Unsupervised Event and Extremism Detection in Open Source Data Streams

A Dissertation
submitted to the Faculty of the
Graduate School of Arts and Sciences
of Georgetown University
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in Computer Science

By

Yifang Wei, M.E.

Washington, DC
August 7, 2017

Unsupervised Event and Extremism Detection in Open Source Data Streams

Yifang Wei, M.E.

Dissertation Advisor: Lisa Singh, Ph.D.

Abstract

Social media, blogs, and newspapers are all example of noisy, open source data streams. The amount of data in these sources is multiplying, making it challenging to make sense of them. In this dissertation, we focus on extracting two types of signals from these noisy data sets, events and extremists. While these two signals seem unrelated, they are actually useful for understanding potential movement in areas of conflict.

We have three major contributions in this dissertation. First, we develop methods for detecting targeted events, i.e. events of a particular type at a particular time and location from different forms of open-source data, specifically newspapers, blogs, and tweets. We propose both an offline and an online approach for identifying and summarizing events of the target domain occurring in a particular location from a large number of different news article sources. Next, we turn our attention to a noisier data stream, tweets. Unfortunately, the variability in sentence structure, vocabulary, and limited length require different methods to be proposed for this data stream. We propose a simple algorithm which leverages geotagged bursty term graphs to detect events from a tweet stream. Because Twitter is such a noisy domain and the Twitter API only gives samples of the tweet stream, we then focus on understanding the impact of sample size and noise level on location-based event detection. Finally, we consider extremist conversation on social media. We begin by identifying potential features about ISIS supporters on Twitter, grouping these features into categories,

and presenting a case study looking at the ISIS extremist group. We then propose an approach for identifying users who engage in extremist discussions online. We conclude the dissertation by discussing future areas of work and highlighting current challenges for using big data to help make progress on societal scale issues.

INDEX WORDS: Event Detection, Extremism Detection, Open Source Data Streams

Acknowledgements

When I started my PhD studies six years ago, a professor told me that getting a PhD would be a tough journey. I did not expect that getting a PhD would be easy, but the difficulties I faced on this journey were far beyond my expectation. As a student majoring in Electrical Engineering (to be specific, Integrated Circuit Design) in college and in a master's program, I stumbled through my transition into Computer Science. I even found myself in a tighter corner because I needed to get accustomed to a foreign language and a foreign culture. Nevertheless, Dr. Singh trusted me. She thinks I am a not only hard working but also a capable student. She always believed that I would earn this degree, eventually. It is her encouragement and support that helped me get through the difficulties, and finally reach the end of this journey. She is a talented researcher, a dedicated advisor, an effective leader, and a trusted friend. I respect her as much as I respect my mom. At this moment, I can imagine her bending over papers and looking at a computer screen, working out our research problems and polishing our research papers, as she has been doing for the past thousands of days. Sometimes I even feel that she gives me more support than I need. I know I can never pay her back for what she has done for me, and I know I may have disappointed or even frustrated her a few times, but I hope she is proud today.

I would also like to thank the other members of my dissertation committee: Dr. Susan, for providing her domain knowledge on forced migration and evaluation of the problems I am studying; Dr. Fineman, for opening the door to computer algorithms for me, which is the foundation of computer science; Dr. Zhou, for offering his insight on applications of big data. I am very grateful to for their advise and support through this process.

I would also like to thank two former colleagues, Dr. Guan, and Amin Teymorian. They helped me picking up the basics of computer science. Without their help, the completion of this dissertation could have been much more difficult. Another two colleagues, Sicong Zhang, and Jiyun Luo, gave me a lot of valuable advice on my research and moral support during my difficult times. Their help will always be remembered.

The work presented in this dissertation is supported in part by the National Science Foundation (NSF) Grant SMA-1338507, and the Georgetown University Mass Data Institute (MDI). I would like to acknowledge NSF and MDI for providing the funding for carrying out this research project.

I am the first generation college student in my family. Throughout all my years in school, my parents have supported me without reservation. My dad took me to extracurricular courses relentlessly day by day, year by year when I was in primary school and middle school. People around me all agreed that my dad made more of an effort on his kids' education than any other father. At least a quarter of the degrees I have earned should be attributed to him. My heartfelt gratitude also goes to my grandparents. Ever since my childhood, they have been telling me that they believed I would achieve a lot. At this moment, I believe they are proud of me, even those in heaven.

I have been studying in school for twenty-five years, i.e., a quarter of a century. That is quite a long time. Looking back over all these years, I find that the years I gained the most are the years I felt the most pain. As Emerson says, "That which we persist in doing becomes easier – not that the nature of the task has changed, but our ability to do has increased."

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Social media, blogs, and newspapers are all examples of noisy, open source data streams. The amount of data in these sources is multiplying, making it challenging to make sense of them. In this dissertation, we focus on extracting two types of signals from these noisy data sets, events and extremists. While these two signals seem unrelated, they are actually useful for understanding potential movement in areas of conflict.

1.1 Domain Motivation

While determining events and identifying extremist conversations from open source data are important problems in and of themselves, this dissertation is partially motivated by an initiative to understand the connection between open source data and leading indicators of forced migration [76]. Given the large amount of data available via social media, search engines, and more traditional data sources, we need to begin discussing how this information can be used to make progress toward impacting large societal scale problems. Our research group has an interest in considering how these data can be used to identify and forecast forced migration. Martin and Singh [76] have proposed a model that combines traditional casual factors of forced migration with the changing dynamics of variables that can be extracted from big data sources. The three main types of variables they suggest extracting and using as indirect indicators are events, buzz, and perception. Toward this goal, this work looks at targeted

Figure 1.1: Various types of open source data streams.

event detection (identifying events relevant to violence and other casual indicators of forced migration) and extremism detection (identifying individuals who are propagating extremist conversation to better understand the changing perceptions toward those individuals and the groups they support). Terrorist organizations, especially Islamic State of Iraq and Syria (ISIS), have been leveraging Twitter to encourage supporters to initiate terrorist attacks worldwide [75]. Their activities have also led to large-scale displacement in the Middle East. In order to better understand the relation between extremist discussions and migration, we need to identify users who engage in extremist discussions on Twitter.

1.2 Events, Extremism, and Open Source Data

An event is an occurrence that happens at a particular time and location. For example, the Paris terrorism attack on November 14, 2015 is an event. Events can be grouped together into types. This can be useful for individuals interested in finding particular

Figure 1.2: Forced migration in the Middle East.

Figure 1.3: The Paris terrorism attack on 2015-11-14.

types of events, e.g. violence, sports, etc. or what we will refer to as a *targeted event*. The main focus of this dissertation is developing methods for detecting events of a particular type from different forms of open-source data, e.g. newspapers, blogs, and social media. A secondary focus given our domain interest is on understanding the impact of extremism toward movement. Toward that end, this dissertation also investigates features and methods for identifying extremist conversation and those propagating the conversation on Twitter.

As we investigate these problems, we find that analysis of social media text differs from analysis of news articles or blogs because of the following differences in the types of data shared and the environment for sharing news:

Coverage: Social media have hundreds of millions of users. Each user can share messages with their friends and followers, and thus each user can serve as a *micro* news agency. However, unlike traditional news articles, social media coverage includes much more than news events. Platforms like Twitter include personal messages, ideas, feelings, real world events, and discussions related to almost every aspect of daily life. This makes it more challenging to identify events on social media. On the other hand, the broad coverage of social media posts provides an opportunity to find events that traditional news organizations ignore or may be unaware of. Examples include personalized events [99] and demonstration planning [89].

Immediacy: Most people access social media frequently, enabling them to post about events occurring around them in real-time. For example, the first tweet reporting the earthquake in Japan on Aug 18, 2009 was released within 2 minutes after the earthquake occurred [94].

Richness: Social media posts do not just contain textual descriptions. They come with rich metadata. For example, tweets are associated with geo-tags, exact timestamps, urls, etc. Also, Twitter contains different forms of network structures, which include, but are not limited to, following/follower networks, retweet/reply networks, and geotag/favorite networks. Previous literature has shown that integrating network structures can be useful for identifying events from social media [24] [74] [53] [47] and understanding extremism on social media [113].

Opinion/Perception: On social media, opinion, sentiment and stance are frequently shared, potentially obscuring event detection, but useful for extremism detection. In other words, data that is noisy for one task is useful for another.

While open source data are readily available and contain potentially rich information, there are challenges in using them, including:

Lack of Structure: News articles are free-form texts. They do not follow any pre-specified format when discussing an event. Also, a news report might cover a single event or multiple events. In a lot of cases, the description of one event is interwoven with references to other relevant events. When it comes to social media posts, this problem becomes even worse. Because of their brevity, social media posts usually consist of only a few sentences, or even one sentence. This lack of context adds difficulty to text understanding. Many existing approaches [54] [65] [66] [71] [73] [96] use a set of terms, phrases, or text segments with high occurrences to represent an event; however, such representations of events might not be informative enough for human readers: readers might find it difficult to reconstruct the whole picture of event based on these information fragments.

Unorthodox Text: Social media posts tend not to follow normal grammar rules. For example, a tweet "#expelturkeyfromnato #traitorstan #istandwithrussia" reports the event that Turkey Air Force shot down a Russian warplane on November 24, 2015; however, it is a just a set of keywords. To add to the difficulty associated with the lack of language structure, social media posts sometimes contain a lot of user-made-up words, or phrases, such as #expelturkeyfromnato in the above tweet.

Noise: Noise is pervasive in social media posts. An analysis conducted by a San Antonio-based market-research firm Pear Analytics of 2,000 English tweets originating from the United States shows pointless babbles make up 40% of the tweets, while only 4% are about news [1]. With respect to news articles, even those written by well known journalists are not completely free of noise. When it comes to a domain-specific event detection task, articles/posts reporting events beyond the target domain are another

[1] https://en.wikipedia.org/wiki/Twitter

source of noise and need to be filtered away. Understanding what is noise for different tasks is necessary for determining accurate features for the specific task.

Data Massiveness: Approximately 2 million blogs were posted online on a daily basis in 2012 [2]; thousands of tweets are sent every second on average nowadays [3]. Event and extremism detection in data of such a scale necessitates a highly efficient algorithm and a big data processing engine like Spark [18].

Reliability: Not everyone publishing news articles or social media posts is trustworthy. On Twitter, fake accounts are created, mostly for impersonating celebrities. A lot of high profile celebrities have found themselves impersonated on Twitter [4]. We need a way to determine the reliability of different sources.

1.3 Research Questions

This dissertation investigates the following research questions:

- How do we accurately detect events of a target domain occurring at a particular location from a news article corpus? From a Twitter data stream? How do we represent the detected events in an informative way so human readers can understand what the event is?
- Given the pervasive noise on Twitter, what are the effects of noise on detecting events from Twitter data? What are the effects of sampling on detecting events from Twitter data?
- How do we accurately identify Twitter accounts that engage in extremist discussions? What features can be used to improve the accuracy?

[2] http://www.digitalbuzzblog.com/infographic-24-hours-on-the-internet/
[3] http://www.internetlivestats.com/twitter-statistics/

1.4 Contributions

We make the following contributions in this dissertation:

- In Chapter 2, we analyze a large newspaper collection and investigate the strengths, weaknesses, and biases associated with open-source, big data analysis. This work was published at the 2014 ACM SIGKDD Workshop on Data Science for Social Good [111].

- In Chapter 3, we propose an offline approach that uses a sentence level semantic graph for and well known graph properties for identifying and summarizing events of the target domain occurring in a particular location from a large number of different news article sources. This work was published at the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) [112], and was awarded an Honorable Mention at DSAA. Based on our offline approach, we propose an online approach to identify targeted events that more efficiently updates the semantic graph. This work is currently under review at the International Journal of Data Science and Analytics [114].

- In Chapter 4, we propose a simple algorithm which leverages geotagged bursty term graphs to detect events associated with a particular location from a tweet stream. This work will be presented at the 2017 ACM SIGKDD Workshop on Mining and Learning with Graphs [107]. Because Twitter is such a noisy domain and the Twitter API only gives samples of the tweet stream, we then focus on understanding the impact of sample size and noise level on location-based event detection. This work is currently under review at the 2018 ACM International Conference on Web Search and Data Mining (WSDM) [109].

- In Chapter 5, we analyze potential features about ISIS supporters on Twitter, group these features into categories, and present a case study looking at the ISIS extremist group. Part of this work was published at the 2016 IEEE/ACM ASONAM Workshop on Social Network Analysis Surveillance Techniques [113]. Another part of it will be published as a book chapter in the Springer Surveillance in Action Collection [108]. We also propose an approach for identifying users who engage in extremist discussions online. Our approach uses detailed feature selection to identify relevant posts and then uses a novel weighted network that models the information flow between the publishers of the relevant posts. This work was published at the 2017 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) [110].

While each chapter details the contributions made, our most significant contributions that advance the state of the art are summarized below. On the problem of identifying events from open source data streams:

- We investigate the problem of overlapping location-based event detection. Our work is the first to investigate this problem.

- We propose a variant of traditional semantic graph structure for event detection that not only improves event detection accuracy, but also allows for easy event summarization.

- We study the impact of sample size and noise level on event detection. This work is the fist study of this type for event detection.

On the problem of identifying extremism from open source data streams:

- We identify and analyze features useful for identifying users with extremist views on Twitter.

- We introduce a novel information flow graph for this problem domain, allowing us to detect extremist conversations in an unsupervised setting.

1.5 Outline

The rest of the dissertation is organized as follows: Chapter 2 introduces the data corpa used in this dissertation, and describes a number of our interesting findings related to the strengths, weaknesses, and biases associated with open-source, big data analysis. In Chapter 3, we propose an offline approach and an online approach for identifying target events from a large number of different news article sources. Chapter 4 introduces our proposed approach for identifying events on Twitter and includes a study on the impact of sample size and noise level on location-based event detection. Chapter 5 analyzes potential features about ISIS supporters on Twitter, and introduces our approach for detecting people engaging in extremist discussions on Twitter. We conclude this dissertation and show the future directions in Chapter 6.

Chapter 2

Corpus Exploration

While it is unusual to describe data sets at the outset of a dissertation, we choose to do so to describe some of the issues associated with the data sources and to see if a connection exists between the data sources and our domain of interest, forced migration.

Approaches presented in this dissertation are evaluated on data from three open sources: the Expandable Open Source (EOS) news document archive [1], Text REtrieval Conference (TREC) document summarization track [13], and Twitter [14]. Corpora from EOS and TREC are used in Chapter 3 and corpora from Twitter are used Chapter 4 and Chapter 5 to evaluate the proposed methods. In this chapter, we first introduce these three open data sources, and the corpora we obtain from these sources. Then we present our understanding of possible connections between open source data and the possible leading indicators of forced migration.

2.1 EOS News Document Archive

Hosted and maintained at Georgetown University, EOS is a vast unstructured archive of over 600 million publicly available open-source media articles that have been actively compiled since 2006. New articles are added at the rate of approximately 300,000 per day by automated scraping of over 22,000 Internet sources in 46 languages across the globe.

As explained in Chapter 1, this dissertation is partially motivated by an initiative to understand the connection between open source data and leading indicators of forced migration. Specifically, we focus on forced migration in Iraq. Iraq has been exposed to continuing insecurity and displacement in recent years [95] [46], allowing for both retrospective and prospective analysis. We use a set of queries, all of which are Iraq-related, including Iraq, Iraqi, Gulf War, Saddam Hussein, etc, to retrieve 2.6 million English documents broadly relevant to Iraq published from Nov 22, 2013 to Feb 26, 2014. Figure 2.1a shows the number of documents per day in this corpus. We can see that approximately 20,000 documents are published per day on average. We will see that while these articles are relevant to Iraq, many of them are not relevant to the forced migration domain.

Table 2.1 shows the top 12 media outlets publishing articles in this corpus. There are a few key take aways. First, this corpus is not skewed towards a few sources. Instead, documents from these top 12 sources take up only 16% of the whole corpus, and these top 12 sources are from 11 countries, which is a strong signal that this corpus incorporates voices and views from all over the world. Figure 2.3a lists the top 10 countries in which media outlets publishing the most documents in the EOS corpus reside and the percentages of EOS documents associated with each of the 10 countries. Most of these countries are English speaking countries, which meets our expectation, since we are focusing in on an English news article subset of EOS. Documents from these top 10 countries make up only 38% of the whole corpus, further demonstrating the scattering of document sources. Figure 2.4a shows these top 10 countries on a world map.

While EOS does well with overall source distribution broadly, it does not do as well with respect to regional sources. None of the top sources are from Iraq. Because we want access to local information, local sources are important. When this issue

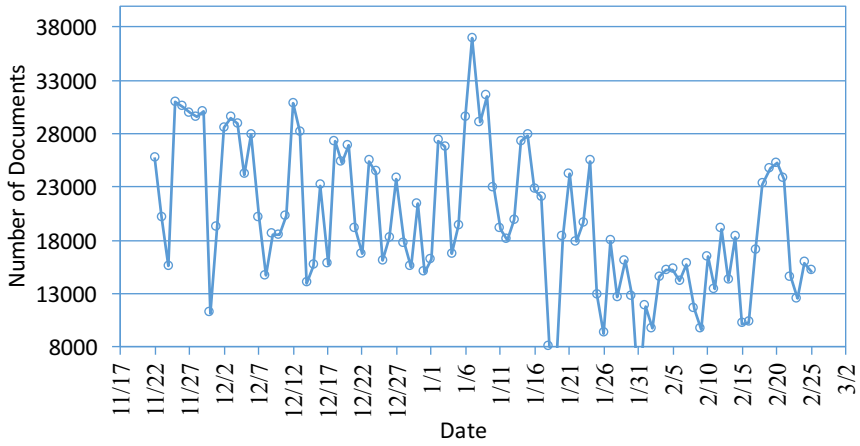
Table 2.1: The top twelve news outlets publishing the most documents in our EOS new article corpus.

News Outlet	Country	# of Documents
Times of India	India	48,278
ANSA English	Italy	36,799
South African News Net	South Africa	33,794
Kenya News.net	Kenya	31,085
Uzbekistan Newsnet	Uzbekistan	30,745
Auckland News	New Zealand	29,759
Kuala Lumpur News	Malaysia	29,317
Kazakhstan News Net	Kazakhstan	28,970
Reuters	United Kingdom	26,746
Taiwan's news.net	China	25,327
The Herald Sun	Australia	23,752
The Daily Telegraph	United Kingdom	22,599

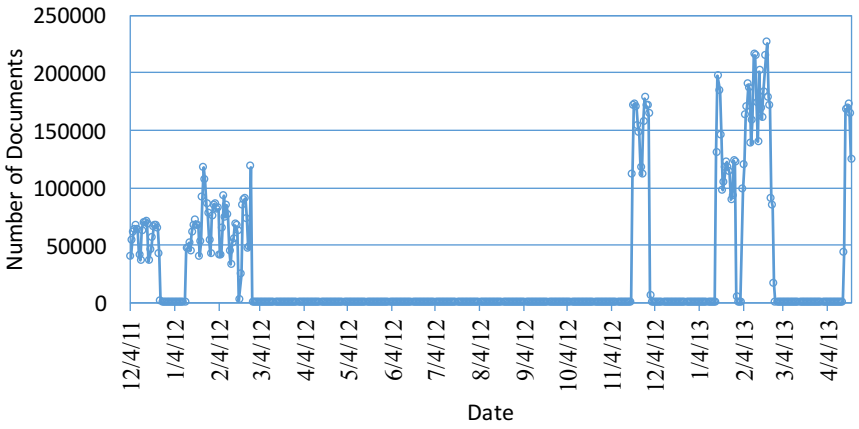
was identified, domain experts began adding sources to EOS to improve the local presence of the data set. However, when using this data set, we need to keep in mind the importance of understanding this distribution before analysis. This issue also explains the importance of considering Twitter or other social media data.

2.2 TREC Document Summarization Track

TREC organizes a document summarization track, aiming to develop systems capable of tracking events over time. For this purpose, TREC provides a standard dataset, named TREC-TS-2014F, which is a news article corpus. It includes 20 million news articles published from November 2011 to April 2013. Along with the documents, TREC provides a list of 15 ground truth incidents, covering a variety of topics, including natural disasters, civil unrests, and hostage crises. Most of the documents in TREC-TS-2014F are believed to be relevant to one of the 15 ground truth events

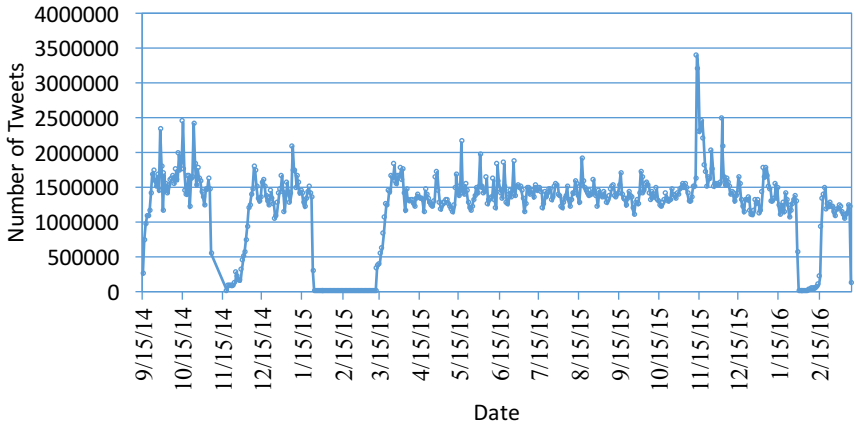


(a) The number of documents per day from 2013-11-22 to 2014-02-26 in the EOS news article corpus.

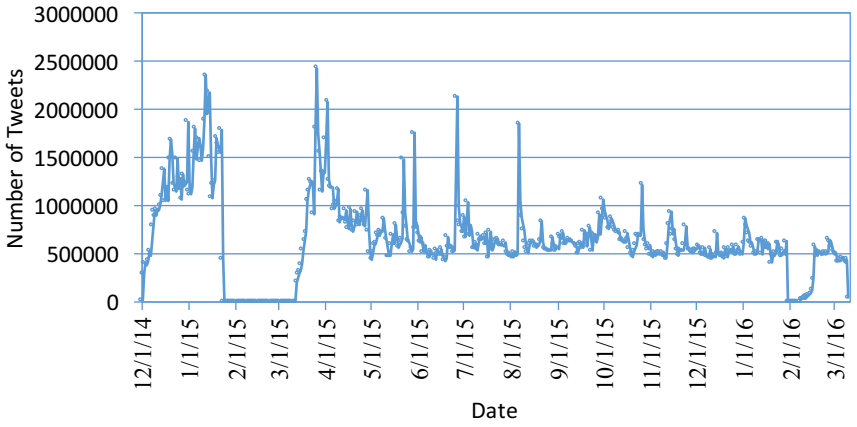


(b) The number of documents per day from 2011-12-04 to 2013-04-20 in the TREC news article corpus.

Figure 2.1: The number of documents per day in Expandable Open Source (EOS) corpus and Text REtrieval Conference (TREC) corpus.

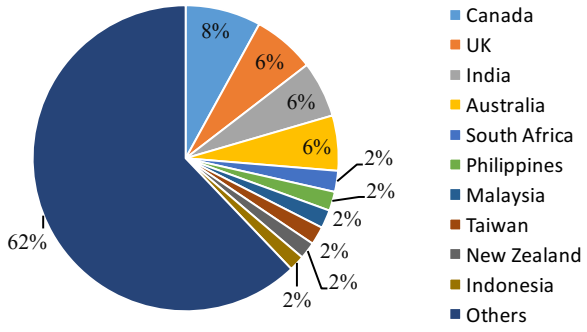


(a) The number of tweets per day from 2014-09-15 to 2016-03-10 in the English tweet corpus.

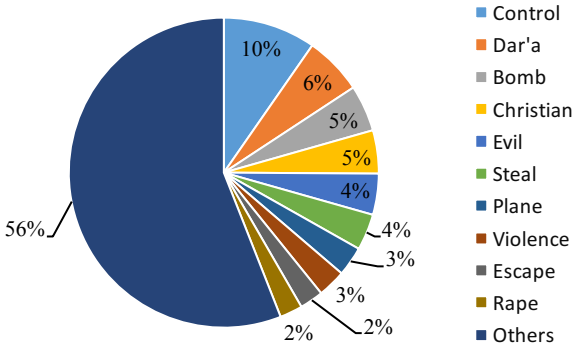


(b) The number of tweets per day from 2014-12-01 to 2016-03-10 in the Arabic tweet corpus.

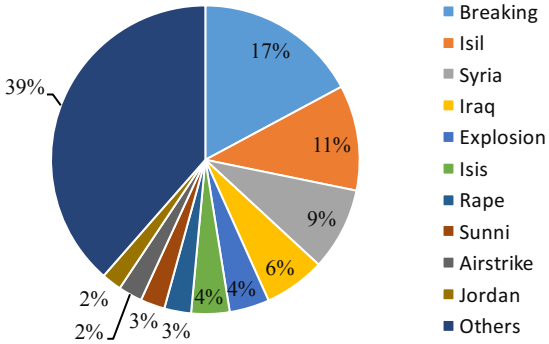
Figure 2.2: The number of tweets per day in the English tweet corpus and the Arabic tweet corpus.



(a) The top ten countries in which media outlets publishing the most documents in the EOS corpus reside, and the percentages of EOS documents associated with each of the 10 countries.

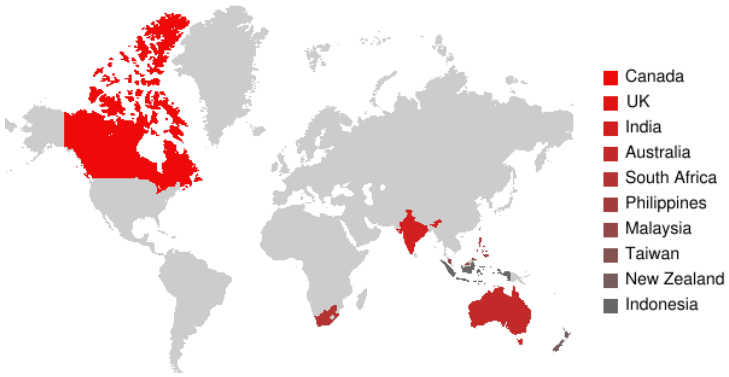


(b) The top ten English hashtags most being referred to in the English tweet corpus, and the percentages of tweets associated with each of the 10 hashtags.

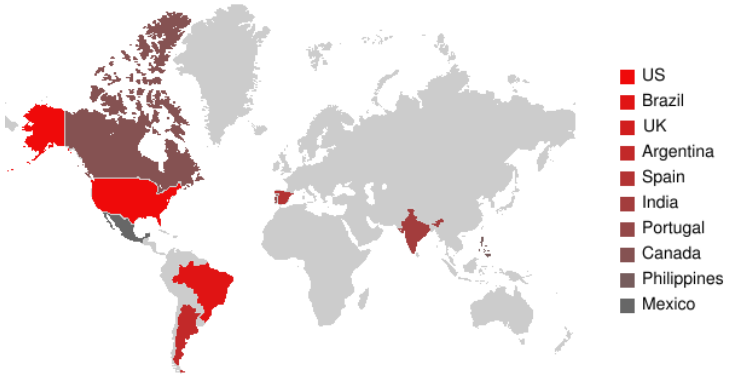


(c) The top ten Arabic hashtags (shown with English translations) most being referred to in the Arabic tweet corpus, and the percentages of tweets associated with each of the hashtags.

Figure 2.3: The top ten countries in which media outlets publishing the most documents in the EOS corpus resides, and the top ten English/Arabic hashtags most being referred to in the tweet corpus.



(a) The top ten countries in which outlets publishing the most documents in the EOS corpus reside.



(b) The top ten countries in which most English tweets in our tweet corpus are tweeted.

(c) The top ten countries in which most Arabic tweets in our tweet corpus are tweeted.

Figure 2.4: The top ten countries in which outlets publishing the most documents in the EOS corpus reside, and the top ten countries in which most English/Arabic tweets in our tweet corpus are tweeted.

to some extent. Figure 2.1b shows the number of documents per day in this corpus. It clearly shows that the majority of articles were written in the beginning of 2012, and the beginning of 2013, leaving a huge gap between these two time periods.

2.3 Twitter Data

The Twitter API provides easy access to posts containing specific hashtags/keywords [14]. Subject matter experts on our research team collect 258 English Twitter hashtags, and 169 Arabic Twitter hashtags. Most of them are names of places, prominent politicians, and religious leaders in Iraq and Syria. Others are concepts relevant to forced migration, such as migration, violence, execution, etc. Figure 2.2a and Figure 2.2b show the the number of tweets published per day in the English tweet corpus and Arabic tweet corpus, respectively. On average, 1.5 millions English tweets and 0.5 million Arabic tweets are added to our tweet corpus every day. There are two time periods (Jan 24, 2015 to Mar 11, 2015, and Jan 30, 2016 to Feb 6, 2016) when no data were collected, due to infrastructure failure. We observe some obvious peaks of the data volume, and most of them correspond to real world events occurring, e.g., the peak of English tweet volume around Nov 15, 2015 maps to a series of coordinated attacks carried out by ISIS in Paris.

Figure 2.3b and Figure 2.3c show the top 10 most used hashtags in our English tweet corpus and the Arabic tweet corpus respectively, and the percentages of tweets associated with each of the 20 hashtags. We can observe some word selection preferences across languages. For example, while both ISIS and ISIL refer to Islamic State of Iraq and Syria, ISIS is more popular among English users, and ISIL is more widely used among Arabic users. Figure 2.4b and Figure 2.4c show the top 10 countries where most English/Arabic tweets in our tweet corpus are tweeted on two maps. It meets

our expectation that most of countries tweeting Arabic tweets are in the Middle East. We do not find it surprising that the United States is among the top 10 countries tweeting most Arabic tweets, given that the huge Muslim community in this country.

2.4 An Exploration of Connection Between Open Source Data and Forced Migration

In this section, we present one study which investigates identifying forced migration related factors in open source data.

2.4.1 Overview

Forced migration is triggered by a humanitarian crisis in most cases. A humanitarian crisis can be defined as "any situation in which there is a widespread threat to life, physical safety, health or basic subsistence that is beyond the coping capacity of individuals and the communities in which they reside" [77]. It can be triggered by (1) acute events, including natural disasters nuclear and industrial accidents, "acts of terrorism" and armed conflict, or by (2) slower-onset processes, including environmental degradation, general violence or political instability. More often than not, humanitarian crises occur due to a combination of these triggers, in addition to underlying stressors, such as poverty and deficient governance [77].

In recent decades, progress has been made in establishing early warning systems to alert the international community, as well as national and local actors, of impending humanitarian crises [51]. For example, tsunami and famine early warning systems monitor and analyze data relevant in anticipating acute and slow-onset crises, respectively, relying on scientific, technological, economic, social and other indicators. Predicting crises involving other triggers, such as conflict and generalized violence, has

proven more difficult, although organizations such as the International Crisis Group put out regular alerts of worsening conditions [21]. Lagging behind these systems are effective early warnings of movements of people in response to humanitarian crises. More effective early warning of displacement will help governments and international organizations plan for such movements, as well as directly aid displaced persons before, during and after their exodus.

Patterns of forced migration in anticipation of, during, and following humanitarian crises are notoriously difficult to predict. Because detailed local data is difficult to obtain in a timely manner, we are interested in exploring whether or not open-source, online data can be used to help identify indirect, leading indicators of displacement/forced migration. Indicators relevant to this project include: economic, political, social, demographic and environmental changes affecting movements; intervening factors such as government refugee policies; and community and household characteristics. Parsing irrelevant information from the true indicators, calibrating results, understanding how these indicators change through time, and identifying and removing potential bias, requires large-scale data analysis and potentially, new computational methods for developing meaningful descriptive and predictive models.

In this study, our focus is a case study of Iraq. We are interested in determining whether or not a data-driven approach using open-source data can be combined with domain expertise in a scalable manner to identify possible indirect indicators of forced migration. While many different data sources are considered for the problem at large, we begin by focusing on what we can determine using our EOS corpus. Using the 2.6 million English documents that are broadly related to Iraq, the first step is to ascertain if a clear mapping exists between terms and concepts in EOS and important events in Iraq and more broadly, the Middle East. Once we feel confident that this relationship exists and can be extracted efficiently, we can use the mapping to understand which

concepts and changes in concepts can serve as leading indirect indicators of forced migration. In this section, we describe our approach for identifying this mapping and present the techniques we have used to better understand the strengths, weaknesses, and biases associated with open-source, big data analysis.

