of that storytelling, countermands that idea. The rapid spread of conspiracy theories such as Pizzagate and the capacious QAnon, coupled to real world actions that people have taken based on a belief in these narratives, are no longer purely a fringe phenomenon.

Actual conspiracies and conspiracy theories threaten Democracy each in their own particular way. An actual conspiracy usually comes to light because of the investigative capacities of a free and independent press, and reveals corruption in government or industry; as such, the discovery of an actual conspiracy confirms the power of democratic institutions. Conspiracy theories, on the other hand, seek to undermine the very premise of democratic institutions. As Muirhead and Rosenblum note, "There is no punctilious demand for proofs, no exhausting amassing of evidence, no dots revealed to form a pattern, no close examination of the operators plotting in the shadows. The new conspiracism dispenses with the burden of explanation"[90]. Given the challenges that conspiracy theories present to democracy and a free and open society, we believe that the ability to automatically discover the underlying narrative frameworks for these accounts will be useful. Such an awareness will, at the very least, provide insight into the type of muddled thinking promoted by propaganda campaigns[91] or other disinformation initiatives. It will also offer a clear overview of the domains of knowledge that are linked together through interpretation of hidden knowledge. Identification of certain structural aspects of a conspiracy theory narrative framework fueling online conversations, such as the weak connection of multiple domains, can alert us to whether an emerging narrative has the hallmarks of a conspiracy theory. Finally, these methods can provide insight into the potential strategies that adherents may be considering for dealing with the various threats identified in the narratives. Taken as a whole, the automated narrative framework discovery pipeline can provide us with a better understanding of how stories help influence decision making, and shape the contours of our shifting political environment.

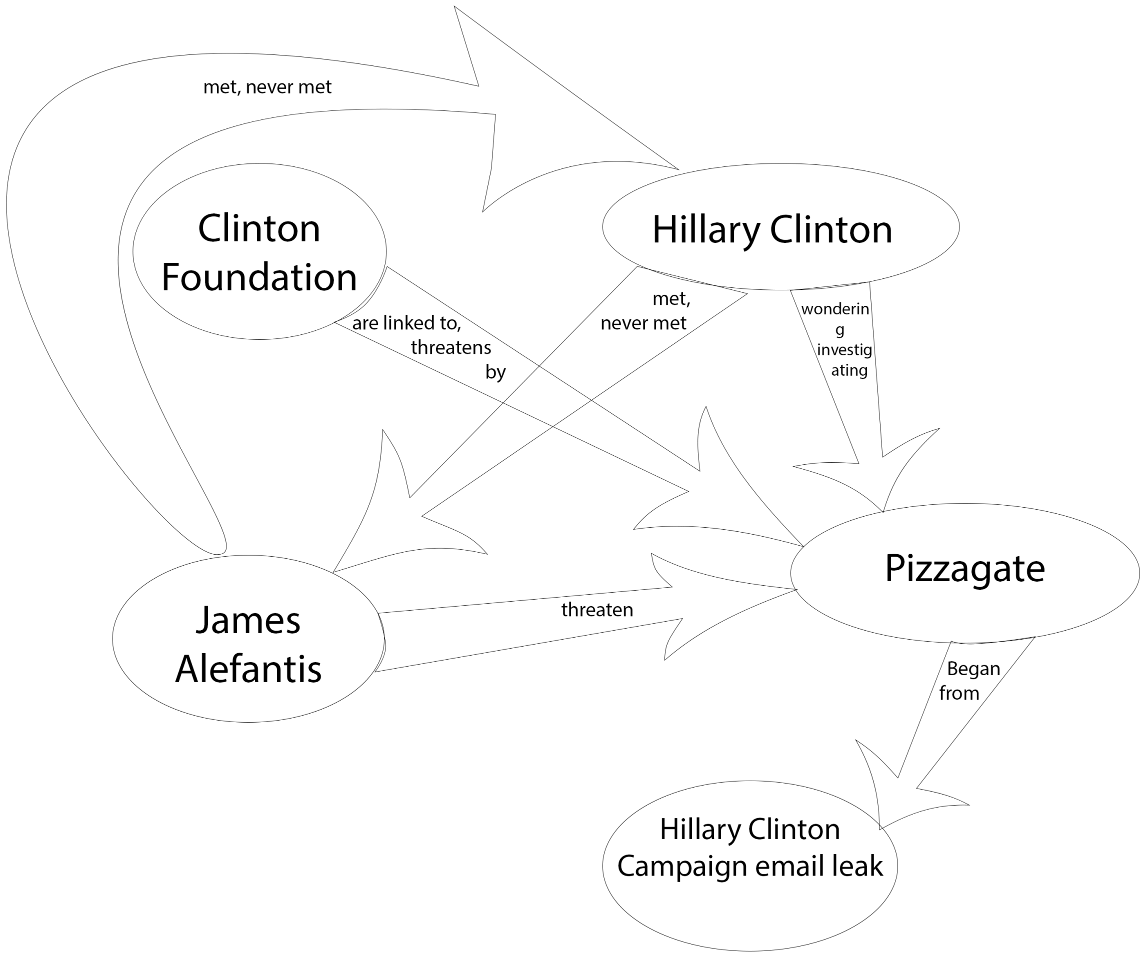

Figure 4.3: **A Subnetwork of the Pizzagate Narrative Network.** Some of the nodes are subnodes (e.g. the one labeled with "Clintion Foundation"), and some others are supernodes (for example the one labeled with "Pizzagate"). Because we only pick the lead verbs for labeling edges, the contextual meaning of relationships become more clear when one considers the entire relationship phrase. For example, the relationship, "began" connecting "Pizzagate" to "Hillary Clinton Campaign email...." originates from sentences such as, "*What has come to be known as Pizzagate began with the public release of Hillary Clinton campaign manager John Podesta's emails by WikiLeaks...*". Similarly the edge labeled with "threaten" connecting "Alefantis" to "Pizzagate" supernode, comes from sentences such as, "*James Alefantis threatens Pizzagate researcher....*". Here the supernode, "Pizzagate" includes the entity "Pizzagate researcher," which appears as a subnode.

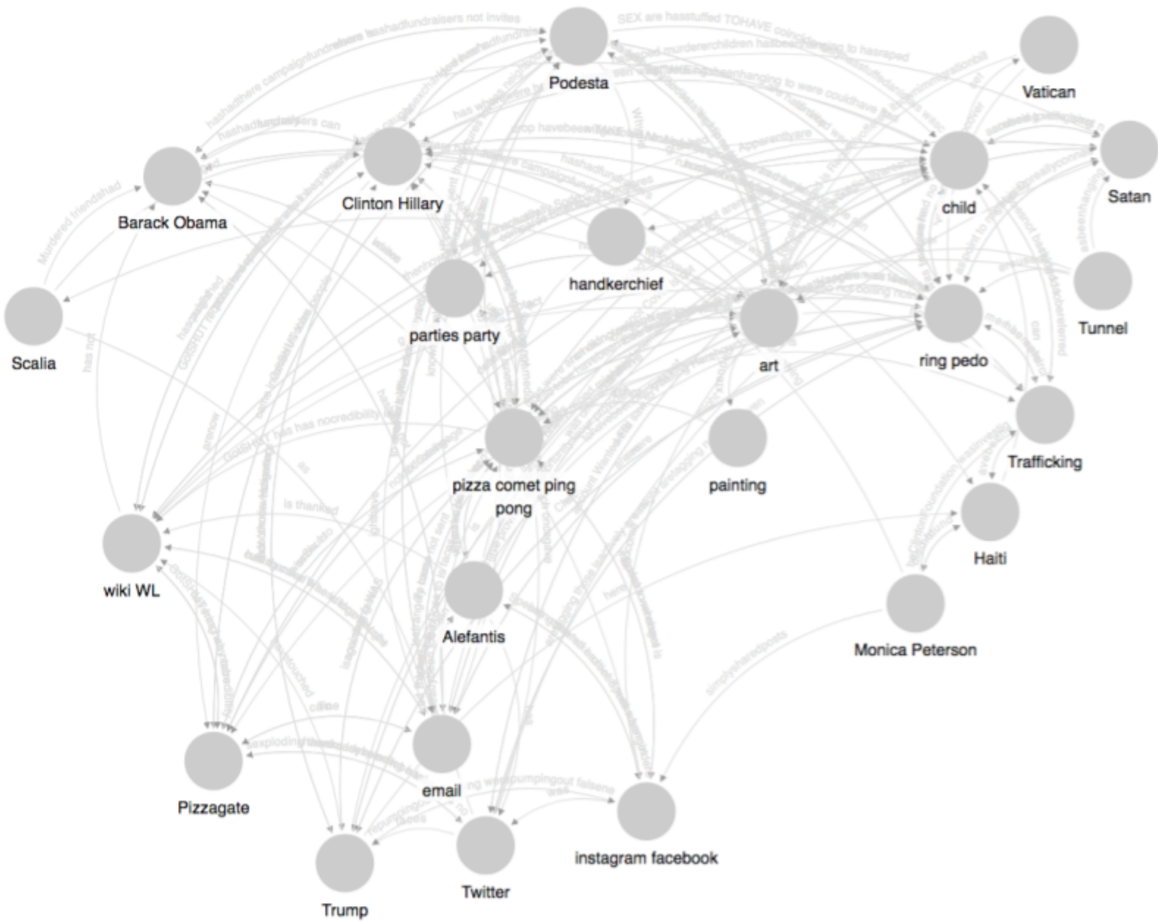

Figure 4.4: **Supernode graph for the Pizzagate conspiracy theory.** The automatically generated labels of the nodes show that all the major actants were discovered.

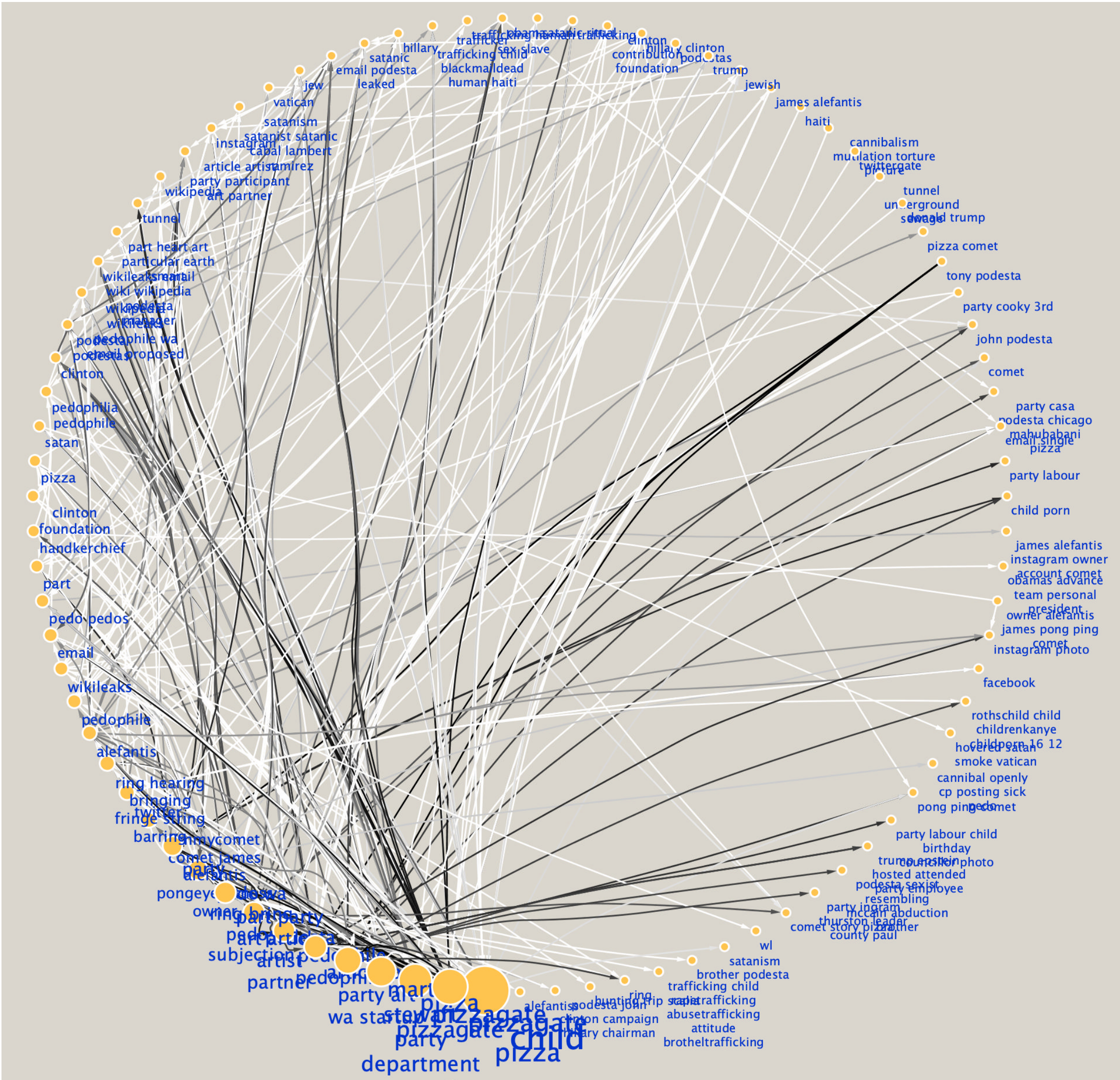

Figure 4.5: **Complete subnode graph of Pizzagate.** Complete subnode graph of Pizzagate visualized using Cytoscape's circle layout algorithm, with bundled edges and nodes sized according to betweenness centrality. Node sizes and labels are also scaled by betweenness centrality.

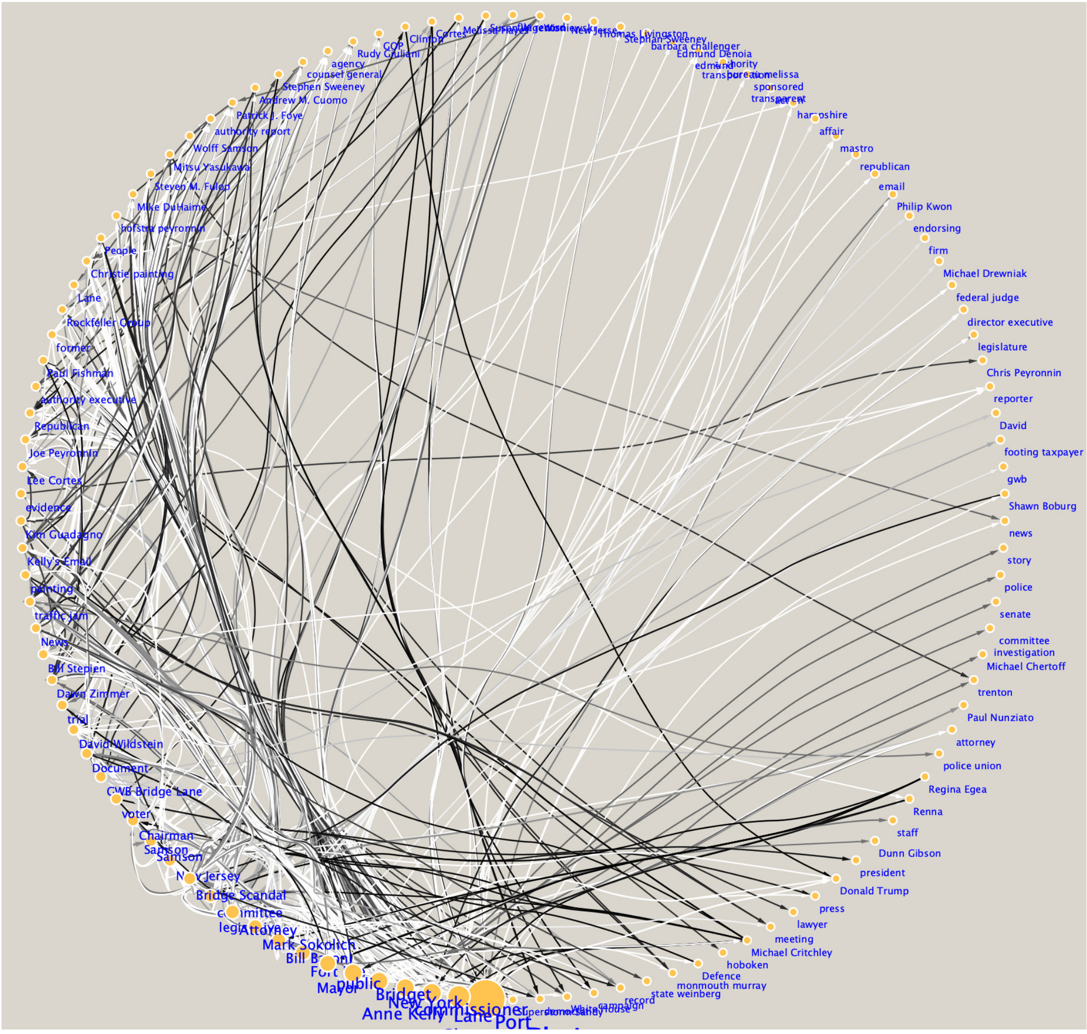

Figure 4.6: **Complete subnode graph of Bridgegate.** Complete subnode graph of Bridgegate, visualized using Cytoscape's circle layout algorithm, with bundled edges and nodes sized according to betweenness centrality. Node sizes and labels are also scaled by betweenness centrality.

Figure 4.7: **Ego network graph visualization for John Podesta.**

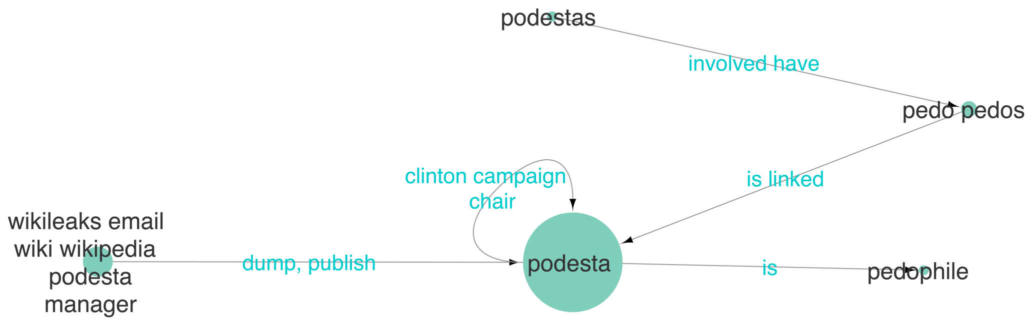

Figure 4.8: **Subselection of nodes from the Podesta subnode egonet subgraph.** The self-loop edge for the node "Podesta" is labeled with an automatically derived description of John Podesta as the Clinton Campaign Chair.

