Spotting Icebergs by the Tips: Rumor and Persuasion Campaign Detection in Social Media

by

Zhe Zhao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2017

Doctoral Committee:

Associate Professor Qiaozhu Mei, Chair
Associate Professor Michael J. Cafarella
Research Scientist Ed H. Chi, Google
Associate Professor Rada Mihalcea
Professor Paul Resnick

Zhe Zhao

zhezhao@umich.edu

ORCID iD: 0000-0002-6847-0186

To my PhD friends.

ACKNOWLEDGEMENTS

My PhD journey have begun since September 2011, and I would not be able to complete it without all the help and guidance from my dear advisor, Professor Qiaozhu Mei. Every step of my growth as a PhD student reflects his great caring and influences. From his insightful vision and research style, I learn how to think systematically, how to act effectively, and how to conclude precisely not only for conducing world-class research but also for solving real-life problems.

I would like to sincerely thank my committee members: Michael J. Cafarella, Ed H. Chi, Rada Mihalcea, and Paul Resnick, who spent their valuable time on my dissertation. The influences of their feedbacks are significant to my growth. This dissertation would not be possible without their careful reviews and valuable comments. I want to thank Yunliang Jiang and Lichan Hong, who provided great mentorships when I interned at Twitter and Google. They helped me understand the industrial standard of conducting large scale data analysis and buidling scalable machine learning algorithms. I also want to thank my collaborators at Google, they are Zhiyuan Cheng, Peng Dai, Jilin Chen, Sagar Jain, Shuo Chang, Alex Beutel and Xinyang Yi. By working with them, I learned to conduct effective research to impact the real product. I would like to thank my collaborators at the Indiana University, including Filippo Menczer and Alessandro Flammini, etc. They inspired me on developing the rumor and persuasion campaign detection systems discussed in this dissertation.

It is a great honor to work with all these fantastic people at the University of Michigan. I want to express my gratitude to all the past and current Michigan

Foreseer research group members: Lei Yang, Yang Liu, Xin Rong, Xiaodan Zhou, Yue Wang, Cheng Li, Wei Ai, Sam Carton, Tera Reynolds, Tao Sun, Jian Tang, Danny Tzu-Yu Wu, and V.G.Vinod Vydiswaran. The valuable discussions with them and their help are essential for my success. I want to mention that I enjoined my five years of PhD life in Ann Arbor knowing and learning from my friend Xin Rong, who always has a high standard in conducting research and has an active attitude in making progress. I will never forget our shared happiness and sadness, and wish one day I could be a more active person like him.

I am fortunate to have many friends in the School of Information and the department of Computer Science and Engineering. Zhe Chen, Li Qian, Lujun Fang, Huan Feng, Fei Li, Young Park, Xiaoxiao Guo, Ning Jiang, Xin Fan and many others from CSE have provided me great help and shared many experiences including taking all the required classes and discussing each other's research topics, etc. Shiyan Yan, Houyang Hou, Tao Dong, Xuan Zhao, Teng Ye, Sangseok You, Daphne Chang, Priyank Chandra, Wenjing Xu, Yunchen Shen, Jingzhu Yan, Xiaochen Li, and many others from SI make me feel included as a member of the UMSI family. I want to thank all of them for the wonderful memories including hotpot parties, off-sites, and writing groups, etc. They make my PhD life colorful.

I would also like to thank people who I met before my Ph.D. and inspired me all the way through this process. I thank Junjie Yao, who gave suggestions on many of my life decision, including the pursuing of a PhD. I also thank Bin Cui who was my advisor in Peking University and have taught me the meaning of doing research while I was an computer science undergraduate.

Finally, I would like to express my deepest gratitude to my parents and my family. I thank my parents Ping Zhao and Jingyun Zhao for the continuous understanding and respecting on any decision I made. I wouldn't go anywhere close to where I am right now without their support along the way.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Spotting Icebergs by the Tips:
Rumor and Persuasion Campaign Detection in Social Media

by

Zhe Zhao

Chair: Qiaozhu Mei

Identifying different types of events in social media, i.e., collective online activities or posts, is critical for researchers who study data mining and online communication. However, the online activities of more than one billion social media users from around the world constitute an ocean of data that is hard to study and understand. In this dissertation, we study the problem of event detection with a focus on two important applications—rumor and persuasion campaign detection.