2.4.2 Interdisciplinary Community

To address massive global issues, we advocate approaches that include a team of researchers from multiple disciplines. Without interdisciplinary insight, it becomes difficult to (1) fully understand the problem; (2) understand the data and the gaps; and (3) analyze the data effectively. The subject matter experts understand the factors that contribute to forced migration at the macro, meso, and micro levels, while computer scientists and statisticians understand how to mine, and analyze mass amounts of data. The co-authors of this work are a subset of the researchers working on this problem. Currently, our research community consists of more than 25 researchers, technicians, policymakers and humanitarian practitioners from around the world. We mention this important collaboration because we want to emphasize that an approach using a team of computer scientists that understand how to process big data but lack subject matter expertise, will likely miss important and possibly even obvious, insights that domain experts are able to spot.

2.4.3 Approach

Here we highlight the methodology and algorithms used to determine whether or not EOS is a viable source for data that can be used as indirect indicators of forced migration. All of the described steps have been investigated in the database, data mining and information retrieval communities and serve as an initial exploration of the data.

Data Extraction

In order to understand the data set, we process each document in the corpus by extracting named entities and then stemming words and extracting unigrams, bigrams, and trigrams. We use the Stanford Named Entity Recognizer to extract named entities [7] and the Snowball implementation [8] of the Porter stemmer to stem words. The stopword list is custom developed. To reduce the number of distinct concepts, we also merge synonyms into a single concept using Wordnet [82]. Once this document processing is complete, approximately 4 million unigrams and 3.7 million named entities are maintained in our concept list. The number of concepts is particularly high because of the large number of foreign names and words identified.

Documents in the EOS corpus are from over 1600 different sources, including many from the Middle East. While the majority of articles are still from newspapers outside of the Middle East, there are a large number of articles from countries in the Middle East, including the United Arab Emirates, Iran, Turkey, Egypt, and Saudi Arabia. Only 6,600 articles come from Iraqi sources and even fewer come from other countries in the Middle East, e.g. Syria. Our domain experts do not find this result unexpected since there has been a history of censorship of controversial issues such as forced displacement in many of these countries, particularly Iraq [95]. This means that future work will need to consider less traditional forms of media in areas where censorship of traditional media is high. If Internet penetration is high, social media is a good option.

Domain Knowledge

Manually building models and ontologies is very labor-intensive. Because our goal is a scalable process, one challenge is to identify the most useful manually collected data

for this task. We decide to collect three data sets that could be easily provided by domain experts: a small set of ground truth documents; a domain vocabulary; and a timeline of events in Iraq that are perceived by subject matter experts to be directly or indirectly relevant to forced migration.

The ground truth document set consists of twenty English articles that are considered early indicators of forced migration. While a larger ground truth document collection would be useful, manually identifying a large number of articles reduces the overall scalability of our approach. Therefore, we limit the number to twenty.

The subject matter experts also develop a relevant domain vocabulary list. This vocabulary list is divided into ten domains of knowledge, e.g. demographical, displacement, economics, governance, etc. Each domain lists between ten to fifty concepts relevant to the domain in the context of Iraq. Again, we limit the amount of manual labor, requesting a few hundred relevant words when the number of concepts in the entire corpus is over 7 million.

Finally, the subject matter experts create a timeline that contain very brief explanations of potentially relevant events, e.g. acts of violence, civil unrest, political developments, weather-related events, religious holidays, developmental initiatives, etc. This timeline serves two purposes. The primary purpose is to provide computer scientists necessary background about the types of events that are relevant and needed to be identified in EOS. Second, it serves as a possible set of relevant concepts that could be used during the noise reduction process.

Noise Reduction

While we anticipate that the amount of *noise* in the corpus would be substantial compared to the amount of *relevant signal* related to forced migration, we verify this empirically. We consider the frequency of concepts and the frequency of concepts that

Table 2.2: Noise reduction comparison.

Concept Threshold	Using Event Concept Seeds	Using Domain Concept Seeds	Using Ground Truth Doc. Concept Seeds	Using Location Seeds	Overlapping Documents
1	2,403,536 - 20	2,284,032 - 20	2,412,006 - 20	48,478 - 20	2,279,112
5	2,257,822 - 20	1,898,790 - 20	2,276,797 - 20	826 - 5	1,898,676
10	2,148,835 - 20	1,385,391 - 20	2,182,860 - 20	1 - 0	1,385,381
50	936,341 - 18	76,883 - 10	1,107,688 - 20	NA	76,883
100	234,030 - 11	2,698 - 0	376,520 - 17	NA	2,698

appear together. We compare the frequent set of concepts in the corpus to concepts generated by our domain experts and concepts in our ground truth documents using the FP-Growth frequent itemsets algorithm [2] with a support of 0.05. We find that the types of concepts appearing frequently are very general and too broad to be used effectively. Example concepts in this frequent set include year, time, news, people, 2013 and report.

Therefore, to reduce noise we consider the data provided by the subject matter experts. We use the concepts from each of the three types of domain data as *seeds* and identify documents that have a sufficient number of these concepts in them. The intuition is that relevant documents would have similar concepts as those in one of the lists provided by the subject matter experts. As an additional comparison, we also consider a list that contains only locations in Iraq. We pause to mention that even though it may seem logical to maintain all the documents in which any relevant concept from the three expert generated lists or the location list are found, doing so results in little reduction in the number of maintained documents.

Table 2.2 contains the comparison of these approaches. The first column specifies the number of concepts required to be in the document for it to be maintained. The

next four columns show the number of documents retained (not considered noise) based on the number of concepts identified using different expert knowledge seeds. It also shows the number of ground truth documents retained in the final document set. Recall that there are initially twenty ground truth documents. The final column shows the overlap in the documents retained by each of the first three different seeding options (we exclude the location seeded comparison in the overlap since the number of retained documents is significantly lower than the other three methods). As an example, the first row of the table states that if only one of the concepts in the different concept list exists in a document in the corpus, the document is maintained. In the cases of the event, domain, and ground truth concept lists, over 2.2 million of the 2.6 million documents are maintained. There is very little reduction in the document set. In the case of the location list, only 96,000 documents are retained. In all cases, all twenty ground truth are still in the reduced corpus. From the table, we see that all the seeds do not lead to the same amount of noise reduction. The location seeds remove the most noise, followed by the event concept seeds (using 34 event concepts, all of the ground truth documents are maintained and the corpus is reduced to 878,386 documents). So while using a small number of concepts generated by domain experts helps us remove a large amount of noise, we still need to use other techniques to find the most relevant documents.

Topic Modeling

Because we have ground truth documents, we build a topic model using those twenty documents. While we consider a number of different topic models, Latent Dirichlet Allocation (LDA) [36] seems effective on this corpus. Also, using a bag of words model produces topic lists that domain experts consider logical. We use the Mallet implementation [5] of LDA to generate ten topics each containing twenty different

concepts. We choose ten topics to see if the learned concepts overlap with the ten domains developed by the subject matter experts. We then identify those documents in our reduced corpus that have a large amount of overlap with the topic model generated from the concepts in the ground truth document set. We consider these to be the set of possible relevant documents. As a comparison, we also run LDA on the reduced corpus and then use Manhattan distance with a threshold of 0.1 to identify the most similar documents to the ground truth documents.

2.4.4 Preliminary Findings

While we have a number of interesting findings, we present a few that can be useful for others conducting similar analyses.

Since completely unsupervised methods do not produce as meaningful results, it is important for these types of endeavors to build a team of subject matter experts that can provide guidance. In our case, our subject matter experts develop three types of domain knowledge. Preliminary findings suggest that doing so results in much better precision of relevant documents. Some of the relevant documents found are broadly relevant to forced migration in the region, but are not as relevant to Iraqi forced migration. This is an interesting finding and that suggests the need for a weighting scheme to determine the overall relevance. Further, the number of concepts used to guide the document selection process ranges from between a few hundred to a few thousand depending upon the specific domain knowledge used in the seeding process.

Location in conjunction with the domain seeds is a consistent piece of information that is present in relevant documents. Further, if we focus on locations as seeds for finding relevant documents, we are able to remove significantly more noise while still maintaining the ground truth documents in the corpus.

When analyzing the topics generated from the ground truth documents, there are two interesting findings. Most of the twenty documents have concepts from multiple topics in them. Second, we find that when mapping topics to domains, the concepts in the topic model cover eight of the ten domains.

2.4.5 Data science recommendations

While this initial analysis indicates that relevant documents exist in the corpus, there are a number of challenges that we encounter. We now make recommendations for more effective use of open-source data for grand-scale data science challenges.

Data quality: Take time to assess the quality of the data and the data sources. Our initial analysis show us that our data sources are biased toward countries outside the region of interest. To deal with this, we add more data from sources in the Middle East.

Dynamic data changes: Because all the sources are independently owned, the availability and type of data that is accessible may change over time. Be prepared to spot changes and compensate for them.

Expect missing data: While we find that some of our domains have ample coverage in terms of concepts, there are a few that are not well covered. While every attempt should be made to obtain full coverage, it is better to design algorithms that still work in the presence of this missing data.

Processing power: Regular standalone servers do not have the processing power to handle parsing and analyzing big data. Expect to setup or purchase time on a distributed cloud infrastructure. In our case, we use five nodes of a 28 node distributed cluster, where each node has 32 GB of memory.

2.4.6 Conclusions

This initial analysis shows the promise of using open-source data for identifying documents and containing topics that may be useful for understanding more about the movements of people. Specifically, we find forced migration-related documents for our Iraq case study do exist within our EOS data set. Not all the documents contain leading indicators for identifying movement of people when a humanitarian crisis occurs. However, a clear mapping that can be extracted exists between data in EOS and important concepts related to internal and cross-border movement relevant to Iraq.

Chapter 3

Event Detection in News Articles

Forced migration is triggered by a humanitarian crisis in most cases. A humanitarian crisis can be caused by (1) acute events, including natural disasters nuclear and industrial accidents, "acts of terrorism" and armed conflict, or by (2) slower-onset processes, including environmental degradation, general violence or political instability. Patterns of forced migration in anticipation of, during, and following humanitarian crises are notoriously difficult to predict. Because detailed local data is difficult to obtain in a timely manner, we are interested in exploring using open source data to detect events of certain targeted types, which are leading indicators of forced migration. These targeted event types include economic, political, social, demographic and environmental changes affecting movements; intervening factors such as government refugee policies; and community and household characteristics.

Event detection from text data is an active area of research. While the emphasis in the literature has been on event identification and labeling using a single data source, this work considers event and story line detection when using a large number of data sources. In this setting, it is natural for different events in the same domain, e.g. violence, sports, politics, to occur at the same time and for different story lines about the same event to emerge. To capture events in this setting, we propose an offline algorithm that detects events and story lines about events for a target domain given a news article collection. Our algorithm leverages a multi-relational sentence level semantic graph and well known graph properties to identify overlapping events

and story lines within the events. We then extend this algorithm for an online setting. Both the offline and the online approaches are evaluated using two large data sets containing millions of news articles from a large number of sources. Our empirical analysis shows that methods using the proposed semantic graph beat the state of the art in terms of precision and recall, while providing more complete event summaries.

3.1 Introduction

Since early 2011, online news readership has surpassed traditional newspaper readership in the US [6]. Given this transition to online news, it is not surprising that the timeliness of online news has continued to improve, also surpassing that of traditional paper sources [17]. While many services exist for finding articles that have certain keywords in them, organizing news into events helps streamline the process of finding information of interest. It can also be useful for identifying unusual events, e.g. civil unrests, or understanding the changing dynamics of topics of interest, e.g. political events/changes in a particular location of the world.

Much literature exists on event detection and story line extraction (document summarization) [26] [119] [38] [54] [65] [66] [115] [94] [89]. Two gaps in the literature that we focus on in this work are discovering *overlapping* events and determining event story lines when there are a large number of newspaper sources. Our goal is to extract events of a particular theme/target domain (e.g. sports, violence, flu, etc) even if they occur at the same time and effectively summarize story lines associated with each event, thereby providing users with richer context. While many news events discuss a single story, some have multiple story lines (subplots). Our approach attempts to distinguish story lines when an event has more than one - differentiating our work from traditional document summarization methods. For example, suppose we identify

the Super Bowl event from a newspaper collection. Different story lines related to the event may include the game summary, the effect of an injury to a key player, the half time show, etc.

For accurate event detection and understanding, it is necessary to track and reason about the connections between related event elements. We leverage a graph data representation for this purpose. Graphs are well-suited for representing complex connections between related entities, and graph analysis algorithms have been developed for reasoning about these connections. Our approach constructs a graph based on a topic and location of interest using documents in a newspaper collection (node labels are document sentences and edges are based on semantic similarity and sentence proximity between nodes), maps events to partitions of the graph using different heuristics based on well-known graph properties, and summarizes the event using high frequency node labels. We propose both an offline and online version of our approach. The main difference between them is the construction of the graph. Our offline approach builds the semantic graph using all the documents in the newspaper collection. Since the offline version can be viewed as taking a retrospective view of the data, we will also refer to it as our *retrospective approach*. Our online approach maintains and updates the graph dynamically through time, maintaining only the nodes and edges associated with a small number of time windows. Evaluation using two large data sets containing millions of news articles from a large number of sources shows that our offline and online approaches beat the state of the art in terms of precision and recall while providing complete event summaries.

To summarize, **our contributions to the literature are as follows:** (1) we propose a comprehensive framework that utilizes a location ontology and a domain dictionary to identify events using relevant news articles from a large, noisy news corpus generated from multiple news sources; (2) we propose a new event detection

algorithm that takes advantage of a multi-relational semantic graph to identify and summarize events and propose two additional heuristics that improve the detection quality in different situations; (3) we propose an offline retrospective event detection algorithm and extend it to an online setting; (4) to the best of our knowledge, our method is the first targeted event detection algorithm that detects and summarizes different events occurring at the same time; (5) an empirical evaluation on two data sets demonstrates the accuracy of our event detection algorithms when compared to the state of the art; (6) we compare story lines generated using different event detection methods and show that subject matter experts rate our event story line synopses higher than other methods; (7) we extend our offline algorithm to an online setting, and evaluate the extension using two large data sets..

3.2 Related Literature

We first review the existing literature on event detection, which can be classified into two broad categories, those that focus on detecting all events (non-targeted event detection) and those that detect domain specific events (targeted event detection). Then we briefly discuss the work on event/document summarization and online event detection.

3.2.1 Non-targeted event detection

The majority of literature related to event detection focuses on identifying events that span a broad range of themes or categories [26] [119] [38] [54] [65] [66] [115] [71] [73] [96] [74] [116]. Allan et al. [26], Yang et al. [119], and Brants et al. [38] propose variants that stem from the *TF.IDF* model. Researchers have also proposed models based on term level analysis [54] [65] [66]. Fung et al. [54] propose an algorithm that

identifies groups of bursty terms by considering both document frequency of the terms and co-occurrence across documents over time. Variants have been proposed that consider burstiness by comparing to the expected frequency [65], considering spatial proximity of document streams when grouping words [66], and evaluating burstiness using wavelet transforms [115]. Segment level event detection approaches have also been proposed [71] [73] [96]. Leskovec et al. [71] track *memes*, a quoted text segment, in a news document stream, and use a group of *memes* to represent an event. Li et al. [73] divide a tweet into consecutive n-grams that represent semantically meaningful phrases from which bursty segments are selected. Sayyadi et al.'s work [96] is most similar to ours in spirit. They consider approaches for partitioning a graph (their nodes are noun phrases), so that each partition maps to an event. We will show that even though our partitioning strategies are similar, our graph construction approach leads to more accurate event detection and more interpretable story line identification.

Previous literature also considers detecting events at a latent topic level [74] [116] [106]. Lin et al. [74] model an event as a series of topics over time, and a topic is defined as a multinomial distribution of words, regularized by the Gibbs Random Field. [116] detects events using topic modeling. In their model, words are generated by a topic according to a underlying distribution. The frequency acceleration of these words can be calculated directly from the raw tweets in tweet stream,. They can also be derived from the acceleration of the underlying topics. By minimizing the error between them, the authors determine the acceleration of the underlying topics, from which the bursty topics can be determined. Similar to our work, Wang et al. [106] investigate event detection across multiple news article streams. While we focus on detecting events, viewing multiple news article streams as an ensemble data stream, Wang and colleagues focus on correlating bursty patterns across different news article streams.

More recent work considers using network structure to improve event detection [24] [74] [53] [47]. Aggrawal and Subbian [24] construct a graph to represent the interactions between entities in a social stream. Recently, Wang et al. [105] proposed a dynamic hierarchical model to learn multiple aspects (opinions) of news events in Twitter. Finally, Guralnik et al. [58] detect events in numerical time series data, by capturing change points in time series.

Our work differs from all the above mentioned works in the following ways. First, we focus on targeted event detection where the target domain is prespecified. Second, we identify events having the same target theme that may occur at the same time (overlapping events). Finally, previous work generates event summaries using a set of terms, phrases, or text segments. In contrast, our event story line summaries are composed of a small number of sentences, offering readers a more comprehensive understanding of the detected events. We accomplish this trivially since we generate story line summaries using our semantic graph node labels.

3.2.2 Targeted Event Detection

Previous work on targeted event detection includes [117] [87] [90] [94] [89] [70] [84]. One direction of research considers using lexico-syntactic or lexico-semantic patterns to identify events [117] [87] [90]. Xu et al. [117] uses syntactic patterns as seeds to learn more patterns from document stream with a bootstrapping strategy, and leverages the learned patterns to extract targeted events. Nishihara et al. [87] define a pattern as a tuple of three elements: *object, place and action*, to extract events from blogs. Ritter et al. [90] define a pattern as a four element tuple: *entity, event phrase, date, and type*. These patterns are used to extract named entities, dates, and temporal expressions from tweets. Each of the entries extracted using the four element pattern is considered an event. These methods rely on the assumption that the text segments

describing targeted events match one of these patterns; however in real world data, a significant part of text associated with targeted events may not match any of these patterns, resulting in a non-trivial miss rate. Leveraging such multi-element patterns can seriously restricts the text scope to explore, resulting in a non-trivial miss rate. Wang et al. [104] propose learning patterns, as opposed to relying on pre-specified patterns, to forecast extreme weather events from spatial-temporal numerical data. In contrast to these works, our approach does not rely on pre-specified patterns. This allows us to handle variation in text more readily.

The second thread of research can be categorized as binary predictors [94] [89] [70] [84]. These methods do not detect or summarize a specific event. Instead, they detect the existence of an event within a document stream. They do not distinguish between different events of the same type or events that are overlapping. A typical domain specific approach begins with a keyword vocabulary collected by domain experts, filters the raw corpus with the domain vocabulary, and uses an increase of the number of retained documents in a time window to signify occurring of a targeted type event [94] [89] [70]. Instead of using keywords in the target domain, Muthiah et al. [84] start with a few seed patterns and use a bootstrapping strategy to learn more patterns, which are then used to identify documents (tweets) relevant to target events. Similar to these works, we use a vocabulary to represent a specific theme of interest as a component in our methodology. Our approach differs from these because we discriminate overlapping targeted events occurring at the same and consecutive time windows, we consider simple graph properties of a dynamic semantic graph to identify events, and we trivially generate story line summaries for each detected event.

3.2.3 Event & Document Summarization

Most of recent work on summarizing detected events focus on events using Twitter data [43] [86] [49]. Because we are generating storylines using newspaper articles that are longer and may be discussing multiple events in a single article, we are unable to leverage these Twitter-centric methods. However, document summarization has a long research history. Here we focus on a few representative methods [29] [100] [81]. Barzilay et al. [29] generate a summary for multiple documents by identifying and synthesizing similar elements across related sentences in documents using sentence dependency trees. Shen et al. [100] summarize documents using a Conditional Random Field that labels each sentence within a document with a 1 (summary sentence) or 0 (non-summary sentence). Mihalcea and Tarau [81] build a graph with weighted edges in which nodes represent sentences within a document, and edge weights represent the textual similarities between sentences. Sentences having the highest PageRank scores are used as the summary for the document. While document summarization is relevant to our story line detection, our story lines are for events that cross multiple newspaper articles, as opposed to a summary of a single document.

3.2.4 Online Event Detection

While most work on event detection from text is for an offline setting, the online setting has also received some attention [119] [20] [30]. Yang et al. [119] propose maintaining a set of word vectors, each of which represents an event. When a new document arrives, [119] represents the document as a word vector, and calculates the new document's similarities to all the existing events. A new event is identified if the similarity does not exceed a threshold; otherwise the document is regarded as discussing an existing event. [119] uses the group of documents classified as belonging

to a detected event to represent the event; in comparison, our online approach uses a sentence level graph to detect events and generate a concise summary composed of a small number of sentences as the synopsis of an event.

Abdelhaq et al. [20] propose a 4-step approach for detecting localized events in an online setting: (1) extract words bursty in the present time window; (2) only retain the words having a localized spatial distribution; (3) group the retained words into clusters, each of which represents an event; and (4) update those clusters in an online fashion. [20] generates event summaries using a set of busty terms. With respect to topic-involved online event detection, Becker et al. [30] train a classifier based on 4 types of features: (1) temporal features, (2) social features, (3) topic features, and (4) Twitter-centric features. The trained classifier is then used to classify tweet clusters into topics, and each topic represents an event. Our approach uses a different algorithm and data structure than these methods. Also, since our event summaries are composed of a small number of sentences, it offers readers a more comprehensive understanding of the detected events.

3.3 Notation and Definitions

Here, we present definitions, assumptions, and our problem statement. An event is *something that happens at a particular time and location* [26]. We define a *targeted event* to be an event that is associated with a particular domain or topic of interest to the user, e.g. politics, violence, football, etc.

3.3.1 Assumptions and Notation

A newspaper collection $\mathbb{D}$ is a set of articles that occur through time. D^t denotes the set of articles that occur in a time window t. Each newspaper article $d_j \in D^t$ is

decomposed into a vector that is a bag of sentences. We maintain a vector, $\mathbb{S}$, of the number of occurrences of sentences $\{s_i\}_{1 \leq i \leq N}$, where s_i is the number of occurrences of sentence i in $\mathbb{D}$, and N is the size of sentence vocabulary.

We assume the following about collection articles:

1. We know the time stamp of the article being published.
2. Each article specifies at least one location.
3. An article may be discussing zero, one, or more events.
4. Articles are composed of paragraphs.
5. A paragraph in an article discusses only one event[1].

When different news agencies describe an event, they may choose to describe different aspects of it. To capture this, we define a *story line* to be a theme or subplot of an event. We also allow an event to take place over one or more consecutive time windows, and assume that an event is reported with temporal continuity. In other words, once it begins being reported, the reporting continues until the event is completed (there are no time window skipped in the reporting).

3.3.2 Problem Statement

The task of overlapping target event and storyline detection has two subtasks: (1) identifying events in the target domain even if they are overlapping; (2) identifying the themes or story lines of the events that have been discovered. As we will show in Section 3.4, identifying storylines is trivial using our proposed data structures. Therefore, the majority of this work will focus on subtask 1.

[1]We have empirically validated this assumption across 1000s of articles. While articles may discuss multiple events or multiple themes of a single event, paragraphs generally focuses on a single story line in a single event.

The goal of offline or retrospective detection is to identify events and storylines using the entire newspaper article collection $\mathbb{D}$; the goal of online detection is to identify events and storylines for a new time window of newspaper articles D^t at its arrival, based on articles occurring during or before time window t.

3.4 Offline Retrospective Targeted Event and Story Line Detection

In order to identify and summarize events, we propose a methodology that contains the following steps: location identification, target domain mapping, semantic graph creation, and event detection (see figure 3.1). Algorithm 1 presents a high level view of our retrospective event detection method. The input to the algorithm is our document collection $\mathbb{D}$, a location ontology $\mathbb{O}$ containing major localities around the world (countries, governorates, and cities), a location $\mathbb{L}$ of interest to the user, and a domain $\mathbb{P}$ of interest to the user. Here, $\mathbb{P}$ is a small set of words and phrases that describe a topic the user wants to monitor. The output of the algorithm is a set of events $\{E_k\}$, represented as story line summaries of the documents discussing them. The algorithm begins by going through the document collection and identifying the subset $\mathbb{R}'$ of documents that include the target location (line 1) and the domain of interest (line 3). $\mathbb{R}'$ is then used to create a semantic graph G (line 4). We then look for connected components in G (line 5). These connected components are the basis of the event and story line detection. After identifying the connected components, we consider different heuristics for improving the quality of the detected events (line 7). The remainder of this section goes through the major components shown in figure 3.1.

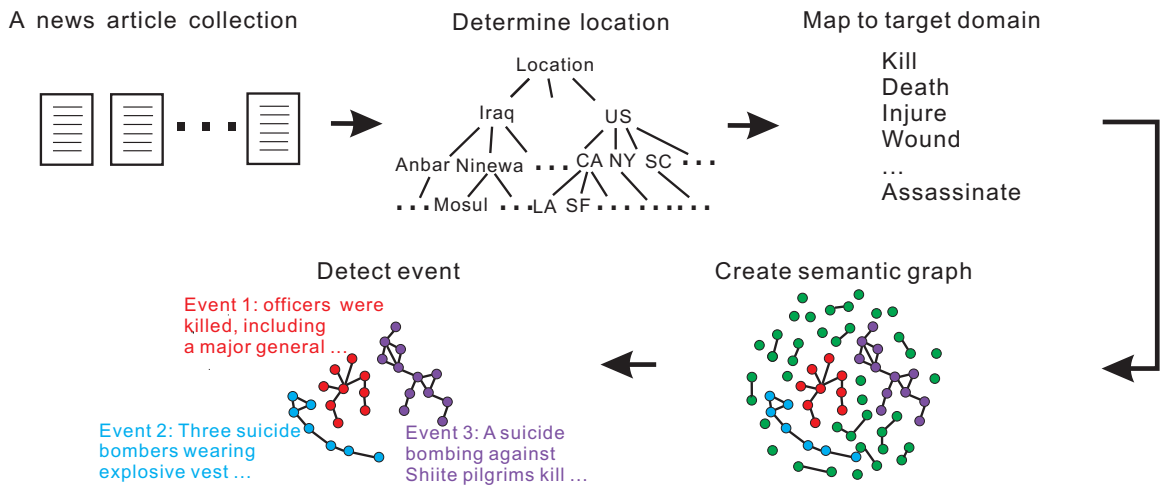


Figure 3.1: The framework of our proposed approach.

3.4.1 Location Identification

It is not uncommon for the same event to occur in different locations, but for a user to only be interested in events in a particular location. Therefore, this step identifies the location associated with each document. There are a number of different approaches for location identification. Our approach begins by constructing a hierarchical ontology $\mathbb{O}$ using open-source data (described in Section 3.6) that contains countries, governorates, and cities. Using this ontology, we then determine the location of each document by counting the occurrences of each location and aggregating the occurrence numbers of the child locations to their parent locations iteratively. The location with the highest frequency count is considered the predominant location of the article. Ties are broken using the location in the title. If the predominant location does not map to the location of interest or the document does not contain a location, it is removed from further analysis. The processing cost of location identification is

$O(|\mathbb{D}| \times |log(\mathbb{O})|)$, where $|\mathbb{D}|$ denotes the number of documents in $\mathbb{D}$, and $|\mathbb{O}|$ denotes the size of the location ontology $\mathbb{O}$.

Algorithm 1: Our retrospective event detection approach at high level.

Input:
A document collection: $\mathbb{D}$
A location ontology: $\mathbb{O}$
A target location: $\mathbb{L}$
A target domain: $\mathbb{P}$

Output: *A set of events*: $\{E_k\}$

1 $\mathbb{R} = identify\_geographically\_relevant\_documents(\mathbb{L}, \mathbb{D}, \mathbb{O})$
2 $T = generate\_domain\_dictionary(\mathbb{P}, \theta)$
3 $\mathbb{R}' = identify\_domain\_relevant\_documents(\mathbb{R}, T)$
4 $G(V, E) = create\_semantic\_graph(\mathbb{R}')$
5 $C = extract\_connected\_components(G)$
6 **for** $C_k \in C$ **do**
7 $\quad C_k = improve\_component\_quality(C_k)$
8 $\quad E_k = identify\_event(C_k)$
9 $\quad E_k = generate\_storyline(E_k)$
10 return $\{E_k\}$

3.4.2 Target Domain Mapping

Since our interest is in identifying events in a particular target domain, we construct a dictionary that contains domain keywords and phrases. Beginning with a set of seed keywords in $\mathbb{P}$, we extract additional related keywords using online thesauri and ontologies. When the size of the dictionary is small or moderate, we have subject matter experts to validate the final dictionary. While a unsupervised approach may be preferred, we have found this semi-supervised approach more promising since it begins with expert knowledge, then expands the domain dictionary using online sources, and finally concludes with expert validation. In cases when the dictionary is very large, subject matter experts validate a sample of the dictionary. The validation is repeated until the accuracy is above a predefined threshold θ (Line 2 of Algorithm

1). Once the dictionary is constructed, we retain articles that contain at least one dictionary keyword in the title. As will be discussed in Section 3.6, we empirically find that articles themselves are noisier than titles when considering a target domain (Table 3.2). The processing cost of target domain mapping is $O(|\mathbb{R}| \times |T|)$, where $\mathbb{R}$ denotes the documents retained after the location identification, and T denotes the constructed domain dictionary. In practice, this dictionary is small, tens to hundreds of words and phrases.

3.4.3 Semantic Graph Creation

As previously mentioned, graphs are well suited for representing and reasoning about entities and connections between them. While there are many different representations of text, we choose to model it in a *semantic graph*. We propose using this semantic graph G to identify and summarize events. The semantic graph we propose keeps track of sentences in relevant articles and their relationships to each other. More precisely, the semantic graph $G = (V, E)$ is composed of a set of nodes $V(G) = \{v_1, \ldots, v_n\}$ and a set of edges $E(G) = \{e_1, \ldots, e_m\}$. Each node v_i represents sentence i in the sentence vocabulary $\mathbb{S}$ of $\mathbb{R}'$. An edge (v_j, v_k) is added to G if one of the following conditions is true: (1) two sentences are consecutive in the same paragraph (proximity edge) OR (2) two sentences appear in documents that are temporally close (occur on the same day or on consecutive days) and have high semantic similarity (semantic edge).

Proximity similarity is based on the assumption presented in section 3.3 that sentences in the same paragraph of an article are discussing the same event. Therefore, an edge is added between nodes in G when the nodes represent two sentences that appear next to each other in a document. *Semantic similarity* is based on the assumption that sentences containing similar vocabulary are semantically similar. Semantic similarity can be measured in many different ways. We consider two different criteria,

relative edit distance (RED) [122] and relative common sequence length RCS [92], where $RED(i,j) = edit(i,j)/n_l$ and $RCS(i,j) = seq\_len(i,j)/n_s$. Here (i,j) denotes a sentence pair, $edit(i,j)$ is the edit distance between i and j, seq_len is the common sequence length between i and j, n_l is the length of the longer sentence ($|i|$ if $|i| > |j|$, otherwise $|j|$), and n_s is the length of the shorter sentence ($|i|$ if $|i| < |j|$, otherwise $|j|$). An edge is added to the semantic graph to connect the sentence pair (i,j) if the semantic similarity is high and they are temporally close. In the next section, we discuss scores that are reasonable for both of these similarity metrics.

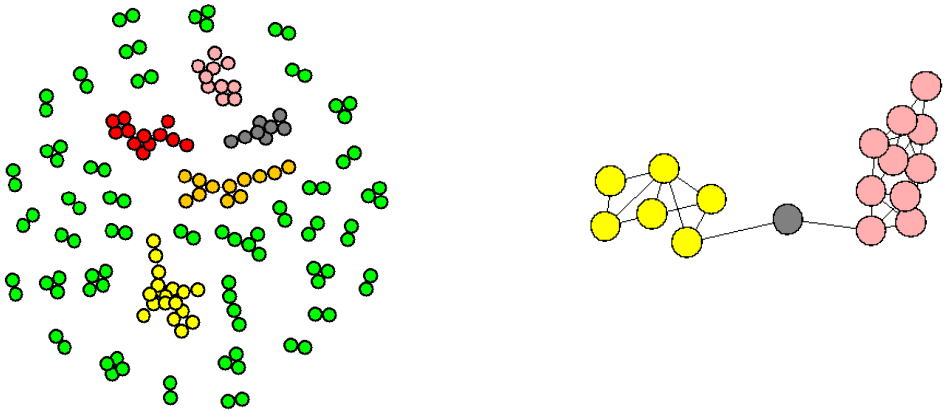
Notice that G is a multi-relational graph since it contains two different types of edges, proximity edges and semantic edges. Considering the semantics of different edges will be useful when we detect events. We pause to mention that while we could construct the semantic graph using keywords, named entities, and/or noun phrases, we will show the strengths of a sentence level semantic graph in Section 3.6. [2]

Finally, if we assume sentence length and document length are constants, then the processing cost of semantic graph creation is $O(\mathbb{R}'^2)$, where $\mathbb{R}'$ denotes the documents retained after the target domain mapping. We will show that $|\mathbb{R}'| << |\mathbb{D}|$ in Section 3.6, because only a small portion of documents in $\mathbb{D}$ supports the target location and maps to the target domain.