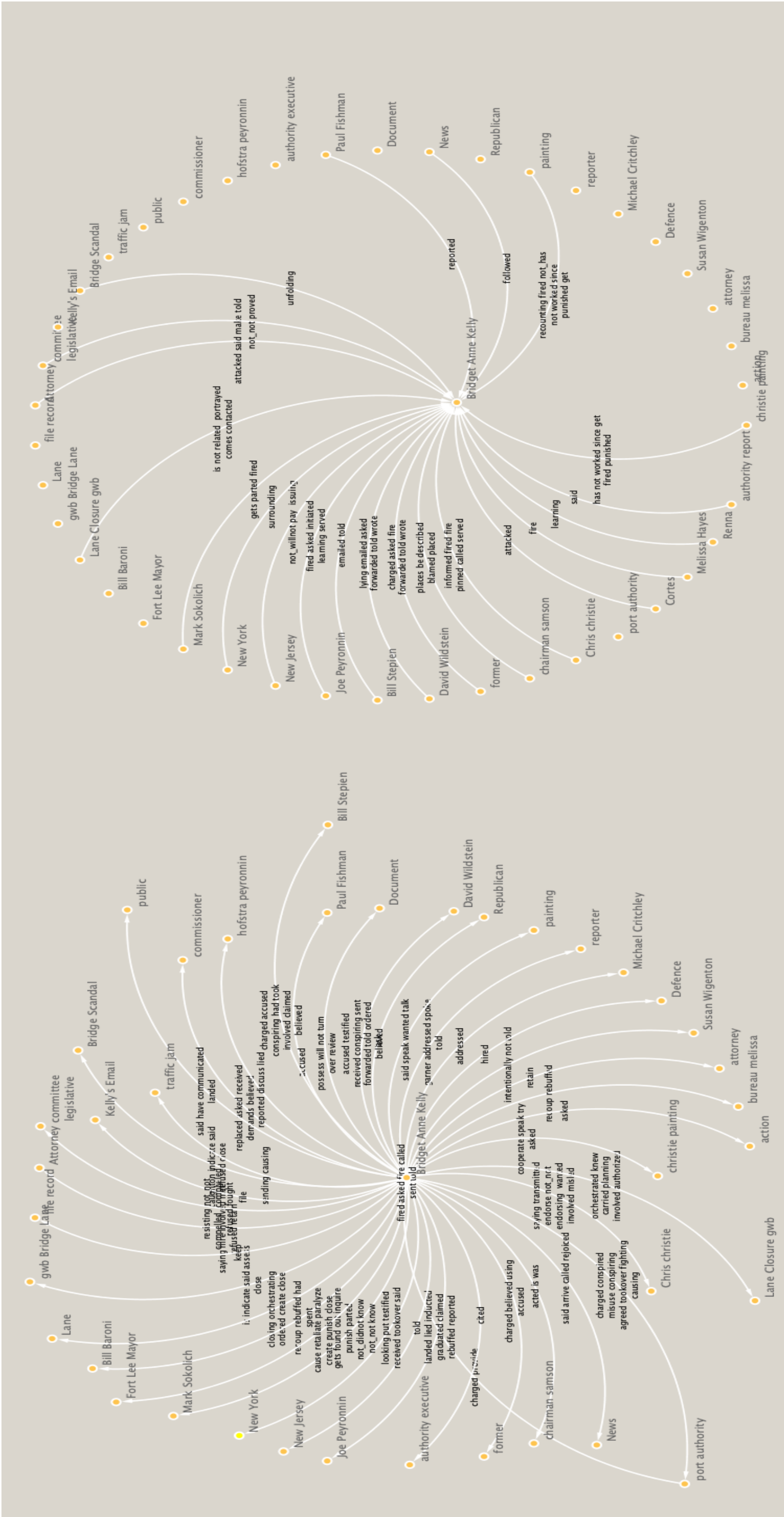

Figure 4.9: **Ego network graph visualization for Bridget Anne Kelly.** A: Labeled edges with Kelly as target. B: Labeled edges with Kelly as source.

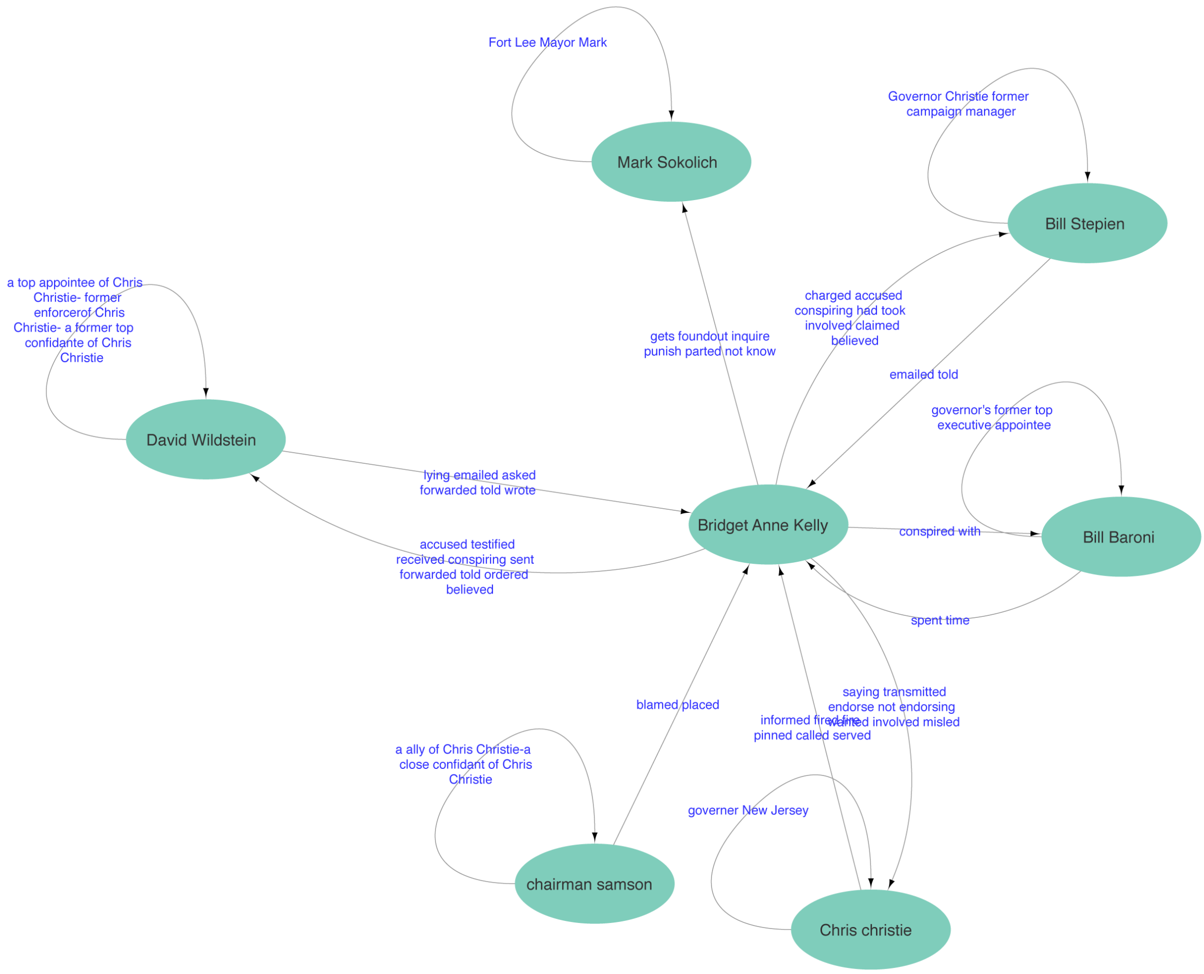

Figure 4.10: **A subset of the ego network for Bridget Anne Kelly.** This graph reveals the specific relationships between her and other important (as determined both by their frequency and centrality in the narrative network) named individuals in the ego network subgraph."

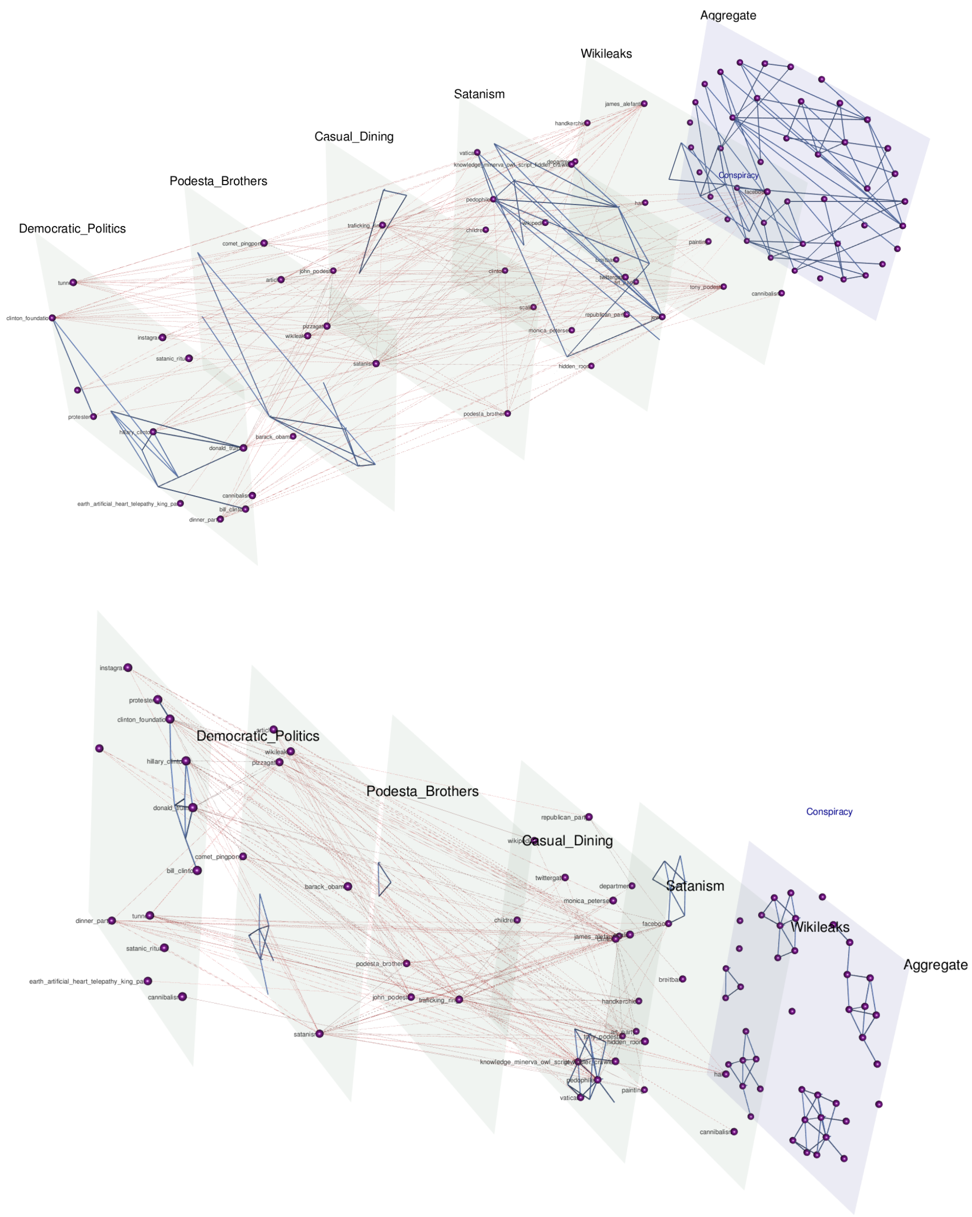

Figure 4.11: **A visualization of the narrative framework for Pizzagate in terms of domains.** On the top, **A:** the graph with the inclusion of relationships generated by Wikileaks–the aggregate graph in blue shows a single large connected component. On the bottom, **B:** the graph with the Wikileaks relationships removed, shows on the aggregate level the remaining domains as disjoint components. Thus, in the PizzaGate conspiracy theory, the different domains have been causally linked via the single dubious source of leaked emails dumped by Wikileaks. No such keystone node exists in the Bridgegate narrative Network.

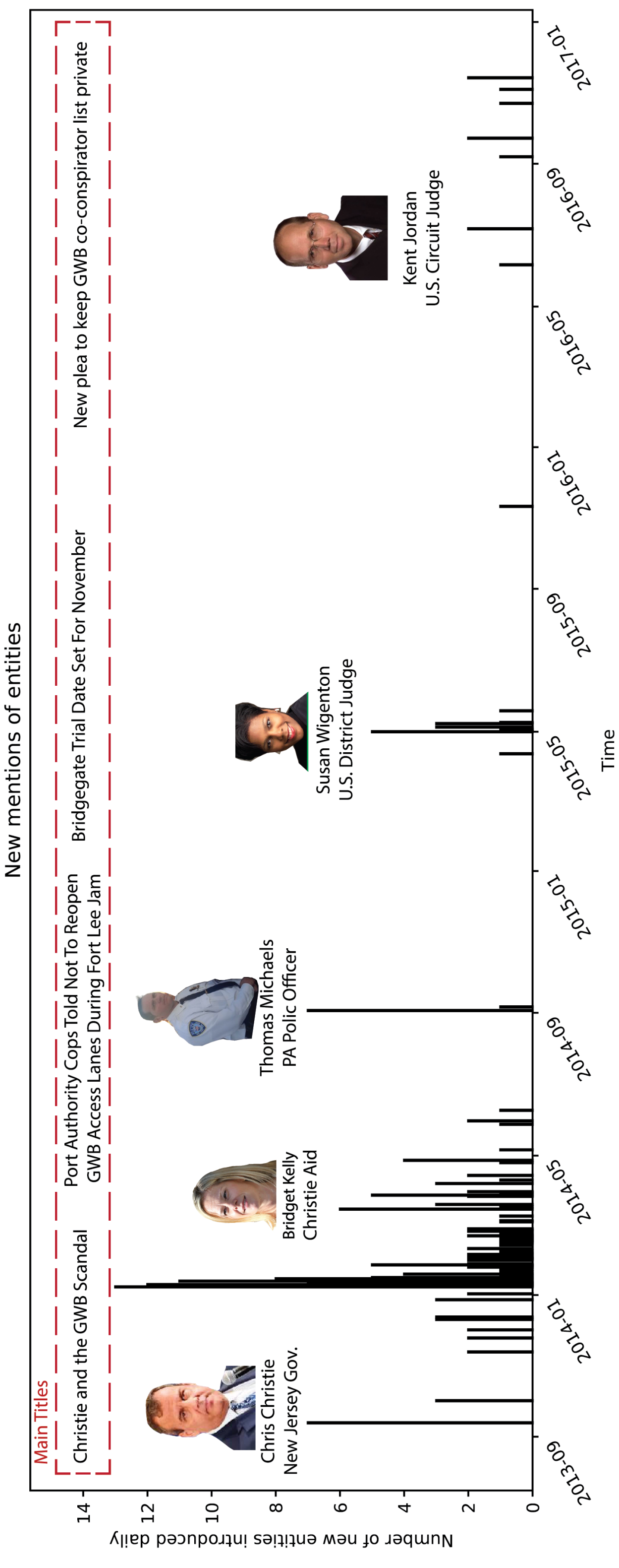

Figure 4.12: **Time series of the first mention of Bridgegate entities.** Starting with the events of September 2013.

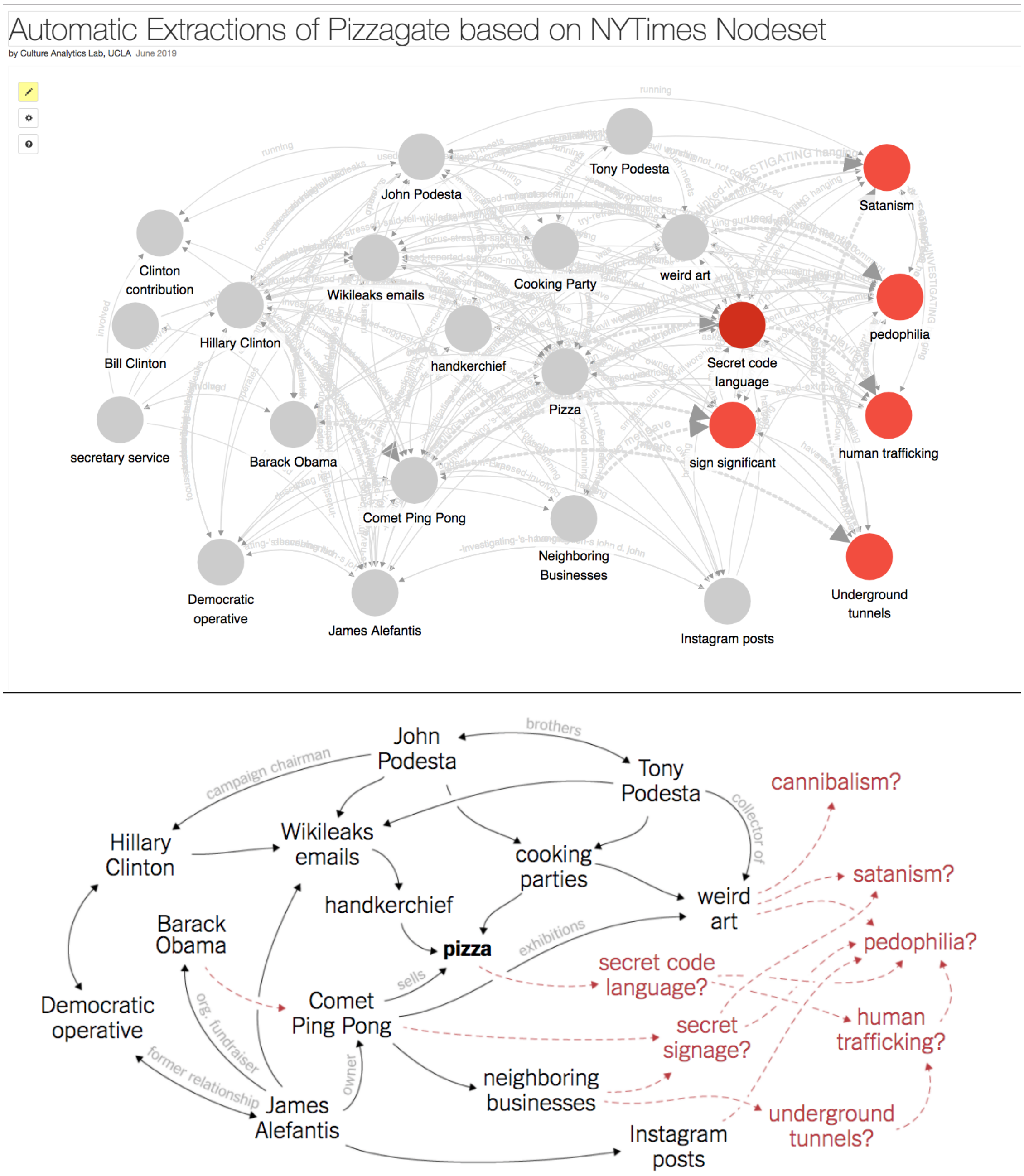

Figure 4.13: **Comparison of NY Times graph and auto-generated graph.** A: Simplified version of the NY Times hand drawn graph of the Pizzagate conspiracy theory. B: A version of our narrative framework graph limited to those supernodes and relationships in the NY Times graph that we discovered automatically. Visualized using Oligrapher. A summary relationship network where edges with most significant labels and subnodes/supernodes of interest is shown in Fig. 4.17.