Detecting events such as rumors and persuasion campaigns is particularly important for social media users and researchers. Events in social media spread and influence people much more quickly than traditional news media reporting. Viral spreading of specific events, such as rumors and persuasion campaigns, can cause substantial damage in online communities. Automatic detection of these can benefit analysts in many different research domains.

Existing work on detecting rumors and persuasion campaigns faces challenges stemming from the uncommonness of these events compared to other events, the uncertain ways in which they spread, and the sheer magnitude of social media user

activities. Presently, there are two main approaches to event detection. One approach first identifies clusters of similar user activities or posts as social media events and then applies a classifier to these to identify specific types of events, such as rumors or persuasion campaigns. This approach typically incurs a high computational cost, since all clusters must be identified before the irrelevant ones are filtered out. The other approach first trains a classifier to identify rumor/persuasion campaign posts and then groups these posts into clusters. While this approach avoids clustering irrelevant posts, it cannot identify rumors and individual persuasion campaign posts accurately due to the absence of collective information.

In this thesis, we extend the existing research on social media event detection of online events such as rumors and persuasion campaigns. We conducted content analysis and found that the emergence and spreading of certain types of online events often result in similar user reactions. For example, some users will react to the spreading of a rumor by questioning its truth, even though most posts will not explicitly question it. These explicit questions serve as signals for detecting the underlying events. Our approach to detecting a given type of event first identifies the signals from the myriad of posts in the data corpus. We then use these signals to find the rest of the targeted events. Different types of events have different signals. As case studies, we analyze and identify the signals for rumors and persuasion campaigns, and we apply our proposed framework to detect these two types of events.

We began by analyzing large-scale online activities in order to understand the relation between events and their signals. We focused on detecting and analyzing users' question-asking activities. We found that many social media users react to popular and fast-emerging memes by explicitly asking questions. Compared to other user activities, these questions are more likely to be correlated to bursty events and emergent information needs.

We use some of our findings to detect trending rumors. We find that in the case of

rumors, a common reaction regardless of the content of the rumor is to question the truth of the statement. We use these questioning activities as signals for detecting rumors. Our experimental results show that our rumor detector can effectively and efficiently detect social media rumors at an early stage.

As in the case of rumors, the emergence and spreading of persuasion campaigns can result in similar reactions from the online audience. However, the explicit signals for detecting persuasion campaigns are not clearly understood and are difficult to label. We propose an algorithm that automatically learns these signals from data, by maximizing an objective that considers their key properties. We then use the learned signals in our proposed framework for detecting persuasion campaigns in social media. In our evaluation, we find that the learned signals can improve the performance of persuasion campaign detection compared to frameworks that use signals generated by alternative methods as well as those that do not use signals.

CHAPTER I

Introduction

By using online social media platforms such as Facebook and Twitter, anyone can create and share information through the internet. Social media users include college students, middle-class workers, scholars, celebrities, advertisers, politicians, and even terrorists. As online social media platforms have become a major information source, their impacts on areas such as commerce and politics as well as their trustworthiness have raised people's concern.

Many companies, organizations, and governments track and monitor different types of social media events and trending topics, so that whenever an event that is related to their interests emerges, they can quickly react to maximize its positive effects and minimize its negative effects. Social media events reflect social media users' collective activities, such as posting and sharing a meme, discussing the same piece of breaking news, and so forth. With different topics and levels of popularity, some social media events are much more influential than others and are thus more meaningful to track and monitor.

The emergence and spreading of rumors and of persuasion campaigns are two types of social media events that can be extremely influential and sometimes harmful. Figure 1.1 shows an example of a social media rumor: a tweet posted by the Associated Press's (AP) official Twitter account after it was hacked. The tweet reported two

Figure 1.1: A fake tweet posted by the Associated Press's hacked Twitter account

Figure 1.2: Examples of persuasion campaigns on social media platforms

explosions in the White House and stated that the President was injured. Even though the account was quickly suspended, the rumor spread to millions of users. In such a special context, the rumor raised an immediate panic, which resulted in a dramatic, although brief, stock market crash [25].