3.4.4 Event Detection

We detect events using the constructed semantic graph G. We begin by identifying the non-trivial connected components. We then consider different heuristics to improve the quality of the non-trivial connected components by pruning and separating weakly connected parts of the subgraph.

[2] It should also be noted that phrase and sentence level topic graphs have been analyzed in other fields. For example, these type of graphs are sometimes called *complex graphs* and have been used for text summarization [27].



(a) Semantic graph. (b) Subcommunity.

Figure 3.2: Graph examples.

Connected component event detection

We define a connected component C_k to be a subgraph containing a set of nodes $V(C_k)$ and edges $E(C_k)$ such that every node in $V(C_k)$ has a path to every other node in $V(C_k)$. We define a non-trivial connected component to be a connected component whose total occurrences of its consisting sentences is *reasonably* large, where *reasonable* will be evaluated empirically in Section 3.6. To provide a little intuition now, we show an example in Figure 3.2a of connected components identified during our analysis. Trivial connected components are shown in green. The non-trivial connected components are depicted using other colors. We then directly map an event to a non-trivial connected component.

Because some connected components contain weak connections, we also propose two heuristics that attempt to further improve the quality of the non-trivial connected components. We refer to the first one as the *subcommunity heuristics* and the other as the *inheritance pruning heuristics*.

Subcommunity heuristics:

We observe that in some cases, a connected component has clear sub-communities (see Figure 3.2b). This heuristics attempts to identify these subcommunities. While any reasonable community detection algorithm will work, we choose to use the edge betweenness algorithm proposed by Girvan and Newman [57], because its mechanism of detecting communities allows us to favor specific edges, and we will explore other clustering algorithms in the future work. We apply edge betweenness clustering on each non-trivial connected component in G. This algorithm removes edges that have the largest number of shortest paths going through them. Recall that G contains two types of edges, proximity edges and semantic edges. Our goal here is to maximum the chance of semantically similar sentences staying in the same connected component. Therefore, we only consider removal of proximity edges when detecting communities.

Inheritance pruning heuristics:

Sometimes two nodes with a semantic edge between them contain sentences in which one sentence is clearly subsumed by the other. A shorter sentence may be connected to a number of longer sentences even though the subsumption relationship only exists between the shorter sentence and *one* of the longer sentences. For each connected component, this heuristics retains the semantic edge that maintains the inheritance relationship and removes the other semantic edges as well as all proximity edges. The final set of non-trivial connected components discovered after inheritance pruning map to the set of identified events. Intuitively, we maintain connections to 'more detailed' sentences.

3.4.5 Story Line Extraction

Each of the heuristics results in a non-trivial set of connected components, each of which maps to an event E_k. While all the sentences (nodes) in the connected component could be used to summarize the event, this leads to redundancy. We reduce redundancy by: (shown in Algorithm 2):

1. Nodes directly connected via semantic edges are reduced to one node - the node with the highest semantic similarity to the other nodes is maintained.

2. The remaining nodes within each connected component are ranked according to the number of occurrences of the sentence in $\mathbb{R}'$.

3. The top-m sentences are selected to be the story line summary of the event.

Algorithm 2: Story line extraction.

Input:
A set of non-trivial connected components: C
The sentence count for the synopsis: m

Output: *Event storylines*: E

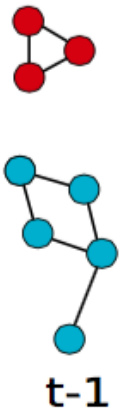
```
C' = extract_semantic_cc(C)
for C'_k ∈ C' do
    for v_j ∈ C'_k do
        v_j.sim = semantic_similarity(v_j, C'_k)
    E_k = extract_synopsis(v, m)
return E
```

3.5 Online Detection

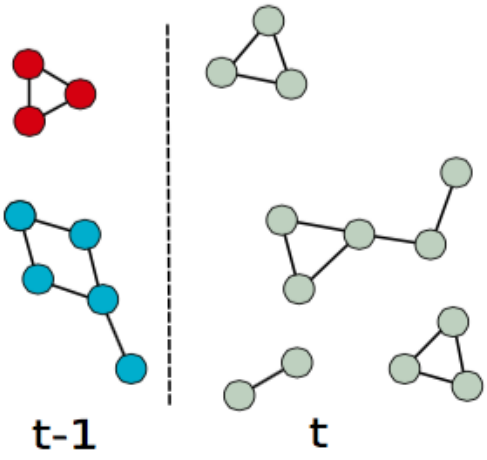
Recall that our offline version considers the entire document collection when constructing the graph. However, for online event detection, we can only consider documents occurring during or before time window t. The general steps associated with our

online method parallels our offline approach: location identification, target domain mapping, community extraction, and event detection. Among these steps, location identification, target domain mapping and event detection are the same as to those employed by our retrospective approach. The difference is during the community extraction step. That will be the focus of this section.

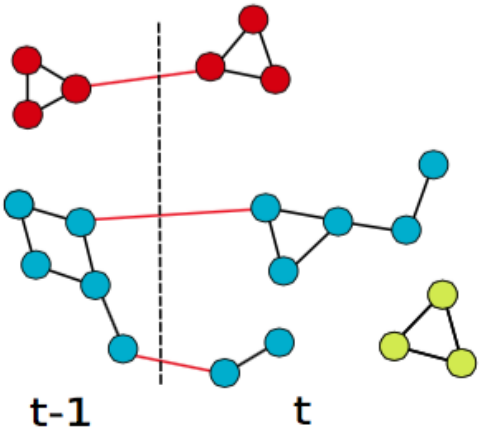
Algorithm 3 presents a high level view of our online event detection method. The input to the algorithm is a news article stream $\{D^t\}$ at time t, a location ontology $\mathbb{O}$ containing major localities around the world, a location $\mathbb{L}$ of interest to the user, and a domain $\mathbb{P}$ of interest to the user. The output of the algorithm is a set of events $\mathbb{E}$, represented as synopses of the documents discussing them. The algorithm begins by processing the documents d^t arriving at time t to identify the subset $\mathbb{R}_t$ of documents that include the target location (line 3) and the domain of interest (line 4). Once the relevant documents have been identified, a semantic subgraph g_t is generated using those documents (line 5). The approach for creating the semantic subgraph is the same as that of the offline algorithm. Once the subgraph has been generated, we need to look for communities within the particular time window t. We accomplish this by merging the semantic subgraph g_t for time window t with the detected communities C_{t-1} from the previous time window $t-1$.The three operations needed to extract events from the semantic subgraph are the following: add new nodes and edges operation (line 6), update communities operation (line 7), and remove outdated nodes and edges operation (line 8). The remainder of this section goes through these three operations. An example of detecting events for a time window of a document stream is shown in Figure 3.3.



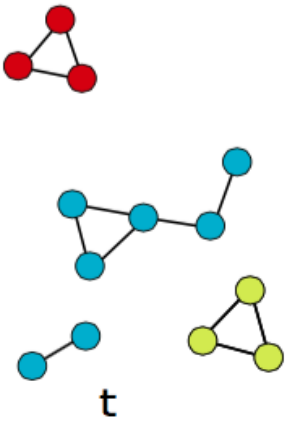
(a) Communities identified in time window $t-1$.



(b) Add new nodes and edges operation - time window t.



(c) Update communities operation - time window t.



(d) Remove outdated nodes and edges operation - time window t.

Figure 3.3: An example of event extraction for a time window in a document stream.

Algorithm 3: Our online event detection approach at high level.

Input:
A document stream: $\{D^t\}$
A location ontology: $\mathbb{O}$
A target location: $\mathbb{L}$
A target domain: $\mathbb{P}$

Output: *A set of events*: $\{E\}$

$T = generate\_domain\_dictionary(\mathbb{P}, \theta)$
for $d^t \in \{D^t\}$ **do**
  $\mathbb{R}_t = identify\_location\_relevant\_docs(\mathbb{L}, d^t, \mathbb{O})$
  $\mathbb{R}_t = identify\_domain\_relevant\_docs(\mathbb{R}_t, T)$
  $g_t = create\_semantic\_graph(\mathbb{R}_t)$
  $C'_t = add\_new\_nodes\_edges\_operation(g_t, C_{t-1})$
  $C''_t = update\_communities(g_t, C_{t-1}, C'_t)$
  $C_t = remove\_outdated\_nodes\_edges(g_{t-1}, C''_t)$
  for $C_{t,k} \in C_t$ **do**
    $E_{t,k} = identify\_event(C_{t,k})$
    $E_{t,k} = generate\_synopsis(E_{t,k})$
return $\{E_{t,k}\}$

3.5.1 Add New Nodes and Edges Operation

A set of articles arrive during time window t. Using this information, a semantic subgraph g_t for time window t is generated based on documents in $\mathbb{R}_t$, i.e. the documents retained after location identification and target domain mapping. The *add_new_nodes_and_edges* operation adds the nodes and edges of g_t to the communities C_{t-1} identified from the previous time window $t-1$. The result is denoted as C'_t. Figure 3.3a shows the two communities (red and blue) identified for the previous time window $t-1$, and Figure shows that nodes and edges in g_t that are added to C_{t-1} (a node colored gray indicates that it has not been assigned to a specific community).

3.5.2 Update Community Operation

The update community operation calculates the Semantic Similarity between nodes in g_t and C_{t-1}, and inserts corresponding semantic edges into C'_t. Recall that a semantic edge connects two nodes having a high semantic similarity. The result after applying the *update_community* operation is denoted as C''_t. In Figure 3.3c, the three added semantic edges connecting g_t and C_{t-1} are shown in red. Then nodes in g_t that connect to C_{t-1} are are assigned to their corresponding communities, while other nodes in g_t form their own communities, e.g., the nodes colored yellow form a new community since none of these nodes connect to C_{t-1}.

3.5.3 Remove Outdated Nodes and Edges Operation

The *remove_outdated_nodes_edges* operation goes through all the nodes and edges in C''_t, removing nodes and edges belonging to g_{t-1}, i.e., only the nodes and edges in g_t are retained. The result is the final detected communities for time window t, denoted as C_t. Figure 3.3d shows the retained nodes, which form three communities. Two of them have predecessors in C_{t-1}, and the third one is a newly formed community. Notice that if communities were not done with the nodes and edges from $t-1$ and t, then there would have been four resulting communities instead of three.

3.6 Offline Detection Evaluation

We now evaluate our proposed retrospective event detection approach on two distinct data sets. We begin this section by describing the data sets and specific target domain event detection tasks. We then empirically evaluate different steps of the methodology, comparing our approach to other state of the art methods.

3.6.1 Data Sets & Tasks

For our empirical analysis, we consider two data sources (EOS and TREC) and four event detection tasks. The remainder of this subsection describes them.

Population Displacement Using EOS data: The EOS archive contains over 600 million publicly available open-source media articles that have been actively compiled since 2006. New articles are being added at the rate of approximately 100,000 per day from over 20,000 Internet sources in 46 languages. For this analysis, we use a subset of 5 millions English news articles published in 2013 and 2014 that are related to the Middle East, with a focus on Iraq[3].

For the task of identifying events from these 5 million Iraq-related EOS new articles, we work with subject matter experts (SMEs) studying population displacement in the Middle East. Since Iraq has been experiencing renewed security and displacement for the past decade, we are interested in identifying events and story lines related to two different topic areas or domains: violence and governance. We do not have a ground truth event catalog for the 5 million articles. Therefore, we use a restricted subset of the data to create a ground truth data set. We consider the subset of articles published from Dec 9 2013 to Dec 31 2013 reporting on Anbar, a province of Iraq, where Islamic State of Iraq and Syria (ISIS) has been active since 2011. Out of the 5 million articles, approximately 500,000 articles from over 1200 sources fall within the target time period. Subject matter experts create a detailed timeline of events in Anbar for the violence domain (39 events) and the governance domain (30 events) during this 3 week period. For this part of the evaluation, we apply our proposed event detection approach to identify events occurring in Anbar having either the

[3]These documents are either published by Iraqi news agencies, or contain a term or a phrase related to Iraq, e.g., Iraq, Baghdad, Gulf War.

target domain of violence or governance. The identified events are evaluated against the ground truth events manually by SMEs.

Civil Unrest Using TREC data: The TREC-TS-2014F data set is a news article corpus provided by NIST[4]. It includes 20 million news articles published from November 2011 to April 2013. As part of the data set, NIST provides a list of 15 ground truth incidents. Given that articles in this data set are collected during different periods (as opposed to continuously) and the ground truth incidents occur in a number of discontinuous time periods, we select two time periods to conduct our analysis: Dec 4, 2011 to Dec 25, 2011, and Jan 13, 2012 to Jan 25, 2012. During these time periods we have at least 100,000 articles per day. These two time periods contain two ground truth incidents (Russian civil unrest and Romanian civil unrest). The 2011 Russian civil unrest contains 25 events, while the 2012 Romanian civil unrest contains 13 events. In total, we have 4.4 million TREC articles in this evaluation. Our goal is to identify events related to these two incidents from the documents.

3.6.2 Location Identification

In this subsection, we explain the construction of our location ontology and test the accuracy of our approach for determining the primary location of a news article. We build our location ontology using *Wikipedia* and *Statoids* [11]. *Wikipedia* has a set of pages listing all the major cities around the world by country, while *Statiods* lists governorates and the governorates' capitals for each country. Leveraging these two sources, we construct an ontology containing approximately 7,600 locations that include countries, governorates, governorates' capitals, and other major cities. Recall that the location with the highest frequency in an article is considered the primary

[4]Available at `http://www.trec-ts.org`

location the article is discussing. Due to space limitations, we do not show a complete evaluation of our ontology accuracy for determining the primary location. In general, our approach led to accuracies of over 80%. We will show in Section 3.6.6 that this is sufficient accuracy for the event detection task since processing a few additional documents does not impact an event detection approach that considers both semantic content and frequency when determining events.

We pause to mention that location is important for the geographical mapping of the event AND for reducing the search space of events. For example, in our location ontology, the subtree rooted at Anbar, Russia, and Romania have 30, 229, and 78 locations, respectively. This pruning allows for the construction of considerably smaller semantic graphs (with 100s to 1000s of nodes) than if the construction was done using the complete corpus across all locations.

3.6.3 Target Domain Mapping

Using the semi-supervised methodology described in section 3.4, we construct three domain dictionaries with help from our subject matter experts - one for violence, one for governance, and one for civil unrest. Table 3.1 shows the number of concepts identified during each step of domain dictionary construction. We see that the thesaurus and ontology augmenting adds a large number of relevant concepts (approximately a factor of 10) and very little noise. On average, 95% of the generated concepts are considered relevant by SMEs.

For each event detection task, we maintain articles with titles that contain at least one concept from the corresponding domain dictionary (title domain mapping strategy). We also considered a strategy that retains articles if the concept appears in the body of the article (body domain mapping strategy). Table 3.2 shows a comparison between the two strategies. The articles identified by each approach are hand

Table 3.1: Domain dictionary creation statistics.

	# Seed Concepts	# Concepts Generated During Augmenting	# Concepts Retained after SMEs' Validation
Violence	3	28	28 (100%)
Governance	10	115	111 (97%)
Civil Unrest	3	28	25 (89%)

evaluated by our project team. We assess the quality of each strategy using a Signal to Noise Ratio (SNR), where SNR is defined as the ratio between the number of documents correctly identified as relevant to the target domain and the number of documents falsely identified as relevant to the target domain. The higher the SNR, the stronger the result. We also consider the miss ratio for our method, where the miss ratio is defined as the number of documents not identified when employing the title domain mapping strategy divided by the number of documents correctly identified by the body domain mapping strategy.

The results show that the title domain mapping strategy has a high SNR compared to the body domain mapping strategy. However, we miss between 8% and 20% of the articles that are relevant. While this number seems high at first glance, we will show, that this miss rate does not result in significant deterioration of the event detection results. However, the additional noise associated with adding documents that are not relevant does lead to a reduction of accuracy for event detection in these data sets. Because of this, we use the title domain mapping strategy as part of our methodology. In future work, we will consider hybrid approaches that may lead to a reduction in the miss rate, while limiting the amount of noise added to the retained documents.

Table 3.2: Article body vs. title domain mapping strategies.

		Anbar, Violence	Anbar, Governance
Using article body	Retained	432	354
	Correct	181	162
	SNR	0.72	0.84
Using article title	Retained	166	131
	Correct	166	130
	SNR	Inf	130
	Miss Rate	8.28%	19.75%

3.6.4 Semantic Graph Generation

We generated a semantic graph G for each of our tasks. Table 3.3 shows the average number of nodes and edges each day for the different cases. The proximity edges are straightforward to determine using the proposed method in Section 3.4. Recall that the semantic edges are determined using two parameters, the relative edit distance (RED) and the relative common sequence length (RCS). Both of these parameters require threshold settings. This remainder of this section considers different setting values and their sensitivity.

To better understand the effect of these threshold settings, we collect 23,000 random pairs of sentences, each of which consists of two sentences with different meanings, and 2,500 random pairs of sentences, each of which consists of two sentences with the same meaning. The similarity and differences in the sentences were manually determined. For each pair, we calculate the relative edit distance (RED) and the relative common sequence length (RCS). Figure 3.4 shows a sensitivity analysis for these two parameters. The x-axis represents the relative edit distance (left) and the relative common sequence length (right). The values are between 0 and 1.

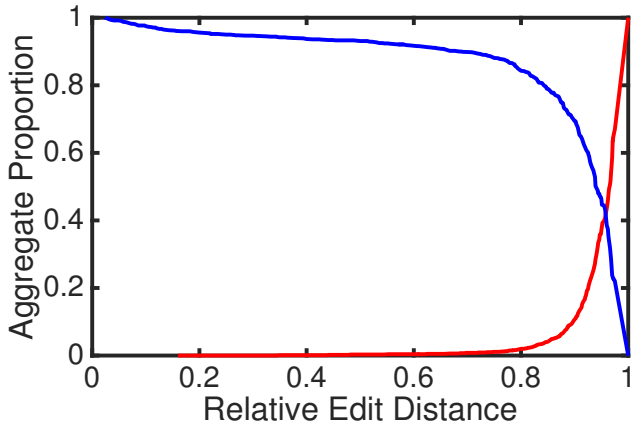
Table 3.3: Semantic graph statistics - averages per day.

	#Nodes	#Semantic Edges	#Proximity Edges
Anbar Violence	67	17	9
Anbar Gover.	103	33	9
Russia Unrest	1,075	224	185
Romania Unrest	234	52	46

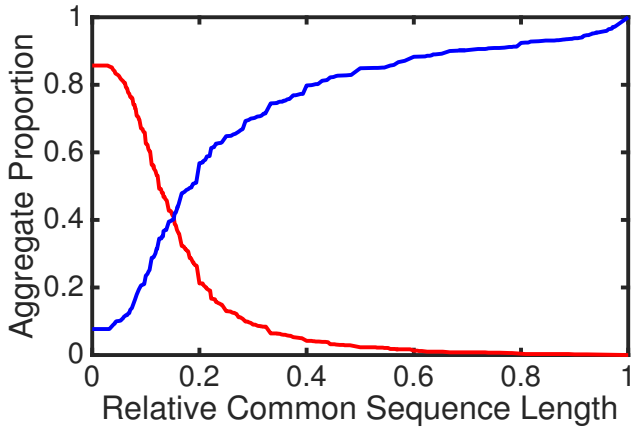
The y-axis represents the percentage or proportion of sentences that are considered similar or different for each RED or RCS value. As the plot shows, the majority of sentence pairs with the same meaning (blue line) can be identified if the threshold for the relative edit distance is below 0.8. The majority of sentence pairs with different meanings (red line) are not considered the same until the RED is larger than 0.8. For our experiments, we initially choose to be conservative and use a RED threshold of 0.2. We apply the same approach when considering relative common sequence length and find that 0.8 is a good conservative threshold. Both of these plots suggest that the sensitivity of these two thresholds is low in general. Additional extensive empirical analysis shows that the optimal threshold setting for relative edit distance and relative common sequence are 0.1-0.2 and 0.8-0.9 for our detection tasks, respectively.

3.6.5 Event Detection

We begin this subsection by comparing the event detection accuracy of our approach, Dynamic Sentence Graph (DSG), to state of the art methods. We then empirically



(a)



(b)

Figure 3.4: Plots of the aggregate proportion of sentence pairs with different meanings (Figure 3.4a–red line); Complement of aggregate proportion of sentence pairs with same meaning (Figure 3.4a–blue line); Aggregate proportion of sentence pairs with same meaning (Figure 3.4b–blue line); Complement of aggregate proportion of sentence pairs with different meanings (Figure 3.4b–red line).

evaluate the proposed heuristics to better understand their impact for event detection in different domains. Finally, we discuss different parameter settings, focusing on their sensitivity for the event detection task.

Event detection experiment details

We compare DSG to four state of the art event detection approaches described in Section 3.2, [71] (Meme), [54] (Bursty), [96] (KeyGraph) and [84] (Pat). Different from others, the Pat approach is a binary detector, i.e., it only determines whether an event in the target domain exists or not at a specific time and location, without providing a description of the event. This method also requires subject matter experts to manually construct seed patterns to detect events. For the two tasks of detecting civil unrest events in Russia and Romania, we use the same seed patterns as [84], since the types of events being detected are the same. For the two tasks related to detecting violence and governance events in Anbar, we have our subject matter experts create the same number of seed patterns as the civil unrest task. In evaluating the detected events, for the two tasks of detecting violence and governance events in Anbar, we evaluate the top eight events identified by each of the five approaches against the significant ground truth events in terms of precision and recall. For the two tasks of detecting civil unrest events in Romania and Russia, we compare the top 12 and top 20 identified events, respectively, since these two tasks involve significantly more ground truth events (refer to Table 3.4). We focus on this subset of events because they each have over five documents in the corpus mentioning them. We will refer to these events as 'representative' or 'popular' events. Table 3.4 shows the number of ground truth events, representative/popular ground truth events, and the number of relevant documents for each task. The majority of other ground truth events have at most one or two documents discussing them in the document collections.

Table 3.4: The number of retained documents, ground truth (GT) events, and significant GT events.

	#Documents	#GT Events	# Popular GT Events
Anbar Violence	166	42	6
Anbar Gover.	131	24	4
Russia Unrest	675	25	15
Romania Unrest	135	13	7

Binary event detection accuracy

We begin by evaluating binary event detection, i.e. determining whether events in the target domain exist or not, using the five event detection approaches (DSG, Meme, Bursty, KeyGraph, Pat). A *Hit* is recorded if events in the target domain are detected and there is at least one popular ground truth event in the target domain occurring on the same day. In this analysis, we ignore the context of the events. Table 3.5 gives the precision and recall of the detected events for each method for each task (Pat-4 uses 4 seed patterns and Pat-10 uses 10 seed patterns). Note that for this experiment, we do not use the subcommunity or inheritance pruning heuristics. The table shows that DSG and Pat perform significantly better than the other methods. The Pat approach's weakness is the lack of context about the detected event. We know that it is a civil unrest event, but details about it are unknown. The Meme approach has a lower recall because of the method's reliance on quoted text segments. The Bursty approach has the worst precision and recall. The KeyGraph method identifies most

of the ground truth events; it has trouble when the ground truth events are the same type, even if the time period does not overlap.

Content-based event detection accuracy

For the next experiment, we evaluate the content of events detected by the four non-binary event detection approaches (DSG, Meme, Bursty, KeyGraph). A *Hit* is recorded if a detected event maps to a ground truth event when considering both the content and time of the event. The results are shown in Table 3.6. We see that our approach (DSG) significantly outperforms the other approaches in terms of both precision and recall.

Overlapping event detection identification

Recall that one of our tasks is to detect events even if they are overlapping in time. Table 3.7 shows the number of sufficiently represented ground truth events (over 5 supporting documents) per day, and the number of detected events mapping to each of these ground truth events in a six-day window across the four tasks. For example, the first cell tells us that there are two ground truth events on 2013-12-21, i.e., they are overlapping in time. The cell below (labeled D) indicates that at least one story line for each of the two ground truth events is detected. From the table, we can see that our approach does detect overlapping events well for all the event domains except for the Russian unrest events. We believe this is a result of the skewed frequency distribution of the documents. The data set has a strong frequency skew toward a few significant events. The other ground truth events are overwhelmed by those. So the missed overlapping events have less to do with overlaps and more to do with the highly skewed document distribution.

3.6.6 Event detection additional heuristics

Table 3.8 shows the precision and recall when incorporating the subcommunity and inheritance pruning heuristics. We see that we get an improvement in some cases, but not others. Because this initial analysis does not give us enough insight about when these heuristics are beneficial, we consider a second approach for evaluating them. We use the notion of *semantic purity*.

An event can have multiple storylines. If we do not want to separate them, then our basic approach without the added heuristics is sufficient. However, if we want to separate them, then we want each connected component to focus in on a smaller number of ideas. To measure the number of ideas in a storyline, we introduce the notion of a semantic group S_g. A semantic group is a group of nodes connected by semantic edges in a connected component C_i. We define semantic purity S_p as the number of semantic groups in a connected component: $S_p = |\{S_g | S_g \in C\}|$.

The lower the semantic purity, the less semantic diversity a connected component contains. Therefore, if the goal is to separate storylines, we want a lower semantic purity. Table 3.9 shows the average semantic purity of the connected components when using the basic connected component algorithm, the sub-community heuristics, and the inheritance pruning heuristics. We find that both heuristics improve the purity of the connected component. However, neither is consistently better on different domains. We observe that connected components in the Russia civil unrest graph always have the highest semantic purity. We attribute this to the fact that the articles reporting on the Russian civil unrest are usually much longer than the articles associated with the other three tasks (the average number of words per article is 243, 297, 1214, and 895 for the four tasks, respectively). Longer articles increase the semantic diversity of a connected component, thus increasing the semantic purity.

Table 3.5: Event detection precision (P) and recall (R) of different algorithms working as binary detectors.

		DSG	Meme	Bursty	Key-Graph	Pat-4	Pat-10
Anbar Violence	P	**100%**	100%	33.3%	100%	**100%**	**100%**
	R	60%	40%	20%	60%	**100%**	**100%**
Anbar Gover.	P	**100%**	100%	25%	100%	**100%**	**100%**
	R	**75%**	50%	50%	75%	50%	75%
Russia Unrest	P	**100%**	**100%**	33%	37.5%	**100%**	NA
	R	83.3%	50%	33%	**100%**	83.3%	NA
Romania Unrest	P	**100%**	**100%**	33%	50%	**100%**	NA
	R	**100%**	**100%**	25%	75%	75%	NA

In contrast, the articles reporting on Anbar are usually much shorter, and some are reprints of other reports, thereby reducing the semantic diversity. In general, we recommend the inheritance pruning heuristics if most of supporting documents of an event derive from a few original reports, because inheritance relationships between sentences in such documents are more common than in ordinary ones. In contrast, the subcommunity heuristics may be a good option if most of supporting documents for an event are from a large number of original reports.

Determining non-trivial connected components

We now discuss the parameter setting related to determining non-trivial connected components. Using the parameter values specified in the previous section, the constructed semantic graphs for the four tasks are shown in Figure 3.5. The trivial connected components are green and the non-trivial connected components are other colors. To determine the cutoff between the trivial and non-trivial connected components, we plot the total occurrences of sentences in the detected events (each connected

Table 3.6: Event detection precision (P) and recall (R) of different algorithms when taking event content into consideration.

		DSG	Meme	Bursty	KeyGraph
Anbar	P	**87.5%**	87.5%	12.5%	75%
Violence	R	**66.7%**	33.3%	16.7%	50%
Anbar	P	**100%**	87.5%	25%	75%
Gover.	R	**75%**	50%	50%	75%
Russia	P	**100%**	90%	35%	30%
Unrest	R	**40%**	26.7%	20%	26.7%
Romania	P	**100%**	**100%**	33%	50%
Unrest	R	**85.7%**	71.4%	42.9%	28.6%

Table 3.7: The number of sufficiently represented ground truth events (G) per day, and the number of detected events (D) mapping to each of the ground truth events in a six-day window.

		D1	D2	D3	D4	D5	D6
Anbar, Violence	G	2	0	1	1	0	0
13-12-21 to 13-12-26	D	2	0	1	1	0	0
Anbar, Gover.	G	0	0	1	0	1	1
13-12-26 to 13-12-31	D	0	0	1	0	1	1
Russia, Unrest	G	1	2	3	0	4	0
11-12-6 to 11-12-11	D	0	1	0	0	2	0
Romania, Unrest	G	0	1	0	0	0	4
12-1-19 to 12-1-24	D	0	1	0	0	0	3

Table 3.8: Event detection precision (P) and recall (R) leveraging different heuristics.

		Connected Components	Sub-community	Inheritance Pruning
Anbar	P	87.5%	**100%**	**100%**
Violence	R	66.7%	**83.3%**	66.7%
Anbar	P	**100%**	**100%**	**100%**
Gover.	R	**100%**	**100%**	75%
Russia	P	**100%**	**100%**	**100%**
Unrest	R	**40%**	33.3%	33.3%
Romania	P	**100%**	**100%**	**100%**
Unrest	R	**85.7%**	71.4%	71.4%

Table 3.9: Semantic purity of connected components.

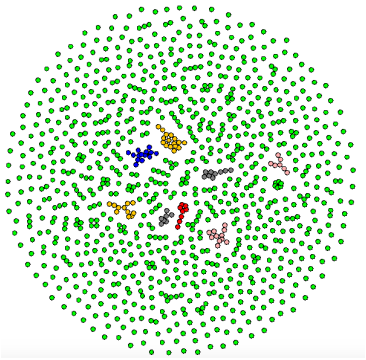
	Connected Component	Sub-community	Inheritance Pruning
Anbar, Violence	4.5	2.1	2.3
Anbar, Governance	4.1	1.9	1.8
Russia, Unrest	8.5	3.9	4.2
Romania, Unrest	6.7	3.7	4.3

component is a detected event) in Figure 3.6a and Figure 3.6b. The x-axis represents the identified events sorted by total frequency of sentences in them and the y-axis represents the total frequency of sentences in the detected events.

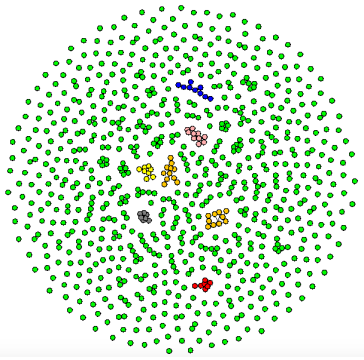
Observing these plots, we see that they follow a power-law distribution. Thus, we set the threshold which determines whether a connected component is significant or not to the spot where the long tail starts (marked by the vertical black bar intercepting each line). Using this approach, both the Anbar, violence semantic graph, and the Anbar, Governance graph have 8 significant components, while the Russia civil unrest graph and the Romania civil unrest graph have 12 and 20 significant components, respectively.

Location Identification Sensitivity

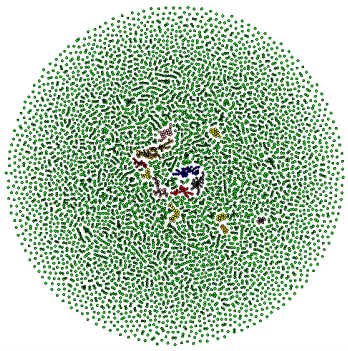
To assess the effect of location identification accuracy on our event detection task, we conduct a location sensitivity analysis. As explained in Section 3.6.2, 632 EOS documents are determined by our location identification approach to be discussing Anbar. The violence event in Anbar best represented in EOS is supported by 120 EOS documents. We then add a different number of EOS articles known to be discussing other locations to the 120 documents. These added articles are considered *noise*. By



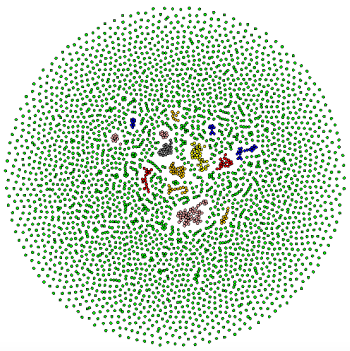
(a) Anbar, Violence, 2013-12-09 to 2013-12-31.



(b) Anbar, Governance, 2013-12-09 to 2013-12-31.