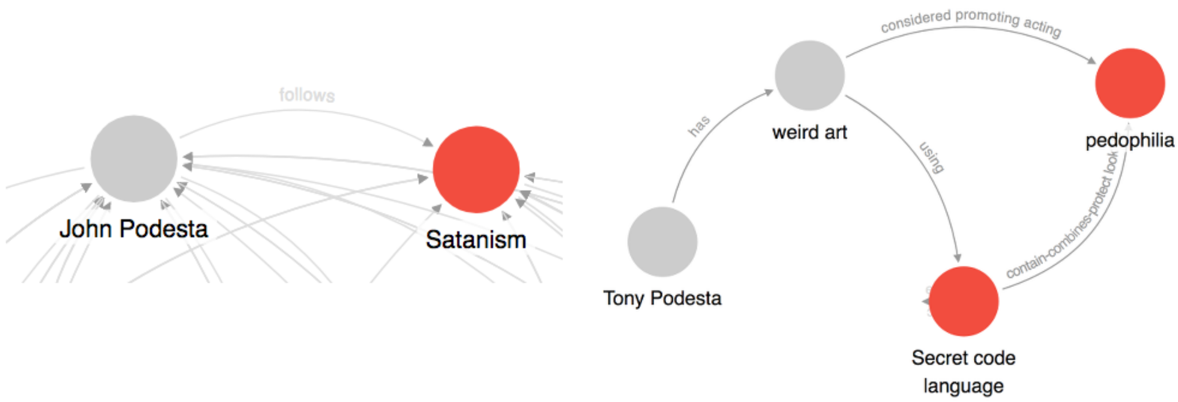

Figure 4.14: **A closeup of the labeled edges.** Excerpt from our auto-generated NY Times matched Pizzagate graph, revealing the relationships between a subset of nodes. A: the graph reveals that John Podesta follows Satanism, and B: that Tony Podesta owns weird art that uses coded language to promote pedophilia.

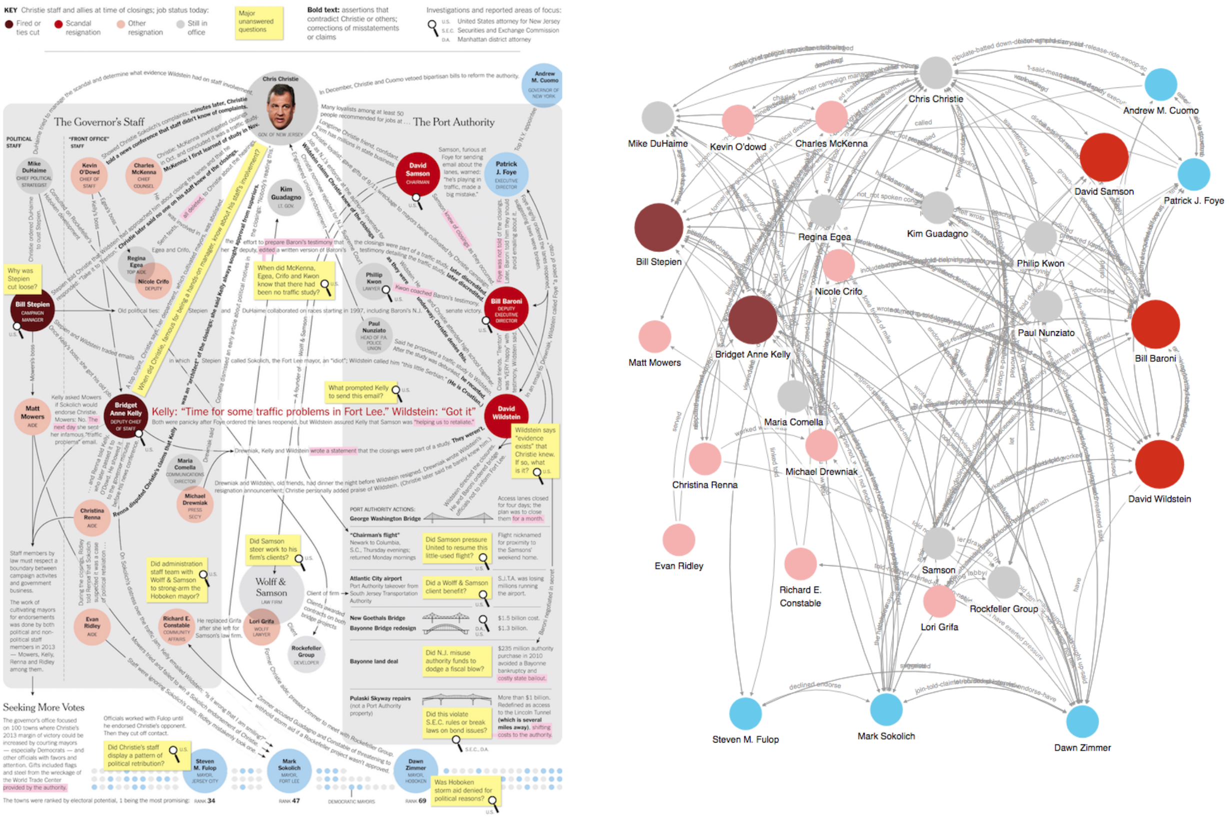

Figure 4.15: **Comparison of the NY Times Bridgegate graph and auto-generated graph.** A: Simplified version of the NY Times hand drawn graph of the Bridgegate conspiracy B: Version of our narrative framework graph limited to those supernodes and relationships in the NY Times graph that we discovered automatically. Nodes on the right have been sized and colored based on the NY Times graph to facilitate comparison. See Fig. 4.16 for significant relationships extracted by our automated method.

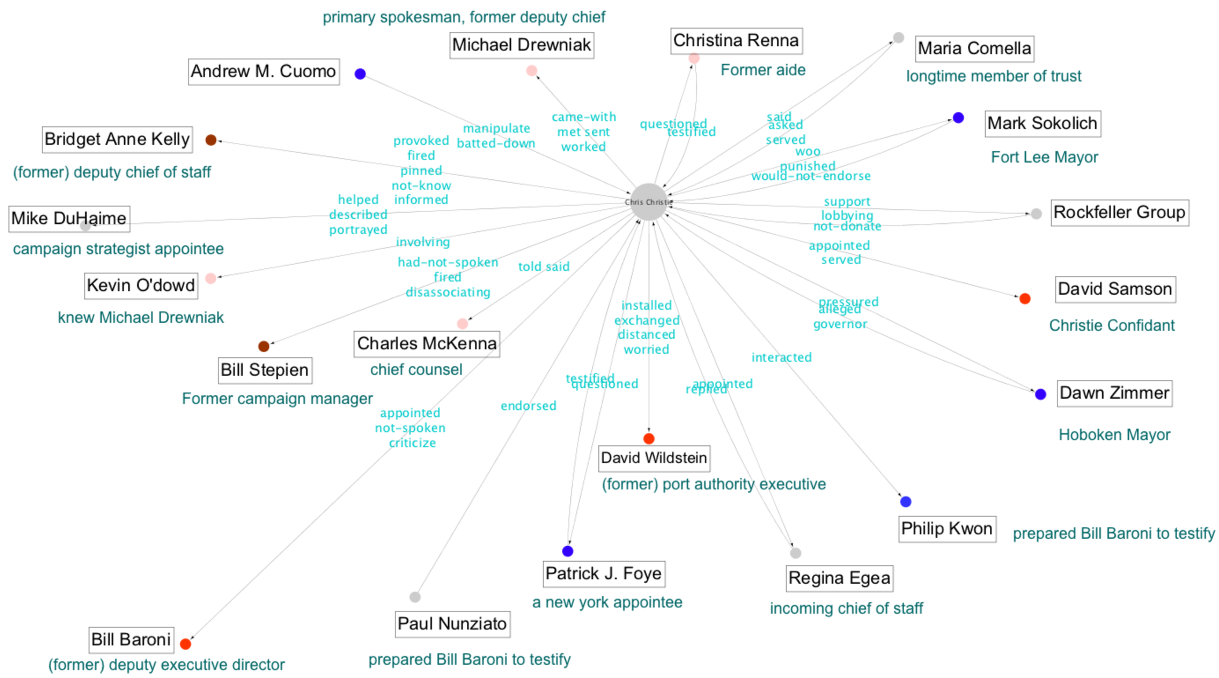

Figure 4.16: **Comparison of Relationship labels generated by our automated methodology with the the NY Times Bridgegate graph (see Fig. 4.15).** Most significant relationship labels from the "Chris Christie" node to other nodes in Fig. 4.15 are displayed here. For each node, we also include one descriptive phrase that was found in an automated manner by our software. These descriptive phrases match very closely the roles portrayed in the NY Times Bridgegate graph (see Fig. 4.15). As in other figures, the edge labels only pick the most important verbs for the associated relationship phrase. The rest of the words in the corresponding phrases provide the necessary context for meaningful interpretations of these verbs, For example, the verb "pinned" connecting Christie to Anne Bridgett Kelly, is part of the phrase, "pinned the blame on," which we extracted from the text.

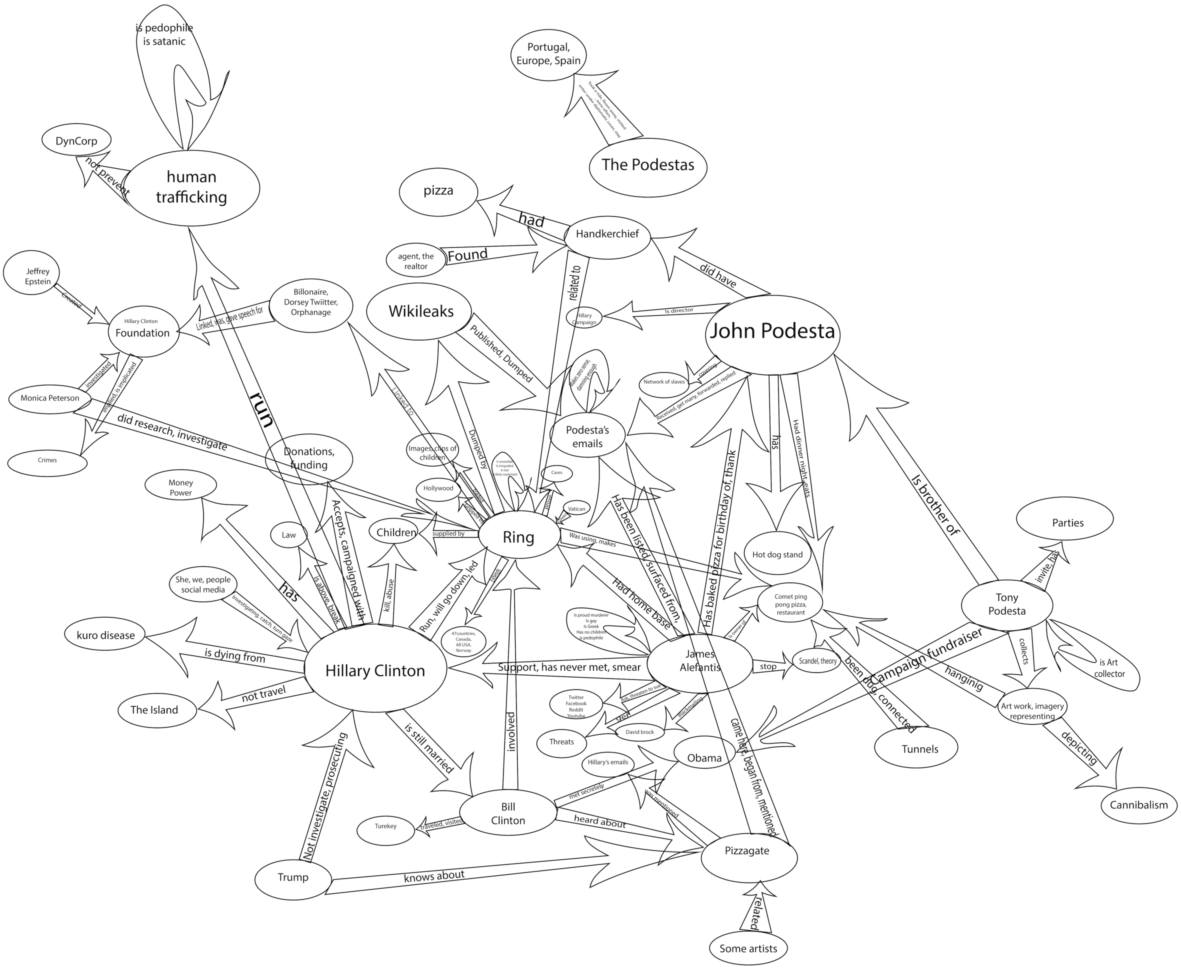

Figure 4.17: **Comparison of Relationship labels generated by our automated methodology with the the NY Times Pizzagate graph (see Fig. 4.13).** A subset of important supernodes and subnodes (as guided by the NY Times Pizzagate graph) and corresponding significant edges show that the automated discovery process not only covers most relationships that were summarized manually in the NY Times graph, but also provides nuanced relationships that can be used to provide different perspectives on how the conspiracy theory unfolded.

CHAPTER 5

StoryMiner for Entity-Relationship Extraction, Summarization, and Tagging from Temporal Transactional Tweets

"By not tweeting you're tweeting.
You're sending a message."

- Anonymous

Recent trends on Twitter's content reveal that users tend to share their daily experiences related to products and services. Their tweets provide information about certain interactions between several classes of entities, e.g. a customer experiencing failure with a mobile payment transaction at a merchant. Turning unstructured users' data into structured knowledge provides meaningful insights about public opinion on products and services. In this work, we tune StoryMiner to develop a Twitter-specific machine learning framework to summarize narratives in tweets. This summarization consists of four major phases; (i) building an entity/relationship (ER) extraction framework, (ii) designing a semi-supervised learning method to infer entity/relationship types, (iii) constructing and visualizing the summarizations in the form of story graphs with entities as nodes and their interactions as edges, (vi) monitoring networks' evolution and their trends in sentiment changes. We employ this model to a set of 527K tweets related to transactions between three entity classes: banks, mobile payments, and merchants. Our results show high precision in/recall on discovering the entities in these three classes, key relationships among them, as well as the sentiment and evolution of the pairwise relationships over time. [1]

5.1 Introduction

Inferring relations between entities in a transactional setting with regional and temporal variations enables us to discover certain structures and correlations. The structures are useful in many applications and decision-making processes such as product opinion retrievals, question & answering systems, semantic search engines, and text summarization platforms. Consider the scenario that a new wireless network provider goes public and attracts Twitter users' attention. People start propagating this news and discuss the locations this provider covers as well as the cell-phones it offers. Then people start comparing it with the other providers such as AT&T and T-Mobile and share their experience. In this scenario a possible set of entity types of interest could be "wireless network providers", the "cell-phones" they offer, and the "locations" they cover. Although the context and categories in these tweets are defined, the entity instances of these categories and their interactions are still unknown. Therefore, an automated system is in need to extract the entity mentions such as AT&T, Sprint's, iPhone 6, Los Angeles, tag them with the proper type, and analyze their pairwise relationships. For example, from the sentence "AT&T covers Los Angeles" it can extract "AT&T" and "Los Angeles" as the entities, tag them with "wireless service provider" and "location", and identify "cover" as the relationship between them. It can also discover whether "AT&T provides a contract for iPhone 6 users", or whether "this new provider serves Los Angeles" from the twitter users' post. Furthermore, such a knowledge extraction system can monitor the sentiment of people's experience with the service over time and identify their issues, concerns, or feedback. Note that the extracted information can help with mining people's opinions in various levels of granularity. As such, one can obtain a general assessment of sentiment polarity regarding a particular product or service, which can be invaluable for marketing or reputation management. In another case, a more granular objective could be

[1] In addition to my advisors, I would like to acknowledge the following people for their contributions to this Chapter: Misagh Falahi, Ehsan Ebrahimzadeh, Arash Vahaabpour.

to answer specific questions, such as "Which particular features do customers like best about a given product?"

There have been research trends towards the development of Twitter-specific information extraction frameworks [92, 93, 94]. However, these systems have a limited view of the data: none of these methods provide a comprehensive summarization framework to discover and aggregate entities, relationships, and their sentiment in a transactional environment. In this chapter, we discuss how StoryMiner could be used to address such needs. In a novel way, we take advantage of common limited structures in various transactional environments. These limitations are due to the low number of entity classes involved in the transactions, and their limited pairwise relationship types. For instance, in our setting, the data mainly consist of three classes of entities: banks, mobile payments, and merchants. These class of entities have certain type of relationships: whether a mobile payment works at a merchant, or whether it is supported by a certain bank. In particular, we are interested to mine user experiences with various mobile payments. Thus, we analyze the pairwise relationships that mobile payments have with banks and merchants from 527K tweets. We further monitor how these interactions evolve over time. Using relationship extraction along with sentiment analysis, we identify common issues in mobile payment systems. These issues may occur on either the user or merchant end. In the following sections, we provide further details on each component of our framework along with experimental results.

5.2 Related Work

Transforming unstructured tweets to structured knowledge has been studied extensively from different angles, including problem phrase detection, opinion mining, event discovery and entity/relationship extraction. Opinion mining and problem phrase detection are extensively studied on datasets of user reviews where reviews tend to be longer, more targeted and comprehensive than twitter reviews where there is a stringent limit on the tweet length [95, 96, 97]. Opinion mining in Twitter is primarily focused on sentiment analysis and is very well studied [98, 99, 100, 101]. Previous work on problem phrase detection in Twitter proposes an extraction method [102] to identify descriptions of problems from a corpus of tweets related to AT&T products. There are, however, no summarization efforts after problem phrase detection in [102]. In contrast to tweet-level analysis of these works, we focus on providing a holistic view of the entire narrative coupled with sentiment analysis.

Twitter event detection methods are mainly rooted in co-occurrence based topic modeling approaches [18, 19, 20], or information extraction techniques [103, 104, 105]. Generally speaking, events can be defined as real-world occurrences that unfold over space and time [106, 107]. These events can be earthquakes, concerts, a new phone release, or supporting a mobile payment by a bank. In [108], a topic-based summarization system is proposed for tweets about 6 entities - Obama, Lady Gaga, David Cameron, Nokia, Apple, Microsoft. Topics are identified as the set of related hash-tags to each entity. Such topic-based summaries, however, do not provide a granular view of the underlying interactions between entities mentioned in tweets.

Finally entity/relationship extraction has been previously studied in Natural Language Processing [4, 5, 6]. The common approaches are based on syntactic structures [5], or dependency tree structures [4, 7, 8]. Some people have used knowledge bases like Freebase [10], DPpedia [11], or Yago [12] to disambiguate entities and relationships [13]. There has been relatively little effort in designing Twitter-specific entity/relationship extraction frameworks [92, 93]. While extracting entities and relationships are an integral part of our framework, the main focus of this chapter is how to aggregate pieces of information to summarize the tweets' stories. Therefore, as we explain in section 5.3, we design our own Twitter-specific methods for extraction and aggregation of entities and relationships. These targeted methods will help us identify stories and narratives that are discussed in short, with at most 140 characters, which would not be possible with previously-used generic methods. At the time of this analysis, none of the previous work uses methods beyond frequency counts to aggregate the key relationships. For instance, [94] uses ClauseIE [8] to extract relationships, and aggregate them according to string matchings, and report the frequent extractions as important events. However, in this work, we go beyond frequency measures and design a machine learning model to capture rare yet important events.

To our knowledge, we are the first to propose Twitter-specific framework which couples knowledge graph information and contextual entity/relationship extractions with a semi-supervised learning model to summarize the narratives and opinions over time.