Figure 1.2[1] shows examples of social media persuasion campaigns supporting or opposing terrorist groups. Millions of people can be exposed to and affected by viral

[1]Image sources: http://www.tomcotton.com/2015/02/tell-obama-stop-isis/, www.givebackourfreedom.com, and http://www.marieclaire.co.uk/news/546482/now-this-is-how-you-use-the-cannes-red-carpet-to-do-good-1.html

persuasion. In May 2015, roughly 30,000 foreign fighters were fighting for the Islamic State of Iraq and Syria (ISIS), a group of military jihadists in the Middle East. The foreign fighters came from more than 100 countries around the world, including Australia, Canada, and the US.[2] Behind this is ISIS's successful persuasion campaign on social media. With more than 90,000 Twitter accounts around the world, a very large amount of propaganda content for ISIS circulates much more quickly than it can be identified and deleted by Twitter or other officials.[3] In the absence of effective methods for identifying and removing ISIS's persuasion campaigns, the number of social media users who turn into fighters for ISIS is continually increasing.

It is impossible to manually identify all of these events, given that thousands of posts[4] and images[5] and hundreds of hours of videos[6] are created every second on different social media platforms. Automatic rumor and persuasion campaign detection systems must be able to go through this huge amount of social media content, filtering out irrelevant content and events, in order to accurately identify potential rumors and persuasion campaigns.

Researchers working on social media event detection, especially rumor and persuasion campaign detection, have made promising progress, with many prototype systems having been built at different granularities. In rumor detection, for example, a classifier trained using posts of a specific rumor is able to identify more posts of that rumor, e.g., two explosions in the White House, based on each individual post's content, author, and other context features [86]. A similar classifier has been trained and used to identify rumorous posts about a specific topic, e.g., the Boston Marathon [43]. Then the identified posts related to that topic can be further analyzed and grouped into different rumor statements, e.g., fake news that $1 will be donated to

[2] http://www.iraqinews.com/arab-world-news/isis-30000-foreign-fighters-100-countries
[3] http://www.cbsnews.com/news/why-so-difficult-counter-isis-social-media
[4] http://www.internetlivestats.com/twitter-statistics/
[5] http://blog.wishpond.com/post/115675435109/40-up-to-date-facebook-facts-and-stats
[6] https://fortunelords.com/youtube-statistics/

victims per retweet and rumors about suspects. However, these approaches will not be effective for detecting new rumors that are intentionally crafted to avoid some of the significant features used by the classifier. Moreover, a single classifier cannot be used to detect rumors about general unspecified topics, where the characteristic features of individual posts associated with different rumors can be entirely different.

To build systems that can detect rumors or persuasion campaigns in general, without training different classifiers for different topics or content, many existing approaches treat their emergence and spreading as social media events that collectively involve multiple posts and reposts by users. To this end, many recent event detection systems for rumors or persuasion campaigns train classifiers on groups of posts (generated by different types of user activities, such as posting and sharing). These systems can therefore use the collective features of each group of posts, e.g., statistics such as the number of shares or the temporal patterns of spreading, to determine whether that group of posts corresponds to a social media rumor/persuasion campaign. As a preprocessing step, these systems rely on general event detection algorithms to cluster posts into groups based on their content, topics, and other features such as network features. Each cluster or group corresponds to the emergence and spreading of one social media event, such as a popular meme, trending topics, or breaking news. Since our detection targets in this dissertation are specific types of events—i.e., rumors and persuasion campaigns—which form a small subset of events in general [123], it would be time consuming and inefficient to cluster all posts in this preprocessing step. Moreover, the performance of the classification can suffer from imbalanced positive and negative examples during training. Additionally, some of the most helpful features for classification—such as the time series of a number of posts or network features—can only be determined after a cluster is large enough, or after outbursting. It would be difficult to identify these at an early stage, which is a potentially critical stage for preventing viral spreading.

One way to address these challenges is to efficiently filter out the majority of irrelevant content before clustering posts into groups and extracting their collective features. This will improve both the effectiveness and the efficiency of automatic event detection systems that aim to identify only a small subset of all events from a large-scale social media post stream. In this dissertation, we build such a system through understanding the key signals for different types of events and discovering filtering strategies based on these. Specifically, we build two systems, one to detect rumors and one to detect persuasion campaigns. Understanding some of the similar properties of these two types of events is the key for building our systems and identifying features for filtering. The single general framework we used to build the two systems can also be adapted to detect other types of social media events with similar properties.