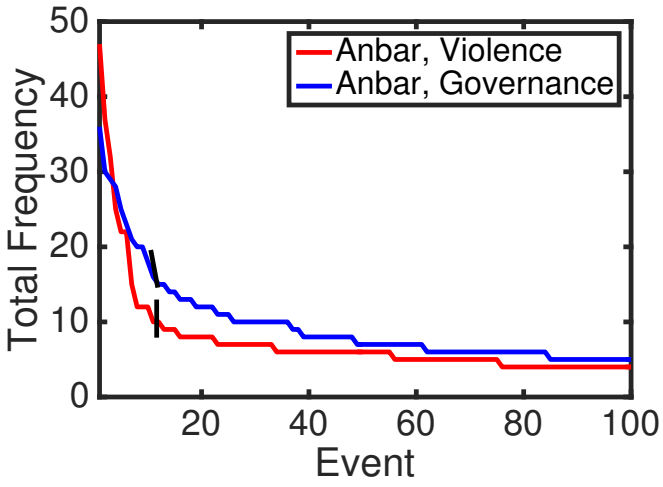


(c) Russia, Civil Unrest, 2011-12-04 to 2011-12-25.

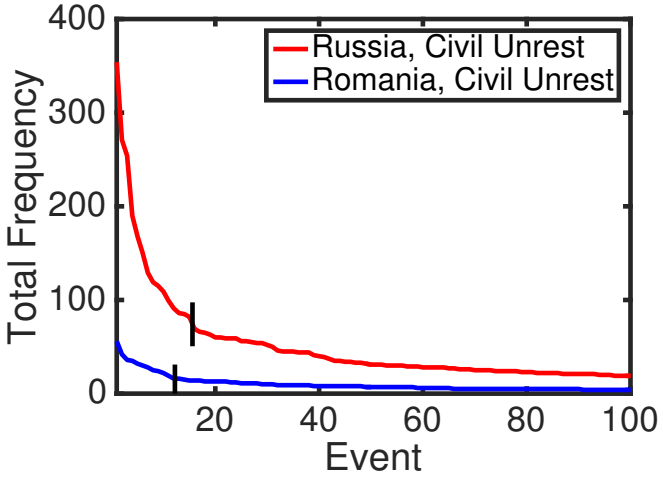


(d) Romania, Civil Unrest, 2012-01-13 to 2012-01-25.

Figure 3.5: Semantic graphs for different tasks. Green – trivial; other colors – non-trivial.



(a) Anbar.



(b) Russia, Romania.

Figure 3.6: The total occurrences of sentences in connected components.

Table 3.10: Event detection accuracy with varying location identification accuracy.

# Noise Documents	0	30	40	80	120
Location Accuracy	100%	80%	75%	60%	50%
Event Detection Precision	87.5%	87.5%	75%	50%	50%

adding different levels of noise to the target documents, we can better understand the impact of noise on the final event detection results. In the first experiment, we add 30 noise documents (reducing the location accuracy to 80%). In the second experiment, we add 40 noise documents (reducing the location accuracy to 75%). In the third and fourth experiments, we add 80 and 120 noise documents, respectively. We apply our event detection approach to the constructed document sets to detect violence events occurring in Anbar, and then evaluate the detected events against the ground truth events. The results are shown in Table 3.10. We see that an 80% accuracy in location identification is sufficient for target domain event detection. This makes sense since documents from other locations are not likely to have the same themes are those in the target location. However, when the amount of noise gets large, it impacts the quality of the detected events.

3.6.7 Event Story Lines

We evaluate the story line summaries generated by our approach against those generated by the other content-based event detection approaches. We also compare all the summaries to a "gold standard" summary obtained from a well known document summarization approach (PageRank) introduced by Mihalcea and Tarau [81]. For the gold standard, we use the most relevant document to summarize the event. Two SMEs

Table 3.11: The average rating of different event detection approaches compared to a gold standard for the quality of event summary. NA means that an approach failed to detect the ground truth event.

	DSG	Meme	Bursty	Key Graph	Gold Standard
GT Event 1	3.5	3.5	2	3.5	5
GT Event 2	4.5	1.5	1	2.5	5
GT Event 3	3	NA	NA	1	5
GT Event 4	4.5	2	1	3	4
GT Event 5	4	NA	NA	1	4
GT Event 6	2	NA	2	NA	3.5

rated all the summaries using a scale from 1–5, where 1 is the lowest and 5 is the highest rating based on informativeness, readability and accuracy. Since the state of the art approaches only detected 6 out of the 10 representative ground truth events associated with Anbar, this experiment focuses on those 6 events. The average ratings of the four approaches and the gold standard are shown in Table 3.11. We see that the SMEs almost always prefer the gold standard. However, our approach results in the highest average rating of the event detection methods. We attribute this to the sentence level nodes in the semantic graph. Table 3.12 shows the first four lines of the event storyline summaries given by DSG, Meme, Bursty, and KeyGraph approach for one of the six events. In general, the methods using keywords resulted in less detailed summaries, while the methods using phrases were more informative, readable, and accurate to the SMEs.

Table 3.12: The story line summaries given by DSG, Meme, Bursty, and KeyGraph approach for a representative ground truth event and their averaged rating (AR) by SMEs.

Approach	AR	Story Line Summary
DSG	4.5	Iraqi police officials say Alwani's brother and three guards were killed after they opened fire on security forces at dawn on December 28 as they arrived to arrest him. Alwani a Sunni lawmaker who had ...
Meme	1.5	Army troops with police special forces were trying to arrest Alwani. We told him that we had a warrant for his arrest and arrested him. I call upon Sunni's protesters and sons of Ramadi to insist upon your ...
Bursty	1	amid, government, group, Maliki, prime minister, city, Anbar, December
KeyGraph	2.5	Alwani's release, Anbar's provincial council, the death of his brother, a strong critic of Maliki, minority Sunni leaders, attacks that killed Iraqi soldiers, a clear violation, the core of the Iraqi constitution, its articles ...

3.6.8 Depicting Event Dynamics

Our event detection approach is also helpful for identifying the evolving dynamics of events. We can accomplish this by looking at the identified events and their sentence overlap through time. Figure 3.7 shows the total number of occurrences of sentences in detected events over multiple days. The x-axis is the date and the y-axis is the frequency of the sentences (nodes in G) associated with an event. Fluctuations in the frequency of these sentences highlights the rise and fall of an event's media popularity. While event dynamics is not the focus of this work, this figure highlights an additional value of our event detection approach.

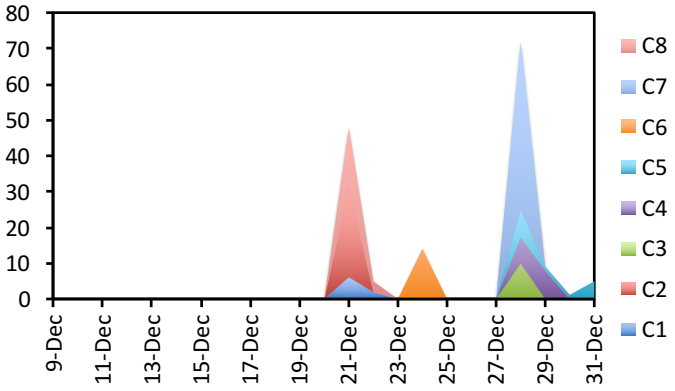
3.7 Online Detection Evaluation

Since our online approach uses the same location identification component and target domain mapping component as our offline approach, this section focuses on semantic graph creation and event extraction.

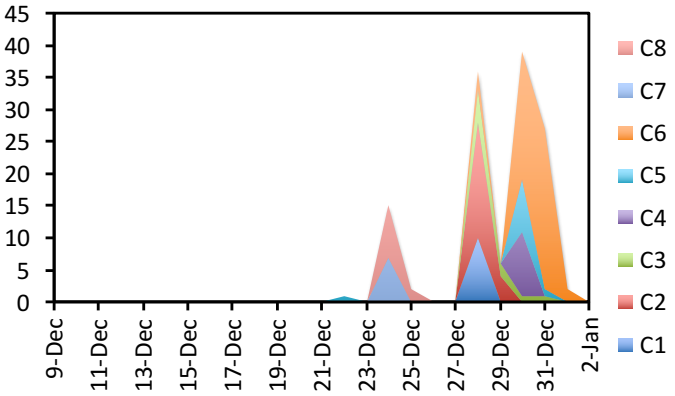
3.7.1 Semantic Graph Generation

We evaluate our online event detection approach (referred to as Online in the following experiments) on the same four tasks as we evaluate our retrospective approach. We set the length of a time window to one day, two days, and three days, and conduct three groups of semantic graph generation with these three time window lengths, respectively.

For the communities generated by the Online approach, we sort these communities by the total occurrences of sentences in them, and intercept the plot where the long tail starts. The communities prior to where the long tail starts are considered to be



(a) Anbar, Violence.



(b) Anbar, Governance.

Figure 3.7: Frequency dynamics of sentences in detected events.

Table 3.13: The number of events detected by the Online approach at different time window lengths.

Task	Window Length	#Events
Anbar, Violence	1	40
	2	33
	3	29
Anbar, Governance	2	22
	2	19
	3	19
Russia, Unrest	1	42
	2	36
	3	25
Romania, Unrest	1	36
	2	29
	3	18

non-trivial communities, and each non-trivial community represents an event. Table 3.13 shows the number of events detected by the Online approach in different tasks.

3.7.2 Event Detection

We compare the Online approach to our retrospective approach (referred to as Offline in the following experiments) and state of art methods in terms of binary event detection accuracy and content-based event detection accuracy.

Binary Event Detection Accuracy

We begin by evaluating binary event detection, i.e., determining whether events in the target domain exist or not, using four different approaches (Online, Offline, Meme, and KeyGraph). We select the Meme approach and the KeyGraph approach as baselines in evaluating our Online approach, because both of them could be easily adapted to

detect events in an online fashion, and their online adaption have similar principles to the approaches specifically designed for online event detection [20] [30]. For the Meme approach and the KeyGraph approach, we use them to detect events for each time window of a news article stream. We do not include the Bursty approach in this evaluation because it cannot be adapted to an online setting.

Stated in Section 3.6, in evaluating the events detected by the Offline approach, we use the top 8 events identified for the two tasks of detecting violence and governance events in Anbar, and the top 12 and top 20 identified events for the two tasks of detecting civil unrest events in Romania and Russia. In this evaluation, we also select the 8, 8, 20, and 12 most significant events (having the most occurrences of sentences in them) from the non-trivial communities generated by the Online approach, as well as the most significant 8, 8, 20, and 12 events detected by the Meme approach and the KeyGraph approach, and compare them against the popular ground truth events in the 4 tasks.

Besides comparing the detected events against the popular ground truth events, we are also interested in how well our proposed approaches capture all the ground truth events. For this purpose, we use all the non-trivial events generated by the Online approach (refer to Table 3.13) and compare them against all the ground truth events. To ensure a fair comparison, we select the same number of significant events generated using the Offline approach, the Meme approach, ands the KeyGraph approach as shown in Table 3.13, and compare the detected events to these ones.

Table 3.14 shows the event detection precision, recall, and F1 of the Offline approach and the Online approach with different time window lengths. We can see that (1) for the Online approach, setting the time window to different lengths does not have a significant impact on the event detection accuracy; (2) in comparison to the popular ground truth events, the performance of the Online approach is almost

as good as the Offline approach in terms of precision and recall; (3) in comparing against all the ground truth events, the Online approach has better F1 score than the Offline approach for all the four tasks: it outperforms the Offline approach with respect to recall, with a slight sacrifice of precision. The Online approach's success in recall can be attributed to the fact that it detects events at the arrival of a new time window, instead of considering all the time windows as a whole. In the Online approach, top events in a time window can always be spotted, even if they are much more trivial than top events in some other time windows. In contrast, the Offline approach pays most of its attention to significant events from a few time windows, ignoring events from other windows. The Online approach has an advantage over the Offline approach in that it adaptively adjusts its criterion for determining communities as non-trivial for each time window, while the Offline approach uses the same criterion to determine whether or not a component is trivial or not. Figure 3.8a and 3.8b highlight the comparison of event detection F1 value of different approaches. We can see that both the Online approach and Offline approach outperform the Meme approach and the KeyGraph approach significantly.

Content-based Event Detection Accuracy

For the next group of experiments, we evaluate the content of events detected by the different event detection approaches. Table 3.15 shows the event detection precision, recall, and F1 of the Offline approach, and the Online approach with different time window lengths. We see that when considering content, the online approach has the same or better accuracy, precision and recall as the offline approach.

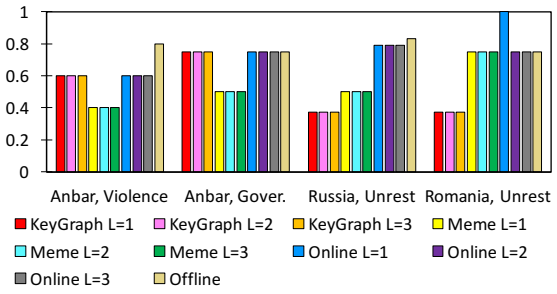
Figure 3.8c and 3.8d compares the event detection F1 value for the different approaches. Similar to the binary event detection results, the offline and online approaches we propose perform as well or better than the other methods.

Table 3.14: Event detection precision (P), recall (R), and F1 of different algorithms working as binary detectors with different time window lengths (L).

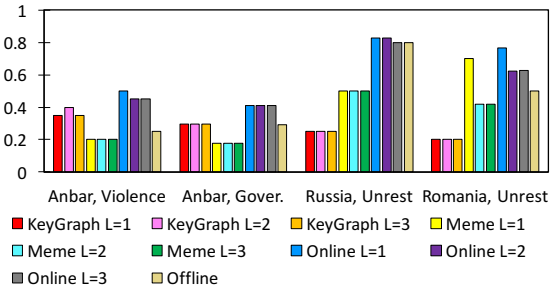
			Online			Offline
			L = 1	L = 2	L = 3	
Popular GT Events	Anbar, Violence	P	100%	100%	100%	100%
		R	60%	60%	60%	80%
		F1	0.6	0.6	0.6	**0.8**
	Anbar, Gover.	P	100%	100%	100%	100%
		R	75%	75%	75%	75%
		F1	**0.75**	**0.75**	**0.75**	**0.75**
	Russia, Unrest	P	95%	95%	95%	100%
		R	83.3%	83.3%	83.3%	83.3%
		F1	0.791	0.791	0.791	**0.833**
	Romania, Unrest	P	100%	100%	100%	100%
		R	100%	75%	75%	75%
		F1	**1.0**	0.75	0.75	0.75
All GT Events	Anbar, Violence	P	100%	100%	100%	100%
		R	50%	45%	45%	25%
		F1	**0.5**	0.45	0.45	0.25
	Anbar, Gover.	P	100%	100%	100%	100%
		R	41.1%	41.1%	41.1%	29%
		F1	**0.411**	**0.411**	**0.411**	0.29
	Russia, Unrest	P	92.1%	92.1%	100%	100 %
		R	90%	90%	80%	80%
		F1	**0.828**	**0.828**	0.8	0.8
	Romania, Unrest	P	87.5%	83.3%	100%	100%
		R	87.5%	75%	62.5%	50%
		F1	**0.765**	0.624	0.625	0.5

Table 3.15: Event detection precision (P), recall (R), and F1 of different algorithms working with different time window lengths (L) when taking event content into consideration.

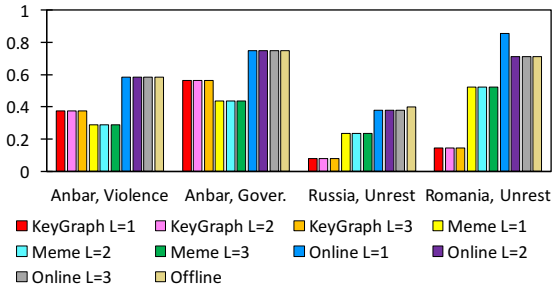
			Online			Offline
			L = 1	L = 2	L = 3	
Popular GT Events	Anbar, Violence	P	87.5%	87.5%	87.5%	87.5%
		R	66.7%	66.7%	66.7%	66.7%
		F1	**0.583**	**0.583**	**0.583**	**0.583**
	Anbar, Gover.	P	100%	100%	100%	100%
		R	75%	75%	75%	75%
		F1	**0.75**	**0.75**	**0.75**	**0.75**
	Russia, Unrest	P	95%	95%	95%	100%
		R	40%	40%	40%	40%
		F1	0.38	0.38	0.38	**0.40**
	Romania, Unrest	P	100%	100%	100%	100%
		R	85.7%	71.4%	71.4%	71.4%
		F1	**0.857**	0.714	0.714	0.714
All GT Events	Anbar, Violence	P	77.5%	78%	75.8%	87.5%
		R	26%	23.8%	23.8%	14.2%
		F1	**0.201**	0.185	0.18	0.125
	Anbar, Gover.	P	90.9%	94.7%	94.7%	100%
		R	33.3%	33.3%	33.3%	20.8%
		F1	0.303	**0.315**	**0.315**	0.208
	Russia, Unrest	P	92.1%	92.8%	95.2%	100%
		R	60%	52%	48%	36%
		F1	**0.552**	0.482	0.456	0.36
	Romania, Unrest	P	87.5%	96.5%	100%	100%
		R	61.5%	46.1%	46.1%	38.4%
		F1	**0.538**	0.444	0.461	0.384



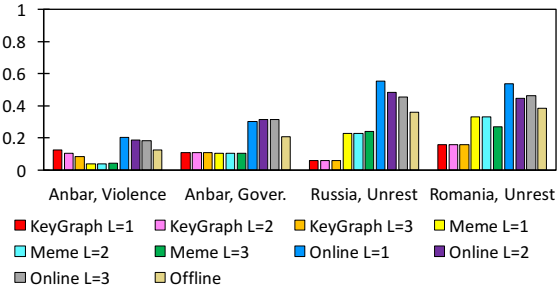
(a) Binary event detection against the popular ground truth events.



(b) Binary event detection against all the ground truth events.



(c) Content-based event detection against the popular ground truth events.



(d) Content-based event detection against all the ground truth events.

Figure 3.8: Event detection F1 value of different algorithms.

Finally, we present a case study which highlights a potential benefit of the Online approach to the Offline approach in terms of capturing non-popular ground truth events. For the task of detecting civil unrest events in Russia, Table 3.16 lists the top 3 events identified by the Offline approach, and the top 3 events identified by the Online approach on 12-21-2011. We can see that the events identified by the Offline approach have high numbers of supporting documents. In contrast, each of the 3 events identified by the Online approach have only 2 supporting documents, but the Online approach still identifies them, because they are the top ones in their own time window. The Online approach is capable of capturing much less significant events overall, as long as these events are the most important events in their own time window.

3.8 Conclusions

In this chapter, we propose a comprehensive framework that utilizes a location ontology and a domain dictionary to identify overlapping, target events using news articles from a large, noisy news corpus generated from multiple new sources. Using this framework, we propose an offline event detection approach, and an online event detection approach. To the best of our knowledge, our method is the first targeted event detection algorithm that detects events and story lines occurring at the same time. For both the offline and online approaches, we make use of a semantic graph constructed from sentences within articles from the corpus. We use a set of graph invariants (connected components, community structure, and node subsumption) on this semantic graph to help us identify popular events. Using this multi-relational graph allows us to capture different types of relationships between sentences in the document. The difference between our offline and online approaches is that our online

Table 3.16: The top three events identified by the Offline approach, and the top three events identified by the Online approach in the time window of 2011-12-21 for the task of detecting civil unrest events in Russia.

	# Supp. Docs.	Event Synopsis
Online	2	Two leaders of the protests ... were on wednesday released from prison in moscow
	2	A close ally of vladimir putin has been elected ... Speaker of russias state duma at its first sitting ...
	2	Putin probably wont lose in march ... His dominance of the country significantly curbed ...
Offline	354	Police clashed Tuesday on a central moscow ... trying to hold a second day of protests
	354	Police clashed with demonstrators protesting alleged election ... Anger boiled over against strongman Prime Minister ...
	67	The Kremlin has come under strong international pressure ... US secretary of state Hillary Clinton ...

approach maintains and updates the graph dynamically throughout time, instead of building it using all the documents in the newspaper collection as a whole. We believe this type of graph is the reason we perform better than the state of the art.

Extensive experiments on two large data sets demonstrate the strengths of our event detection method when compared to the state of the art. We also conduct detailed sensitivity analyses on different parameters to give researchers intuition about their settings, and show that our event synopses are effective in helping readers gain a better understanding of the detected events when compared to synopses generated by other methods. Finally, we compare our online approach to our offline approach, and show that the performance of our online approach is almost as good as the offline. We believe that this area of research is fruitful and necessary for not only identifying and understanding events in large, noisy corpora, but also understanding the types of information people find important to discuss.

Chapter 4

Detecting Events on Twitter and Understanding the Impact of Sampling and Noise on Event Detection

Detecting events rapidly is important for many different application domains, e.g. humanitarian crises, extremist attacks, etc. It is no surprise that social media, especially open platforms like Twitter propagate events faster than other traditional forms of media, e.g. newspapers. Event detection on Twitter has attracted a significant amount of research attention. Even so, detecting events that occur at a particular time and location is still in its infancy. In this chapter, we propose a simple algorithm which leverages geotagged bursty term graphs to detect events from a tweet stream. We conduct an extensive empirical evaluation of our approach against the state of the art and show that our simple modifications significantly improves the detection precision and recall when compared to the state of the art approaches. Because Twitter is such a noisy domain and the Twitter API only gives samples of the tweet stream, we then focus on understanding the impact of sample size and noise level on location-based event detection. Our goal is to understand 1) the impact of sample size on event detection accuracy, and 2) the impact of different types of noise and different levels of noise on event detection accuracy. We show that our approach is more robust than others to this noisy, partial data environment. To the best of our knowledge, this work is the first research to analyze the impact of sample size and noise level on event detection accuracy from tweets.

4.1 Introduction

Social media platforms, including Twitter, have been extensively leveraged to report real world events. Event and topic detection using Twitter has also been an active area of research [24] [74] [53] [47] [116] [70] [115] [73] [90] [94] [89] [71].

In this chapter, we propose leveraging geotagged bursty term graphs to detect events from a tweet stream. This location specific semantic graph contains words and relationships that rapidly increase in frequency at a particular location. Each node in this graph represents a *bursty* term, and an edge represents a co-occurrence of two bursty terms within a tweet. While the algorithm is a straightforward extension of state of the art methods, the use of this data structure to identify events significantly improves the precision and recall.

In addition to improving the accuracy of the state of the art event detection methods, we are interested in understanding the impact of data quality and sample size on event detection accuracy. We therefore also conduct two sensitivity analyses, one focused on understanding the impact of sample size on event detection accuracy and the other focused on understanding the impact of different types and levels of noise on event detection accuracy. Knowing this is important for understanding how "clean" the data needs to be and how much data is necessary to effectively detect events. To the best of our knowledge, this work is the first research to analyze the impact of sample size and noise level on event detection accuracy using Twitter.

To summarize, our contributions to the literature are as follows: (1) we propose a new event detection algorithm that takes advantage of geotagged bursty term graphs to identify events; (2) an empirical evaluation on three large real-world data sets shows that combining geography information with changes in bursting words significantly improves the event detection precision and recall when compared to the state of

the art; (3) we quantify the effects of sampling with different sampling fractions on accuracy of different event detection methods using Twitter data; (4) we quantify the effects of various levels of noise on the accuracy of Twitter event detection methods; and (5) we present an extensive discussion of the impact of sampling and noise on event detection.

4.2 Related Literature

This section reviews the literature in four areas: non-geotagged event detection on Twitter (methods for detecting events without considering geography), geotagged event detection on Twitter (methods for detecting events when considering geography), effects of sampling on Twitter event detection, and effects of noises on Twitter event detection.

4.2.1 Non-Location Event Detection

The majority of literature related to unsupervised event detection on Twitter does not consider geography information when detecting events. Instead, it considers bursty textual segments (e.g., terms, phrases) to represent an event [116] [115] [73] [65] [62] [30]. A number of supervised event detection methods for Twitter data have also been proposed [25] [30] [90] that use text feature vectors and dates to identify events.. More closely related to our approach, another set of proposed algorithms leverages different graph structures when detecting events on Twitter [24] [74] [53] [47] [41] [80] [47]. Aggarwal et al. [24] construct a graph to represent the interactions between entities in a tweet stream. This approach uses a tweet cluster to represent an event. Lin et al. [74] leverage an information diffusion network between users, combined with tweet content and temporal information to track popular events on Twitter.

Cataldi et al. [41] employ a Twitter following/follower network to quantify a user's significance in propagating information. Further, these values are used to modulate the *energy* of emergent keywords, and each strongly connected component of emergent keywords represents an event. Ferlez et al. [53] construct a heuristic bipartite graph for a tweet stream, and use changes in the bipartite graph to signal the occurrences of events. Meladianos et al. [80] propose a K-degenerate graph, in which nodes represent terms in tweets, and edges represent co-occurrences of pairs of terms in a tweet. This approach uses terms pertaining to the highest *KCore* to represent the trending event in a time window of the tweet stream. Chen and Neill [47] build a heterogeneous graph for a tweet stream, in which a node could be a tweet, a user, or a term, and an edge could represent a co-occurrence of two terms in a tweet, a communication between two users, a following/follower relationship, etc. An event is detected when nodes or edges exhibit abnormal behaviors when compared to their historical values. Our approach differs from all the above mentioned works since our graph is location specific, i.e. we have different graphs for different locations, and we use bustiness to determine whether or not a word should appear in the graph, only words with large changes in frequency are added to the graphs.

4.2.2 Geotagged Event Detection

Some previous work does consider location-based event detection [123] [66] [20] [89]. Zhou and Chen [123] extend the classical LDA topic model to incorporate the location and time information, such that each topic is drawn from a distribution of words, locations, and time. In order to accomplish this, they extract user location information from user profiles and map tweets based on those locations. Our model does not assume the availability of user location information. Abdelhaq et al. [20] construct a geographical grid, and represent the usage of a term as a distribution in the grid.

Having a small usage entropy suggests the main usage of the term occurs in a geographical space of limited extent. Zhang et al. [121] build a keyword co-occurrence graph, and calculate tweets' semantic similarities using the keyword-occurrence graph. Then they group tweets based on their semantic similarities and their geographical similarities, and use each group to represent an event. This work is the most similar to our work; however, they assume the tweets are geo-tagged, and their keyword co-occurrence graph is built on all the unigrams appearing in the tweets, while we only consider the *bursty* keywords in building the co-occurrence graph.

In general, what distinguishes our work from previous literature is the following: (1) using separate graphs for each location identified in the tweet stream, (2) using a location ontology to label the location associated with a tweet instead of requiring that a tweet has a geo-tag, and (3) generating graphs that only contain bursty terms. We will show that these simple differences in location identification and data representation have a significant impact on the precision and recall of the detected events.

4.2.3 Sampling and Twitter event detection

A few works investigate the effects of sampling on data mining tasks. Yates et al. [120] focus on the correlation between the Google Flu Trend and the sum of Influenza-like illness (ILI) related keywords in tweets. They investigate the effects on the correlation of using publicly available 1% sample (the Twitter public feed provides a 1% sample of all the public tweets) instead of using all the public tweets (Twitter firehose). They find that using the 1% sample does not substantially compromise ILI estimates on the national level, but might cause damaging effects on city-level estimates. Gosh et al. [56] propose to use tweets published by *experts* in data mining tasks, including event detection. They collect tweets from over 500,000 Twitter users identified as experts on a variety of topics, and compare them to a random sample of all the

public tweets along some dimensions, including information timelines, i.e, when an event occurs, to determine which sampling method gives the earliest information about the event. Choudhury et al. [52] propose an alternative sampling method with a goal of constructing a sample satisfying a desired level of diversity for a topic-centric search task. Both [56] and [52] provide alternative sampling methods to the traditional random sampling, and explore the effects of the proposed sampling method on a data mining task; however, neither of them quantify the effects of sampling on the detection accuracy of event detection. Different from all this previous literature, we focus on understanding the impact of different sampling levels on event detection accuracy.

4.2.4 Noise and Twitter event detection

Understanding the impact of noise on different data mining problems is an important research direction. Stafford and Yu [102] study the effects of spam on Twitter trending topics. They collect over 9 million tweets and use a Naive Bayes classifier to identify spam. They find that spammers do not drive Twitter trending topics, even though they target certain topics in their posts. Sedhai and Sun [98] study the impacts of spam on hashtag recommendation for hyperlinked tweets. They find that hashtag recommending approaches that are effective when analyzing non-spam tweets are less effective in the presence of spam tweets. Jones et al. [64] explore whether spam or irrelevant documents will compromise users' satisfaction with search engine result pages. They find that users prefer if a few spam documents are returned as opposed to irrelevant documents. They conclude that web document retrieval should consider both document relevance and document *spamminess*. Different from all the above mentioned works, we study the effects of spam/irrelevant tweets as well as domain spam on the task of event detection.

4.3 Definitions and Assumptions

Let $\mathbb{D} = \{d_i, d_2, ..., d_3\}$ denote a tweet stream, where d_i is the i^{th} tweet in the stream. Let t_i denote the time when a tweet is published. We are interested in detecting events E that occur in a specified time window. Let τ represent the set of time windows for tweets in $\mathbb{D}$. Each window contains a set of times, e.g. $\tau_1 = [t_0, t_j)$, where j is the j^{th} tweet in the first window, and $D(\tau_j)$ represents the tweets that occur in time window τ_j. A tweet is composed of a set of words, $W = w_i, w_2, ..., w_m$ and possibly a location l.

We make the following assumptions about tweets in the tweet stream $\mathbb{D}$:

1. Given its 140 character length limit, a single tweet maps to at most one event [1].

2. Not all the tweets in $\mathbb{D}$ contain a location. In these cases, we say that the tweet is not discussing a particular event.

3. When a tweet has a specified location l, this location tends to be relevant to the event reported in the tweet, most likely mapping to the location where the event occurs.

4. Two different events do not share the same set of keywords and the same location within the same time window τ_i.

Problem Statement: Given a tweet stream $\mathbb{D}$, the task of geotagged event detection is to identify events E from $\mathbb{D}$, during particular time periods τ. The representation of an event e_m is a three-element tuple $\{\Delta, \Phi, \Sigma\}_m$, in which Δ represents the

[1] While this is not always the case, we empirically find that it is a reasonable assumption that does not reduce the quality of the detected events.

location where the event occurs, Φ represents the time when the event occurs, and Σ represents the synopsis of the event.

4.4 Event Detection

Algorithm 4 presents a high level view of our proposed approach. The input to our approach is a tweet stream $\mathbb{D}$, the number of time windows in the training phase p, and a locality ontology $\mathbb{O}$. The output is a set of detected events for each time window $D(\tau_i)$ in the detecting phase, and each detected event e is in the form of a three-element tuple. Here tuple $\{\Delta, \Phi, \Sigma\}$ is represented by tuple $(l, \tau_i, S(l, \tau_i))$, in which l represents the location where the event occurs, τ_i represents the time window when the event occurs, and $S(l, \tau_i)$ represents the synopsis of the event.

Algorithm 4: High level algorithm for online geotagged event detection.

Input: *A tweet stream*: $\mathbb{D}$
The number of training windows: p
A location ontology: $\mathbb{O}$

Output: *Detected events*: E

```
/*******Training Phase*******/
for i ← 1 to p do
    {D(l, τ_i)} = location_identification(D(τ_i), 𝕆)
    for D(l, τ_i) ∈ {D(l, τ_i)} do
        w(l, τ_i) = calculate_term_frequency(D(l, τ_i))
/*******Detecting Phase*******/
for i ← p + 1 to n do
    {D(l, τ_i)} = location_identification(D(τ_i), 𝕆)
    for D(l, τ_i) ∈ {D(l, τ_i)} do
        w(l, τ_i) = calculate_term_frequency(D(l, τ_i))
        B(l, τ_i) = bursty_term_extraction(w(l, τ_i), {w(l, τ_i − p), ..., w(l, τ_i − 1)})
        G(l, τ_i) = create_geotagged_bursty_term_graph(D(l, τ_i), B(l, τ_i))
        e(l, τ_i) = extract_event(G(l, τ_i))
        S(l, τ_i) = generate_synopsis(e(l, τ_i), D(l, τ_i))
    sort({e(l, τ_i)})
```

Our approach begins by identifying the predominant location l for each tweet. Using open-source data (described in Section 4.5), our approach constructs an ontology, in which each node represents a location. Such a location could be a country, a governorate, or a city.