5.3 Methodology

In this section, we describe the pipeline of our system. Figure 5.1 shows the main components of this work: 1. Data preprocessing component which cleans the input text. 2. Automated entity extraction and typing which discovers entities through a Twitter-specific algorithm and tags them according to a knowledge graph - Freebase. 3. Relation extraction component which follows syntactic patterns in dependency trees of sentences to extract relationships. 4. Machine learning ranker which assigns importance measures to relationships, and 5. Results component which aggregates the extracted information to form story graphs, and monitors the sentiment change over time. In the following sections, we describe each component in detail.

5.3.1 Input and Data Preprocessing

A total number of 527K tweets (including retweets) are crawled from Twitter using keywords and hashtags around mobile payments such as Apple Pay and Samsung Pay. Among these, 202K tweets are unique and largely objective (65% of tweets are objective according to the nltk sentiment toolkit). These tweets mainly describe interactions among a) mobile payments and banks, e.g. *"@Barclays works with #ApplePay"*, or b) mobile payments and merchants, e.g. *"Tried using Apple Pay at McDonalds. Didn't work. Rubbish."* and *"Starbucks will soon accept Apple Pay #news #tech"*.

We apply common text cleaning techniques such as removing hashtags, links, non-ASCII characters, fixing encoding issues, and segmenting large sentences into smaller ones. We use regular expression techniques to combine various user accounts and mentions of the same entity. For example, given an entity Barclays, we replace its other accounts such as BarclaysUK and BarclaysHelp with its root representation: "Barclays". These rules mainly com-

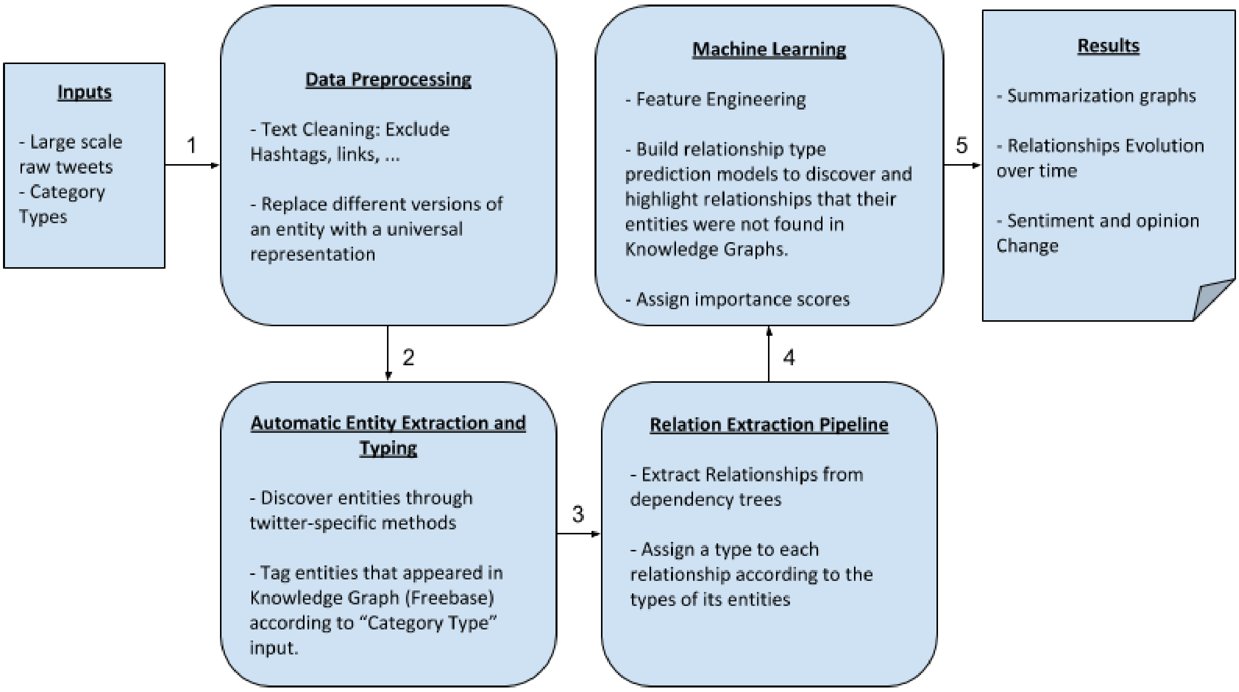

Figure 5.1: Pipeline of our Twitter-specific system.

bine accounts of related channels. Namely, channels whose account name ends with various qualifiers, such as two-letter country codes (e.g. BarclaysUK), or Help/News/Ask/Support keywords (e.g. BarclaysHelp).

5.3.2 Automatic Entity Extraction and Typing

A Twitter-specific entity model is proposed to combine different appearances of the same entity and tag it with the proper type. Our approach for extracting entities comes from the observation that users address entities through direct messaging or mentioning of an account. That is, they use @ sign to address the twitter account corresponding to the entities. Here is an example: *"@CVS_Extra Still no #NFC payment? Get with it. Nobody wants CurrenC. I'm taking my @googlewallet business to @Walgreens."*

Although users also use @ sign for retweeting or replying to someone, the @ sign's frequent occurrences in our data mainly refer to the three categories of banks, mobile payments, and merchants. This is not surprising, as the data was crawled centered around such types of entities. This leads to 1753 entities, which are mentioned at least in 2 unique tweets. Among them, using the Freebase Knowledge Graph API, we identified 86 banks, 194 merchants, and 16 mobile payment services.

5.3.3 Relation Extraction Pipeline

5.3.3.1 Relationships from Dependency Trees

In this work, we extract relationship tuples in the form of (arg1, relationship, arg2) between the main entities using only Stanford dependency trees[34]. We identify verbs as the relation phrases, their subjects ('nsubj' dependency relations or edges) as the first arguments, and their objects ('dobj' or 'prep' dependency relations) as their second arguments. Then similar to Ollie[4], we expand the arguments and relationships to include their attributes/modifiers. For example, given a sentence "Barclays will not enable Apple Pay," we start with the relational verb "enable" and then expand the relationship to include its modifiers "will not" and form the final relation phrase of "will not enable."

Predicating our work on the notations used in Stanford's dependency trees, we expand the arguments on 'neg', 'nn', 'amod', 'det', 'prep_of', 'num', 'quantmod', 'poss' edges. In case the headword of an argument is not a proper noun, we further expand it on 'infmod', 'partmod', 'ref', 'prepc_of' edges. For the relation phrases the expansion is on 'neg', 'advmod', 'mod', 'aux', 'auxpass', 'cop', 'poss', and 'prt' edges. These edges are decribed in detail in the Stanford's dependency annotation manual[85]. For example, from the tweet *"Samsung: LoopPay Breach Did Not Affect SamsungPay http://t.co/7W6C5Bn1pK"* we are able to extract (LoopPay {Breach}, Did Not {Affect}, {Samsungpay}) after cleaning it from its dependency tree (see Figure 5.2).

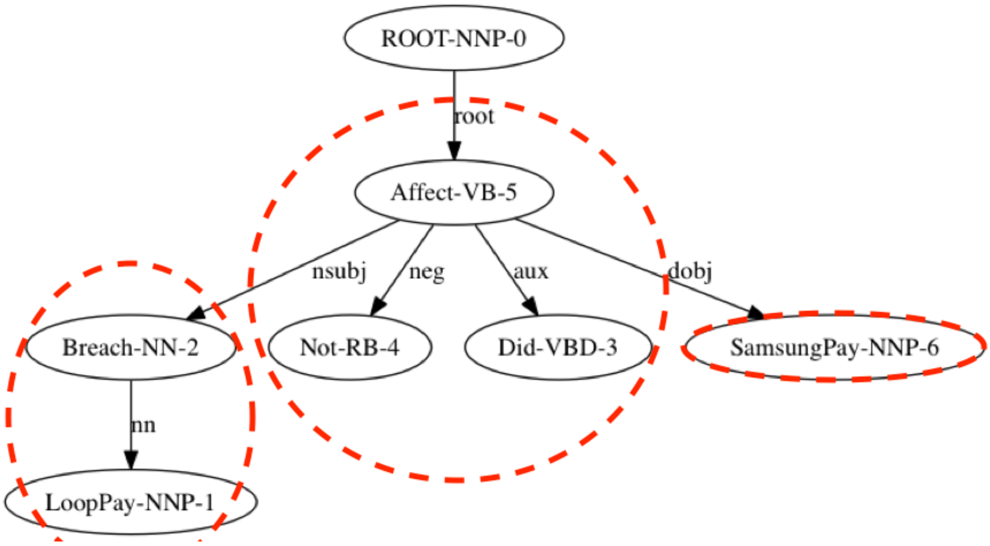

Figure 5.2: Example dependency tree after cleaning the tweet: “Samsung: LoopPay Breach Did Not Affect Samsung Pay http://t.co/7W6C5Bn1pK”

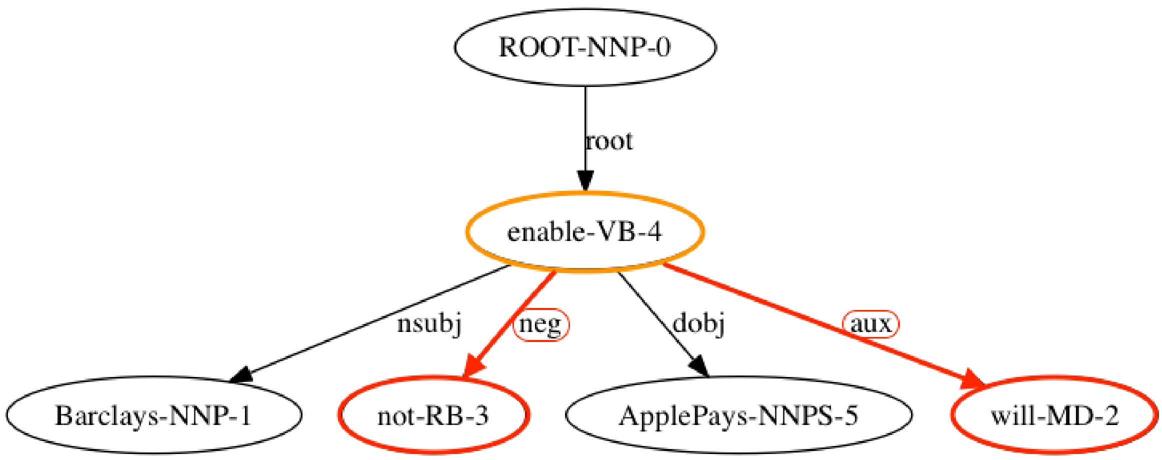

Figure 5.3: Demonstration example for how we expand arguments or relationships. In this example, we expand the relation verb “enable” by including “neg” and “aux” edges into the final relationship phrase - “will not enable”.

5.3.3.2 Relationship Type Assignment

As described earlier, often in analyzing a text corpus, the entity/relationship classes are known. A common problem is to find instances of the predefined classes of entities or relationships. For example, assume while analyzing a biomedical text, it is pre-specified that there are two classes; "disease" and "drug" and certain relationships such as "cure", "cause", or "prevent" between them. However, what needs to be discovered are the specific instances of each class, their corresponding pair, and the characteristic of the relationship between them (e.g. "dexfenfluramine hydrochloride", "is a treatment for", "obesity"). This is also a common scenario in a transactional environment where products (in our case, mobile payments) interact with other entity classes (in our case, banks and merchants). Identifying relationships among mobile payments and other types (banks and merchants) informs us about users' experiences and problems with their transactions. For example, we can identify specific banks that are in support of a particular mobile payment. To this end, we categorize relationships into three classes of interest: a relationship between "a mobile payment and a bank" (class MB), "a mobile payment and a merchant" (class MM), "other type of interactions" (class OTHER). We train a 3-class classifier to predict the category of a relationship. The classifier score represents an importance measure for a relationship. We would refer to this classifier as a ranker due to the ranking task it is performing: it helps in ranking and selecting relationships that are either MB or MM to be present in our final story graph. The training set for the classifier is generated via a semi-supervised approach: Starting from a seed set of entities with known types from querying the Freebase Knowledge Graph, we label relationships as one of the three MB, MM, or OTHER types.

5.3.4 Machine Learning Ranker: Entity-Relationship Classification

This section discusses evaluations and results of our classifier which is trained on the tagged relationships discussed in section 5.3.3.2, and jointly detects the type of a relationship or an individual entity. We further show how such a ranker is integral for transactional environments where there are often stories about entities that are not generally well-known. For example, small stores or recently established places are not yet included in knowledge graphs. However, they carry interesting stories that provide insight about user experiences that we don't yet know. To start, let us introduce our feature sets, machine learning models, and parameters used in our experimentation. Traditional ML models like Support Vector Machines (SVM), Logistic Regression (LR) and Tree-based classifiers have been evaluated to account for different characteristics of the data. Among these models, we identify Logistic Regression as the best model choice. The feature sets and parameters used in our experiments are described in Table 5.1.

Table 5.1: Configuration of our Machine Learning Ranker Pipeline

	Variations
Feature Set	• Unigram, Bigram, Trigram TF-IDF features from tweets (tweet text excluding the known entity mentions) • Unigram, Bigram, Trigram TF-IDF features from the relationship phrases • POS tag features from tweets • POS tag features from relationship phrases
Dimensionality Reduction	• SVD (k = 50, 100, 300)

5.3.4.1 Relationship Type Prediction

As described in section 5.3.4, there are a variety of choices in selecting features, models and parameters in our machine learning ranker pipeline. We compare accuracies of different settings using 10-fold cross-validation to come up with the best choice. A selected number of results are presented in table 5.2. We identified that tree-based models can better explain the outcomes. For example, one can easily trace the decision path of a tree-based classifier and find out the reasoning behind such prediction in a sensible manner. However, tree-based models are not among the top models in terms of performance. In our experiment, SVM and Logistic Regression models outperform the rest in terms of accuracy. SVM performs marginally better than LR, however, due to model simplicity (as our feature set is relatively large comparing to the number of our training examples), we pick LR as our choice for the ML model.

Furthermore, we found that the main features contributing to the performance of our predictions are the tf-idf features from relationships. This validates that there are certain relationships that could hold between two type of entities, and the relationship phrases between two entities reveal their types. This confirms that in a certain context such as interactions between a mobile payment and a merchant, the relationships are centered around specific phrases such as "working", "not working", "accepting", "supporting", etc. Using only tf-idf features from relationships, we achieve 77% accuracy on our test set, which is about 88% performance of our best combination - the 86.8% accuracy shown in Table5.2. To further provide a sense of how different features/configurations affect our prediction results, we summarize how the classification performance of the LR model changes as the configurations vary in Table 5.2.

Table 5.2: LR Classification Accuracy with Different Configurations based on 10-fold Cross-Validation.

SVD Param.	**Features**	**10-fold CV Acc.**
k=100	Tweets: Unigrams Tweets-POS: Included	84.0
k=100	Relationships: Unigrams Relationship-POS: included	76.0
k=100	Relationships: unigram, bigram, trigram Relationship-POS: included	75.5
No SVD	Relationships: unigram, bigram, trigram Relationship-POS: included	76.4
k=100	Relationships: unigram, bigram, trigram Relationship-POS: included Tweets: Unigrams Tweets-POS: Included	85.5
k=100	**Relationships: unigram, bigram, trigram** **Tweets: Unigrams**	**86.8**
k=300	Relationships: unigram, bigram, trigram Tweets: Unigrams	86.7

5.3.4.2 Entity Type Prediction

In this section, we show that the same prediction model can be used to identify the type of an entity. Every relationship in which an entity appeared as one of the arguments provides some insights about the type of the entity. Therefore, we prove that these insights in aggregate reveal an entity type. To formulate this problem, let us define the following notations:

- We let $\mathcal{E} = \{E_i : 1 \leq i \leq N\}$ denote the set of all relationships with size N, where

$E_i = (ent, rel, entc)$ represents the ith relationship (or extraction tuple).

- For each entity ent, we denote by $\mathcal{E}(ent)$ the set of all relationships with ent as an argument.
- We denote c_1 to refer to the class of relationships between banks and mobile payments, c_2 as the class of merchants and mobile payments interactions, and c_3 to be as any other type of relationships.
- We define p_{ij} as the probability given by the relationship type classifier that the jth relationship belongs to class c_i.
- We let *Type* be a mapping from entities to one of the four entity categories: Bank(B), Merchant(M), Mobile Payment(MP), Other(O).
- We denote the case of E_i having the ground-truth label of j as $E_i = c_j$.

As the first step, we take the classifier predictions as probabilities of each relationship E_i belonging to class j. Then we calculate the probability of the relationship belonging to its ground-truth class conditioned on the classifier output by using the Bayes rule (equation 5.2). The formulations are as follows, and they have been calculated empirically by looking at the distribution of p_1s and p_2s for each of the classes.

$$\Pr(E_i = c_j) = \Pr\{E_i \text{ is of class } j\} \tag{5.1}$$

$$\Pr(E_i = c_j | \text{clf outputs}) = \frac{\Pr(\text{clf outputs} | E_i = c_j) \Pr(E_i = c_j)}{\Pr(\text{clf outputs})} \tag{5.2}$$

Next we calculate the probability of an entity belonging to one of the four categories using Bayes theorem.

$$
\begin{aligned}
&\Pr(Type(ent) = \text{Category}|\text{Data, clf outputs}) \qquad (5.3)\\
&= \sum_{E_i \in \mathcal{E}(ent)} \Pr(Type(ent_i) = \text{Category}|\text{clf outputs}, E_i) \Pr(E_i)\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i) = \text{Category}|\text{clf outputs}, E_i)\\
&= \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{j=3} \Pr(Type(ent_i) = \text{Category}|\text{clf outputs}, E_i = c_j)\\
&\times \Pr(E_i = c_j|\text{clf outputs})
\end{aligned}
$$

In equation (5.3), $\Pr(E_i = c_j|\text{clf outputs})$ is the updated probability score calculated from equation (5.2). Furthermore, $\Pr(\text{ent=Category}|\text{clf outputs}, E_i = c_j)$ is calculated for different categories using Bayes' theorem as well. For example, the probability for the merchant category is as follows:

$$
\begin{aligned}
&\Pr(\text{ent=M}|\text{Data, clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{i=k}\sum_{j=1}^{j=3} \Pr(Type(ent_i) = \text{M}, entc_i|\text{clf outputs}, E_i = c_j)\\
&\times \Pr(E_i = c_j|\text{clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i) = \text{M}, entc_i|\text{clf outputs}, E_i = c_2)\\
&\times \Pr(E_i = c_2|\text{clf outputs})\\
&+ \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i) = \text{M}, entc_i|\text{clf outputs}, E_i = c_3)\\
&\times \Pr(E_i = c_3|\text{clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i) = \text{M}, entc_i \in MP|\text{clf outputs}, E_i = c_2)p_{2i}\\
&+ \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i) = \text{M}, entc_i|\text{clf outputs}, E_i = c_3)p_{3i}
\end{aligned}
$$

In the above equation, if $entc_i$ is included as a mobile payment in our predefined knowledge graph list, then the $\Pr(Type(ent_i){=}\mathrm{M}, entc_i \in MP) = 1$. Otherwise, we assume that $\Pr(Type(ent_i){=}\mathrm{M}, entc_i \in MP) = 0.5$.