What properties of social media events such as rumors and persuasion campaigns can be used to filter out most irrelevant posts but retain the important ones for detecting these events? In general, the properties should be unrelated to the content and topic of each rumor or persuasion campaign, but every rumor or persuasion campaign should have these. Those properties should be capable of being effectively and efficiently spotted in some posts, but not necessarily all, that are involved in each rumor or persuasion campaign. One such property for our given types of targeted events, i.e., rumors or persuasion campaigns, is that their emergence and spreading can lead to certain common user reactions. The posts that explicitly reflect those reactions are signals of the targeted events we want to detect. By filtering out nonsignals, we are left with a subset of posts that contains at least some posts for each targeted event. This subset will later be used in our generic framework to extract and detect targeted events.

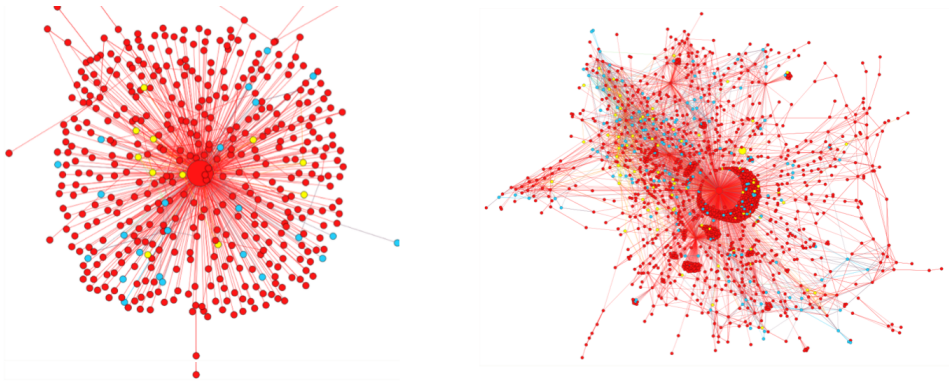
Turning to social media rumors for an example, a content analysis showed that a common user reaction after exposure to a rumorous post is to express skepticism about its truth, due to its controversial content. A rumor in social media usually

Table 1.1: Examples of signal tweets regarding the rumor of explosions in the White House

Oh my god is this real? RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
Is this true? Or hacked account? RT @AP Breaking: Two Explosions in the White House and Barack Obama is injured
Is this real or hacked? RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
How does this happen? #hackers RT @user: RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
Is this legit? RT @AP Breaking: Two Explosions in the White House and Barack Obama is injured
WHAT????!!!! RT @AP Breaking: Two Explosions in the White House and Barack Obama is injured

starts with a post of an unverified claim that purports to be fact. All of the user activities that involve talking about and spreading that rumor form a group of posts. Even though not all users will be skeptical or explicitly express their skepticism in their posts, there are some who will. Figure 1.3 depicts how the two-explosion rumor posted by the hacked AP account (see Figure 1.1) spread in Twitter. In the two graphs, each node represents a tweet. Different colors represent different types of user reactions: retweeting, questioning, and debunking. We can see that most of the tweets are spreading the rumor by retweeting (red nodes). However, some of the tweets are questioning the truth of the rumor and correcting the original post while spreading it. These behaviors can be observed during the first 60 seconds after this rumor emerged. Some examples of the tweets questioning the rumor are shown in Table 1.1. These enquiry posts, which explicitly express users' skeptical reactions, are the signals we use to build our rumor detection system. They exist for almost all social media rumors regardless of the actual rumorous content, and they are easy to identify by their language patterns. Like rumors, social media persuasion campaigns also result in common user reactions such as explicitly supporting or protesting against the campaign, regardless of its goal.