Tweets in $D(\tau_i)$ are divided into groups $\{D(l, \tau_i)\}$ according to their predominant locations (Line 3 and Line 8). Then the term frequency $w(l, \tau_i)$ of each term in each tweet group $\{D(l, \tau_i)\}$ are calculated, where $w(l, \tau_i)$ denotes the term frequency of term w in $D(l, \tau_i)$ (Line 5 and Line 10). A term is considered *bursty* in a detecting time window τ_i if its term frequency compared to the previous p windows is significantly different. We denote a bursty term at location l in time window τ_i as $b(l, \tau_i)$, and the set of bursty terms as $B(l, \tau_i) = \{b(l, \tau_i)\}$ (Line 11). Terms go viral on Twitter because people are using these terms to discuss occurring event(s). When discussing the same event, people tend to use the same set of keywords. When a new event occurs, people will tend to use a new set of words that differs from the set of words they used to discuss previous events. Therefore, we should be able to represent events reported in tweets with bursty terms if these bursty terms are appropriately grouped.

Then, a geotagged bursty term graph $G(l, \tau_i)$ is constructed using $b(l, \tau_i)$ (Line 12), in which a node is a bursty term $b(l, \tau_i)$ and an edge represents the co-occurrence of two bursty terms within a tweet. Since all the nodes in the graph are tagged with the same location, we refer to the graph as a *geotagged graph*. For each geotagged graph $G(l, \tau_i)$, an event $e(l, \tau_i)$ is extracted by identifying a semantically cohesive set of nodes (Line 13). In other words, we do not want the terms that have the highest frequency alone. Instead, we want the terms that are most frequent, but part of a connected component. Therefore, we begin by identifying the node/term with the highest frequency in the graph. We then select its neighbors in frequency order and iteratively continue the process until we have a set of k nodes. Once we have k nodes,

we stop. The subgraph containing the k identified nodes is considered a semantically cohesive group since it is frequently occurring and has relationships among the nodes.

Using the tweets in $D(l, \tau_i)$, the tweet most representative of the detected event $e(l, \tau_i)$ is selected as the event synopsis $S(l, \tau_i)$ (Line 14). To accomplish this, we identify the tweet that contains the highest overall term frequency of bursty terms associated with the detected event. Finally, events $\{e(l, \tau_i)\}$ generated from different geotagged graphs with time window τ_i are sorted (Line 15) since not all the detected events are of equal importance. We define Importance of an event e as

$$I(e) = \sum_i w(\tau_i),\ \tau_i \in S \tag{4.1}$$

where S denotes the synopsis of event e and $w(\tau_i)$ denotes the term frequency of a term in S. Our event importance considers both the event burstiness and the number of tweets supporting the event.

4.5 Evaluation of Geotagged Event Detection

In this section we empirically evaluate our event detection accuracy on three real world data sets.

4.5.1 Data Sets

For our empirical analysis, we consider three tweet streams in three distinct domains: terrorism, migration, and politics. Table 4.1 shows some statistics about them.

Terrorism: The Islamic State of Iraq and Syria (ISIS) is a Sunni jihadist group with a violent ideology. The group is responsible for terrorist attacks worldwide in recent years, emerging as a top security concern for the United States and many other countries [2]. We work with an interdisciplinary team of researchers, students,

[2] `http://law.emory.edu/eilr/content/volume-30/issue-2/comment/isis-largest-threat-world-peace.html`

and policymakers, some of whom have years of in-field research experience in the Middle East. With their help, we identified a set of hashtags that are related to ISIS. When collecting tweets using these hashtags with Twitter API, we find that #isis and #isil are the only two hashtags that each returns over 10 thousand tweets per day on average. This evaluation consists of tweets containing one of these two hashtags published between September 2014 and March 2016.

Migration: More than one million migrants crossed into Europe in 2015 [3], triggering a crisis which many European countries have been struggling to deal with. We worked with the Institute for the Study of International Migration (ISIM) at Georgetown University to identify a set of hashtags related to migration. Two popular ones are #flee and #refugees. In this evaluation, we use tweets containing one of these hashtags between September 2014 and March 2016.

Politics: Our research team collected a set of hashtags related to the 2016 US presidential election: #hillary, #trump, #votehillary, and #votetrump. This evaluation uses tweets containing any of these four hashtags from July 2016 to November 2016, which covers the most crucial part of the campaign season.

Noise is pervasive in tweets. An analysis of 2,000 English tweets originating from the United States conducted by a San Antonio-based market-research firm Pear Analytics shows only 4% of the 2,000 tweets are reporting real world events [4]. Therefore, a well designed preprocessing step is imperative for a data mining task on a tweet stream. Our preprocessing includes (1) filtering the tweet stream using the cybozu library [5] to remove any tweet considered non-English, (2) removing all the retweet signs (e.g., RT), all urls, and all Twitter handles, (3) tokenizing and stemming tweets,

[3] `https://en.wikipedia.org/wiki/European\_migrant\_crisis`

[4] `https://en.wikipedia.org/wiki/Twitter`

[5] `http://developer.cybozu.co.jp/archives/oss/2010/10/language-detect.html`

Table 4.1: Data sets.

	Hashtags	Start Date	End Data	#Tweets
Terrorism	#isis, #isil	2014-09-10	2016-03-10	15,835,184
Migration	#flee, #refugees	2014-09-10	2016-03-10	5,511,455
Politics	#hillary, #trump	2016-07-03	2016-09-13	18,368,438
	#votehillary, #votetrump	2016-09-27	2016-11-13	2,210,202

and (4) removing stopwords (the hashtags employed to collect tweets are also considered stopwords) and non-alphanumerical characters.

4.5.2 Location Identification

We build our location ontology using Wikipedia and Statoids [6]. Wikipedia has a set of pages listing all the major cities around the world by country, and Statiods lists governorates and the major cities in governorates and their populations for each country. Leveraging these two sources, we construct a three-level ontology, including countries, governorates, and cities. The raw ontology has 37,379 nodes with 55,816 locality names. After applying the population-based duplicate name removal, the ontology is left with 46,560 distinct locality names.

To evaluate our location identification approach, we obtain a random sample of 300 tweets from the tweets having predominant locations for each of the three data sets, and manually check whether the determined location of a tweet maps to the location where the event reported in the tweet actually occurs. We find that for some tweets the

[6] `http://www.statoids.com`

Table 4.2: Three example tweets whose determined predominant locations are relevant to the events reported in the tweets, but not the locations where the events occur.

	Tweet	Determined Locality	Actual Locality
Terrorism	raqqa the airstrikes today is by russian warplanes and they don t hit isis hq most of the places are civilian they destroy bridge	Russia	Syria
Migration	the three year old was killed as his family made a desperate bid to flee syria and mirror columnist carole malone says we must learn	Syria	Turkey
Politics	breaking donald trump jr just pledged 89 delegates from new york officially placing donaldtrump over 1237 as the gop nominee	New York	Ohio

determined predominant locations are the locations where the events reported in the tweets occur (Type A), whereas for some other tweets the determined predominant locations are relevant to the events reported in the tweets, but not the locations where the events occur (Type B). As an example, the first row in Table 4.2 gives a tweet reporting Russian airstrikes in Syria on Nov 3, 2015. The determined location of the tweet is Russia, but the airstrikes actually occurred in Syria. We consider this a relevant location, but it is not the predominant one. This type of location labeling is a mistake in certain contexts, but reasonable in other contexts. Therefore, we keep track of how often this occurs.

If we consider both types of answers as correct answers, our location identification approach achieves accuracies of 97.0% and 94.7% for the terrorism data set and migration data set, respectively (shown in Figure 4.1). The accuracy is 79.7% for

Table 4.3: The proportion of tweets with predominant locations for different data sets and the number of ground truth (GT) events.

Data Set	#English Tweets	Tweets W/ Locations	Proportion	GT Events: 30 Day Wondow
Terrorism	11,637,327	4,444,508	38.1%	100
Migration	4,797,956	1,524,515	31.7%	98
Politics	19,253,417	920,038	4.7%	107

the politics data set. This lower accuracy occurs because some names of politicians also map to location names, e.g. Lincoln. Table 4.3 lists the number of English tweets and the number of English tweets determined as with predominant locations for the three datasets. We can see for the Terrorism data set and the Migration data set, the proportion of tweets with predominant locations is over 30%; however, the proportion of tweets with predominant locations is only 5% for the Politics data set. This is not surprising since many of the tweets are not about specific events, but are opinions about candidates.

4.5.3 Geotagged Graph Generation

Since an exhaustive and authoritative list of ground truth events is not available, our team manually created a list of ground truth events. We focused in on a 30 day consecutive stream for each of the three data sets. More specifically, we choose Nov 1, 2015 to Nov 30, 2015 for the terrorism data set, September 1, 2015 to September 30, 2015 for the migration data set, and July 13, 2016 to August 12, 2016 for the politics data set. These time periods were chosen because of their comparatively higher volume of data during these time periods (refer to Figure 4.2).

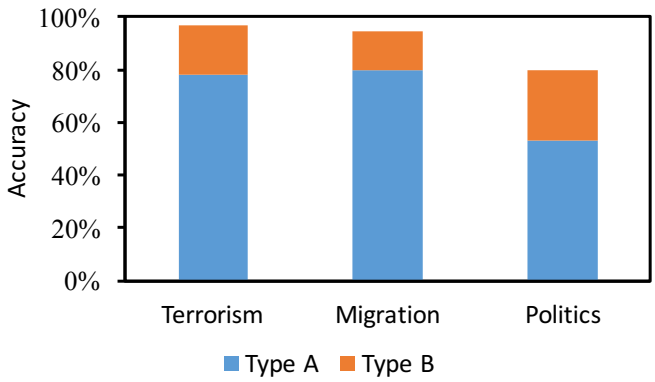
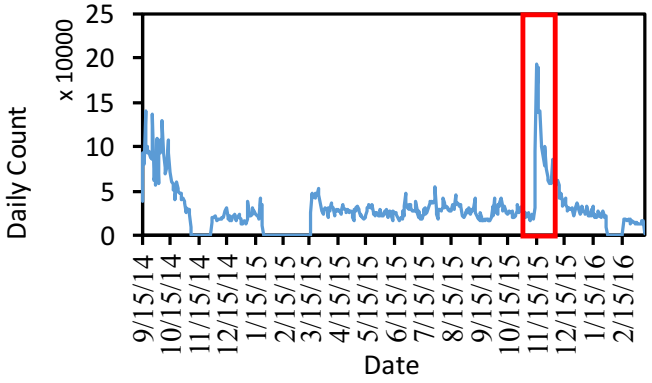
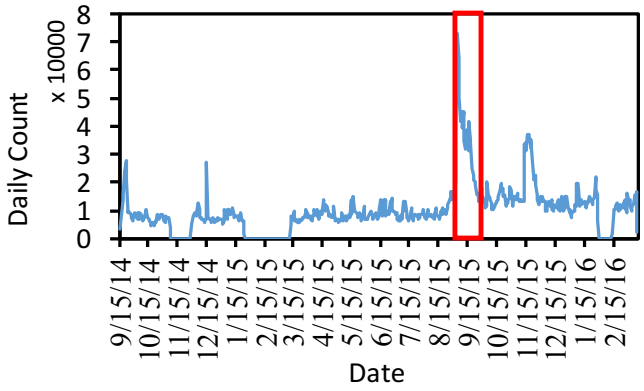


Figure 4.1: Location identification accuracy. Type A: the determined locations are the locations where the events reported in the tweets occur; Type B: the determined locations are relevant to the events reported in the tweets, but not the locations where the events occur.

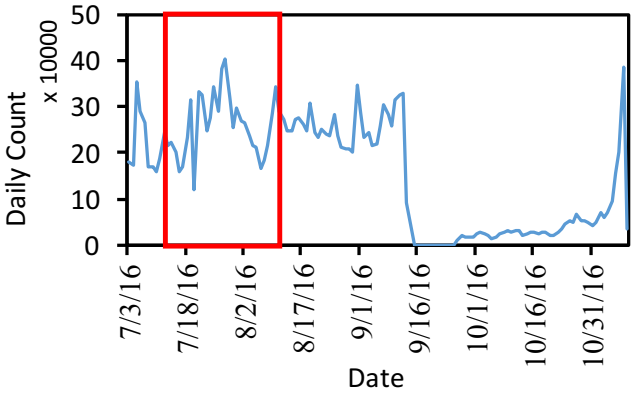
Applying location identification and geotagged graph generation to a time window of a tweet stream yields a list of geotagged graphs. In the rest of this evaluation, we only consider the top 5 geotagged graphs generated for each time window, which correspond to 150 geotagged graphs throughout the 30 day tweet stream. Note that for the terrorism data set and the migration data set, we generate geotagged graphs on the country level. For the politics data set, we generate geotagged graphs on the state level. Figure 4.3 shows the geotagged graph distribution over countries or states. For the terrorism data set, the top 2 countries are Syria and Russia. It confirms our intuition: Syria is the country where most of ISIS's territory is, and Russia is deeply involved in the conflict. For the migration dataset, the top 2 countries are Syria and Germany: Syria is the country where most of the refugees left from to go to Europe, and Germany has accepted a large number of refugees. With respect to the politics data set, the top states include DC, New York, Ohio, Texas, Florida, Colorado, etc. DC is the US capital; New York is the home state of both candidates; Texas has the



(a) Terrorism.



(b) Migration.



(c) Politics.

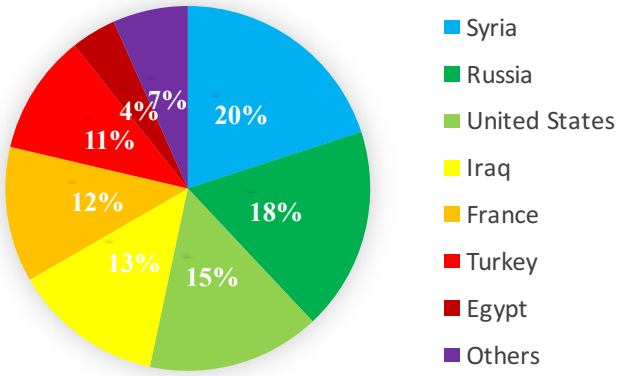
Figure 4.2: Daily tweet count of the three datasets. (Red boxes confine the time periods chosen for detecting events; There are three gaps for the terrorism dataset and the migration dataset, and one gap for the politics dataset in our data collection due to infrastructure unavailability.)

second largest number of electors in the electoral college; Ohio, Florida, and Colorado are three key swing states, where both candidates had a number of events.

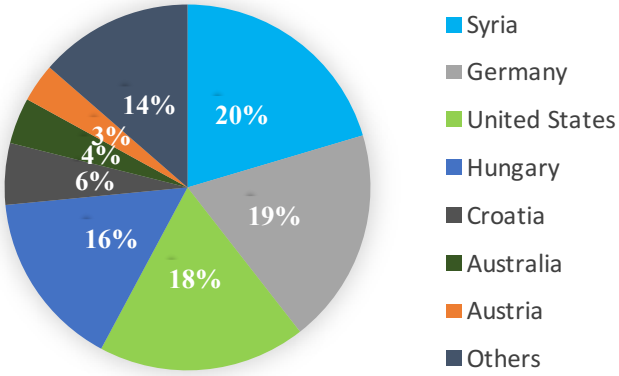
4.5.4 Geotagged Event Extraction

In detecting events using our geotagged event extraction component, we set the length of the time window to a day. We set the number of time windows in the training phase p to 10, and the number of top terms representing a detected event k to 10. These numbers were chosen based on an empirical sensitivity analysis. Note that the 10 training windows are not included in the selected 30 day tweet stream. They correspond to 10 days prior to the 30 day detecting phase. We evaluate the accuracy of the events detected by our approach, and compare our approach (GeoGraph) to four state-of-art event approaches described in Section 4.2: [80] (KCore), [41] (Emerge), [115] (Wavelet), and [121] (GeoBurst). For each of the five approaches, we consider the top 5 events (if available) detected for each day, and manually check whether each of them maps to any real world event by exploring Wikipedia event pages and using their key terms as search queries for Google search. In this way, we build a list of ground truth events reported in the tweet stream. Then we manually check whether a detected event maps to any event on the ground truth event list. Table 4.3 shows the number of events labeled as ground truth events for the three data sets.

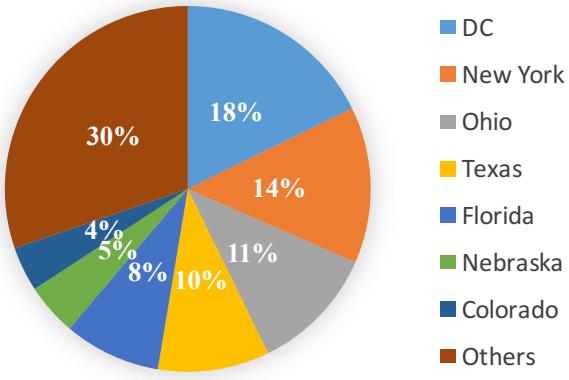
Figure 4.4 shows the precision and recall of our proposed approach and the three baseline approaches for event detection. We can see that our approach achieves over 55% precision and over 65% recall on all the three data sets, significantly outperforming the baselines in terms of both precision and recall. The success of our approach can be attributed to two factors: (1) it provides a tweet as the synopsis for a detected event, which greatly facilitates mapping it to a ground truth event. In contrast, both the Emerge approach and the Wavelet approach provide a set of terms as the synopsis



(a) Terrorism.



(b) Migration.



(c) Politics.

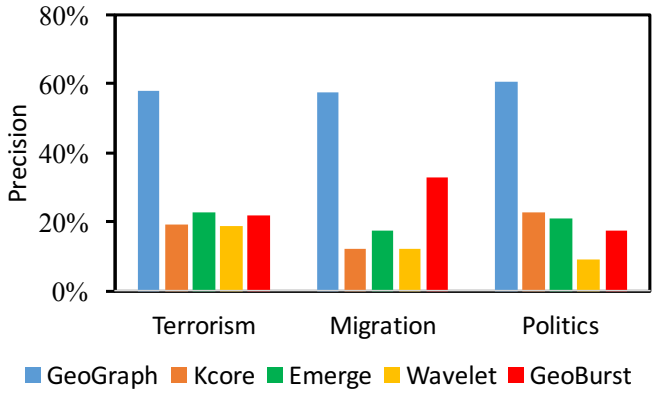
Figure 4.3: Geotagged graphs distribution over countries and states.

of a detected event, which is much less interpretable; (2) our approach detects events only based on the tweets with predominant locations; This helps filter Twitter memes since memes usually do not contain location information. When looking at the results produced by the state of the art methods, we observe that most of the events detected by the KCore approach are Twitter memes. The GeoBurst approach's performance is not as good as expected, because it detects tweets based on tweets with longitudes and latitudes; however, only a very small portion of tweets are encoded with longitudes and latitudes. Take the Terrorism data sets as an example, out of the 15,835,184 tweets, only 13,031 tweets have longitudes and latitudes. Finally, we pause to mention that there could be events that none of the methods detected that we are missing. This is a limitation of our evaluation approach resulting from evaluation data sets not being available.

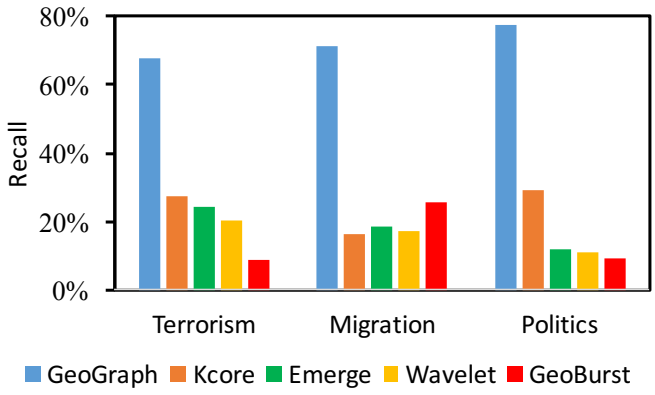
Since our event detection approach also specifies a location for each detected event, we are interested in understanding the how the location maps to the ground truth locations of the events: what is the percentage of the determined locations that map to where the events actually occur (Type A), and what is the the percentage of the determined locations that are relevant to the detected events, but not the locations where the events occur (Type B). The results are shown in Figure 4.5. We can see that by combining both types of answers together, our approach achieves over 90% accuracy for all the three data sets. Again, the terrorism and migration location accuracy are higher than the location accuracy for the politics data set.

4.5.5 Event Synopsis

We now show the event synopses generated by different approaches. Table 4.4 provides the synopses of the top 3 events (if available) detected by different approaches on a single day for the terrorism data set. Table 4.5 and Table 4.6 provide the synopses of



(a) Precision.



(b) Recall.

Figure 4.4: Event detection accuracy of different approaches.

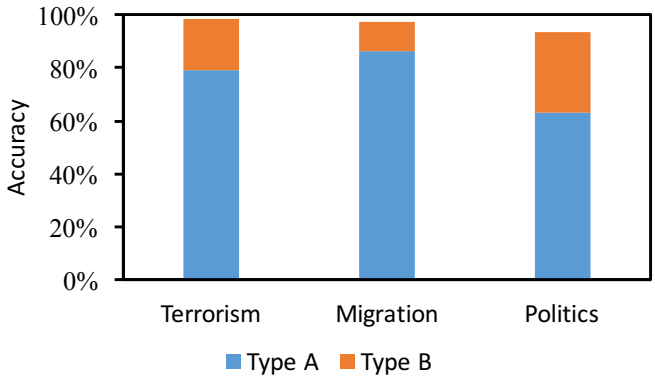


Figure 4.5: Event location accuracy. Type A: the determined locations are the locations where the events occur; Type B: the determined locations are relevant to the events, but not the locations where the events occur.

Table 4.4: Synopses of events detected by different approaches on 2015-11-13 for the terrorism dataset.

	No.	Map GT	Event Synopsis
GT	1	GT 1	ISIS carries out a series of coordinated attacks in Paris, killing 130 people.
	2	GT 2	Pentagon confirms Jihadi John was killed in an airstrike carried out by US.
	3	GT 3	Suicide bomb and road blast kill 26 in Baghdad.
GeoGraph	1	GT 1	FRANCE: good job boys thanks for paris attack france paris attack parisattack jihadist mujahedeen islam terror alqaeda isis is isil
	2	GT 2	SYRIA: us military says reasonably certain that raqqa airstrike killed daesh terrorist jihadijohn reuters is isis syria iraq
	3	GT 3	IRAQ: 150 killed wounded in a triple isis suicide attacks in east baghdad in sadr in one day iraq
KCore	1	GT 1	good job boys thanks for paris attack france paris attack parisattack jihadist mujahedeen islam terror alqaeda isis is isil
	2	GT 1	fuck terrorism fuck isis fuck al queda fuck u and fuck everything you fucking stand for you fucking fuckwit fuckers isis parisattacks
	3	GT 1	parisattacks is a wake up call for every single country in the world except the countries that support isis parisattacks parisshooting
Emerge	1	GT 1	isis, the, to, of, in, ccot, rt, is, on, paris
Wavelet	1	None	cold, usual, carlyfiorina, nuclear, journalneo, experi, thr, constitut, weather, ecentauri
	2	None	ilnewsflash, billion, tag, price, oct, record, victim, star, hunt, shock
	3	None	destruct, true, communiti, thereaperteam, hellfir, oust, journal, boom, worth, european
GeoBurst	1	GT 1	#daesh #islam #isis #mahoma (@ gran mosquee in paris, france) https:// t.co/dswhz8ljv0 https://t.co/n3hdk4trsz
	2	None	no #isis is no jv team. frau #merkel needs to resign for putting europe in danger, with her openness at the border. we are at war! #eusucks
	3	None	#isis isis is not islam isis is not islam isis is not islam isis is not islam

Table 4.5: Synopses of events detected by our approach on 2015-09-05 for the migration dataset.

	No.	Map GT	Event Synopsis
GT	1	GT 1	Alan Kurdi, a three-years-old Syrian child, drown in the Mediterranean Sea.
	2	GT 2	Refugees are stranded at Budapest train station.
	3	GT 3	Refugees applauded in Germany by locals.
GeoGraph	1	GT 1	SYRIA: the three year old was killed as his family made a desperate bid to flee syria mirror columnist carole malone says we must learn
	2	GT 2	HUNGRAY: hungarian families arrive at the railway station with aid for refugees a new wave waiting to leave for their destination budapest
	3	GT 3	GERMANY: check out germany being super cool as ever applauding refugees arriving in munich way to go deutschland

Table 4.6: Synopses of events detected by our approach on 2016-08-11 for the politics dataset.

	No.	Map GT	Event Synopsis
GT	1	GT 1	Hillary Clinton makes a speech in Detroit.
	2	GT 2	Donald Trump holds rally in Kissimmee, Florida.
	3	GT 3	Utah Governor Gary Herbert says he'll be voting for Donald Trump.
GeoGraph	1	GT 1	MICHIGAN: hillary you bill and obama destroyed michigan economy now you have the nerve to speak about creat jobs in detroit
	2	GT 2	FLORIDA: and whichever florida pastors applauded to trump claim about obama founding isis should resign immediately
	3	GT 3	UTAH: utah govgaryherbert is voting for trump so i must not vote for herbert in the fall anyonebuttrump anyonebutherbert

the top 3 events detected by our approach on a single day for the migration data set and the politics data set (events detected by baseline approaches are not shown given the space limitation). In these tables, the GT rows list the corresponding ground truth events, and the *Map* GT column shows which ground truth event a detected event maps to. We can see that the event synopses generated by our approach, the KCore, and the GeoBurst approach are much more informative than the Emerge approach and the Wavelet approach, since both our approach, the KCore approach, and the GeoBurst approach give a tweet as event synopsis, whereas the Emerge approach and the Wavelet approach give a set of terms as event synopsis. On the other hand, our approach achieves a much higher event detection accuracy than the KCore approach and the GeoBurst approach, which can be seen in Table 4.4, Table 4.5, and Table 4.6. Combining the event accuracy and event synopsis informativeness, our approach clearly outperforms all the four baseline approaches.

4.5.6 Case Study: Geotagged vs. Non-Geotagged

As stated previously, one of the reasons our approach performs better than the baseline approaches is that it leverages geography information during graph construction. Although three out of the four baseline approaches do not consider geography information in detecting events, we hypothesize that leveraging geography information could also improve their performance. For this analysis, we investigate the impact of incorporating location information into the baseline approaches. More specifically, we divide a tweet stream $\mathbb{D}$ into multiple tweet streams $\{\mathbb{D}(l)\}$ by using the predominant location of each tweet in $\mathbb{D}$. These geotagged tweet streams are then fed into the KCore approach, the Emerging approach, and the Wavelet approach. With respect to the GeoBurst approach, we retrieve the longitude and latitude of the predominant

location of each tweet, and feed the tweets with predominant locations along with the tweets encoded with longitudes and latitudes into the GeoBurst approach.

Similar to Section 4.5.4, we select the top 5 events per time window for each of the four geotagged baseline approaches and our proposed approach, and manually check whether a detected event maps to any real world event. We apply the four geotagged baseline approaches and our proposed approach to the terrorism data set. Figure 4.6 shows the precision and recall of the four approaches. Combining Figure 4.4 (event detection accuracy of non-geotagged baselines) and Figure 4.6 (event detection accuracy of geotagged baselines), we can see that leveraging the location information improves the detection precision by approximately 20% and the recall by approximately 10% for the KCore approach and the Emerge approach; with respect to the Wavelet approach, the improvement is not significant, but still observable. We conclude that including geography information is beneficial to event detection in general. On the other hand, we can see that our proposed approach still outperforms the geotagged baseline approaches. It suggests that other components of our approach, including bursty term extraction, geotagged bursty term graph generation, and event extraction, are important for detecting events accurately.

4.6 Effects of Sampling on Event Detection

The question we want to answer is the following - what fraction of the tweet stream is needed to identify the events we identified in the previous subsections? This is an important question because it is sometimes difficult to get the entire tweet stream and we need to understand the impact of only getting a fraction of it.

For this analysis, we conduct the analysis on the GeoGraph and the GeoBurst approaches for event detection because they are the two best performing ones among

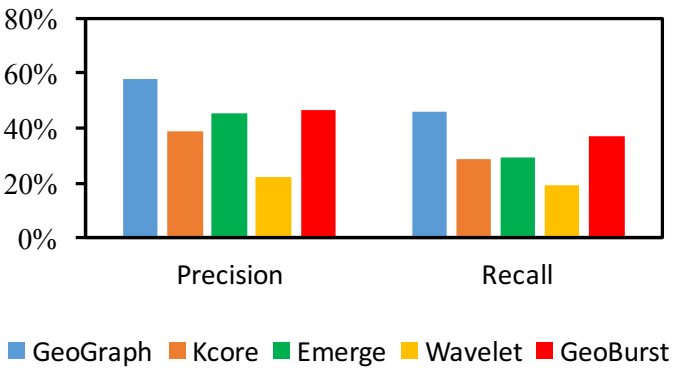


Figure 4.6: Event detection accuracy of the geotagged baseline approaches and our approach on the terrorism dataset.

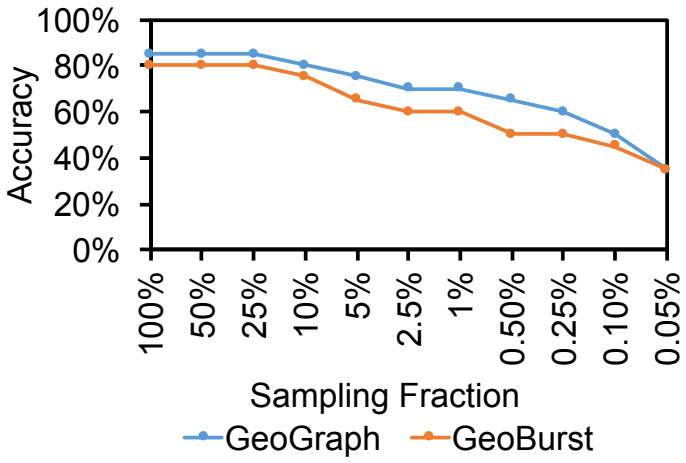
the approaches evaluated in Section 4.5. We sample a tweet stream with a specified sampling fraction using the sampling with replacement strategy. Different from the previous section which compares different event detection approaches on a 30 day consecutive tweet stream, this section uses samples of the whole twitter stream (refer to Table 4.1 for specific stream length). Since no exhaustive and authoritative list of ground truth events is publicly available, we collect 20 major ground truth events on the terrorism topic, the migration topic, and the politics topic respectively during the time period covered by the tweet stream. Note that these 20 ground truth events are collected based on their real world impacts and their publicity in the blogsphere; however, they are not necessarily the biggest events in the tweet stream. Table 4.7 lists the 20 major ground truth events on the terrorism topic. As shown in Table 4.7, most days have one ground truth event, and a few days have two ground truth events. For each day, we select the top one or two detected events, depending on how many ground truth events have been identified for that day.

Table 4.7: Twenty major ground truth events on the topic of terrorism from 2014-09-15 to 2016-03-10.

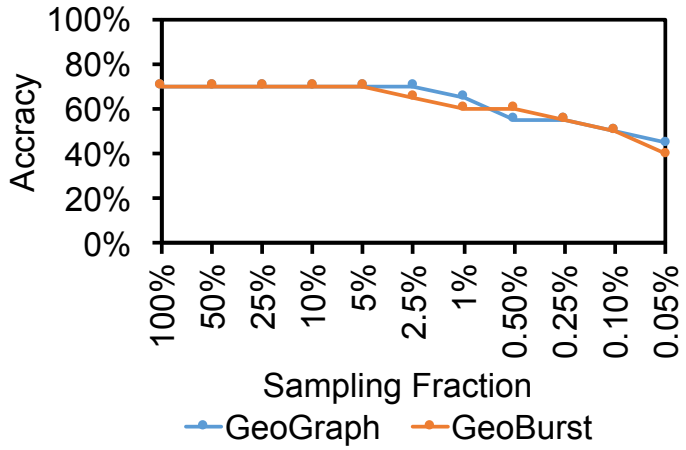
No.	Date	Event
1	2014-09-19	ISIS advances on the Syrian border town of Kobani and thousands flee into Turkey.
2	2014-09-22	France and other countries join US-led coalition to destroy ISIS.
3	2014-09-23	The United States launches its first air strikes against ISIS in Syria.
4	2014-10-03	ISIS releases a video shows the beheading of British aid worker Alan Henning.
5	2014-10-22	A series of shootings occurs at Parliament Hill in Ottawa.
6	2014-12-16	Taliban terrorists enter school in Peshawar, Pakistan, and hold at least 500 kids hostage.
7	2015-01-07	Two gunmen attack the offices of newspaper Charlie Hebdo in Paris, killing 11 people.
8	2015-03-20	ISIS claims responsibility for an attack in Tunis, 22 people killed.
9	2015-03-20	ISIS-linked militants bomb two mosques in Sanaa, Yemen, killing 137 people.
10	2015-05-17	ISIS takes over Ramadi, Iraq.
11	2015-05-20	ISIS seizes the ancient Syrian city of Palmyra.
12	2015-07-01	ISIS fighters carry out assaults in Egypt's Sinai Peninsula.
13	2015-09-30	Russia begins airstrikes in Syria.
14	2015-10-10	30 civilians are killed by the blast in Ankara Peace Rally.
15	2015-10-31	ISIS claims responsibility for bombing a Russian passenger plane over the Sinai Peninsula.
16	2015-11-12	ISIS claims responsibility for suicide attacks in Beirut that kills 40 people.
17	2015-11-12	Kurdish forces seize Sinjar, Iraq from ISIS.
18	2015-11-13	ISIS carries out a series of coordinated attacks in Paris, killing 130 people.
19	2015-11-15	France ramps up its airstrikes on ISIS targets in Raqqa, Syria.
20	2016-01-14	ISIS claims responsibility for an attack in Jakarta, Indonesia.