The term $\Pr(Type(ent_i){=}\mathrm{M}, entc_i|\text{clf outputs}, E_i = c_3)$ depends on the distribution of merchants in the $c3$ class, which, assuming a uniform distribution among categories, is equal to 0.25.

Following the same set of equations, we find the probability of an entity belonging to the other categories by using the following equations:

$$
\begin{aligned}
&\Pr(Type(ent){=}\mathrm{B}|\text{Data, clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{B}, entc_i \in MP|\text{clf outputs}, E_i = c_1)p_{1i}\\
&+ \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{B}, entc_i|\text{clf outputs}, E_i = c_3)p_{3i}
\end{aligned}
$$

$$
\begin{aligned}
&\Pr(Type(ent){=}\mathrm{MP}|\text{Data, clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{MP}, entc_i \in B|\text{clf outputs}, E_i = c_1)p_{1i}\\
&+ \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{MP}, entc_i \in M|\text{clf outputs}, E_i = c_2)p_{2i}\\
&+ \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{MP}, entc_i|\text{clf outputs}, E_i = c_3)p_{3i}
\end{aligned}
$$

$$
\begin{aligned}
&\Pr(Type(ent){=}\mathrm{Other}|\text{Data, clf outputs})\\
&= \frac{1}{k}\sum_{i=1}^{k} \Pr(Type(ent_i){=}\mathrm{Other}, entc_i|\text{clf outputs}, E_i = c_3)p_{3i}
\end{aligned}
$$

The advantage of the above calculations is that it does not force an entity to be of a single type. Instead, it provides the likelihood of each entity belonging to a certain category in an unnormalized way, so that it's not necessary for the summation of these probabilities to be equal to one. This helps when an entity belongs to multiple categories. For example in our data set, "tesco" appeared both as a bank, and a merchant. Our type classifier assigns multi-label tags for "tesco" by providing a 0.70 and 0.51 probabilities for this entity to be of types bank and merchant respectively, which intuitively is aligned with our ground-truth data in which tesco is mentioned as a bank more frequently than a merchant.

Furthermore, in order to test our entity type detection mechanism, we manually create a set of "discoverable" entities. They are "banks", "mobile payments", or "merchants" that were not in Freebase at the time of our analysis. Therefore these entities represent a reasonable test set for evaluating the recall of our system for discovering rare entities. To create this set, we first filter the relationships with only one known entity from Freebase, then we look at the unknown entities with frequencies above 3, and tag them if they belong to one of the categories. This process results in 29 entities; among those, 10 are merchants, 2 are mobile payments, and 17 are banks. Two of these entities belong to multiple categories. We label them as their dominant category in our dataset. Thus, we label "tesco" as its most frequent type mentions - bank. Next we apply the likelihood ratio test (equation (5.4)) to assign a single category to each entity.

$$Type(ent) = \underset{\text{category}\in\{\text{B, M, MP, O}\}}{\arg\max} \frac{\Pr(Type(ent)\text{=category}|\text{Data, clf outputs})}{\Pr(Type(ent)\text{=category}|\text{Data})} \tag{5.4}$$

We achieve about 80% recall on our manually labeled "discoverable" entity set. This shows not only that our ML model is capable of retrieving important relationships, but also discovering entity types. The results of our ML pipeline will be used to generate the final story graph, including both frequently mentioned entities/relationships as well as the important (as of interest) but rarely mentioned ones.

5.3.5 Results

5.3.5.1 Aggregating Relationships to Generate Story Graphs

The abundance of information in a large stream of tweets consequently results in various entities with multiple relationships among them. The stories can be summarized in various granularity levels. In this work, we offer a hybrid approach which combines a fully automated mechanism based on a machine learning pipeline with additional manual inputs to generate the final story graph. In an automated way, the entities in the three classes are selected as the nodes, and their aggregated relationships (after stemming and semantic groupings) form the edges. Then, additional input specifies if the final graph should include only certain entities of interest. For example, "Barclays", "HSBC", "ApplePay" could be listed as the only entities to be shown in the final graph. This feature leads us to a sparse and easy-to-read story graph and allows researchers to focus on sub-stories around predefined sets of entities. Combining what researchers are interested in knowing about along with what the machine learning model thinks is of interest to the user (top rank relationships in MM and MB classes) will result in a story graph that visualizes key entities and their interactions. Figure 5.4 shows an example story graph retrieved from our system. It reveals how mobile payments interact with banks and merchants. The graph construction is performed in two phases: 1. Retrieving interactions where their participant entities are found in the knowledge graph (upper part of the graph). 2. Adding sub-stories around some discovered entities such as CurrentC, Etsy, or Eastern which were not found in knowledge graphs at the time of our analysis using the machine learning pipeline (lower part of the graph with orange nodes)

5.3.5.2 Sentiment Analysis and Relationship Evolution

In objective transactional tweets, both the relationships among entities and their sentiments are constantly changing over time. Thus, to enable deeper analysis, we offer a monitoring system that visualizes the frequent relationships over time and tags them with their sentiment. Therefore for a single or pair of entities, the evolution of their relationship and cumulative sentiment are visualized in a time series. As an example, we demonstrate this trend for one of our entities - Barclays bank. Figure 5.7 consists of two sub-figures. First, a manually created figure with the major tweets and their corresponding relationships over time. Second, an automated trend which aggregates relationships and their sentiment on a daily basis. It can be seen that the sentiment trend identifies frequently mentioned relationships over time and color-codes them according to their sentiment polarity: green for positive sentiment, red for negative, and gray for neutral.[2]

5.4 Concluding Remarks

In this work we propose a Twitter-specific information extraction framework that constructs story graphs based on machine learning models from transactional tweets. More specifically, this work enriches StoryMiners by using three additional components: developing prediction models to identify entity/relationship types and assign importance scores to the extracted relationships, leveraging a Logisitic Regression classifier to aggregate the relationships to form story graphs, and monitoring changes in relationships and their sentiments over time.

[2]Note that the sentiment time series get generated in a fully automated way with proper colors and labels. However, to clearly show the trend with no overlapping labels, we manually re-draw the automatically generated time series.

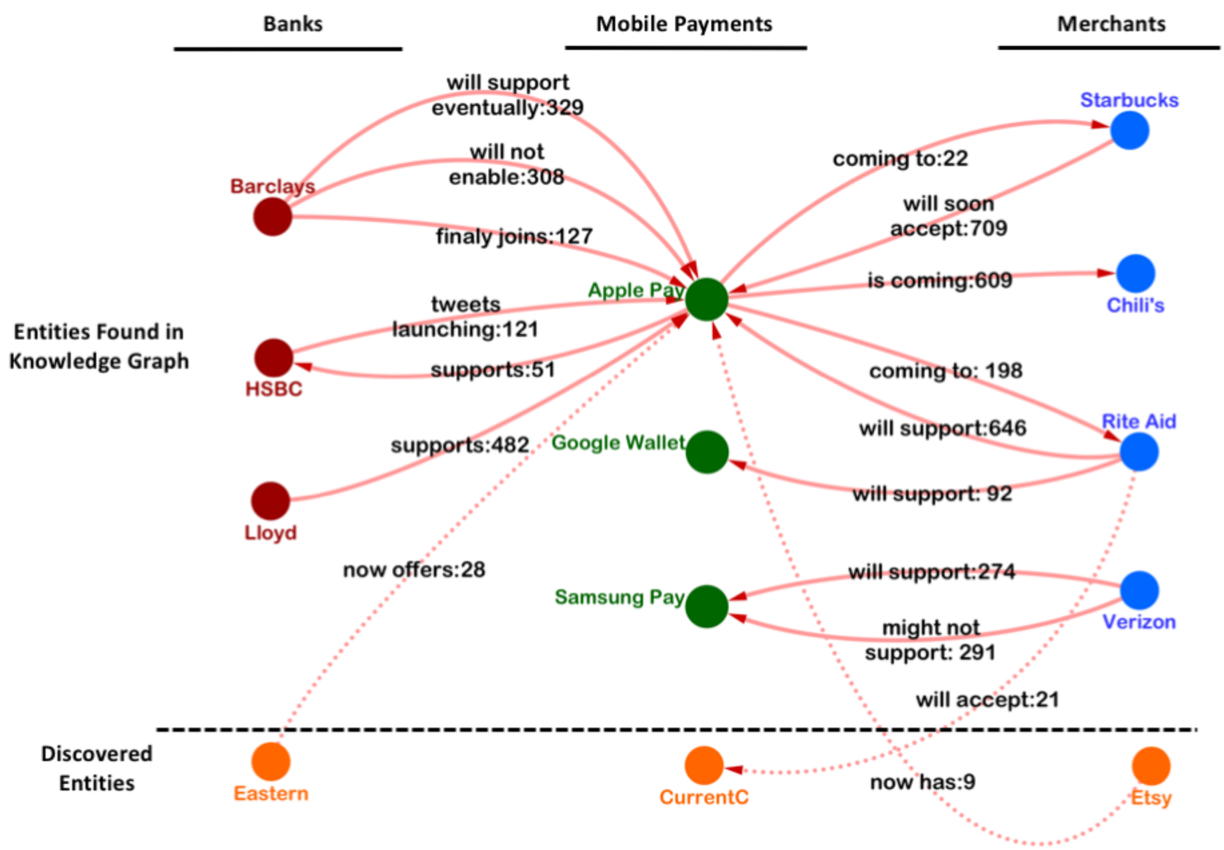

Figure 5.4: Story Graph generated from transactional tweets. The graph construction is performed in two phases: 1. Retrieving interactions where their participant entities are found in the knowledge graph (upper part of the graph). 2. Adding sub-stories around some discovered entities such as CurrentC, Etsy, or Eastern which were not found in knowledge graphs at the time of our analysis using the machine learning pipeline (lower part of the graph with orange-color nodes).

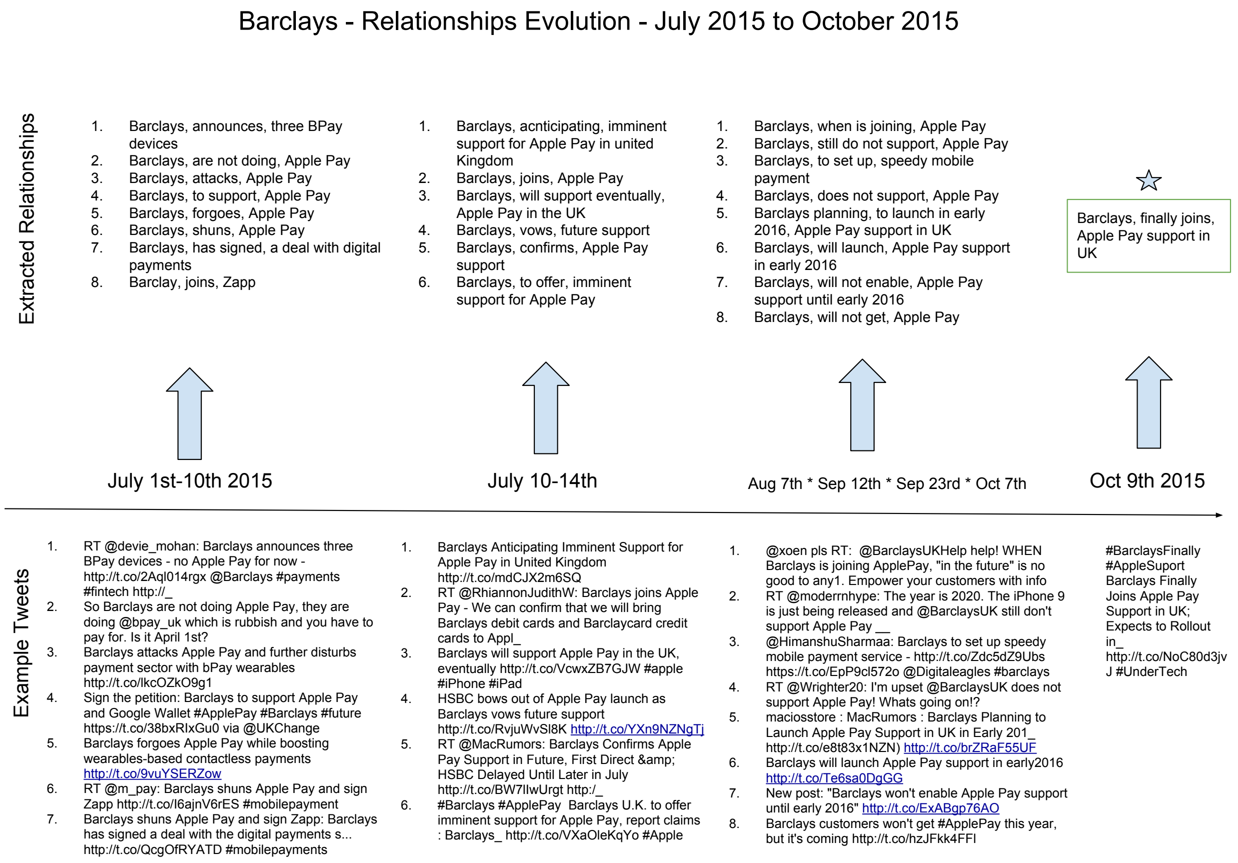

Figure 5.5: Grand-truth tweets about Barclays Bank along with their extracted relationships.

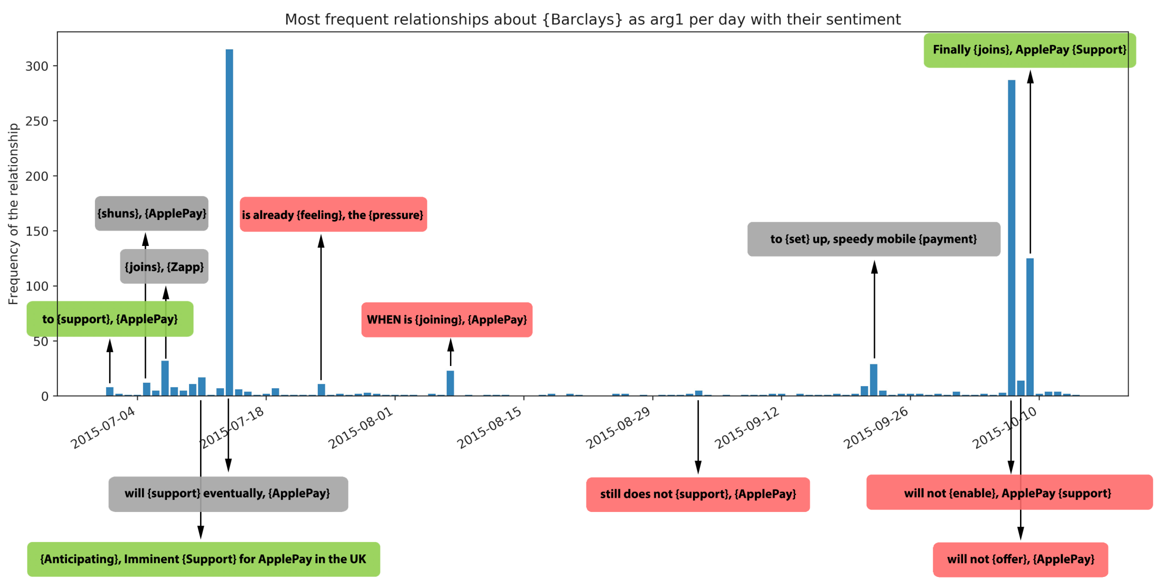

Figure 5.6: Automatically extracted relationships colored with according to their sentiments: green for positive, red for negative, and gray for neutral sentiments.

Figure 5.7: Relationship and sentiment evolution centering around an example entity: “Barclys” Bank. Comparing grand truth and automatically retrieved information.

CHAPTER 6

StoryMiner for Building Consensus Models of Literary Fiction from Reader Reviews

> "So many books, so little time."
>
> - Frank Zappa

In this chapter, we demonstrate how StoryMiner can be used to derive consenses models of literary fiction from Online readear reviews. StoryMiner automatically creates a summary of any novel in the form of a sequential actant-relation story graph based on the thousands of reviews of that novel on the social reading site Goodreads. This graph identifies the characters, places, and objects the readers mention (actants), their relative importance, and the pairwise connections between them (relationships). When possible, we determine the sequencing of events in the novel to derive a minimal plot structure for the work (sequence). The resulting reader consensus abstraction can, in turn, be compared to a professionally created one derived from a literature study guide such as SparkNotes. By comparing the reader consensus model of a particular novel with a ground truth model, we can determine what "sticks" when people read a novel. Readers' emphasis of certain actants also allows us to understand what "sticks out" for readers. In the aggregate, readers are good at reporting on main actants and the relationships between them. We find very low instances of readers introducing spurious actants or inter-actant relationships in their reviews. We also discover a tendency toward simplification of both the number of actants and their relationships, even in the aggregate. The reduction in the number of actants to some smaller subset of actants may be related to the cognitive constraints on social group size identified by evolutionary anthropologists. We hypothesize that, when people summarize what they have read for others, they focus on the main actants and interactant relationships, and that the size of these actant groups roughly correspond to social group sizes that characterize everyday life. Ultimately, this work might provide insight into how people (or classes of people) read and how they recount what they have read to others.[1]

6.1 Introduction

In 2006, Gregory Crane posed the provocative question, "What do you do with a million books?", a question that was motivated by the rapid digitization of library collections throughout the world, and the distribution of more and more books in digital form [109]. Subsequent studies on these massive digital text corpora focused on what was possible when machines did the reading [110]. Overlooked was the fact that people are still doing a great deal of reading. Although a recent National Endowment for the Arts survey reveals the long steady decline in literary reading in the United States, the number of American adults who read at least one work of fiction a year still hovers around 43%, after excluding work read for school or work [2]. Many of these readers are not silent but rather comment on the books they have read by posting reviews on social media venues such as Goodreads, providing an opportunity to understand reader response at internet scale. While studies of reader response to fiction have either been constrained to small groups of readers or focused on the affective qualities of reading, we develop a method for discovering reader-based consensus models of works of fiction based on the aggregation of thousands of reader reviews [111, 112]. Turning Crane's question around, we seek to answer both the general question, "What do you do with a million readers?", and the more specific question, "What does the collective memory of the readers of a novel look like?" To do this, we automatically create a summary of any novel in the form of a sequential actant-relation story graph based on the thousands of reviews of that novel on the social reading site Goodreads (Alexa global ranking 356).

[1] In addition to my advisors, I would like to acknowledge the following people for their contributions to this Chapter: Prof. Tim Tangherlini, Ehsan Ebrahimzadeh, Misagh Falahi.

[2] Results from the Annual Arts Basic Survey (2013-2015): https://www.arts.gov/artistic-fields/research-analysis/arts-data-profiles/arts-data-profile-10, Accessed February 10, 2019.

The graph identifies the characters, places, and objects the readers mention (actants), their relative importance, and the pairwise connections between them (relationships) [70, 74, 75]. When possible, we determine the sequencing of events in the novel to derive a minimal plot structure for the work (sequence). The resulting reader consensus abstraction can, in turn, be compared to a professionally created one derived from a literature study guide such as SparkNotes. By comparing the reader consensus model of a particular novel with the actual novel, we reveal what "sticks" when people read a novel. Since the ground truth abstraction does not include information about the importance of actants, readers' emphasis of certain actants also allows us to understand what "sticks out" for readers. Ultimately, this work might provide insight into how people (or classes of people) read and how they recount what they have read to others [113, 114].