In evaluating the detected events, we use detection accuracy, which is the percentage of the ground truth events captured by the 20 detected events. Figure 4.7 shows the detection accuracies of our approach and the GeoBurst approach at different sampling fractions. We have the following three observations: (1) in general the detection accuracy decreases as the sample fraction decreases; (2) the detection accuracy does not decrease rapidly when the sample fraction is above 1%. Let's consider the Terrorism data set as an example. Our approach's detection accuracy drops by only 15% from using the full dataset to using the 1% sample. With respect to the GeoBurst approach, the accuracy drops by 20% in the same range; (3) the detection accuracy decreases much more rapidly when the sample fraction is below 1%. On the Terrorism data set, the detection accuracy of our approach drops by 35% from using the 1% sample to using the 0.05% sample, and the accuracy of the GeoBurst approach drops by 25% in the same range. We conclude that sampling impacts the accuracy of event detection algorithms more than some of the other data mining tasks previously studied. However, if the sample fraction is high enough, reasonable approximate results can be obtained.

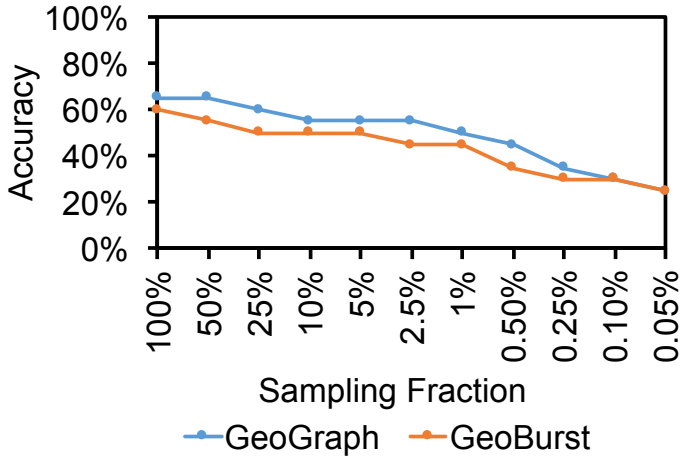
We notice that neither of two event detection approaches achieves 100% accuracy even using the full data set. Further analysis shows that it is due to two reasons: (1) on some days, the selected ground truth events are not the biggest events in the tweet stream; (2) location information is so irrelevant for some ground truth events that tweets reporting these events seldom refer to any location. Take the Politics data set as an example, the 20 ground truth events include the presidential debates and the vice presidential debate; however, people hardly pay any attention to where the debates take place, and thus do not include any location information in their tweets when tweeting about the debates. Detecting events whose locations are seldom mentioned is an important direction for future work.



(a) Terrorism.



(b) Migration.



(c) Politics.

Figure 4.7: Event detection accuracies of our approach (GeoGraph) and the GeoBurst approach at different sampling ratios.

4.7 Effects of Noises on Event Detection

As stated in Section 4.5, noise is pervasive in tweets. In this section, we explore the effects of different types of noise on Twitter event detection methods. In the context of event detection, noise tweets can be categorized into two groups: spam tweets, tweets that have nothing to do with the domain (e.g. ads for iPhones) and tweets that are domain relevant, but are irrelevant to the target events (e.g. tweets generated by political bots). Because of the differences in their nature and the differences in their effects on event detection, which will be shown later, we investigate their impacts on event detection separately. Similar to Section 4.6, we use the GeoGraph approach and the GeoBurst approaches as the benchmark approaches.

4.7.1 Effects of Irrelevant Spam Tweets

Because Twitter does not allow any third party entity to redistribute the contents of tweets as part of a corpus unless explicitly permitted [7], there is no spam tweet data set publicly available online. Therefore, our first step was to collect spam tweets. A lot of approaches for detecting spam posts on social media have been proposed [103] [32] [78]. Among them, we select a state-of-art, highly cited approach [55] and leverage it to detect spam tweets in each of our three tweet data sets.

For each of the 20 major ground truth events, we manually collect 100 to 500 tweets explicitly covering that event, and use these tweets as the pool from which we sample to obtain a random sample of 50 tweets. We use a sampling with replacement strategy. These tweets are considered the event signal. Then we place all the spam tweets into a pool from which we randomly sample from to reach the level of noise we want to evaluate. These tweets are considered noise. Combining the signal tweets

[7] `https://dev.twitter.com/overview/terms/agreement-and-policy`

Figure 4.8: The workflow for evaluating the effects of noise tweets on event detection.

and the noise tweets yields a noisy tweet stream with a specific Signal to Noise Ratio (SNR). For the Terrorism data set and the Migration data set, we generate noisy tweet streams with the SNR ranging from 1 to 0.02. With respect to the Politics data set, we generate noisy tweet streams with the SNR ranging from 0.1 to 0.002 for the following reason. Both the GeoGraph approach and the GeoBurst approach only take the tweets with geography information into consideration; however, the proportions of tweets with predominant locations vary significantly across different data sets (see Table 4.3). With respect to the signal tweets, the proportion of tweets with predominant locations is around 60% for all the three data sets. Decreasing the SNR to the range of 0.01 to 0.002 for the Politics data set makes its generated noisy tweet streams have approximately same volume of noises as the generated noisy tweet streams for the Terrorism data set and the Migration data set. Figure 4.8 shows the workflow for evaluating the effects of noise tweets on event detection accuracy.

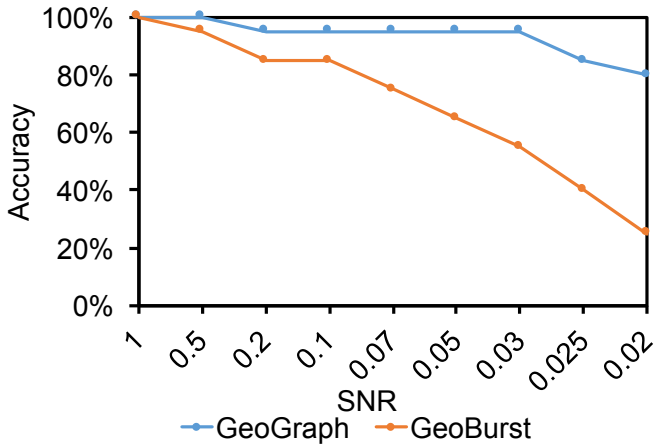
Similar to Section 4.6, for each day we select the top one or two detected events, depending on how many ground truth events are associated with the particular day. Figure 4.9 shows the detection accuracy of the two approaches on noisy tweet streams

generated at different SNRs (all the experiments are repeated three times - the average results are shown). We can see that the detection accuracy remains high when the SNR is above 0.1 (0.01 for the Politics data set, the same below), but deteriorates more rapidly when the SNR is below 0.1. On the other hand, the accuracy drop of the GeoBurst approach is much more significant than that of our approach. Take the Terrorism data set as an example, our approach's detection accuracy drops by 5% when the SNR is 0.2 and by 20% when the SNR is 0.02, while the GeoGraph approach's accuracy drops by 15% when the SNR is 0.2 and by 75% when the SNR is 0.02. We attribute the strength of our approach to two factors: (1) our bursty term extraction component is effective in preventing signals to be overwhelmed by noise; (2) our event ranking scheme considers both the event burstiness and the number of tweets supporting the event. We conclude that while the event detection accuracy will decrease as the SNR decreases, an effective event detection approach can significantly reduce the deterioration of the event detection accuracy.

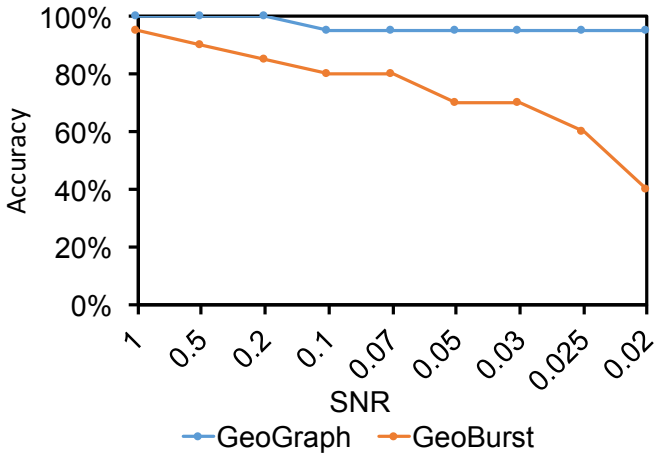
4.7.2 Effects of Domain Specific Noise

We define domain specific noise as tweets that are irrelevant to the target events but are not traditional spam. In other words, they might be Twitter memes, or even report real world events, but their contents are irrelevant to the events we aim to detect. They may be generated by more sophisticated bots or be opinions being shared by people. Therefore, for this experiment, we need to identify tweets that are domain relevant, but not the spam or the event signal identified in the previous subsection.

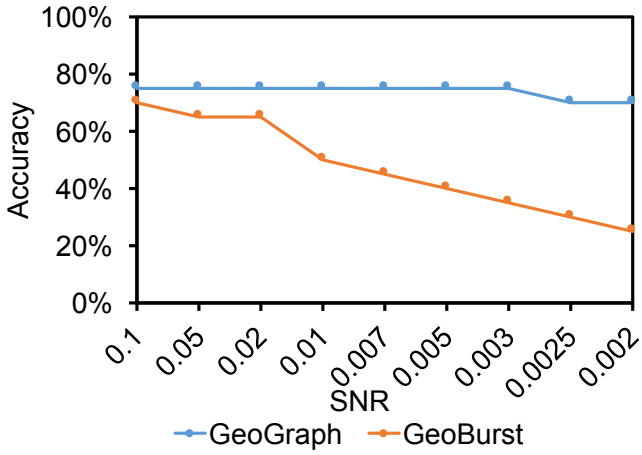
We use the same set of signal tweets as the previous section. We then use a pool of other domain tweets from which we randomly sample a subset. This sample of tweets is considered noise. Combining the signal tweets and the noise tweets yields a noisy tweet stream with a specific SNR. Similar to the previous subsection, for the



(a) Terrorism.



(b) Migration.



(c) Politics.

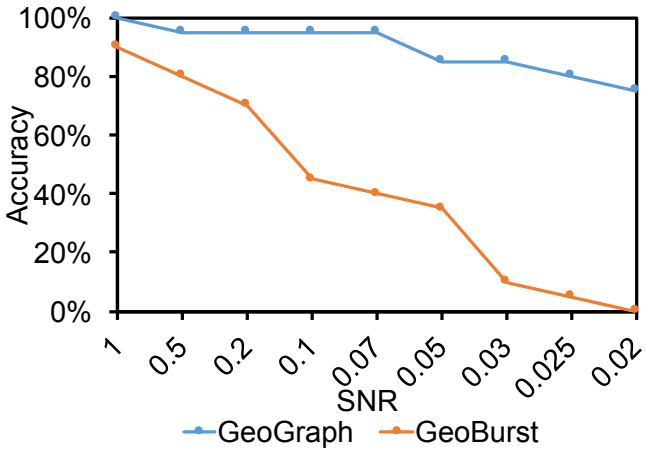
Figure 4.9: Detection accuracy on tweet streams with different fractions of spam.

Terrorism data set and the Migration data set, we generate noisy tweet streams with the SNR ranging from 1 to 0.02 and for the Politics data set, we generate noisy tweet stream with the SNR ranging from 0.1 to 0.002.

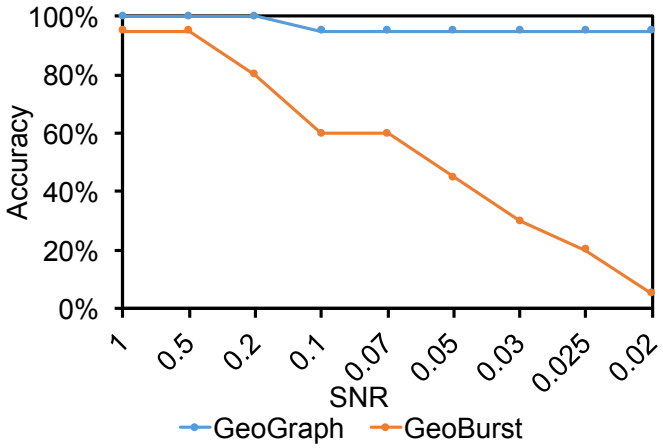
Figure 4.10 shows the detection accuracy of the GeoGraph and GeoBurst approaches on noisy tweet streams generated at different SNRs. We make the following observations: (1) the decrease in detection accuracy is comparably limited when the SNR is above 0.1, but more significant when the SNR is below 0.1; (2) the accuracy drop of the GeoBurst approach is much more significant than that of the GeoGraph approach; (3) the deterioration of the detection accuracy is more significant when adding domain specific noise tweets compared to adding spam noise tweets. This is not surprising since these domain specific noise tweets are likely to contain some signal, even though it is not relevant to the target signal. In contrast, the spam tweets are pure *white* noise, which makes the burstiness of target signals more obvious.

4.8 Conclusions

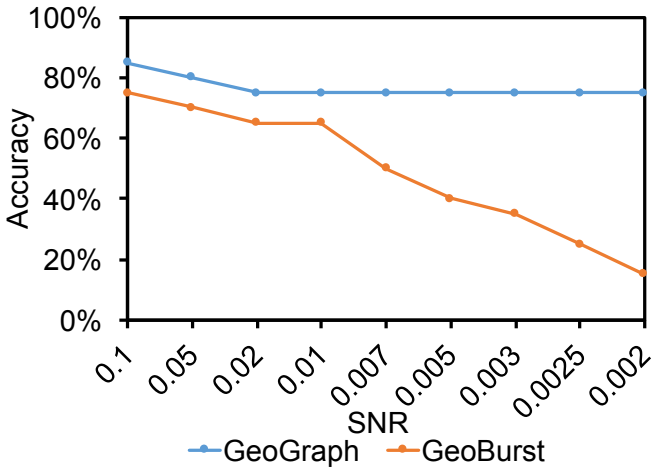
In this chapter, we propose an approach for detecting events from a tweet stream that leverages geotagged bursty term graphs as the main data structure for extracting events. An empirical evaluation on three large real-world data sets shows that our approach and our data structure significantly improves the event detection precision and recall when compared to the state of the art. We also find that events are easier to detect because of our simple approach to labeling them using tweets containing bursty words. Finally, given the noisy, partial nature of Twitter, we conduct a sensitivity analysis that evaluates our approach using different sample sizes, and different levels



(a) Terrorism.



(b) Migration.



(c) Politics.

Figure 4.10: Detection accuracy on tweet streams with different fractions of irrelevant tweets.

of both spam and domain specific noise. We find that our approach is more robust to these scenarios than the state of the art.

Chapter 5

Detecting Extremist Users on Twitter

Terrorist organizations, especially Islamic State of Iraq and Syria (ISIS), have been leveraging Twitter to encourage supporters to initiate terrorist attacks worldwide [75]. Their activities have also led to large-scale displacement in the Middle East. In order to better understand the relation between extremist discussions and migration, we need to first begin by identifying users who engage in extremist discussions on Twitter. Once we have that information, we can track the changing dynamics of the conversations to better understand the relationship between extremist conversation online and potential movement.

Identifying extremist-associated conversations and users engaging in extremist discussions on social media sites and blog forums is still an open problem. In Section 5.1, we analyze potential features that may be useful for identifying ISIS supporters on Twitter, group these features into categories, and present a case study looking at the ISIS extremist group. The second part of the chapter (Section 5.2) proposes an approach for identifying users engaging in extremist discussions. Our approach uses detailed feature selection to identify relevant posts and then uses a novel weighted network that models the information flow between the publishers of the relevant posts.

5.1 Feature Construction for Detecting Users Sharing Extremist Content

We begin this section by analyzing potential features on a small amount of manually labeled data about ISIS supporters on Twitter, and grouping these features into categories related to tweet content, viewpoints, and dynamics. After discussing different state of the art methods for extremism detection and similar problems, we present a case study looking at the ISIS extremist group. Finally, we discuss how one collects these data for a surveillance system and conclude by discussing some current challenges and future directions for effective surveillance of extremism.

5.1.1 Introduction

As Twitter, Facebook, and other social media sites continues to grow in popularity, conversations involving extremism are being recognized as a serious problem. One well known extremist group using Twitter as a platform for sharing its ideas and recruiting members/jihadists to its group is the Islamic State of Iraq and Syria (ISIS), also known as IS or ISIL. ISIS is itself responsible for mass atrocities in Iraq and Syria, and the group uses Twitter to encourage supporters to initiate terrorist attacks worldwide [75]. Its activities have also led to large-scale displacement in the Middle East. At of the end of 2015, almost 6 million persons were internally displaced in Syria and another 4 million were refugees in neighboring countries. ISIS related attacks in Iraq had displaced approximately 2.6 million persons in that country [15, 113].

To reduce extremist conversation by individuals and terrorist groups, Twitter has been suspending accounts which are believed to be associated with terrorist organizations. In February 2016, Twitter announced that it had shut down 125,000 accounts related to ISIS between the middle of 2015 to the beginning of 2016 [3]. The removed

ISIS-related accounts include ISIS-related media outlets, information hubs, and supporters. For obvious reasons, Twitter has never released its algorithm or strategy for determining whether an account is primarily related to ISIS or not [33]. In this work, we consider a generalization of this problem - determining whether a social media account should be classified as exhibiting extremist behavior, e.g. sharing extremist content, or not. We will refer to this problem as *extremism detection*. More formally, given a set of social media publishers U, our goal is to identify the subset of publishers U^+ that exhibit extremist behavior by promoting or disseminating extremist content.

This section is organized as follows. Section 5.1.5 presents as simple analysis of tweets containing extremist content and accounts exhibiting extremist behavior. Given this content, Section 5.1.3 considers different features that may be useful for identifying these accounts exhibiting extremist behavior on Twitter. In Section 5.1.4 we overview recent results and state of the art systems. Section 5.1.5 presents a case study focusing on detecting extremism related to ISIS on Twitter. Finally, Section 5.1.7 presents suggestions for data collection to gather these relevant data set for surveillance.

5.1.2 Twitter Data Analysis

In this section we present a small case study describing extremist related data on Twitter. We discuss both tweet content and extremist accounts.

We begin by looking at some examples of extremist content. Table 5.1 shows examples of some tweets that were classified as supporting an extreme viewpoint. We see that these tweets are retweeted by a small fraction of individuals who exhibit extremist behavior compared to the overall number of individuals who retweet them. When designing algorithms that attempt to identify extremism, we need to consider both content and content propagation. What are the distinguishing words? What are

Table 5.1: Examples of extremist tweets and their retweet propagation.

Tweet	Extremist Retweets	Overall Retweets
These 2 pics will explain to you the meaning of the whole world,1:Jihad =Honor 2:Democracy=Humiliation	3	162
This is how #ISIS supporters in Twitter win against western world :) :) http://t.co/...	3	154
See the difference between orphanages in #ISIS and other countries..thats why we love #IslamicState ...	2	34

the frequent ones? Words like *honor*, *win*, and *love* are not unusual and will be used in many positive contexts as well. To further complicate the situation, extremist content will be propagated by both extremists and non-extremists, so just following the flow of messages is insufficient.

To better understand content on Twitter, we manually identify approximately 1300 individual tweets that exhibit extremist views consistent with views of the ISIS extremist group. Figure 5.1 shows the frequency distribution of the different words as a percentage of the total number of tweets. As expected, the predominant words in these tweets are the hashtags of the extremist group itself, i.e. synonyms for ISIS [`#isis` (97%), `#islamicstate` (38%), `#is` (24%) and `#isil` (3%)]. The next group of popular words are related to religion, specifically Islam [`Muslim` (66%), `Islamic` (59%), and `#islam` (37%)]. There are general words like `world`, `state`, `people`, `never`, that are in 35% to 60% of the tweets. Another set of words are related to locations in Iraq and Syria, as well as general geography words, e.g. `land` also appear regularly in these tweets. `Iraq` (12%) and `Syria` (11%) are the most frequent. Interestingly, the `United States` occurs in 5% of these tweets. Finally, there are a small group of

extremist words that occur in approximately 3% of the tweets [`Abu` (the name of an extremist), `Mujahideen` (a person engaging in Jihad), and `#caliphate` (an Islamic state led by a religious leader who is considered a successor to the Islamic prophet Muhammad.)] It is interesting to note that there are very few "extreme" words in this set of tweets. This reminds us that searching for more unique words that are associated with extremist thought would miss a large number of tweets containing extremist content. The bag of words model will help identify some tweets that contain extremist views, but will also miss a large number of tweets because similar vocabulary is used in different contexts. For this reason, other features like sentiment/tone, and stance need to be considered. These features begin to get at the opinion of the post's author.

Finally, we look at five Twitter accounts that exhibit extremist behaviors. Some consistent properties exist across these accounts. First, all of these users tweet a fair amount, over 4500 tweets per year on average. Second, while they have followers, the median number of followers is 942 and the high is only 2166. The network structure when considering follower and friend count is very similar to regular Twitter users. In other words, the overall network structure is not unique when compared to accounts not exhibiting extremist behaviors for this set of data. This is one reason both network structure and content of the followers and friends are important - counts alone do not always tell the entire story.

5.1.3 Relevant Features for Extremism Detection

Given this small glimpse into the content on Twitter, we see that there are a large number of features that are relevant for identifying extremist behaviors. Unfortunately, social media data are noisy, partial, and biased. This means that there are

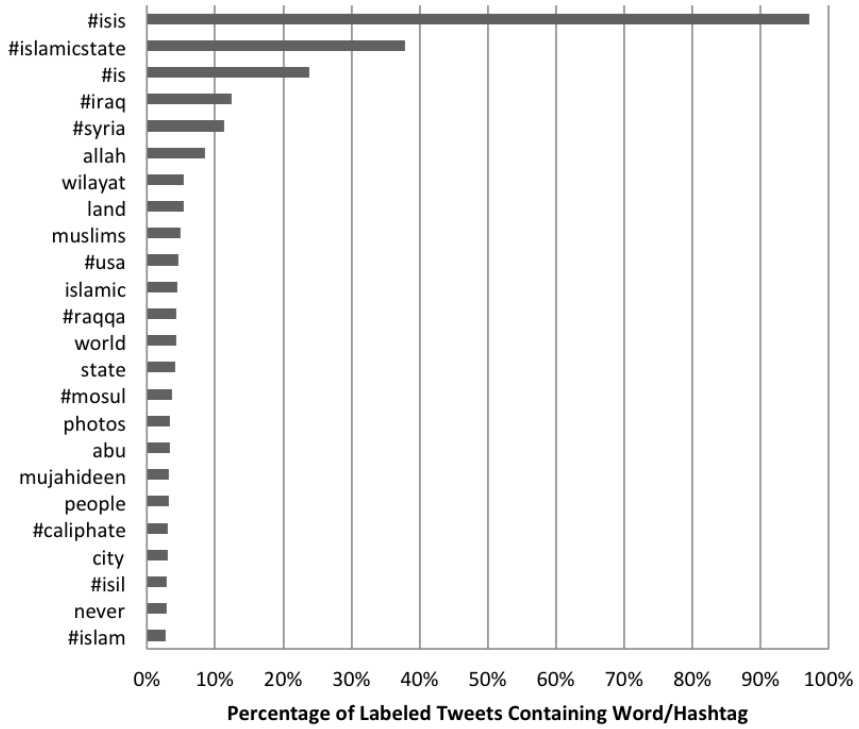


Figure 5.1: Distribution of words across tweets.

additional challenges when building classifiers in this arena. In this section, we identify features that can be used to improve the reliability of different algorithms for extremist detection. Later in the section we discuss data quality challenges. Note that we focus on Twitter features, but they can easily be generalized to features on other social media sites. We discuss a few interesting, specific examples throughout. We organize the relevant features into the following categories: user post basic content, user post viewpoint, user social media profile, user network profile, user content dynamics, and user dynamics.

User Post Basic Content

User post basic content features focus on identifying distinguishing words, symbols, and sentence structure in posts. The simplest feature is word frequency. Are the words that appear more frequently in posts containing extremist content different from words that appear in posts containing non-extremist content. The vocabulary is also important for determining the topic of a post. People do not express the same ideas the same ways. So it is important to group words into topics to better understand the similarities and differences between the content of different posts.

While one can mine the data to determine the relevant set of words for different extremist groups and topics, custom dictionaries developed by experts in the field are another important source for identifying words, synonyms, and topics that may be relevant. Other tweet specific content features, include URLs, punctuation, capitalization, hashtags, mentions, emoticons, emojis, capitalization, geotagged location, photos, photo captions, and photo metadata. While some of these features may seem useless, e.g. punctuation, each of these features has been useful for machine learning tasks related to understanding behaviors on Twitter (in different contexts). Therefore, it is important to determine which are insightful when learning behavioral norms for certain extremist conversations. Finally, the structure of the tweet can also give insight into its purpose - specific features include, the language usage, the sentence structure, and the tweet readability.

User Post Viewpoint

User post viewpoint features attempt to interpret the tone/sentiment and emotion of the post. Is the post negative or positive toward the particular extremist group or toward extremist actions? There are different opinion or viewpoint features that can

be measured including, sentiment/tone, stance, and emotion. Sentiment/tone analysis focuses on whether the post is positive, negative, or neutral. It does not consider opinion in the context of the topic of the tweet. To capture this, a feature that can be measured is *stance* [83]. Stance detection focuses on determining if the post writer is in favor, against, or neutral toward a target. Finally, while sentiment and stance are important, so is emotion. What are the emotions expressed in the post? Anger? Happiness? Despair? Effective content analysis requires that we not only understand the topic of the post, but also its tone, its stance and the emotion expressed by the author of the post. When the user shares a reasonable fraction of posts that have extremist words, negative sentiment, a stance that supports extremist groups or viewpoints, and content that is angry or meant to incite, the user is exhibiting extremist behavior.

User Social Media Profile

The *user social media profile* looks at features specific to a user account. The user's name, affiliation, profile information, social media account name (sometimes these are inflammatory), interest lists, groups joined, and location information, to name a few. These features give us insight into basic demographics and interests of the user. While the features will tend to be less distinguishing, there are some who have very overt extremist behavior that can be identified using these more general features.

User Network Profile

The *user network profile* looks specifically at the composition of the user's local neighborhood on the social media site. In the context of Twitter, features would include the number of followers, the number of followers that retweet extremist content, the

number of people the user follows that share extremist content, the size of the follower network compared to others, the clustering coefficient of the user's network (how many of the user's followers follow each other), and the sentiment, stance, and emotional profile of the user's network. If the user does not post a large amount of content himself, having this network information can be useful for identifying his potential extremist views.

Just as interesting is considering how the network associations change through time. Are more and more individuals with extremist behavior following a particular user? Is the number decreasing? Is the user beginning to follow other users that are sharing extremist viewpoints. Changing dynamics of the network is also a potentially valuable feature for this task.

User Content Dynamics

Extremist content on social media gets varying amounts of attention. *User content dynamics* focuses on features that measure the popularity of content. Is a particular post being discussed and/or retweeted more than the normal post? Are people who are reposting or retweeting it sympathetic to the extremist content in the original post or not? In other words, what is their stance on the post? Here we need to be creative with our feature generation. Possible features include the number of retweets of user content by followers exhibiting extremist features and the stance of those retweeting the content. If the content shows support for extremist ideology or thought, we may see more people denouncing it than supporting it. We also need to understand the accounts being mentioned - perhaps some of those are accounts of individuals sharing extremist content. And finally, the change in topic between the original tweet and the retweets. Are there new themes that are being generated by the tweet, e.g. new hashtags, that people are beginning to add to their own tweets?

User Dynamics

User dynamics focuses on the users distribution frequency of different types of content. Here we focus on developing a profile of the user based on the types of content shared overall (not just a specific post). For a given user, what is the distribution of his post content? What is the distribution of emotion in the content? What types of content does the user retweet? Understanding the distribution of different content streams is important for understanding the similarities and differences between those that share extremist content and those that do not.

5.1.4 State of the Art

In a social media setting, a surveillance system may have access to a restricted set of relevant features. It is unlikely that all the mentioned features will be available. Therefore, it is important to understand what can be determined with different subsets of these features. Here we review the state of the art. We highlight the features used and the results obtained in different relevant studies. At the end of the section we will discuss the future of surveillance in this field given the data challenges.

Extremism detection is a relatively new area of research in the context of social media data. This is partially because data have not been available to the academic community and earlier extremist groups did not use social media as extensively as it has been used in the last few years. One form of abusive behavior that has been studied more extensively on Twitter is spam detection [103] [101] [31] [68]. While relevant, spam propagates differently from extremist content - the campaign style is different. First, a major goal of spam is to increase visibility of a product or idea. It is not recruitment. This means that *conversations* are not needed to have a successful campaign. Second, a spam campaign is usually focused on generating profit.

An extremist campaign promotes an ideology and focuses on convincing others to believe in a particular extreme viewpoint. This leads to extremist posts that have more sophisticated content and tone than traditional spam. Finally, spam in social media is usually circulated by robots, while extremist conversations/campaigns are executed by human supporters, voluntarily in most cases. Previous literature has shown that patterns of human behavior in social media differs significantly from that of robots [67]. Given all these reasons, we believe that the behavioral patterns associated with extremist detection will vary significantly from spam detection, and therefore, different algorithms are necessary for accurate extremism detection.

We pause to mention that there are relevant areas of the literature that we do not explore in this chapter. We refer you to surveys in this area that are broader than this piece [23] [60].

Identifying extremist content

A primary task of extremist content detection on social media is crawling extremist content, for which several solutions have been proposed [79] [37]. Mei and Frank [79] classify a webpage into four sentiment-based classes: pro-extremist, anti-extremist, neutral, and irrelevant using a sentiment and word frequency based decision tree classifer. They propose a web crawler capable of crawling webpages with pro-extremist sentiment, achieving 80% accuracy. Bouchard et. al. [37] identify a set of words that are used to distinguish terrorist websites from anti-terrorist websites using content analysis of different types of websites, e.g. white supremacist websites, jihadist websites, terrorist related news websites, and counter-terrorism websites. This type of dictionary is important for understanding the goal/mission of these different sites. They used this feature analysis to develop a web crawler which automatically searches the Internet for extremist content. Looking at these examples of the literature in this

area, we see that while there are reasonable methods for identifying website extremist content, the features used are fairly basic. As more robust features related to web hyperlinks, stance, and emotion are considered, classification accuracy is likely to continue to improve.

Understanding spread

Beyond simply crawling extremist contents, [45] [40] [124] analyze content exposing extremist ideology. Chatfield et. al. [45] investigate the problem of how extremists leverage social media to spread their propaganda. They perform network and content analysis of tweets published by a user previously identified as an information disseminator of ISIS. Burnap et. al. [40] study the propagation pattern of the information following a terrorist attack. Zhou et. al. [124] analyze the hyperlink structure and the content of the extremist websites to better understand connections between extremist groups. Buntain et al. [39] study the response of social media to three terrorist attacks: the 2013 Marathon bombing in Boston, the 2014 hostage crisis in Sydney, and the 2015 Charlie Hebdo shooting in Paris. Not surprisingly, they found that the use of retweets, hashtags, and urls related to the events increased during and after the events. These different findings reinforce the importance of information dissemination and spread for extremism behavior detection.

Learning from networks using extremist seeds

Understanding the networks of different extremists is an important direction of research. One approach to doing this is to begin with a set of individuals who have been identified (most likely manually) as affiliated with extremist groups or are aliases of known extremist accounts. As an example, Berger and Morgan [33] start with approximately 400 manually selected Twitter accounts which they believe to

be official accounts of ISIS, and use different content and network features of a 2-hop network for each of these accounts to build a classifier that identifies supporters/sympathizers of ISIS. Their classifier had an 80% accuracy on the labeled data. Berger and Strathearn [34] leverage an information flow network to identify accounts serving as information hubs for promoting extremist ideology. They find that if the begin with a few seed individuals, using metrics for influence, exposure and interactivity are sufficient for identifying individuals engaging in extremist conversation.

Learning from content without extremist seeds

Suppose we are not given a set of seed individuals that are known to post extremist content. This means we do not have knowledge of a user's network in advance and must rely on content shared and the flow of the shared content. Wei and Singh [113] identify extremist behavior by dividing the problem into two subproblems: identifying relevant posts and then using those posts to identify individuals sharing content consistent with extremist views. Because different extremist groups use different vocabulary on social media, generic dictionaries are less effective. Therefore, the authors begin by identifying features that best distinguish seed posts exhibiting extremist ideology from seed posts exhibiting anti-extremist ideology. They then use these distinguishing features to identify relevant posts. The posts are then used to construct two weighted networks that model the information flow between the publishers of the identified posts. Different node centrality metrics are considered for evaluating a users' contribution to spreading extremist ideology and anti-extremist ideology. Their approach leads to an accuracy of 90% for the top extremist users, but the accuracy deteriorates for users who have more limited content and message flow. Rowe and Saif [91] investigate ways to identify behavioral changes in an individual when they transit

from a "normal" state to one that is pro-ISIS. Their approach considers changes in retweeting behavior of extremist material and language usage changes, e.g. increased use of keywords that are considered pro-ISIS and/or anti-Western. They find that in a data set of over 150,000 Twitter users, less than 1000 exhibit pro-ISIS behavior.

While progress is being made in this arena, these methods are still in their infancy. They only consider a small fraction of the features we have described and have not effectively calibrated the impact of the different types of features. Future work needs to understand the limits and biases associated with content analysis.

Other types of inappropriate behavior detection

There are many other threads of relevant research related to detecting inappropriate behavior. These include promoting marijuana [42], attacks of identity theft [35], identifying suspicious urls [69], and finding worms that are propagating [118]. These papers leverage different content, including follower demographics, url reuse, etc. While the profiles built have some of the same features at a broad level, the dictionaries and specific features used must be custom designed. Still, it is important to remember that these other tangential areas give us insight into the subsets of features that are distinguishing in similar domains.

Sentiment and Stance

We pause to discuss sentiment analysis since it is important for understanding tone. Tweet sentiment analysis has been an active area of research in recent years. Most sentiment algorithms are supervised. The commonly employed features include word-related features (words, stems, n-grams) [28] [63] [88], word shape features (punctuations, and capitalization) [63], syntactic features (POS taggers, dependency trees) [22] [28] [88], and Twitter-specific features (Twitter handles, hashtags, urls, emojis,

emoticons) [22] [28] [63]. Li et al. [72] propose an unsupervised algorithm that uses a sentiment topic model with decomposed priors. Lexicon-based methods do not require training data. Gutierrez et al. [59] propose to classify Spanish sentence sentiment using support vector machines and a Spanish sentiment lexicon. Instead of simply using the fixed sentiment polarity of words, Saif et al. [93] update the sentiment of words based on co-occurrences of words in different contexts. As previously mentioned, Mei and Frank [79] classify a webpage into four sentiment-based classes: pro-extremist, anti-extremist, neutral, and irrelevant. Their sentiment classifier considers only nouns.

Most of the approaches mentioned above are target-independent sentiment analysis. There is no doubt about an occurrence of a complimentary word strongly indicates positive sentiment. However, our task considers features that are needed for target-dependent sentiment analysis, i.e., we aim to classify the sentiment in a tweet towards an extremist group. This is where the idea of stance comes in. Liang et al. [63] define 7 types of syntactic patterns, to extract co-occurrences of a target object and expressions with sentiment. Similarly, Nasukawa et al. [85] adopt a set of human-created rules for nouns, verbs, and adjectives to extract sentiments towards specific objects. Hu et al. [61] summarize sentiment of customer reviews towards products. Since the customer reviews could be considered towards specific products presumably, all the expressed sentiment is relevant to the target products for their task. Finally, Mohammad, et al [83] use a linear-kernel SVM classifier that combines features drawn from training data with more general features to determine stance. Their classification performance is around 80% and they show the difference in sentiment and stance and the need for both.

5.1.5 Extremism Detection Case Study

This section is a summary of work in Wei et al. [113]. We conduct a small exploratory case study using approximately 2 million accounts, identifying the number of accounts exhibiting extremist behavior.

5.1.6 Features for Identifying Extremist Behavior

For this analysis, we consider the following features: sentiment/stance of tweets, the polarity of the user's ego-network, and user mentions as different proxies for misbehavior.

Sentiment/Stance Tendency Feature Anecdotal observations support the idea that a user with extremist views consistent with an extremist group will show positive sentiment towards that group in his/her posts, while an ordinary person will show negative/neutral sentiment towards the group. Based on this observation, our sentiment tendency feature $\mathcal{S}$ measures the sentiment tendency for an individual based on the overall sentiment of his/her published tweets. To create this feature, we begin by identifying the tweets associated with a particular extremist group. One simple approach is to look for any variant of the name of the group in the tweet. For ease of exposition, let T equal the set of tweets that are relevant for a particular user. Each tweet is assigned a sentiment value (positive or negative/neutral). Then $\mathcal{S} = \frac{\sum_{i=1}^{|T|} sentiment(t_i)}{|T|}$, where $sentiment(t_i)$ is the sentiment of tweet t_i. A score of 1 is assigned if the sentiment of a tweet is positive. A score of 0 is assigned if the sentiment of the tweet is neutral or negative. Therefore, if $\mathcal{S} > 0.5$, the sentiment is classified as positive.

Ego-Network Extremism Support Feature We make the following two observations about extremist content on Twitter: (1) a user with extremist views con-

sistent with an extremist group is highly likely to be followed by at least one user exhibiting similar extremist views, and (2) an ordinary user might follow a user exhibiting extremist-associated behaviors. Adversaries, socialists, journalists, researchers, including researchers on our team, do this to monitor/learn behaviors of individuals with extremist views. Based on these two observations, if a user has one or more followers that have a positive sentiment tendency $\mathcal{S} > 0.5$ and the user has a positive sentiment tendency, our ego-network extremism support feature $\mathcal{E} = 1$. Otherwise, it is zero.

Mention-Network Feature When analyzing the data, we observed the following: (1) Users with extremist content in tweets are highly likely to be mentioned by other users with a similar viewpoint, and (2) an ordinary user might mention a user exhibiting extremism-associated behaviors in his/her tweets. Based on these two observations, our mention network feature $\mathcal{M} = 1$ if at least one other user exhibiting extremist sentiment tendencies mentions the user and the user has a positive sentiment tendency.

Data Set Exploration

We collected tweets with references to any of these hashtags between September 2014 and April 2016 using Twitter API. Table 5.2 shows the number of tweets associated with the main ISIS related hashtags in both English and Arabic during our data collection period. Tweets containing an English hashtag pertinent to ISIS were published by approximately 2 million Twitter users, and tweets containing an Arabic hashtag pertinent to ISIS were published by approximately 1.2 million Twitter users.

We use the Twitter API to obtain tweets with references to ISIS-relevant hashtags as the first round of data collection. The second round of data collection focuses on

Table 5.2: The number of tweets containing different ISIS related hashtags.

Language	Hashtag	#Tweets
English	#ISIS	13 million
	#ISIL	2.5 million
	#IS	3.2 million
	#Islamic_State	20 thousand
	#Islamic_State_in_Iraq_and_Al-Sham	1.5 thousand
	#Islamic_State_in_Iraq_and_the_Levant	5 thousand
Arabic	ISIS	14 million
	ISIL	44 million
	IS	6.7 million
	Islamic State	25 thousand
	Islamic State in Iraq and the Levant	1.2 million
	Islamic State of Iraq and Al-Sham	46 thousand

collecting profiles of the users who published one or more of the downloaded ISIS-related tweets. Since we began data collection, 11% of the 2 million Twitter users that published tweets using one or more of the English ISIS-related hashtags have been suspended, while 23% of the accounts using the Arabic ISIS-related hashtags have been suspended. These suspensions occurred during our data collection. For those subset of users, we do not have their network information or any tweets after the account suspension.

Tweet Sentiment/Stance Classification

When using a basic sentiment analyzer that considers just content for measuring sentiment and stance, we found that the majority of tweets did not contain any sentiment or stance, i.e. they are more objective or neutral. Based on previous literature, we build a custom stance classifier that contained the following features: unigrams (except

stopwords), emoticons, URLs, hashtags, and negation expressions. Our project team containing Middle East experts manually labelled 3800 English tweets using the following categories: positive (positive stance towards ISIS), negative (negative stance towards ISIS), neutral, and noise (tweets considered noise are removed from the later experiments). Our experts were very specific with their labeling. For example, negative stance towards ISIS is different from negative sentiment in a tweet discussing President Obama's policies related to ISIS.

Since our goal is to detect posts supporting extremist views, a tweet with positive stance towards ISIS is labeled *positive*, while a tweet with negative/neutral stance towards ISIS is labeled *negative*. We consider four different binary classifiers: Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Table 5.3 shows the accuracies achieved by these classifiers using 10-fold cross validation. We also test three popular sentiment analysis tools (Stanford CoreNLP Sentiment Analyzer [10], vaderSentiment [16], and SentiStrength [9]) on the same dataset. All of these three tools are set in binary mode (positive VS negative/neutral), and their accuracies are shown in Table 5.3 . We see that our classifiers (NB, LR, and SVM) outperform these state-of-art tools. Given these results, we use Naive Bayes for the remaining experiments related to this case study.

Extremism Detection

In order to evaluate our extremism detection results, researchers on our project team manually identified 100 accounts with content consistent with pro-ISIS views among suspended accounts, 100 with content consistent with pro-ISIS views among unsuspended accounts, 100 with content consistent with anti/neutral-ISIS views among suspended accounts, and 100 with content consistent with anti/neutral-ISIS among

Table 5.3: Accuracy of classifiers based on different models and state-of-art sentiment analysis tools in classifying tweet sentiment.

Classifier	Accuracy
NB	85%
LR	85%
SVM	70%
KNN	51%
Stanford CoreNLP	68%
vaderSentiment	53%
SentiStrength	57%

unsuspended accounts. The extremist detection decisions given by different combinations of features presented in the previous subsection were then evaluated against the hand labeled results: sentiment/stance-tendency (SENT), the ego-network extremism support (EGO), and the mentioned-network (MENTIONED). The results are shown in Figure 5.2. SENT only considers the user's sentiment/stance tendency. EGO only considers the user's ego-network extremism support. MENTIONED only considers the user's mentioned-network. SENT+EGO considers the user's sentiment/stance tendency and the user's ego-network extremism support. SENT+MENTIONED considers the user's sentiment/stance tendency and the user's mentioned-network. ALL combines all the features. Note that we separate our analysis of the suspended accounts from the not suspended accounts because we do not have EGO features for the suspended accounts. From Figure 5.2, we see that when classifying the unsuspended users, the detector combining all the three features outperforms the others; for classifying the suspended accounts the detector combining the SENT and the MENTIONED features has the highest accuracy.

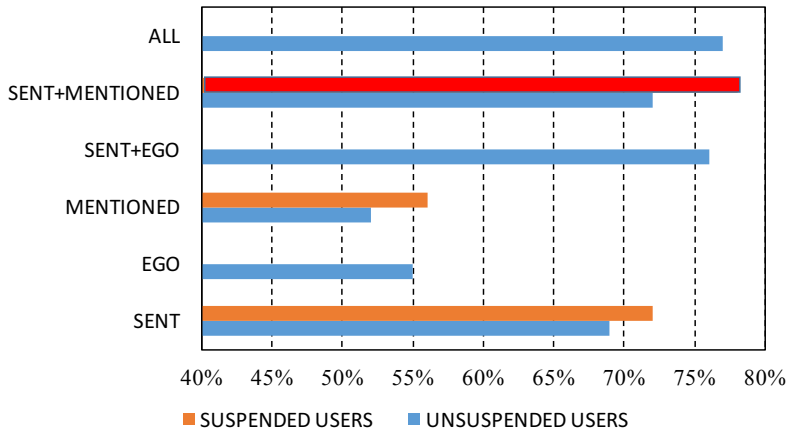


Figure 5.2: Extremism detecting accuracy by different detectors.

Our final analysis was using all the proposed features to detect the number of extremist accounts in our full data set. Table 5.4 shows the number of users classified as having content consistent with pro-ISIS views and anti-ISIS views. While we do not have a ground truth to validate the results, the numbers confirm our intuition that suspended accounts have a much higher percentage of users having content consistent with pro-ISIS views than unsuspended accounts; While misclassification does exist, the proportion of misclassification should be relatively similar in the 2 data sets. Using this approach for surveillance may help Twitter identify these accounts more rapidly.

5.1.7 Capturing Data for Surveillance Systems for Extremism Detection

Surveillance of social media extremist behavior is very challenging. Social media sites limit the amount and type of data they are willing to let people download using their free APIs. This adds a new level of complexity to surveillance. Even in the best

Table 5.4: The number of accounts determined as having content consistent with pro-ISIS views.

		Suspended Accounts	Unsuspended Accounts
SENT	Positive	24,761	49,951
	Negative	174,100	1,769,559
EGO	Positive	NA	36,784
	Negative	NA	1,782,726
MENTIONED	Positive	15,832	23,549
	Negative	183,029	1,795,961
SENT+EGO	Positive	NA	42,158
	Negative	NA	1,777,352
SENT+MENTIONED	Positive	18,542	40,352
	Negative	180,319	1,779,158
ALL	Positive	NA	30,671
	Negative	NA	1,788,839

case scenario, a group does not have funds to pay for all the data a social media site has. So a feature selection process needs to take place. Here we suggest different options for collecting relevant Twitter data: (1) Identify a small number of Twitter accounts exhibiting extremist behavior manually and download the tweets, followers and friends for those accounts. Those accounts can be used for a small case study, but they can also be considered seed accounts. It is best if the seed accounts are not friends and followers of each other. Once all the available data are collected from these seed accounts, we can begin collecting tweets, profile information and summary network statistics for the users that are friends and followers of these seed individuals. Good collection will continue building these ego networks from this snowball sample until the data collection period is over; (2) Identify hashtags that correspond to the names or ideals of different extremist groups. Use the API to collect data from these streams;

(3) Randomly select the names of individuals who post negative content on one of the hashtags related data streams. Conduct a 2-hop snowball sample beginning with these accounts as seeds. Each of the accounts captured need to be continually collecting data every few weeks. Otherwise, any emerging dynamics or changing behaviors will not be detected.

Keep in mind while this process will result in raw data that is meaningful, many of the features described need to be constructed from these raw data using state of the art methods described in the previous section and by developing new methods as well. Another consideration is the processing power and storage space needed for these types of surveillance systems. We try to keep older data to help us understand the changing dynamics of the extremist groups. It is important to continually update models based on current data. At the same time, the historic data gives us an opportunity to establish ground truth for different predictive tasks. As long as data sets are not available for this purpose for these types of analyses, we must create our own data sets using different social media APIs.

5.2 Identifying Users Sharing Extremist Content

In this section, we propose an approach for identifying users who engage in extremist discussions online. Our approach uses detailed feature selection to identify relevant posts and then uses a novel weighted network that models the information flow between the publishers of the relevant posts. An empirical evaluation of a post collection crawled from a web forum containing racially driven discussions and a tweet stream discussing the ISIS extremist group shows that our proposed method for relevant post identification is significantly better than the state of the art and using a network flow graph for user identification leads to very accurate user identification.

5.2.1 Introduction

Users endorsing extremist ideology have been increasingly leveraging social media to spread their viewpoint and promote their agenda. For example, Islamic State of Iraq and Syria (ISIS) has been using social media platforms to share their ideas and recruit members/jihadists to their groups. This work presents a method for identifying users who share extremist viewpoints on social media. Our hope is that early identification will provide law enforcement options for early identification of individuals before they become dangerous.

Previous literature concerning identification of users sharing extremist content [34] [33] [50] [19] assumes that the target user is a friend (or within a few hops of friends) of validated accounts affiliated with extremist groups [34] [33], or alias accounts of these validated accounts [50] [19]. While an important direction, our approach focuses on a method that does not require knowledge of the network structure in advance.

Specifically, we divide the problem into two subproblems: identifying relevant posts and then using those posts to identify individuals sharing content consistent with extremist views. Because different extremist groups use different vocabulary on social media, generic dictionaries are less effective. Therefore, we propose an approach that begins by identifying features that best distinguish seed posts exhibiting extremist ideology from seed posts exhibiting anti-extremist ideology. We then use these features to identify relevant posts. The posts are then used to construct two weighted networks that model the information flow between the publishers of the identified posts. Different node centrality metrics are considered to evaluate users' contribution to spreading extremist ideology and anti-extremist ideology. Users with more contribution to sharing content and/or spreading extremist ideology than anti-extremist ideology are regarded as promoting extremist content.

The main contributions of our work include: 1) we propose a new method for identifying relevant content that results in a much higher accuracy than the state of the art; 2) we propose using information flow networks to find users sharing extremist and anti-extremist viewpoints; 3) we empirically evaluate our method on a web forum and a tweet stream and find that our method leads to accuracies above 90% in some case.

5.2.2 Notation, Assumptions, and Definitions

This section presents definitions, assumptions, and a formal problem statement.

Notation and Assumptions Let P be a set of posts associated with a particular forum or discussion stream on a social media site. These posts are written by a set of users U. A particular post p is written by a specific user u and is denoted p_u. For ease of exposition, we will use positive as a proxy for posts or users exhibiting extremist ideology, and negative as a proxy for posts or users exhibiting anti-extremist ideology.

We make the following assumptions about the post collection P:

1. There are features differentiating positive posts P^+ and negative posts P^-.

2. A user u might publish posts containing content that has contradictory viewpoints on extremism, e.g., a user endorsing extremist ideology might publish tweets exhibiting extremist ideology, while at the same time retweeting and replying to tweets containing content of the opposite position.

3. While user u may post a range of differing ideological messages, we assume that there is a direct relationship between the number of positive posts u propagates and the probability of promoting an extremist ideology. Similarly, we assume there is a direct relationship between the number of negative posts u propagates and the probability of promoting an anti-extremist ideology.

4. A user u that has an extremist viewpoint would play a more important role in spreading positive information than spreading negative information.

Problem Statement Given a post collection P and the set of their publishers U, the task of identifying users sharing extremist content has two subtasks:

1. *Relevant Content Identification*: Identify the posts from P that have content consistent with extremist ideology (P^+) and anti-extremist ideology (P^-).

2. *Extremist User Identification*: Identify the users from U that have a viewpoint consistent with extremism (U^+).

5.2.3 Extremism Detection

The framework for our proposed approach is divided into two subtasks: Relevant Content Identification and Extremist User Identification. Figure 5.3 shows the major steps associated with each subtask: feature selection, post retrieval, network creation, user centrality calculation, and user centrality integration.

Algorithm 5 presents a high level view of this proposed approach. The input to our approach is a post collection P, a set of positive seed posts S^+, and a set of negative seed posts S^-. The output is a set of users U^+ identified as having a viewpoint consistent with extremism. The approach begins by identifying features F^+ best distinguishing positive seed posts from negative seed posts (Line 1), and features F^- best distinguishing negative seed posts from positive seed posts (Line 2). From P, posts containing F^+ and F^- are retrieved, respectively, denoted as P^+ and P^- (Line 3 and Line 4). An information flow network $G^+(V, E)$ is constructed, in which each node in V represents a user of a post in P^+ (Line 5). A directed edge in $G^+(V, E)$ is added to the network if a node v_i responds or reposts a message sent by node v_j. For nodes in $G^+(V, E)$, centrality metrics C^+ are calculated (Line 7).

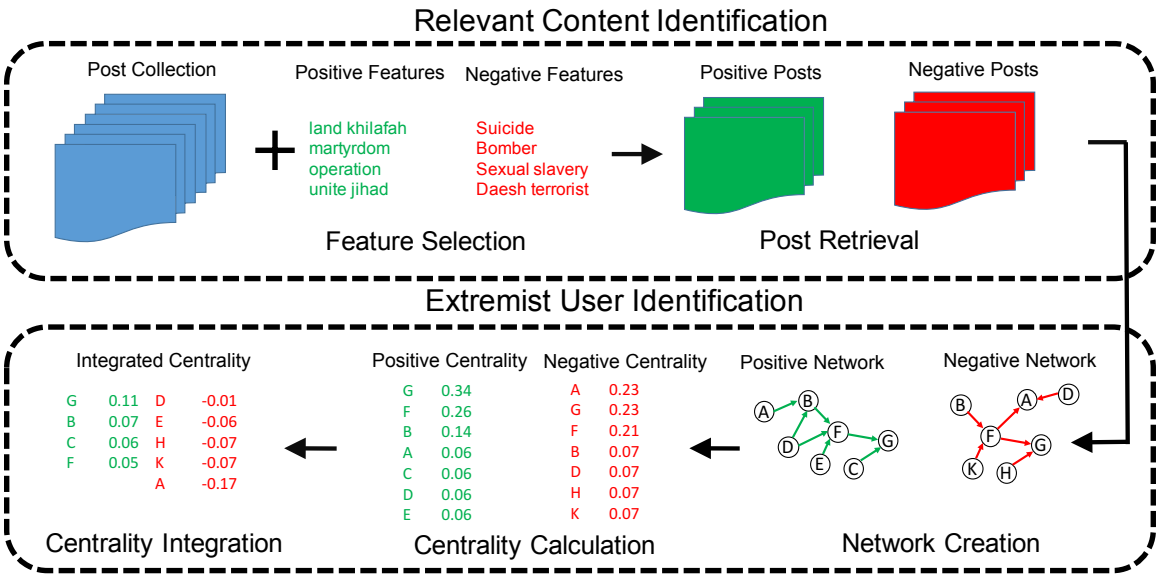


Figure 5.3: The framework of our proposed approach.

Similarly, using P^-, an information flow network $G^-(V, E)$ is constructed. For all the nodes in $G^-(V, E)$, centrality metrics C^- are calculated (Line 6 and Line 8). For all the nodes in $G^+(V, E)$ and in $G^-(V, E)$, their centrality C_u^+ and C_u^- are integrated into a single score centrality score C_u (Line 9). Users with positive integrated centrality are regarded as individuals sharing content containing extremist views.

Feature Selection

Our approach uses all the ngrams in the seed posts S as the feature pool. The goal of feature selection is to select features best distinguishing positive seed posts from negative seed posts, or vice versa. A basic way to accomplish this is to compute the difference between the number of occurrences of an n-gram f in S^+ and in S^-: $N(f, S^+) - N(f, S^-)$, where $N(f, S^+)$ denotes the number of occurrences of f in S^+, and $N(f, S^-)$ denotes the number of occurrences of f in S^-. $N(f, S)$ only considers the

Algorithm 5: High level algorithm for extremist user detection.

Input:
A post collection: P
A set of positive seed posts: S^+
A set of negative seed posts: S^-

Output: *Users identified as endorsing extremism ideology*

$F^+ = select\_positive\_features(S^+, S^-)$
$F^- = select\_negative\_features(S^-, S^+)$
$P^+ = retrieve\_positive\_posts(F^+, P)$
$P^- = retrieve\_negative\_posts(F^-, P)$
$G^+(V, E) = create\_positive\_information\_flow\_network(P^+)$
$G^-(V, E) = create\_negative\_information\_flow\_network(P^-)$
$C^+ = calculate\_centrality(G^+)$
$C^- = calculate\_centrality(G^-)$
$C = integrate\_centrality(C^+, C^-)$
return $\{u|C_u > 0\}$

intensity of a feature being used, ignoring its popularity among users. This can result in noisy features being retained. In an extreme case, a small number of users may repeatedly publish the same post. Considering only $N(f, s)$, most of words (excluding stopwords) in this post would be selected as features, including words that may be less relevant to extremism. Therefore, we consider incorporating user coverage of a feature. A higher user coverage indicates a feature being associated with posts written by different people. To evaluate user coverage of a ngram, we define a feature's Author Entropy ($\mathcal{E}$): $\mathcal{E}(f, S) = \sum_u (N(f, S_u) \log \frac{\sum_u N(f, S_u)}{N(f, S_u)})$, where $N(f, S_u)$ denotes the number of seed posts published by user u containing feature f. We use this notion in conjunction with intensity of feature usage to define a feature's importance ($\mathcal{I}$): $\mathcal{I}(f, S) = \mathcal{E}(f, S) \times N(f, S)$ We calculate each feature's importance $\mathcal{I}(f, S^+)$ for positive seed posts and $\mathcal{I}(f, S^-)$ for negative seed posts. Then we rank all the features in a descending order according to their $\mathcal{I}(f, S^+)$ score. We only consider those where

$\mathcal{I}(f, S^+) > 0$ and $\mathcal{I}(f, S^-) = 0$. We select the top k features, denoted as F^+ and use these features to retrieve positive posts from P in the next step. In a similar way, we rank all the features in a descending order according to their $\mathcal{I}(f, S^-)$, and select the top k features, denoted as F^-, with the constraints that $\mathcal{I}(f, S^+) = 0$ and $\mathcal{I}(f, S^-) > 0$. These features are leveraged to retrieve negative posts from P.

Post Retrieval

This step focuses on retrieving posts containing the selected features F. To measure a post's relevance to a set of features F, we define Feature Relevance (Fr) as: $Fr(p, F) = \sum_{f \in F} Tf(f, p) Ae(f, p)$ where $Tf(f, p)$ represents the term frequency of feature f in a post p. We do not incorporate the commonly used inverse document frequency (IDF) into Feature Relevance. IDF prioritizes items most different from the rest of the corpus. Its success relies on the premise that a query term occurring less frequently in a corpus contains more information, and thus, is of more importance. However, this assumption does not hold in our case, e.g., the occurrence of white nationalist in a post would be a strong indicator of promoting white supremacy; however, it would not have a low IDF value on a forum discussing ethnic/racial issues.

For each post, we calculate its Feature Relevance to positive features $Fr(p, F^+)$ and to negative features $Fr(p, F^-)$, respectively. We retain all the posts satisfying $Fr(p, F^+) - Fr(p, F^-) > 0$, denoted as P^+, and retain all the posts satisfying $Fr(p, F^-) - Fr(p, F^+) > 0$, denoted as P^-.

Information Flow Network Creation

While there are many different representations of networks, we choose to leverage an information flow network. Posts on social media are information flows between users, e.g., on Twitter, a user can reply to, retweet, or mention other user(s) in a tweet; on

a forum, a user can reply to or quote another user's post. We propose constructing information flow networks to identify the flow of extremist views.

A weighted information flow network $G = (V, E)$ is composed of a set of nodes $V(G) = v_1, ..., v_n$ and a set of edges $E(G) = e_1, ..., e_m$. Each node v_i represents a user u_i, and an edge (v_j, v_k) is added to the network G if user u_j responds to (e.g., retweets, replies to, quotes, etc) a post of user u_k. If the post does not result in an edge between two users, a self edge is added to the graph. This is important because it allows us to capture extreme content that is being posted, but not necessarily propagating. G is also an edge weighted graph $\mathcal{W}(e_i)$, where an edge weight represents absolute information flow (described below).

While a single information flow graph can be constructed with edges containing both positive and negative post information, we choose to separately analyze positive and negative information flow by constructing two more focused graphs, G^+ and G^-. For the positive posts P^+, we construct an information flow network $G^+(V, E)$ and define the edge weight to be the the difference between the positive and negative features that are relevant to the post $Fr(p, F^+) - Fr(p, F^-)$. We choose this edge weight scheme since it reflects the *absolute* amount of positive information flowing along the edge. Similarly, for the negative posts P^-, we construct an information flow network $G^-(V, E)$ and define the edge weight to be the the difference between the negative and positive features that are relevant to the post $Fr(n, F^-) - Fr(p, F^+)$.

User Centrality Calculation

We use node centrality to measure each user's importance in sharing relevant positive and negative content. Among node centrality metrics, we consider degree (number of connections of u_i), node betweenness (fraction of shortest paths going through node v_i), pagerank (importance of node v_i based on importance of connections of v_i),

and personalized pagerank (customized importance for specific types of graphs). In computing personalized pagerank [44], we need to designate a user-custom adjustment to pagerank in each iteration: $C = \alpha \mathbf{A} \times C + (1-\alpha)C'$, where $\mathbf{A}$ denotes the transition probability matrix, C' is a user-custom vector to adjust the pagerank vector, and α is a user-custom weighing factor. We use the sum of the weight of outgoing edges incident to nodes as the user-custom vector: $C'_u = \sum_p Fr_u(p, F)$. In other words, we are increasing a user's score if he/she is sharing more content in G^+ or G^-. We calculate centrality C^+ and C^- for nodes in the positive $G^+(V, E)$ and the negative $G^-(V, E)$ networks.

User Centrality Integration

As stated in Section 5.2.2, we make the assumption that a user might publish posts containing content having contradicting viewpoints. We also assume that a user posting content consistent with extremist views would play a more important role in spreading positive information when compared to spreading negative information. In this step, we integrate C_u^+ and C_u^- into a single score C_u to measure a user's *absolute* importance for sharing/spreading positive information: $C_u = C_u^+ - C_u^-$. Users satisfying $C_u > 0$ are considered to be users promoting extremist views.

5.2.4 Evaluation

In this section we begin by describing the data sets, and then evaluate the different steps of our framework.

Data Sets

For our empirical analysis, we consider two distinct types of social media: microblogs and forums. The microblog data set is a Twitter stream. The forum data set is the Stromfront [12] data set.

Microblog Data Set: We work with an interdisciplinary team consisting of students, researcher, and policymakers. Some of them have years of in-field research experience in the Middle East. With help from our subject matter experts, we identified a set of hashtags that are related to ISIS. Using the Twitter API, we collected tweets containing these hashtags between September 2014 and April 2016. In total, this data set consists of 23 million tweets, published by approximately 2 million users. In previous work, we have shown that this data set contains extremist content [113]. For this evaluation, our task is to identify users sharing extremist views consistent with Islamic fundamentalism, or showing support for jihadist groups, including ISIS, Al-Qaeda, Jabhat Al Nusra, etc.

Forum Data Set: Stromfront [12] is a web forum that includes many radically driven discussions with a right-wing extremist focus. The most prevailing extremist ideologies are racism and antisemitism. The forum consists of scores of sub-forums, and each sub-forum has an explicitly-stated focus. From the sub-forums having an explicitly-stated focus on philosophy and ideology, we crawled 2.9 million posts. Our task here is to identify users promoting and/or sharing content that is racist or antisemitic.

Feature Selection

Subject matter experts on our team manually identify 1,300 tweets containing content promoting extremist ideology, 1,300 tweets containing content consistent with anti-extremist ideology, and 2,600 neutral tweets from the tweet collection. We use these

2,600 positive/negative tweets as seed posts to select features best distinguishing positive seed posts and negative seed posts. Basic pre-processings, including punctuation removal, stop word removal, and non-English word removal, are applied to these seed posts. Using the feature selection approach described in Section 5.2.3, the top 20 distinguishing features are identified. We set $k = 20$, since 20 is commonly regarded as an appropriate number of query terms for retrieval tasks [48]. Table 5.5 shows the top 5 positive features and negative features. We can see that the positive features have a clear focus on martyrdom and caliphate, while the negative features focus on terror, Daesh (a derogatory term for ISIS), and Yezidi (ISIS is holding thousands of Yezidi girls as slaves).

For the Stromfront post collection, we identify 500 seed posts containing extremist views and 500 seed posts containing anti-extremist views. Feature selection is applied to these 1,000 seed posts. The results are also shown in Table 5.5. We see that the positive features have a theme of white knights, while the negative features focus on evil and hate. We pause to mention that we also considered the simpler approach for feature selection that only uses the intensity of the word to identify features. Using this approach resulted in a larger number of noisy, information poor words, e.g., hey, entire, good claim, agenda.

Post Retrival

For the tweet collection, using the top 20 positive features and top 20 negative features as query terms, 24,452 tweets and 462,436 tweets are retrieved as positive tweets and negative tweets; for the Stromfront post collection, 56,498 posts and 12,696 are retrieved as positive posts and negative posts.

To better understand the accuracy of using our post retrieval method (referred to as XtremePost), we compare our method to four other methods:

Table 5.5: Top five features selected using feature importance $\mathcal{I}$.