6.2 Resources

We use reader reviews from Goodreads of four works of fiction: Frankenstein (1818); Of Mice and Men (1937); The Hobbit (1937); and To Kill a Mockingbird (1960) [115, 116, 117, 118]. The works were chosen from the list of the most frequently rated books on the site (number of ratings > 500,000). For each of the novels, we downloaded the maximum allowed three thousand reviews [3]. Although our initial corpus comprised sixteen novels, we chose these four novels for detailed analysis on the basis of the broad disparity in their narrative structures, large variability in the number of characters, and a broad range of character relationships. For example, The Hobbit can be characterized as a multi-episodic, linear narrative, that takes place across many different settings in an elaborate fantasy world, and includes a large cast of both human and non-human characters, instantiating an elaborate version of a standard fairy tale plot. Of Mice and Men, by way of contrast, is a short novella with a limited cast of characters that takes place in a highly localized realistic setting, and is a straightforward

[3]Followed Goodreads terms of service.

version of Vonnegut's "From bad to worse" plot [4]. Frankenstein , although told partly in flashback, has a strongly linear plot and a limited cast of characters, with a strong central figure and a relatively clear villain. Finally, To Kill a Mockingbird has an overlapping set of complex characters with multiple subplots. Reviewers who post to Goodreads have a variety of motivations for posting. The majority of reviewers use the site as part of a social network focused on reading, with the gender balance of active reviewers skewing slightly toward women [113][5]. There appear to be several categories of active reviewers on the Goodreads site, including students reviewing books as part of school assignments, members of book clubs, and people who aspire to becoming book reviewers. We make no discrimination as to classes of reviewers, but rather consider each review equally, as our goal is to understand the aggregate consensus model of a reviewed book [6]. At the same time, we recognize that reviews of a book are often conditioned by the pre-existing reviews of that same book. In certain cases, we recognize that these reviews may be influenced by the filmed adaptations of the target novels or professionally written summaries. [7]

6.3 Methodology

The reviews were harvested using a crawler specifically designed for this project. Not all reviews were useful since numerous posts were not reviews at all, but rather posts on different topics, simple spam, or not written in English. Other reviews were either too short to include meaningful content, or so garbled as to be unintelligible. After filtering the reviews, we were left with a corpus of 8693 usable reviews: *Frankenstein* (2947), *Hobbit* (2897), *Of Mice and*

[4]We use Vonnegut here simply as a means of shorthand, and do not attempt to fit any of these narratives to his plot models [119].

[5]These reviews were acquired prior to the acquisition of Goodreads by Amazon, and the subsequent changes in the Goodreads site to one used by authors to promote their own work. Dimitrov et al provide excellent comparative statistics for reviews on Goodreads and Amazon in the category biography, and suggest that Goodreads reviewers are more active than their Amazon counterparts, although their reviews tend to be shorter [120].

[6]We reserve the discussion of how different classes of readers reflect on their reading for future work.

[7]To wit, in the actant discovery process, Gregory Peck, who was the lead in the filmed adaptation of To Kill a Mockingbird, appears in the list of highly ranked actants in reviews for that novel.

Men (2956), and *To Kill a Mockingbird* (2893). We discovered two types of phrases in the reviews: (i) Opinion phrases that reflected the reader's opinions about the book, the author, or the various characters and events. Relationships extracted from these phrases are the dominant ones when aggregated over all readers' posts, which is not surprising given that these posts are intended to be reviews. (ii) Plot phrases that described what happened to a subset of the actants, and how they interacted with each other. These phrases contain both the actants and their relationships, and are of primary interest to us. The events in these fragmentary story summaries are not always presented sequentially. In an expert generated summary, such as SparkNotes, the textual rendition and sentence sequencing closely follow the actual flow of events in the book. In contrast, most reviewers summarize events that they find memorable, adding other pieces of the plot to complete their thoughts. To compensate for nonlinear recollection, reviewers frequently rely on changes in verb tense to indicate sequential ordering. Consequently, understanding the sequence of events presented in these reviews can be difficult even for a human reader. Standard Natural Language Processing (NLP) tools have notoriously low accuracy on this type of close-reading task, and can accurately extract only a subset of the actants, relationships and sequencing from any individual post.

Accordingly, we broke the consensus-based story reconstruction task into two steps: 1) the summary story graph step, and 2) the story sequencing step. In the summary story graph step, we aggregated the actants and their relationships detected in each post. First, we extracted the dependency trees over the corpus, and then extracted relationship tuples in the form of (argument 1, relationship, argument 2) between the main characters. We identified verbs as the relation phrases, their subjects as the first arguments, and their objects as their second arguments. Similar to the approach presented by the Open Language Learning for Information Extraction, we expanded the arguments and relationships to include more contextualize information [4]. These SVO (subject, verb, object) relationships were then tuned to capture story-specific syntactic forms of expressions. For example, we broke up three way relationships into multiple pairwise relationships, rendering the sentence, "Bilbo steals the ring from Gollum in the Misty Mountains," as three pair-wise relationships: : (Bilbo, steals, the ring); (Bilbo, steals << the ring>> from, Gollum); (Bilbo, steals <<the ring>> in , Misty Mountains).

In contrast to the expert-generated ground truth where relationships are expressed using only the most representative verbs, readers often express the same relationships using different phrases. For example, reader generated relationships between the actants “Bilbo Baggins” and “Gollum” include ‘what if had killed’, ‘meets’, ‘could have easily killed’, ‘tricks’ all of which express the same intention as the ground truth labels “Encounters” and “Escapes”. We used a combination of manual inspection and automated tools such as Verbnet to align the extracted relationship verb phrases with the summary verbs most closely matching the ground truth labels.

We used a similar process in mapping the extracted entity noun-phrases to actants. For example, the ground truth actant Gollum is referred to by various noun phrases in the reviews, including “gollum”, “smeagol”, “smagol”, and “smegol”, all of which are resolved as Gollum. Finally, the summary story graph was created by aggregating noun-phrases into actants, and relationship phrases into relationship verbs. This graph constitutes an end-state ranked consensus model of all actants and relationships. By ignoring sequencing information, each post can be viewed as the outcome of a random surfer model: An actant node is picked according to a certain distribution, and then a transition is made to another actant based on the probability or importance of the relationship edges connecting them. These transition probabilities, reflecting the importance of different relationships, are unknown parameters to be estimated from data. Since we have a large number of users, we can assume that in aggregate this random walk reaches a steady state distribution, and that an actant’s probability of being mentioned in a post is proportional to its occurrence frequency in the data. We then solve for the transition probabilities, allowing us to reconstruct the relative importance of relationships among actants from the data. These computed weights for relationships can be verified by an expert against the ground truth model to determine if they correspond to their importance in the underlying story. In turn, we can define a more accurate measure for what is retained collectively by the readers: instead of simply counting the ground truth relationships that are present in the reviews, we rank these discovered relationships by their importance weights. In the story sequencing step, sequencing information present in each post is aggregated to reconstruct a story line. For each review, we determine a precedence order: starting with the first sentence, we number the sentences in a post in increasing order. Next we construct an *aggregate relationship precedence network* as follows: For any post, if a relationship A occurs in a sentence that precedes the occurrence of relationship B, then we put a directed edge from a node representing A to a node representing B. Aggregating these relationships over all the posts produces a directed network where, for every pair of relationships (A,B), we can compute the number of directed paths of a fixed length from A to B and from B to A. Assuming that, on average, the order of recollection of relationships follows the original sequencing, any asymmetry in these pairwise counts is a good measure of their relative sequence in the story line. For stories with linear plots, such as *The Hobbit*, our approach yields surprisingly accurate timelines.

6.4 Evaluation

For evaluation purposes, we used the *SparkNotes* for each target work as the "ground truth" summary of that work. The character lists in *SparkNotes* aim to be comprehensive and the plot summaries, while abbreviated, provide an overview of the main inter-character relationships and a sequencing of the main events in the novel. [8]

Following standard machine learning practices, we evaluate our reconstruction results based on both the rate of recall of relationships and the false discovery rate. For each pair of actants (i, j) we define the following quantities: (i) the number of ground truth relationships between the pair, $g_{i,j}$, (ii) the number of ground truth relationships between the pair that are detected by our algorithm, $d_{i,j}$, (iii) the number of detected relationships between the pair that are correct but not included in the ground truth, $e_{i,j}$, and (iv) the number of detected

[8] Particularly in the case of students being asked to write a book review on Goodreads as part of a school assignment, we recognize the possibility that the reviewer may be relying on a reading of SparkNotes as opposed to the target novel. This phenomenon may be more likely to occur with Frankenstein (Open Syllabus rank 2 for US English classes at colleges and universities), Of Mice and Men (seventh most assigned book at US high schools) and To Kill a Mockingbird (fifth most assigned book at US high schools) given that these novels are frequently included in the curricula of US schools [121, 122].

relationships between the pair that are factually incorrect, $f_{i,j}$. $P_{i,j}$ is the estimated transition probability that a relationship involving actant i will also involve actant j (i.e., the relative importance of the directed relationship edge from i to j), and $\pi_{i,j}$ is the probability that actant i is mentioned in a post (i.e., the relative importance of an actant node i). First, we estimate each π_i by the relative frequencies of the actants in the detected relationships over all posts. Assuming the network structure of the ground truth network is true (i.e., the transition probability $P_{i,j} = 0$ for any relationship edge missing in the ground truth story network, ε_G), we estimate the remaining transition probabilities $P_{i,j}$ by solving the following optimization problem, estimating the relationship probabilities such that all are the steady-state probabilities of the actants in a random surfer model:

$$\left.\begin{array}{l}\sum_j P_{i,j}\pi_j = \pi_i \\ \sum_j P_{i,j} = 1\end{array}\right\} \text{for all } i \tag{6.1}$$

$$P_{i,j} \geq 0 \ \forall (i,j) \tag{6.2}$$

$$P_{i,j} = 0 \ \forall (i,j) \notin \varepsilon_G \tag{6.3}$$

These transition probabilities serve as a measure of importance for the relationships between the corresponding actants.

We define a Weighted Detection Ratio (WDR) as a collective recall measure where the numerator is the expected number of relationships per mention of any actant in the ground truth in the readers' posts:

$$WDR = \frac{\sum_i \pi_i \sum_j P_{i,j} d_{i,j}}{\sum_i \pi_i \sum_j P_{i,j} g_{i,j}} \tag{6.4}$$

We also compute the unweighted Detection Ratio (DR):

$$DR = \frac{\sum_{i,j} d_{i,j}}{\sum_{i,j} g_{i,j}} \tag{6.5}$$

where the sums are over all the actant pairs in the ground truth networks. [9]

For the false rate detection, we define analogous quantities, the Weighted False Detection Ratio (WFDR) and the False Detection Ratio (FDR):

$$WFDR = \frac{\sum_i \pi_i \sum_j P_{i,j} f_{i,j}}{\sum_i \pi_i \sum_j P_{i,j}(f_{i,j} + e_{i,j} + d_{i,j})} \tag{6.6}$$

$$FDR = \frac{\sum_{i,j} f_{i,j}}{\sum_{i,j}(f_{i,j} + e_{i,j} + d_{i,j})} \tag{6.7}$$

We find that most false relationships are due to mistakes in NLP processing, and primarily due to coreference resolution errors. Thus, for a sentence in a review for Of Mice and Men, "He ended up killing Lenny", the pronoun "He" resolves to "Lenny" and not "George", creating an error. Sometimes what appear to be improperly extracted relationships are actually erroneous recollections by the readers. (Table 1)

Eval vs	Relation Evaluation				False Discovery			
Story	Mice and men	Hobbit	Frankenstein	Mockingbird	Mice and men	Hobbit	Frankenstein	Mockingbird
WDR/WFDR	93%	59%	97%	57%	2%	2%	4%	7%
DR/FDR	59%	48%	60%	45%	12%	7%	7%	3%
DR/FDR (main actants)	77%	51%	71%	59%	8%	4%	4%	5%

Table 6.1: Evaluation results

6.5 Results

Many of the highest frequency actant-relationship pairs are "meta-pairs" that report on the relationships between the reader and the book, the author and the book, or the reader/author and one or more of the actants. These relationships largely motivate the review, yet provide little information about the story itself. Although simple actant extraction is thwarted

[9] We tried various other similar measures, both weighted and unweighted, that emphasized the counts $d_{i,j}$ and $e_{i,j}$ differently, but for the sake of brevity, we report only the WDR and DR measures here as they are balanced measures for evaluating importance-based recall of relationships.

by the high frequency use of pronouns in reader summaries, standard pronoun resolution methods greatly improve the recall of story actants, while also providing insight into the most commonly mentioned actants for each target novel. In one of the more dramatic examples, for the character "Bilbo" from The Hobbit, we were able to increase the capture of Bilbo-related interactions by 5407 instances. Individual reader reviews include a smaller number of actants than are present in the comprehensive ground truth actant list. These actants tend to be peripheral to the main plot or, as in the case with the group of dwarves with whom Bilbo travels, individual members of a larger group. In this case, while readers may refer to the entire group collectively, or only one or two members of the group by name, the ground truth includes both collective appellations, such as "the dwarves," as well as all the individual names, such as the names of all the dwarves including the popular Fili and Kili. A very small group of the discrepant actants that appear in reader reviews but not in the ground truth consist of "external actants", such as the author or actors who acted in film versions of the novel; since these actants have no relationship to the plot, we do not include them in our relationship discovery.

The relationships between actants reveal a high degree of consistency with the ground truth graph. The largest divergences consist of missed relationships, rather than the identification of non-existent relationships, although these occur occasionally. This latter group of relationships is often the attribution of a relationship, such as the killing of Smaug the dragon in *The Hobbit*, to an important character, such as Bilbo Baggins. Another small set of spurious relationships, including one that suggests that Jem killed Bob Ewell in *To Kill a Mockingbird*, are caused by reader confusion, "what-if" scenarios in reviews or, more commonly, incorrect pronoun resolution and aggregation. Apart from the relatively infrequent misattribution of relationships, the reduction in relationships aligns with the corresponding reduction in the number of actants connected to the central component of the story graph that is characteristic of reader reviews. Consequently, the aggregate story consensus graphs represent a subset of the ground truth graph, with missing edges, rather than missing nodes, being a main discriminating feature.

A relationship graph for the *Hobbit* highlights the reductive nature of reader reviews yet, even without sequencing, the overall plot of The Hobbit can be easily grasped (Figure 6.1).

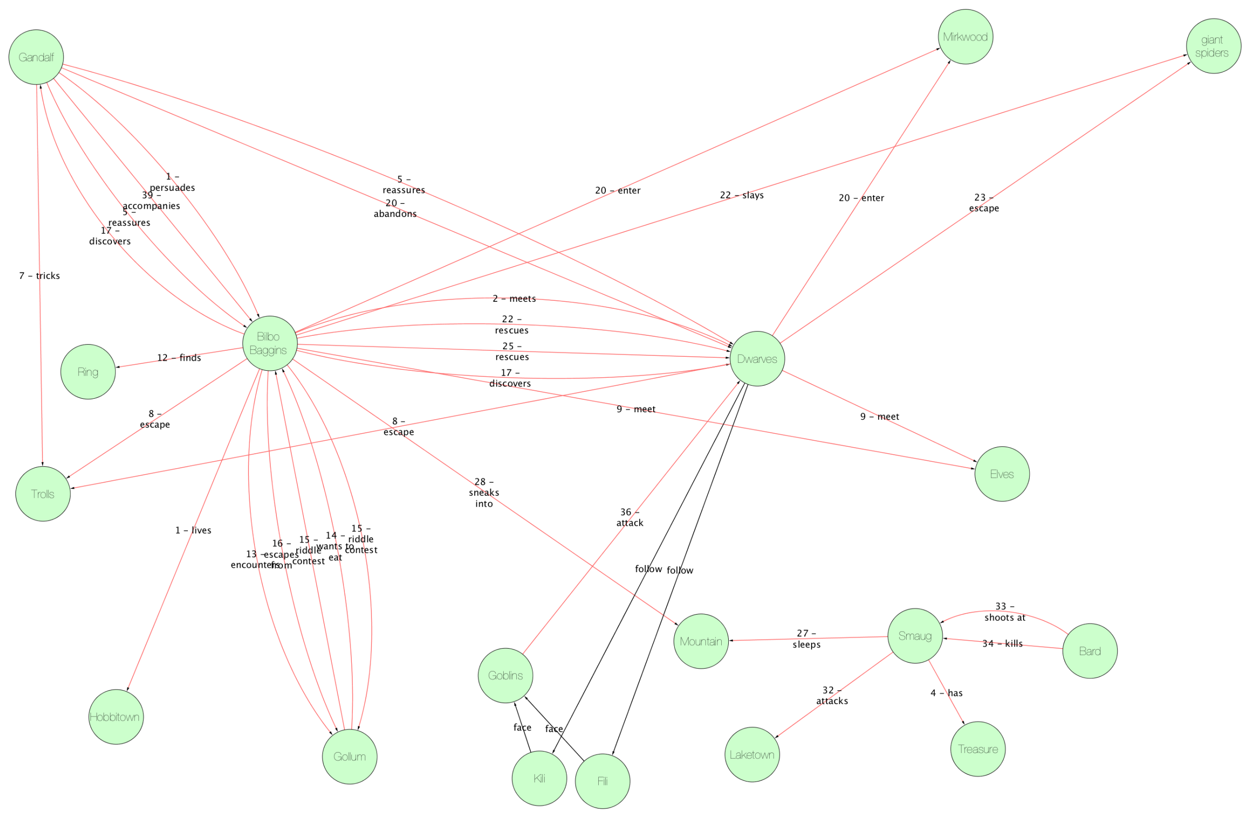

Figure 6.1: The story summary graph for The Hobbit. The edges in pink are those detected in the reader reviews and in the ground truth, while the black edges are those present in the reader reviews as separate edges (the relationship is more generically directed at dwarves in the ground truth than the specific dwarves, Kili and Fili).

The graph can be overlayed on the ground truth graph to highlight the differences between the reader reviews and the ground truth (See Figure 6.2).

Sequencing the events allows us to draw a series of minimal plot lines linking a series of relationships in the proper time sequence. Because of the brevity of reader reviews, and because not all reviews include the same sets of actants and relationships, we often discover branching plot lines that, in the aggregate, still provide a clear, albeit minimal, summary of the novel. (figure 6.3)

Following the central line in the graph above results in an abbreviated version of the novel, yet captures the proper sequence of many of the key events in the novel. Even for the more

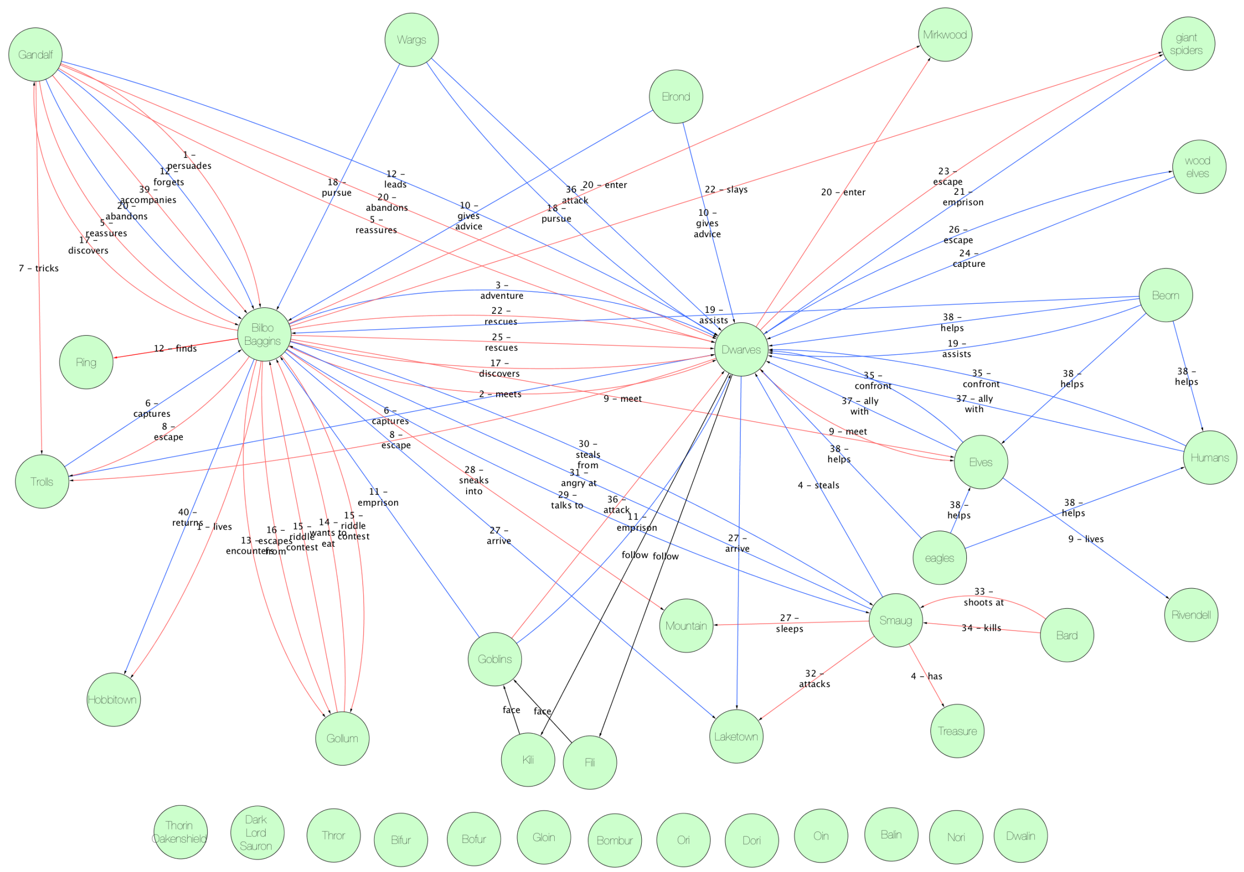

Figure 6.2: The story summary graph for The Hobbit. The common actant relationships between the reader reviews and the ground truth are drawn as red edges, while the unmatched relationships in blue The ground truth only relationships are far more numerous than the review-based graph.

complicated plot of The Hobbit, we discover a similarly abbreviated plot summary. (figure 6.4)

Similar reductive plot sequences are derived for *Frankenstein* and, despite its complexities and misdirections, *To Kill a Mockingbird*.

6.6 Discussion

The results support the notion that readers, when summarizing a novel, tend to reduce the scope of the story and to focus on the most memorable aspects of the plot, here modeled

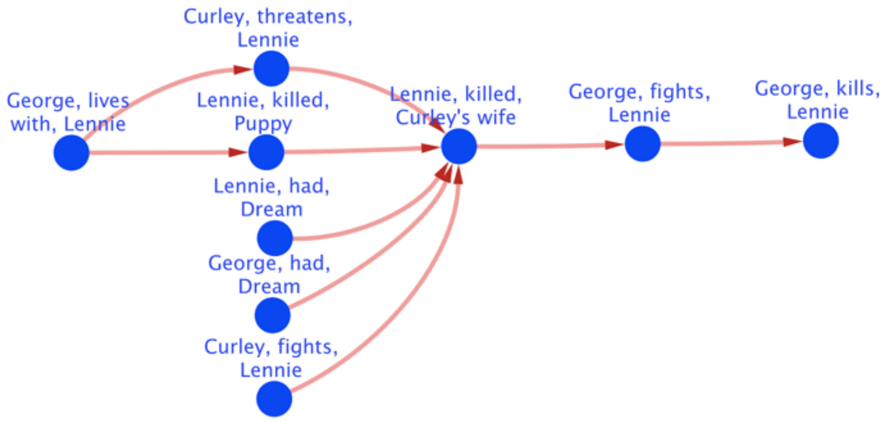

Figure 6.3: The story sequencing graph for *Of Mice and Men*.

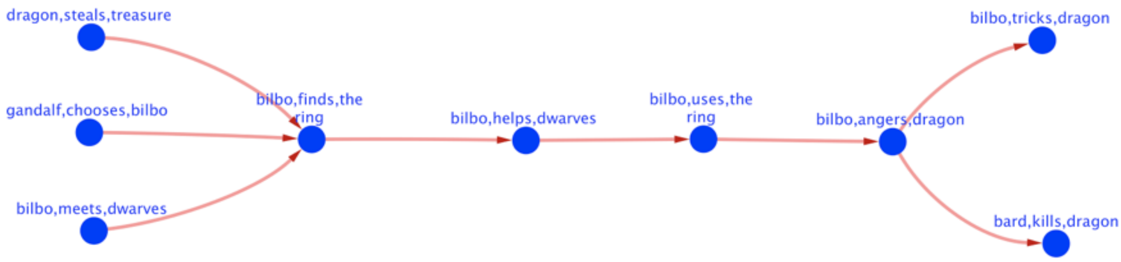

Figure 6.4: The story sequencing graph for *The Hobbit*.

as actant-relationships. In these reviews, people converge on a set of main actants and relationships that map well to a core set of actants and relationships in the ground truth summaries, suggesting that people are relatively adept at summarizing even complex plots. As part of this summarization, however, people tend to simplify. This simplification may be closely linked to cognitive limits on the number of real-world relationships that a person can keep in mind.

While the novels are fiction, there is general acknowledgement that readers "lose themself" in the taleworld of the novel. Robin Dunbar has suggested that the human brain imposes physiological limits on the number of friends one can keep in mind, and that there is a scaling ratio of roughly three that imposes limits on the numbers of increasingly more important relationships that a person can remember [123]. The upper limit in Dunbar's model is one hundred and fifty people, far beyond the number of characters in the books under consideration (and more in line with the Icelandic family saga), but the smaller geometric series of 3-5, 9-15, 30-45 that he derives are in line with the observed number of actants in our study. The larger groups align with the comprehensive actant lists for works such as *The Hobbit* and *To Kill a Mockingbird*, while the smaller groups of 9-15 are consistent with the more focused works of *Of Mice and Men* and *Frankenstein*. The smallest groupings of 3-5 are consistent with the main actants for whom there are multiple relationships in the reader consensus story graphs. It may well be that physiological constraints on memory coupled to the alignment of the experience of a taleworld to the lived experience of readers may help explain the tendency toward simplification in these reviews.

The story plots are also simplified in the reader reviews. Readers appear adept at reducing even complex plots, such as that in *To Kill a Mockingbird*, into relatively simple stories of conflict, strategies to address that conflict, and the result of the use of those strategies. The reduction of plot complexity may also be influenced by the abstraction of the novel in other media. For certain books, such as *The Hobbit*, recent films have been highly successful, and it is quite possible that movie watching has had some impact on reader reviews. The same may apply to the other books in this study, given the references to the actor Gregory Peck in reviews of *To Kill a Mockingbird*. Although we have not done so here, it may be interesting to compare reader reviews to the summary story graphs for those films as well.

6.7 Concluding Remarks

The approach we describe here is widely applicable to other crowd-sourced response sites such as *Rotten Tomatoes* and *Metacritic* that, much like *Goodreads*, allow viewers to present their own reviews of fiction, be it literature or film. An intriguing aspect of many of these sites is the propensity of reviewers to provide "plot summaries" as opposed to critical engagements of more sophisticated thematic analysis. While this may drive most literary scholars to the brink of insanity, it does allow us to consider questions regarding the popular engagement with literature and other forms of artistic production. Given the responses that people post, we can use the scale of these sites to derive insight into how people (or groups of people) not only read but also remember. Turning the process around, one may also be able to develop a dynamically updated crowd-sourced summary of a novel or film–as more people write reviews, the consensus summary would update, capturing the emphasis on actants, relationships, and events that commentators add. Such a system could act as a cultural response barometer since what people remember, and what they forget (or choose to leave out), can be telling indicators of popular engagement with art.

CHAPTER 7

Concluding Remarks and Future Work

> "You must be the change you wish to see in the world."
>
> \- Mahatma Gandhi

In this dissertation, we proposed StoryMiner, an automated and scalable framework that discovers emerging narratives on social media and news sites predicated on an understanding of narrative models. Rooted in narrative theory, StoryMiner derives stories and narrative structures by automatically 1) extracting and co-referencing the actants and their relationships from the text by proposing an Open Information Extraction system, 2) assigning named-entity types and importance scores for entities and relationships using character-level neural language architectures and traditional machine learning models, 3) making use of context-dependent word embeddings to aggregate actant-relationships and form contextual story graphs in which the nodes are the actants and the edges are the actant-relationships, and 4) enriching the story graphs with additional layers of information such as sentiments or sequence orders of relationships. In this work, StoryMiner's methods and applications were described throughout three use-cases: summary of user product opinions and experiences from tweets, reconstruction of plot summaries of famous novels from online reader reviews, and identification of differences in narrative structures between fake and real conspiracies. Specifically, the main contributions of this work have been explained in detail throughout the chapters. In summary, they can be listed as follow:

- A sentence-level relation extraction system called **StoryMiner RelEx** that is particularized for story-specific relationships. It achieves comparable results with the state-of-the-art relation extraction systems in general domains (see Section 3.3). StoryMiner RelEx, however, is the only extraction system that is specifically designed to retrieve story-specific relationships and thus, offers an additional set of novel procedures compared to the common relation extraction methods. For example, StoryMiner RelEx a) simultaneously couples sentence-level relation extraction and paragraph-level co-reference resolution to resolve pronoun arguments to the nouns they refer to, b) uses argument headwords and dependency tree information to map arguments to actants - the nodes in story graphs, - and c) breaks down n-ary relationships into pairwise relationships to form edges in story graphs (described in Sections 3.2.1, 4.5, 5.3.3.1).

- A **hierarchical actant model** to partition entities into hierarchical groups with similar contextual roles based on context-dependent word embeddings. In our group, we originally proposed an embedding approach based on explicit factorization of suitably generated entity-relation matrices along with a new exterior point method to solve the factorization problem. Our approach demonstrated superior clustering performance over embeddings obtained by the optimal matrix completion approach based on SVD (see [1] for more information). However, over the course of this dissertation, we further utilize our work with the state-of-the-art context-dependent word embeddings such as BERT and Flair [2, 3]. We further propose models and algorithms to learn actants hierarchy by clustering the context-dependent embeddings. The hierarchical actant model offers a novel way to identify narratives in various granularity levels, ranging from a broad story to a more specific one (described in Sections 3.2.1, 4.5).

- A **Story Model** to represent narratives in the form of networks (aka story graphs). Story graphs reveal narratives, narrative structure, the sequence orders of relationships, and other fundamental aspects of a narrative. Performing graph theory techniques on story graphs provides meaningful interpretations and results. For instance, ego networks or networks comprising a target set of actants reveal sub-stories surrounding certain actants, and graph connectivity distinguishes between a fake and a real news (described Chapters 3 and 4).

- A **fake news detection and summarization system**. Story Model captures useful information about real news, and fake news is often characteristically different in this model. Specifically, in real news the concepts in the narrative are more connected, whereas in fake news - because of the way people construct it - the concepts tend to be less connected. People cook up stories in parts, glue them together, and align otherwise unrelated domains of human interaction. Chapter 4 discusses our experimental results on Pizzagate and Bridgegate, a fake and a real news. We not only discover what was mentioned in summaries published by online newspapers, but we additionally identify the distinction in their narrative structures. (described Chapter 4).

- A **machine learning framework (aka StoryMiner) for story narrative detection** from text. This framework will be publicly available on GitHub repositories as well as on a demo webpage. StoryMiner achieves empirically powerful results in detecting stories from fragmentary posts. For instance, it automatically retrieves 93% and 97% of story plots from two famous novels - Of Mice and Men and To Kill a Mockingbird, respectively - from online reader reviews. The accuracy and effectiveness of StoryMiner have been verified via a set of computational experiments. Depending on the nature of the input text and the research questions, StoryMiner offers additional analysis techniques. For example in Chapter 5, StoryMiner summarizes user experiences with contact-less payment methods from tweets. Thus, it develops classification models to detect the type of an entity and a relationship and performs sentiment analysis to monitor views prevalent among the general public opinion (described in Chapters 4 to 7).

Throughout a series of experiments, this dissertation verifies that the underlying stories that are discussed in a large set of fragmentary posts are computationally detectable. Although NLP techniques such as dependency tree construction are noisy when dealing with malformed social media posts, but in aggregate high confidence narratives can be robustly retrieved in the presence of noise. Furthermore, we identify that the pairwise relationships and the structure of story graphs are important features for down-stream NLP tasks. For example, relationships are significant signals for detecting the type of an entity and network structure can reveal the type of a narrative (e.g. being fake or real).

The StoryMiner pipeline consists of multiple components, each of which include challenging research problems with active future directions. These ongoing NLP research topics include pronoun resolution, entity discovery, and relationship extraction. StoryMiner must consistently adjust its underlying models and methods according to future directions of its individual components. Most prior research on these components have been conducted separately, extracting relationships from a sentence or co-referencing an entity mention within a short paragraph without considering the corpus-level information. On the other hand, StoryMiner can establish research opportunities to improve the separate components by providing its holistic corpus-level knowledge. For instance, co-reference resolution can be done more accurately in our specific setup: a pronoun and the actual entity that it gets resolved to, should have similar relationships with the other entities throughout the corpus. Therefore we can do multiple passes over the text to make sure that pronoun resolutions are consistent with the relationships extracted for the entities they get resolved to. The corpus level information could also be useful for other components such as entity and relationship extractions. StoryMiner can refine and denoise the identification of its underlying components, and consequently the story graphs iteratively. Moreover, it is proven that looking at both left and right contexts while learning language representations is significantly important. As described in [2], the major improvement of BERT embeddings is due to their bidirectional pre-training of language representations as opposed to previous unidirectional language models. The context, however, is often defined as a set of surrounding words limited to one or few sentences. We suggest that by appropriately expanding the context to cover a larger set of left and right sentences (or even stories), the performance of language representations can be further enhanced. In this direction, StoryMiner can be used to represent what happened before and after a text that is being analyzed; and thus, it can provide a better representation of the text given its holistic context.

The years of the Trump presidency including the 2016 presidential election, have been marred by what increasingly has come to be known as fake news. Fake news became a means to deliberately produce erroneous narratives to polarize communities and encourage various types of real-world actions that might cause hazardous scenarios. For instance, rumors surrounding the Democratic party and the trafficking of children in what became known as Pizzagate, led not only to the resurgence of virulent and polarizing political discourse, but also motivated a young man to arm himself and attempt to "investigate" the trafficking ring at the pizza parlor. As we refine the system, we have strong indications that there are structural differences in the narratives of emerging news stories and emerging fictional stories. However, in order to distinguish between fake and real stories, the narrative structures must couple with other features such as linguistic features and user information. Fake News detection is a challenging research problem, and an impactful future direction to this work.

Finally, in order to provide public access to this work, we offer a set of github repositories under the Big Data and Complex Group group at UCLA[1]. We also designed a demo page[2] to further facilitate researchers to quickly use our system. The current workflow of the demo website is as follows: A user can upload or input a text document on our webpage via the interface shown in Figure 7.1. The user may add an additional input specifying "actants" (or group of entities) in the story graph. For instance, it can specify that in the context of the input text, the words "framework" and "StoryMiner" refer to the same actant and should be considered the same. After clicking on the submit button, the rankings of the relationships and entities are extracted (Figure 7.2). A user may download these rankings as csv files. Also, the page visualizes the final story graph as a d3 interactive graph in which the nodes represent the actants and the edges represent their relationships (Figure 7.3). This network can be exported as a json file for further offline analysis. Additional functionalities such as sentiment annotation could be included as part of the future work.

In conclusion, StoryMiner extracts narratives, which can shape ideological orientations and real-world decision-making in an automated and scaleable manner. This research could facilitate future directions to detect and avoid fake news, which has become a major issue in society in recent years. StoryMiner's implementation and documentation has been

[1]https://github.com/Roychowdhury-group

[2]http://big-data.ee.ucla.edu/demo

made publicly available. We hope that this research will allow other academic and industry researchers to extract structured knowledge from unstructured text to inform practical decisions.

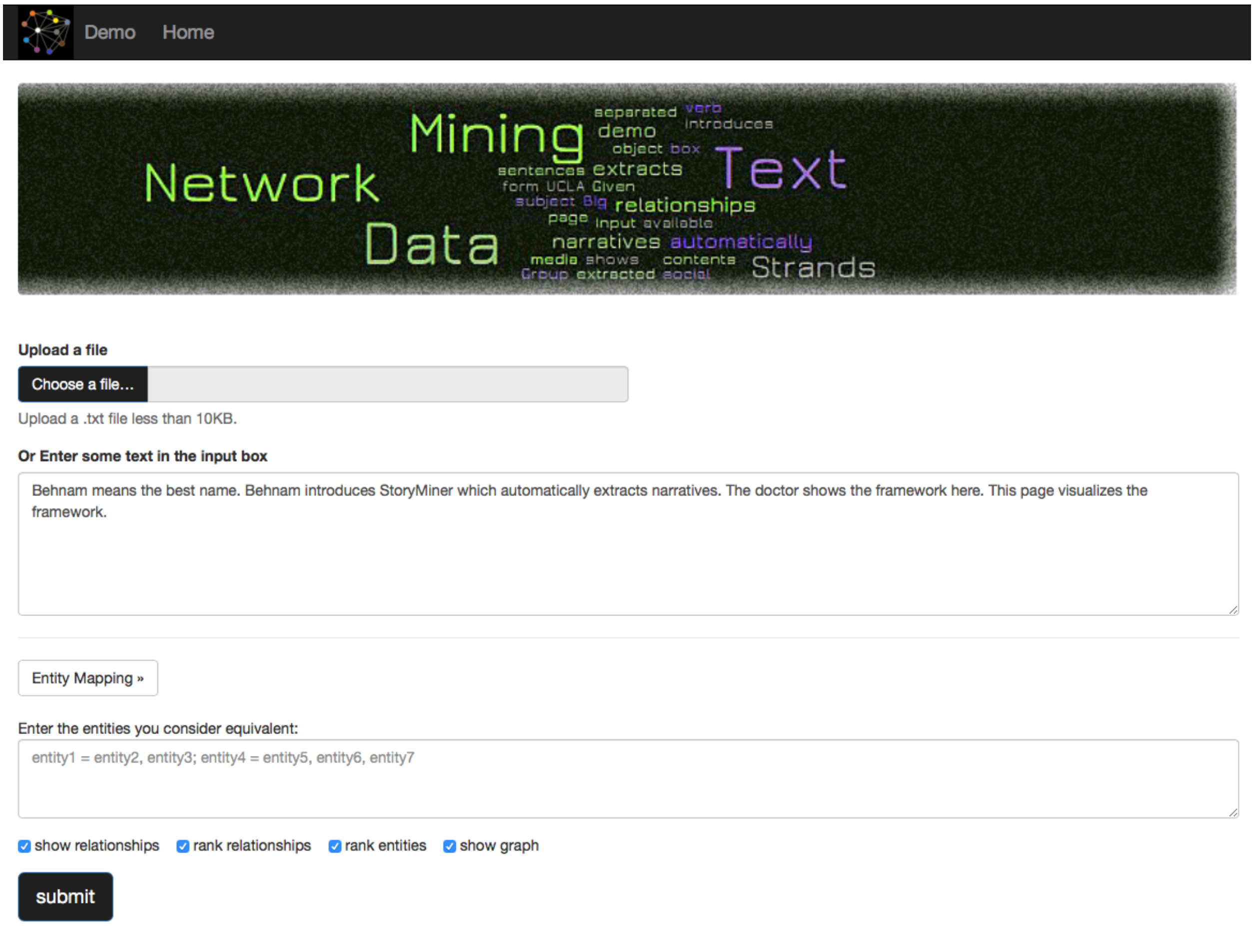

Figure 7.1: StoryMiner's demo website, in which a user can input their text document by copying and pasting into the input box or uploading it. If needed, the user can group some entities/mentions by specifying the groups in the Entity Mapping box. For instance, "StoryMiner" and "the framework" (or Behnam and Shahbazi) are considered to be the same, given the previous groups specified in the example input text.