	Twitter		Stromfront	
	Positive	Negative	Positive	Negative
1	mujahideen	tcot	knights	nazis
2	martyrdom	yazidi	united white	evil nazis
3	allah accept	bombers	klux klan	hate crime
4	alhamdulilah	suicide bombers	white knights	warmongers
5	martyrdom operation	kittens daesh	jews	liberals

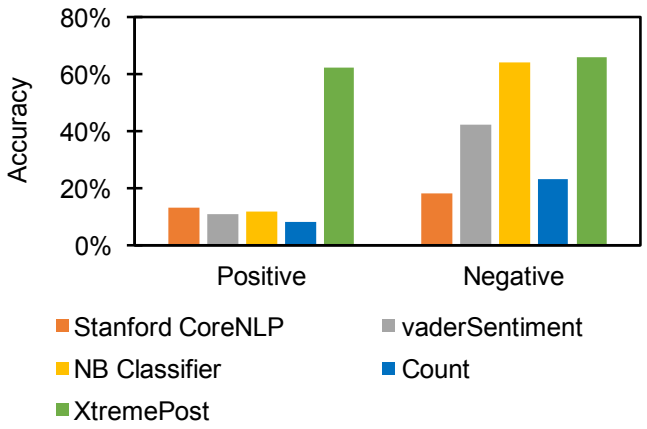
1. A Naive Bayes classifier (NB Classifier) that incorporates unigrams, emoticons, urls, and POS taggers to identify extremism. It was built using the labeled ground truth data. Note, we did experiment with other classic machine learning algorithms, including Logistic Regression, Support Vector Machines, K Nearest Neighbors and Decision Trees. Naive Bayes performed better than the other models.

2. Two state of the art sentiment detection tools - Stanford CoreNLP [10] and vaderSentiment [16] [1]

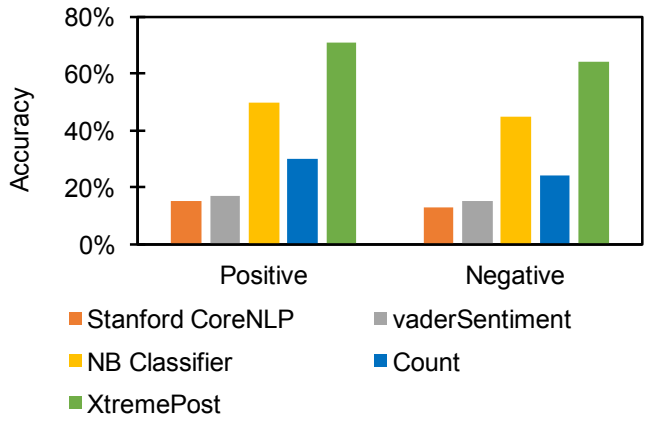
3. Using features generated by computing the difference in frequency intensity of positive and negative posts (referred to as Count).

Due to the lack of ground truth labels of the retrieved tweets, we randomly sample 200 posts from each of the two classes. The results are shown in Figure 5.4. We can see that our approach achieves 62%/66% accuracy in identifying positive/negative posts in the tweet collection, and 71%/64% accuracy in the Stromfront post collection, significantly outperforming other methods. When analyzing the results, we find

[1] vaderSentiment is also the tool employed by [97] to identify extremist users.



(a) Twitter.



(b) Stromfront.

Figure 5.4: Accuracy of extremism post identification by different approaches.

that our approach performs well because: 1) it is insensitive to abnormal grammar structures and made-up words; 2) it is insensitive to data with a skewed distribution across classes. The Naive Bayes classifier has the second best performance, but it has a very low accuracy in identifying positive posts in the tweet collection. This is not surprising since only 0.5% of the tweets in this collection are positive. The underperformance of two sentiment analyzers can be attributed to the prevalence of noise in social media posts. Social media posts tend to contain made-up words and do not to follow normal grammar rules.

Centrality Calculation

Based on the retrieved positive/negative posts, we build the information flow network. For the tweet network, an edge is added to the network if a user is retweeted, replied, or mentioned in a tweet. For a tweet that does not refer to another user, a self-edge incident to its author is added to the network. For the Stromfront post network, an edge is added to the network if a user is quoted or replied to in a post. Similarly, self-edges are added if the post does not refer to another user.

Network analysts use different measures of centrality to define importance. As mentioned in Section 5.2.3, we consider degree, betweenness, pagerank, and personalized pagerank. We compare all of these methods to a simple method that only considers the frequency of positive and negative posts. All the methods return a comparable number of users. For the tweet collection, the number is around 7,000; for Stromfront post collection, the number is around 20,000.

We sample 100 users from all the users identified by the different methods and evaluate what percentage of the 100 users post extremist content. The results are shown in Figure 5.5. We can see that using the information flow graph and the centrality metrics of degree and personalized pagerank result in the highest accuracies.

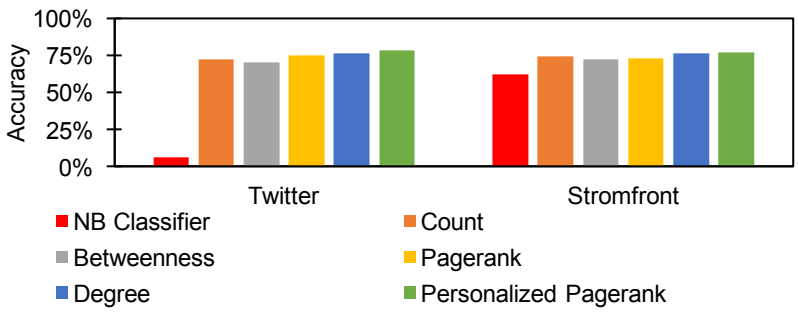


Figure 5.5: Accuracy of extremist user identification by different methods.

However, all of the information flow methods and the simpler count method have comparable accuracies.

In order to better understand the percentage of all the users endorsing extremist ideology, we take a 400 user random sample from each post collection, and manually check their posts. We find that about 38% users in the Stromfront post collection endorse extremist ideology in their posts, while only 0.5% users in the tweet corpus endorse extremist ideology in theirs posts.

In another experiment, we rank the identified users according to their centrality scores for each method. We are interested in the accuracy of our proposed approach at different positions along the scale. We take a 50 user sample from users identified as endorsing extremist ideology by each method at different positions along the scale, and manually evaluate what percentage of them post extremist content. For example, we take all the top 50 users, sample 50 users from the top 1,000 users, sample 50 users from the 1,001 to 2,000 ranking, etc; then we classify them as sharing extremist content or not. The results are shown in Figure 5.6. We observe a trend that the higher an identified user ranks in the scale, the more likely the user posts contain

Table 5.6: Cost of centrality computation. ×100 represents 100 cores allocated.

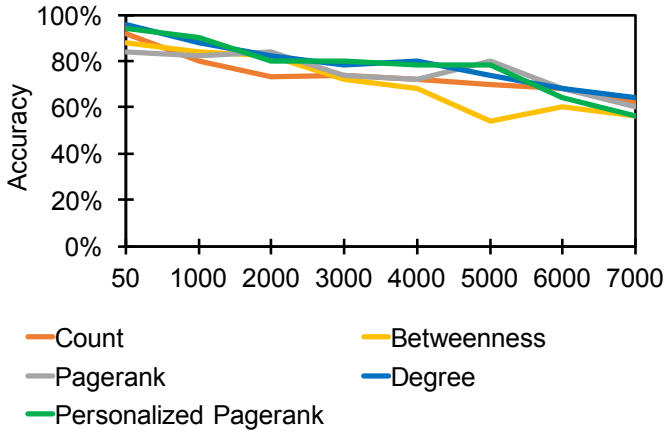
	Twitter	Stromfront
Count	0.6s	0.9s
Degree	0.8s	1s
Betweenness	744s×100	86s×100
Pagerank	911s×100	261s×100
Personalized Pagerank	942s×100	302s×100

extremist views. This means that all the information flow methods can identify the most extreme users effectively.

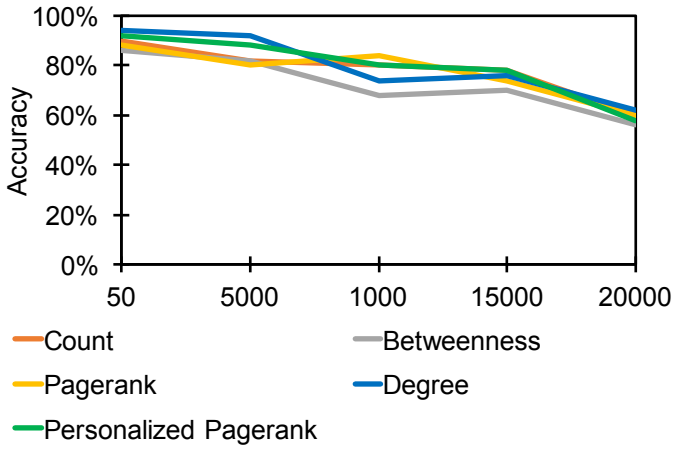
Finally, we evaluate the efficiency of different methods. Since Betweenness, Pagerank, and Personalized Pagerank are computationally intensive, we implement them on a distributed yarn cluster, which consists of 12 nodes. Each node has 16 CPUs. In computing these metrics, we designate 10 executors, with 10 cores for each executor, which result in 100 cores. On the other hand, since both the Count and Degree method are computationally trivial and the overhead of initializing the yarn cluster would cost more than calculating these two metrics, we implement them as single thread programs. Table 5.6 shows the time cost of the different methods. We can see that the Betweenness, Pagrank, and Personalized Pagerank methods are considerably more expensive than the Count and Degree methods. Considering both accuracy and computational cost, the Degree method ends up being the best performer.

5.2.5 Conclusions

In this section, we propose an approach for identifying users endorsing extremist ideology on social media. Our approach first identifies posts exposing extremist ide-



(a) Twitter.



(b) Stromfront.

Figure 5.6: Accuracy of extremist user identification at different positions along the scale of users ranked according to different methods.

ology and posts exposing anti-extremist ideology, then constructs two weighted networks to model the information flow between the publishers of the identified posts. Different node centrality metrics are considered to evaluate users' contribution to spreading extremist ideology and anti-extremist ideology. Users with more contribution to spreading extremist ideology than anti-extremist ideology are labeled as individuals sharing extremist views. We empirically evaluate our approach on two social media post collections. We find that our approach for identifying posts that contain extremist views is significantly better than the state of the art. We also show that using an information flow graph can achieves over 90% accuracy when identifying the top scoring users sharing extremist content.

Chapter 6

Conclusions and Future Directions

This dissertation focuses on developing methods for detecting events of a particular type from different forms of open-source data, e.g. newspapers, blogs, and social media. This dissertation makes the following contributions:

1. In Chapter 2, we analyze a large newspaper collection and investigate the strengths, weaknesses, and biases associated with open-source, big data analysis. This initial analysis shows the promise of using open-source data for identifying documents and topics that may be useful for understanding more about the movements of people.

2. In Chapter 3, we propose a comprehensive framework that utilizes a location ontology and a domain dictionary to identify overlapping, target events using news articles from a large, noisy news corpus generated from multiple new sources. Using this framework, we propose an offline event detection approach, and an online event detection approach. To the best of our knowledge, our method is the first targeted event detection algorithm that detects events and story lines occurring at the same time. For both the offline and online approaches, we make use of a semantic graph constructed from sentences within articles from the corpus. Detected events are represented as sets of sentences, which are much more informative than event representations adopted by most of state-of-art approaches. Evaluating the identified events against

ground truth events collected by subject matter experts show that our proposed approaches are capable of capturing most of the significant ground truth events, and our approaches outperforms state-of-art approaches.

3. In Chapter 4, we propose a simple algorithm which leverages geotagged bursty term graphs to detect events from a tweet stream. We conduct an extensive empirical evaluation of our approach against the state of the art and show that our simple modifications significantly improves the detection precision and recall when compared to the state of the art approaches. Because Twitter is such a noisy domain and the Twitter API only gives samples of the tweet stream, we then focus on understanding the impact of sample size and noise level on location-based event detection. We find that our approach is more robust to these scenarios than the state of the art. To the best of our knowledge, this work is the first research to analyze the impact of sample size and noise level on event detection accuracy using Twitter.

4. In Chapter 5, we analyze potential features about ISIS supporters on Twitter, group these features into categories, and present a case study looking at the ISIS extremist group. We then propose an approach for identifying users who engage in extremist discussions online. Our approach uses detailed feature selection to identify relevant posts and then uses a novel weighted network that models the information flow between the publishers of the relevant posts. An empirical evaluation of a post collection crawled from a web forum containing racially driven discussions and a tweet stream discussing the ISIS extremist group shows that our proposed method for relevant post identification is significantly better than the state of the art and using a network flow graph for user identification leads to very accurate user identification.

This dissertation has the following directions for future work:

1. In Chapter 3, we propose an approach for detecting events from a news article corpus. It leverages the title domain mapping strategy to identify events reporting events in a specific domain. The title mapping strategy has a high Signal-to-Noise Ratio (SNR) compared to the body domain mapping strategy. However, we miss between 8% and 20% of the articles that are relevant. In future work, we will consider hybrid approaches that may lead to a reduction in the miss rate, while limiting the amount of noise added to the retained documents.

2. In Chapter 4, we propose a location-based Twitter event detection approach. It detects events based on tweets explicitly referring to location names; however, for some events, location information are so irrelevant that tweets reporting these events seldom refer to any location. Detecting events whose locations are seldom mentioned is an important direction of our future work.

3. In Chapter 5, we propose an approach for detecting users engaging in extremist discussions on Twitter. It uses detailed feature selection to identify relevant posts and the information flow between the publishers of the relevant posts. Future work will consider other types of information as we discuss in Section 5.1, especially user network profiles, and user dynamics.

Integrating Event and Extremist Signals

Our proposed news article event detection approach, Twitter event detection approach, and extremist user identification approach help identify potential leading indirect indicators of forced migration. When integrating these different components into a holistic framework for identifying possible movement, a number of issues need

to be considered and resolved. First, the time scales must be aligned. Newspaper data time scales are larger than Twitter data time scales in terms of changing dynamics. Therefore, we must begin by selecting one. While we have not yet integrating these signals, we believe that selecting the most detailed scale may be too noisy and selecting one that is daily may be reasonable for our applications. Because the newspapers signals are more reliable than the social media ones, we also believe it is important to weight them higher. Another issue will be that events may contradict each other when they are detected from different sources. To deal with this, we need to assign reliability scores to different data streams and then trust those streams with higher scores. These scores need to be verified by domain experts.

The other signal of interest is extremist conversation. This signal will potentially be higher weighted than the event signals since changes to the extremist conversation dynamics may correlate highly with events that lead to movement. We will monitor not only the extremist conversation, but also the conversations related to movement to understand how well correlated they are in different locations in the world. We expect a high correlation and will be looking for changes in conversation amounts as a variable for predicting possible movement. The event dynamics variables and the extremist conversation variables are a subset of the full set of variables we will be extracting from text data to better understand how we can use this information to determine movement.

Multi Disciplinary Consideration

We advocate approaches that include a team of researchers from multiple disciplines. Without interdisciplinary insight, it becomes difficult to (1) fully understand the problem; (2) understand the data and the gaps; and (3) analyze the data effectively. The subject matter experts understand the factors that contribute to forced migration

at the macro, meso, and micro levels, while computer scientists and statisticians understand how to mine, and analyze mass amounts of data. Our research community consists of more than 25 researchers, technicians, policymakers and humanitarian practitioners from around the world. We emphasize the multi-disciplinary component of this dissertation because an approach using a team of computer scientists that understand how to process big data but lack subject matter expertise, will likely miss important and possibly even obvious, insights that domain experts are able to spot.

Our multi-disciplinary team provides domain knowledge, including domain vocabularies, ground truth events of the targeted types, etc. These domain knowledge are irreplaceable in this dissertation. Domain vocabularies are employed to identify documents reporting events of the targeted types; ground truth events are used to evaluate the detected events. Generally, in a multi-disciplinary project, incorporating more domain knowledge will lead to a more complete solution. On the other hand, getting expertise from domain experts is not always possible - there is too much manual work if everything is done by domain experts. Our approach has been to leverage domain experts for the most critical tasks so that we can make working on large-scale problems practical and scalable.

Current Challenges For Use of Big Data for Societal Scale Issues

When working on extracting variables for prediction related to societal scale issues, e.g. forced migration, we are reminded about the importance of quality predictions and the need to qualify all of our results with confidence levels. How reliable is the result and are there biases that need to be considered? The more precise we are about the quality and confidence of the results, the more likely it is that we can reduce the potential for misinformation.

Analyses also need to be conducted in a way that preserves the privacy of individuals who share their data. To help mitigate challenges with private data, we focus on analyses using only public data. Everyone, regardless of viewpoint, deserves to have their private data protected. Without user consent, using private, sensitive data, should not be considered an option.

As a community, if we are going to make progress on these larger societal-scale problems, we need to work together - share algorithms, data sets, and testing platforms. We need to look at the problem from more than a computational perspective. Algorithms need to incorporate changing behaviors by working with social scientists who have studied extremist or migration behavior from psychological, social, and political perspectives. We will not succeed in building realistic models from these noisy, partial data sets without getting expert knowledge from those who have been studying these domains for decades.

Bibliography

[1] Eos, a vast unstructured archive of publicly available open-source media articles. https://osvpr.georgetown.edu/eos.

[2] Find frequent item sets with fp-growth algorithm. http://www.borgelt.net/doc/fpgrowth/fpgrowth.html.

[3] Twitter's new isis policy. http://www.theatlantic.com/international/archive/2016/02/twitter-isis/460269/.

[4] The ethics of fake twitter accounts. http://www.thewire.com/technology/2012/02/learning-cormac-mccarthy-twitter-hoax/48147/.

[5] Machine learning for language toolkit. http://mallet.cs.umass.edu/.

[6] Nbcnews. http://www.nbcnews.com/technology/online-news-readership-overtakes-newspapers-124383.

[7] Stanford named entity recognizor. http://nlp.stanford.edu/software/CRF-NER.shtml.

[8] The english (porter2) stemming algorithm. http://snowball.tartarus.org/algorithms/english/stemmer.html.

[9] Sentistrength. http://sentistrength.wlv.ac.uk/.

[10] Stanford corenlp sentiment analyzer. http://stanfordnlp.github.io/CoreNLP/sentiment.html, .

[11] Statoids. http://www.statoids.com, .

[12] Stromfront. https://www.stormfront.org/forum/.

[13] Trec document summarization track. http://www.trec-ts.org.

[14] Twitter api. https://dev.twitter.com/overview/documentation.

[15] Un high commissioner for refugees (2016) global trends: Forced displacement in 2015. http://www.unhcr.org/en-us/statistics/unhcrstats/576408cd7/unhcr-global-trends-2015.html?query=Global\%20trends\%202016.

[16] Vader sentiment analysis. https://github.com/cjhutto/vaderSentiment.

[17] Inma. http://www.inma.org/article/index.cfm/23899-credibility-of-online-newspapers.

[18] Spark. http://spark.apache.org/documentation.html, 2015.

[19] Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75, 2005.

[20] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6 (12):1326–1329, 2013.

[21] Howard Adelman. Difficulties in early warning: networking and conflict management. 1998.

[22] 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.

[23] Swati Agarwal and Ashish Sureka. Applying social media intelligence for predicting and identifying on-line radicalization and civil unrest oriented threats. *CoRR*, abs/1511.06858, 2015. URL `http://arxiv.org/abs/1511.06858`.

[24] Charu C Aggarwal and Karthik Subbian. Event detection in social streams. In *SDM*, volume 12, pages 624–635. SIAM, 2012.

[25] Mohammad Akbari, Xia Hu, Nie Liqiang, and Tat-Seng Chua. From tweets to wellness: Wellness event detection from twitter streams. In *AAAI*, 2016.

[26] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *SIGIR*, pages 37–45. ACM, 1998.

[27] Lucas Antiqueira, Osvaldo N. Oliveira Jr., Luciano da Fontoura Costa, and Maria das GraÃğas Volpe Nunes. A complex network approach to text summarization. *Information Sciences*, 179(5):584 – 599, 2009.

[28] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[29] Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *ACL*, pages 550–557. ACL, 1999.

[30] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. 2011.

[31] Fabrício Benevenuto, Tiago Rodrigues, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. Detecting spammers and content promoters in online video

social networks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 620–627. ACM, 2009.

[32] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[33] JM Berger and Jonathon Morgan. The isis twitter census: Defining and describing the population of isis supporters on twitter. *The Brookings Project on US Relations with the Islamic World*, 3(20), 2015.

[34] JM Berger and Bill Strathearn. Who matters online: Measuring influence, evaluating content and countering violent extremism in online social networks. 2013.

[35] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *WWW*, 2009.

[36] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[37] Martin Bouchard, Kila Joffres, and Richard Frank. Preliminary analytical considerations in designing a terrorism and extremism online network extractor. In *Computational Models of Complex Systems*, pages 171–184. Springer, 2014.

[38] Thorsten Brants, Francine Chen, and Ayman Farahat. A system for new event detection. In *SIGIR*, pages 330–337. ACM, 2003.

[39] Cody Buntain, Jennifer Golbeck, Brooke Liu, and Gary LaFree. Evaluating public response to the boston marathon bombing and other acts of terrorism through twitter. In *ICWSM*, pages 555–558, 2016.

[40] Pete Burnap, Matthew L Williams, Luke Sloan, Omer Rana, William Housley, Adam Edwards, Vincent Knight, Rob Procter, and Alex Voss. Tweeting the terror: modelling the social media reaction to the woolwich terrorist attack. *Social Network Analysis and Mining*, 4(1):1–14, 2014.

[41] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, page 4. ACM, 2010.

[42] Patricia Cavazos-Rehg, Melissa Krauss, Richard Grucza, and Laura Bierut. Characterizing the followers and tweets of a marijuana-focused twitter handle. *Journal of medical Internet research*, 16(6), 2014.

[43] Deepayan Chakrabarti and Kunal Punera. Event summarization using tweets. *ICWSM*, 11:66–73, 2011.

[44] Soumen Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *WWW*, pages 571–580. ACM, 2007.

[45] Akemi Takeoka Chatfield, Christopher G Reddick, and Uuf Brajawidagda. Tweeting propaganda, radicalization and recruitment: Islamic state supporters multi-sided twitter networks. In *Proceedings of the 16th Annual International Conference on Digital Government Research*, pages 239–249. ACM, 2015.

[46] Dawn Chatty and Nisrine Mansour. Unlocking protracted displacement: An iraqi case study. *Refugee survey quarterly*, 30(4):50–83, 2011.

[47] Feng Chen and Daniel B Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *KDD*, pages 1166–1175. ACM, 2014.

[48] Prabhakar Raghavan Christopher D. Manning and Hinrich Schutze. Introduction to information retrieval. 2008.

[49] Freddy Chong Tat Chua and Sitaram Asur. Automatic summarization of events from social media. In *ICWSM*. Citeseer, 2013.

[50] Johan Dahlin, Fredrik Johansson, Lisa Kaati, Christian Martenson, and Pontus Svenson. Combining entity matching techniques for detecting extremist behavior on discussion boards. In *ASONAM*, pages 850–857. IEEE, 2012.

[51] John L Davies and Ted Robert Gurr. Preventive measures: an overview. *Preventive Measures: Building Risk Assessment and Crisis Early Warning Systems*, pages 1–14, 1998.

[52] Munmun De Choudhury, Scott Counts, and Mary Czerwinski. Find me the right content! diversity-based sampling of social media spaces for topic-centric search. In *ICWSM*, 2011.

[53] Jure Ferlez, Christos Faloutsos, Jure Leskovec, Dunja Mladenic, and Marko Grobelnik. Monitoring network evolution using mdl. In *ICDE*, pages 1328–1330. IEEE, 2008.

[54] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192. VLDB Endowment, 2005.

[55] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

[56] Saptarshi Ghosh, Muhammad Bilal Zafar, Parantapa Bhattacharya, Naveen Sharma, Niloy Ganguly, and Krishna Gummadi. On sampling the wisdom of crowds: Random vs. expert sampling of the twitter stream. In *CIKM*, pages 1739–1744. ACM, 2013.

[57] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12): 7821–7826, 2002.

[58] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *KDD*, pages 33–42. ACM, 1999.

[59] Ernesto Gutiérrez, Ofelia Cervantes, David Báez-López, and J Alfredo Sánchez. Sentiment groups as features of a classification model using a spanish sentiment lexicon: A hybrid approach. In *Pattern Recognition*, pages 258–268. Springer, 2015.

[60] W. Chris Hale. Extremism on the world wide web: a research review. *Criminal Justice Studies*, 25(4):343–356, 2012.

[61] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[62] Georgiana Ifrim, Bichen Shi, and Igor Brigadir. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM, 2014.

[63] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics, 2011.

[64] Timothy Jones, David Hawking, Paul Thomas, and Ramesh Sankaranarayana. Relative effect of spam and irrelevant documents on user interaction with search engines. In *CIKM*, pages 2113–2116. ACM, 2011.

[65] Theodoros Lappas, Benjamin Arai, Manolis Platakis, Dimitrios Kotsakos, and Dimitrios Gunopulos. On burstiness-aware search for document sequences. In *KDD*, pages 477–486. ACM, 2009.

[66] Theodoros Lappas, Marcos R Vieira, Dimitrios Gunopulos, and Vassilis J Tsotras. On the spatiotemporal burstiness of terms. In *VLDB*, pages 836–847, 2012.

[67] Junsup Lee, Sungdeok Cha, Dongkun Lee, and Hyungkyu Lee. Classification of web robots: An empirical study based on over one billion requests. *computers & security*, 28(8):795–802, 2009.

[68] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[69] Sangho Lee and Jong Kim. Warningbird: Detecting suspicious urls in twitter stream. In *NDSS*, 2012.

[70] KH Lei, R Khadiwala, and KC-C TEDAS Chang. A twitter-based event detection and analysis system. In *ICDE*, pages 1273–1276, 2012.

[71] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506. ACM, 2009.

[72] Chengtao Li, Jianwen Zhang, Jian-Tao Sun, and Zheng Chen. Sentiment topic model with decomposed prior. In *SIAM International Conference on Data Mining (SDM 2013). Society for Industrial and Applied Mathematics*. SIAM, 2013.

[73] Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. In *CIKM*, pages 155–164. ACM, 2012.

[74] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. Pet: a statistical model for popular events tracking in social communities. In *KDD*, pages 929–938. ACM, 2010.

[75] Tim Lister, Ray Sanchez, Mark Bixler, Sean O'Key, Michael Hogenmiller, and Mohammed Tawfeeq. Isis goes global, 2017. URL `http://www.cnn.com/2015/12/17/world/mapping-isis-attacks-around-the-world/`.

[76] Susan Martin and Lisa Singh. Data analytics and displacement: Using big data to forecast mass movements of people. In *Digital Lifeline? ICTs for Refugees and the Displaced*. MIT Press, 2018.

[77] Susan F Martin, Sanjula Weerasinghe, and Abbie Taylor. *Humanitarian Crises and Migration: Causes, Consequences and Responses*. Routledge, 2014.

[78] Michael Mccord and M Chuah. Spam detection on twitter using traditional classifiers. In *Autonomic and trusted computing*, pages 175–186. Springer, 2011.

[79] Joseph Mei and Richard Frank. Sentiment crawling: Extremist content collection through a sentiment analysis guided web-crawler. In *ASONAM*, pages 1024–1027. ACM, 2015.

[80] Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *ICWSM*, pages 248–257, 2015.

[81] Rada Mihalcea and Paul Tarau. A language independent algorithm for single and multiple document summarization. In *ICJNLP*, 2005.

[82] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[83] Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *arXiv preprint arXiv:1605.01655*, 2016.

[84] Sathappan Muthiah, Bert Huang, Jaime Arredondo, David Mares, Lise Getoor, Graham Katz, and Naren Ramakrishnan. Planned protest modeling in news and social media. In *AAAI*, pages 3920–3927, 2015.

[85] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

[86] Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. Summarizing sporting events using twitter. In *IUI*, pages 189–198. ACM, 2012.

[87] Yoko Nishihara, Keita Sato, and Wataru Sunayama. Event extraction and visualization for obtaining personal experiences from blogs. In *Human Interface and the Management of Information. Information and Interaction*, pages 315–324. Springer, 2009.

[88] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[89] Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, et al. 'beating the news' with embers: Forecasting civil unrest using open source indicators. In *KDD*, pages 1799–1808. ACM, 2014.

[90] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *KDD*, pages 1104–1112. ACM, 2012.

[91] Matthew Rowe and Hassan Saif. Mining pro-isis radicalisation signals from social media users. In *ICWSM*, pages 329–338, 2016.

[92] Horacio Saggion, Simone Teufel, Dragomir Radev, and Wai Lam. Meta-evaluation of summaries in a cross-lingual environment using content-based metrics. In *COLING*, pages 1–7. Association for Computational Linguistics, 2002.

[93] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52(1):5–19, 2016.

[94] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860. ACM, 2010.

[95] Joseph Sassoon. *The Iraqi Refugees: The New Crisis in the Middle-East*, volume 3. IB Tauris, 2008.

[96] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *ICWSM*, 2009.

[97] Ryan Scrivens, Garth Davies, Richard Frank, and Joseph Mei. Sentiment-based identification of radical authors. In *ICDMW*, pages 979–986. IEEE, 2015.

[98] Surendra Sedhai and Aixin Sun. Effect of spam on hashtag recommendation for tweets. In *WWW*, pages 97–98. International World Wide Web Conferences Steering Committee, 2016.

[99] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *KDD*, pages 623–632. ACM, 2010.

[100] Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. Document summarization using conditional random fields. In *IJCAI*, volume 7, pages 2862–2867, 2007.

[101] Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in twitter using sender-receiver relationship. In *Recent Advances in Intrusion Detection*, pages 301–317. Springer, 2011.

[102] Grant Stafford and Louis Lei Yu. An evaluation of the effect of spam on twitter trending topics. In *SocialCom*, pages 373–378. IEEE, 2013.

[103] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

[104] Dawei Wang and Wei Ding. A hierarchical pattern learning framework for forecasting extreme weather events. In *ICDM*, pages 1021–1026. IEEE, 2015.

[105] Jingjing Wang, Wenzhu Tong, Hongkun Yu, Min Li, Xiuli Ma, Haoyan Cai, Tim Hanratty, and Jiawei Han. Mining multi-aspect reflection of news events in twitter: Discovery, linking and presentation. In *ICDM*, pages 429–438. IEEE, 2015.

[106] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, pages 784–793. ACM, 2007.

[107] Yifang Wei and Lisa Singh. Location-based event detection using geotagged semantic graphs. In *KDD Workshop Mining and Learning with Graphs*, 2017.

[108] Yifang Wei and Lisa Singh. Detecting users who share extremist content on twitter. In *Surveillance in Action: Technologies for Civilian, Military and Cyber Surveillance*. Springer, 2017.

[109] Yifang Wei and Lisa Singh. Understanding the impact of sampling and noise on detecting events using twitter. In *The International ACM Conference on Web Search and Data Mining*. Under Review, 2017.

[110] Yifang Wei and Lisa Singh. Using network flows to identify users sharing extremist content on social media. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 330–342. Springer, 2017.

[111] Yifang Wei, Abbie Taylor, N Yossinger, E Singewood, D Quinn, S Martin, S McGrath, J Collman, Sidney Berkowitz, and Lisa Singh. Using large-scale open source data to identify potential forced migration. In *KDD Workshop Data Science for Social Good*, 2014.

[112] Yifang Wei, Lisa Singh, Brian Gallagher, and David Buttler. Overlapping target event and story line detection of online newspaper articles. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 222–232. IEEE, 2016.

[113] Yifang Wei, Lisa Singh, and Susan Martin. Identification of extremism on twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 1251–1255. IEEE, 2016.

[114] Yifang Wei, Lisa Singh, David Buttler, and Brian Gallagher. Using semantic graphs to detect overlapping target events and story lines from newspaper articles. In *International Journal of Data Science and Analytics*. To Appear, 2017.

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

[116] Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. Topicsketch: Real-time bursty topic detection from twitter. In *ICDM*, pages 837–846. IEEE, 2013.

[117] Feiyu Xu, Hans Uszkoreit, and Hong Li. Automatic event and relation detection with seeds of varying complexity. In *AAAI workshop event extraction and synthesis*, pages 12–17, 2006.

[118] Wei Xu, Fangfang Zhang, and Sencun Zhu. Toward worm detection in online social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 11–20. ACM, 2010.

[119] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36. ACM, 1998.

[120] Andrew Yates, Alek Kolcz, Nazli Goharian, and Ophir Frieder. Effects of sampling on twitter trend detection. In *LREC*, 2016.

[121] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *SIGIR*, pages 513–522. ACM, 2016.

[122] Zhenjie Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, and Divesh Srivastava. Bed-tree: an all-purpose index structure for string similarity search based on edit distance. In *SIGMOD*, pages 915–926. ACM, 2010.

[123] Xiangmin Zhou and Lei Chen. Event detection over twitter social media streams. *The VLDB journal*, 23(3):381–400, 2014.

[124] Yilu Zhou, Edna Reid, Jialun Qin, Hsinchun Chen, and Guanpi Lai. Us domestic extremist groups on the web: link and content analysis. *IEEE intelligent systems*, 20(5):44–51, 2005.