Extracted Relationships

Download

	arg1	rel	arg2
0	{Behnam}	{means}	the best {name}
1	{Behnam}	{introduces}	{StoryMiner}
2	{StoryMiner}	automatically {extracts}	{narratives}
3	The {doctor}	{shows} here	the {framework}
4	This {page}	{visualizes}	the {framework}

Ranking of the Extractions

Download

	rel	count
0	(This {page}, {visualizes}, the {framework})	1
1	({StoryMiner}, automatically {extracts}, {narratives})	1
2	({Behnam}, {introduces}, {StoryMiner})	1
3	({Behnam}, {means}, the best {name})	1
4	(The {doctor}, {shows} here, the {framework})	1

Ranking of the Entities

Download

	entity	count
0	{StoryMiner}	2
1	{Behnam}	2
2	the {framework}	2
3	The {doctor}	1
4	This {page}	1
5	{narratives}	1
6	the best {name}	1

Figure 7.2: The demo page provides the extracted relationships, the ranking of the extractions, and the ranking of the entities.

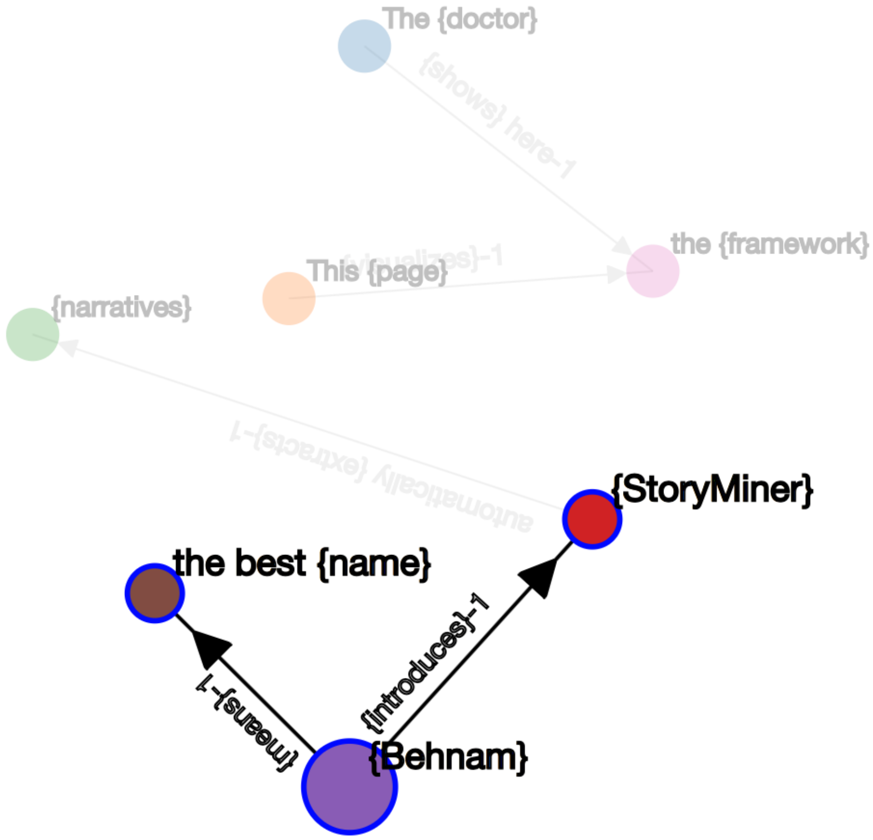

Figure 7.3: The interactive network, visualizing the story graph given the input text.

Bibliography

[1] Ehsan Ebrahimzadeh Houlasou. *A Computational Study of Story Narratives and Dynamics in On-line Social Media*. PhD thesis, UCLA, 2018.

[2] 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.

[3] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[4] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics, 2012.

[5] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[6] Fei Wu and Daniel S Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. Association for Computational Linguistics, 2010.

[7] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2006.

[8] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. ACM, 2013.

[9] Janara Christensen, Stephen Soderland, and Oren Etzioni. Semantic Role Labeling for Open Information Extraction. (June):52–60, 2010.

[10] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[11] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[12] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. 2012.

[13] Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004. ACM, 2015.

[14] Ayoub Bagheri, Mohamad Saraee, and Franciska De Jong. Adm-lda: An aspect detection model based on topic modelling using the structure of review sentences. *Journal of Information Science*, 40(5):621–636, 2014.

[15] Victor C Cheng, Clement HC Leung, Jiming Liu, and Alfredo Milani. Probabilistic aspect mining model for drug reviews. *IEEE transactions on knowledge and data engineering*, 26(8):2002–2013, 2013.

[16] Jonathan Chang and David Blei. Relational topic models for document networks. In *Artificial Intelligence and Statistics*, pages 81–88, 2009.

[17] Yuheng Hu, Ajita John, Fei Wang, and Subbarao Kambhampati. Et-lda: Joint topic modeling for aligning events and their twitter feedback. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[18] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *Icwsm*, 11(2011):438–441, 2011.

[19] Yuheng Hu, Ajita John, Dorée Duncan Seligmann, and Fei Wang. What were the tweets about? topical associations between public events and twitter feeds. In *ICWSM*, 2012.

[20] Jimmy Lin, Rion Snow, and William Morgan. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 422–429. ACM, 2011.

[21] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[22] Ce Zhang, Christopher Ré, Michael Cafarella, Christopher De Sa, Alex Ratner, Jaeho Shin, Feiran Wang, and Sen Wu. Deepdive: declarative knowledge base construction. *Communications of the ACM*, 60(5):93–102, 2017.

[23] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

[24] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[25] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[26] Philipp Cimiano and Universität Bielefeld. Knowledge Graph Refinement : A Survey of Approaches and Evaluation Methods. 0, 2016.

[27] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. pages 1–23, 2015.

[28] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

[29] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[32] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[33] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[34] Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592, 2014.

[35] Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R Bowman, Timothy Dozat, and Christopher D Manning. More constructions, more genres: Extending stanford dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 187–196, 2013.

[36] Sebastian Schuster and Christopher D Manning. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*, 2016.

[37] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[38] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

[39] William Léchelle, Fabrizio Gotti, and Philippe Langlais. WiRe57 : A Fine-Grained Benchmark for Open Information Extraction. 2017.

[40] Nancy L Rosenblum and Russell Muirhead. *A Lot of People Are Saying: The New Conspiracism and the Assault on Democracy*. Princeton University Press, 2019.

[41] Michael Barkun. Conspiracy theories as stigmatized knowledge. *Diogenes*, page 0392192116669288, 2016.

[42] J Eric Oliver and Thomas J Wood. Conspiracy theories and the paranoid style (s) of mass opinion. *American Journal of Political Science*, 58(4):952–966, 2014.

[43] Cass R Sunstein and Adrian Vermeule. Conspiracy theories: Causes and cures. *Journal of Political Philosophy*, 17(2):202–227, 2009.

[44] Richard Hofstadter. *The paranoid style in American politics*. Vintage, 2012.

[45] Robert Brotherton, Christopher C French, and Alan D Pickering. Measuring belief in conspiracy theories: The generic conspiracist beliefs scale. *Frontiers in psychology*, 4:279, 2013.

[46] Jan-Willem van Prooijen and Michele Acker. The influence of control on belief in conspiracy theories: Conceptual and applied extensions. *Applied Cognitive Psychology*, 29(5):753–761, 2015.

[47] Viren Swami, Adrian Furnham, Nina Smyth, Laura Weis, Alixe Lay, and Angela Clow. Putting the stress on conspiracy theories: Examining associations between psychological stress, anxiety, and belief in conspiracy theories. *Personality and Individual Differences*, 99:72–76, 2016.

[48] Karen M Douglas, Robbie M Sutton, and Aleksandra Cichocka. The psychology of conspiracy theories. *Current directions in psychological science*, 26(6):538–542, 2017.

[49] Gordon Pennycook, Tyrone D Cannon, and David G Rand. Prior exposure increases perceived accuracy of fake news. *Journal of experimental psychology: general*, 2018.

[50] Mark Fenster. *Conspiracy theories: Secrecy and power in American culture*. U of Minnesota Press, 1999.

[51] Ralph L Rosnow and Gary A Fine. *Rumor and gossip: The social psychology of hearsay*. Elsevier, 1976.

[52] David Robert Grimes. On the viability of conspiratorial beliefs. *PloS one*, 11(1):e0147905, 2016.

[53] Bill Ellis. *Raising the devil: Satanism, new religions, and the media*. University Press of Kentucky, 2015.

[54] Véronique Campion-Vincent and Jean-Bruno Renard. Conspiracy theories today. *Diogenes*, 249-250(1):1–264, 2015.

[55] Peter Knight. *Conspiracy theories in American history: An encyclopedia*, volume 1. Abc-clio, 2003.

[56] Gordon B Arnold. *Conspiracy theory in film, television, and politics*. ABC-CLIO, 2008.

[57] Panagiotis Metaxas and Samantha T Finn. The infamous# pizzagate conspiracy theory: Insight from a twittertrails investigation. *Wellesley College Faculty Research and Scholarship*, 2017.

[58] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[59] 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.

[60] Alessandro Bessi, Mauro Coletto, George Alexandru Davidescu, Antonio Scala, Guido Caldarelli, and Walter Quattrociocchi. Science vs conspiracy: Collective narratives in the age of misinformation. *PloS one*, 10(2):e0118093, 2015.

[61] Sander Van der Linden. The conspiracy-effect: Exposure to conspiracy theories (about global warming) decreases pro-social behavior and science acceptance. *Personality and Individual Differences*, 87:171–173, 2015.

[62] Daniel Jolley and Karen M Douglas. The effects of anti-vaccine conspiracy theories on vaccination intentions. *PloS one*, 9(2):e89177, 2014.

[63] Michael Barkun. *A culture of conspiracy: Apocalyptic visions in contemporary America*, volume 15. Univ of California Press, 2013.

[64] Timothy Tangherlini. Toward a generative model of legend: Pizzas, bridges, vaccines, and witches. *Humanities*, 7(1):1, 2018.

[65] Linda Dégh. What is a belief legend? *Folklore*, 107(1-2):33–46, 1996.

[66] Petula Dvorak. At a d.c. pizzeria, the dangers of fake news just got all too real. *Washington Post*, Dec 2016.

[67] Bill Ellis. Raising the devil: Satanism. *New Religious Movements, and the Media (Lexington: University Press of Kentucky, 2000)*, 2000.

[68] George Boole. *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. Walton and Maberly, London, 1854.

[69] Gary Alan Fine, Véronique Campion-Vincent, and Chip Heath. Rumor mills. *The Social Impact of Rumor and Legend. New Brunswick, Transaction*, 200:17–182, 2005.

[70] Timothy R Tangherlini, Vwani Roychowdhury, Beth Glenn, Catherine M Crespi, Roja Bandari, Akshay Wadia, Misagh Falahi, Ehsan Ebrahimzadeh, and Roshan Bastani. "mommy blogs" and the vaccination exemption narrative: results from a machine-learning approach for story aggregation on parenting social media sites. *JMIR public health and surveillance*, 2(2), 2016.

[71] Roja Bandari, Zicong Zhou, Hai Qian, Timothy R Tangherlini, and Vwani P Roychowdhury. A resistant strain: Revealing the online grassroots rise of the antivaccination movement. *Computer*, 50(11):60–67, 2017.

[72] Algirdas Julien Greimas. Éléments pour une théorie de l'interprétation du récit mythique. *Communications*, 8(1):28–59, 1966.

[73] AJ Greimas. Sémantique structurale, paris 1966. *ss. On trouvera des applications de la méthode d'analyse du récit de Greimas p. ex. chez H. Quéré et al., Analyse narrative d'un conte littéraire: Le Signe de Maupassant," Doc. de travail*, 9:200–222, 1968.

[74] Algirdas Julien Greimas. Les actants, les acteurs et les figures. *Sémiotique narrative et textuelle*, pages 161–176, 1973.

[75] Tzvetan Todorov. Les catégories du récit littéraire. *Communications*, 8(1):125–151, 1966.

[76] Albert Eskeröd. *Årets äring, etnologiska studier i skördens och julens trooch sed*, volume 6 of *Nordiska Museets Handlingar*. Nordiska Museet, Stockholm, 1947.

[77] Walter Anderson. *Kaiser und Abt: die Geschichte eines Schwanks*, volume n:o 42 of *Folklore Fellows. FF communication*. Suomalainen Tiedeakatemia, Academia Scientiarum Fennica, Helsinki, 1923.

[78] Carol J Clover. The long prose form. *Arkiv för nordisk filologi*, 101:10–39, 1986.

[79] John Laudun. Talk about the past in a midwestern town:“it was there at that time.”. *Midwestern Folklore*, 27(2):41–54, 2001.

[80] Manlio De Domenico, Mason A Porter, and Alex Arenas. Muxviz: a tool for multilayer analysis and visualization of networks. *Journal of Complex Networks*, 3(2):159–176, 2015.

[81] Matthew Skomorovsky and Michael Mintz. Oligrapher. `https://github.com/public-accountability/oligrapher`, 2019.

[82] Christian A Duncan, David Eppstein, Michael T Goodrich, Stephen G Kobourov, and Martin Nöllenburg. Lombardi drawings of graphs. *J. Graph Algorithms Appl.*, 16(1):85–108, 2012.

[83] P Shannon, A Markiel, O Ozier, N S Baliga, J T Wang, D Ramage, N Amin, B Schwikowski, and T Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, November 2003.

[84] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[85] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[86] Gregor Aisch, Jon Huang, and Cecilia Kang. Dissecting the pizzagate conspiracy theories. *New York Times*, Dec 2016.

[87] Bill Marsh and Kate Zernike. Chris christie and the lane closings: A spectator's guide. *The New York Times*, Apr 2015.

[88] Kate Starbird. Information wars: A window into the alternative media ecosystem. *Medium.(Mar. 2017)*, 2017.

[89] 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.

[90] Russell Muirhead and Nancy Rosenblum. The new conspiracists. *Dissent*, 65(1):51–60, 2018.

[91] Peter Pomerantsev and Michael Weiss. *The menace of unreality: How the Kremlin weaponizes information, culture and money.* Institute of Modern Russia New York, 2014.

[92] Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark Greenwood, Diana Maynard, and Niraj Aswani. Twitie: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 83–90, 2013.

[93] Leon Derczynski, Diana Maynard, Guiseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Inf. Process. Manage.*, 51:32–49, 2015.

[94] Gregory Katsios, Svitlana Vakulenko, Anastasia Krithara, and Georgios Paliouras. Towards open domain event extraction from twitter: Revealing entity relations. In *DeRiVE@ ESWC*, pages 35–46, 2015.

[95] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[96] Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[97] Elena Tutubalina and Vladimir Ivanov. Unsupervised approach to extracting problem

phrases from user reviews of products. In *Proceedings of the First AHA!-Workshop on Information Discovery in Text*, pages 48–53, 2014.

[98] Aliza Sarlan, Chayanit Nadam, and Shuib Basri. Twitter sentiment analysis. In *Information Technology and Multimedia (ICIMU), 2014 International Conference on*, pages 212–216. IEEE, 2014.

[99] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.

[100] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics, 2011.

[101] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49, 2016.

[102] Narendra K Gupta. Extracting descriptions of problems with product and services from twitter data. In *Proceedings of the 3rd Workshop on Social Web Search and Mining (SWSM2011). Beijing, China*, 2011.

[103] Swit Phuvipadawat and Tsuyoshi Murata. Breaking news detection and tracking in twitter. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 3, pages 120–123. IEEE, 2010.

[104] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. *ICWSM*, 11:401–408, 2011.

[105] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[106] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

[107] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

[108] Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 379–387. ACM, 2012.

[109] Gregory Crane. What do you do with a million books? *D-Lib magazine*, 12(3):1, 2006.

[110] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

[111] Richard J Gerrig. *Experiencing narrative worlds: On the psychological activities of reading*. Yale University Press, 1993.

[112] Reuven Tsur. Aspects of cognitive poetics. *Cognitive Stylistics: Language and cognition in text analysis*, 1:279–318, 2002.

[113] Mike Thelwall and Kayvan Kousha. Goodreads: A social network site for book readers. *Journal of the Association for Information Science and Technology*, 68(4):972–983, 2017.

[114] Lisa Nakamura. "words with friends": socially networked reading on goodreads. *PMLA*, 128(1):238–243, 2013.

[115] Mary Shelley. *frankenstein*. Pearson Education India, 2007.

[116] J Steinbeck. Of mice and men. new york: Covici & friede, 1937.

[117] John Ronald Reuel Tolkien. *The letters of JRR Tolkien*. Houghton Mifflin Harcourt, 2014.

[118] H Lee. To kill a mockingbird (book club ed.), 1960.

[119] Andrew J Reagan, Lewis Mitchell, Dilan Kiley, Christopher M Danforth, and Peter Sheridan Dodds. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1):31, 2016.

[120] Stefan Dimitrov, Faiyaz Al Zamal, Andrew Piper, and Derek Ruths. Goodreads versus amazon: The effect of decoupling book reviewing and book selling. In *ICWSM*, pages 602–605, 2015.

[121] Sandra Stotsky, Joan Traffas, and James Woodworth. Literary study in grades 9, 10, and 11: A national survey. In *Forum: A Publication of the ALSCW*, volume 4, pages 1–75, 2010.

[122] Arthur N Applebee. A study of book-length works taught in high school english courses. report series 1.2. 1989.

[123] W-X Zhou, Didier Sornette, Russell A Hill, and Robin IM Dunbar. Discrete hierarchical organization of social group sizes. *Proceedings of the Royal Society of London B: Biological Sciences*, 272(1561):439–444, 2005.