Hypothetically, if those signals were entirely absent in non-targeted events, a set



(a) 60 seconds after the hacked Associated Press (AP) Twitter account tweeted the White House rumor

(b) 134 seconds after the rumor (two seconds after the first AP employee debunked the rumor using his personal Twitter account, and two minutes before the official denial came from the AP)

Figure 1.3: Snapshots of the diffusion of the White House rumor. Red, yellow, and blue nodes: users spreading, correcting, and questioning the rumor.

of signals would be a subset of all posts of the targeted events. Then we could identify the content of targeted events by analyzing the content of the signals and, if the signals are reposts or responses, the content of the origin posts. We could also retrieve the nonsignal posts for an event by using that content, so that further analysis and monitoring could be conducted. Realistically, however, since signals are explicit expressions of users' common reactions, the more easily they can be detected and the more common the reactions are, the less exclusive these signals will be. For example, enquiry posts exist not only in rumors but can also appear in connection with any breaking news. To further remove false positives, we build a generic framework for targeted event detection with signal filtering.

As shown in Figure 1.4, the input of the generic framework is a steam of social media posts. The output is a set of detected events. Each detected event is in the form of a group of posts related to that event. In general, the framework consists of the following three steps:

- 1. Signal filtering: In this step, all posts generated by user activities such as posting, sharing, or responding are checked and separated into two groups,

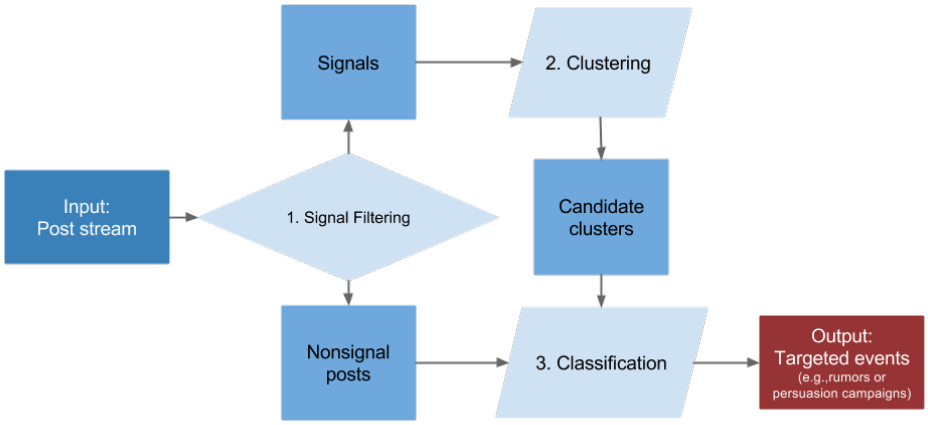


Figure 1.4: A general framework for detecting targeted social media events

signals and nonsignal posts.

- 2. Clustering: In this step, signals are clustered based on a given similarity measure, e.g., content similarity. Thus, signals from the same targeted event are grouped into one cluster. These clusters are candidate event clusters.

- 3. Classification: This step checks whether a candidate cluster is an actual targeted event. Nonsignal posts are matched against and grouped with signals into candidate clusters as appropriate. Then the collective features of each candidate cluster are extracted so that classification algorithms can evaluate how likely it is that a candidate is a target.

Similar to existing general rumor and persuasion campaign detection systems, we train classifiers with the collective features of groups of posts to determine whether a group corresponds to a targeted event. By using a signal filtering step before clustering, we can filter out a large proportion of the posts from the post stream. Only those posts that are likely to be related to targeted events are used to generate candidate clusters, and the candidate clusters will therefore have fewer irrelevant events than clusters generated from the entire post stream. Thus, because our approach avoids clustering all posts, our general framework for rumor or persuasion campaign detection reduces the computational cost as well as the difficulty of event classification.

Additionally, as signals are reflections made in user reactions, it will be difficult for those who create rumors and persuasion campaigns to intentionally avoid detection. Finally, signals can usually be captured at any stage of an event rather than only after it has gone viral. This is important for early detection of targeted events.

As our rumor detection and persuasion campaign detection systems share the same generic framework, the most critical step—that of finding good signals for each of these two types of events—depends on their individuating characteristics. We begin our work in this dissertation by conducting a content analysis of posts on social media to understand what the common user reactions to different types of events are, as well as how users' individual activities or posts can explicitly reflect such reactions. Specifically, we focus on tracking and analyzing question-asking behaviors on social media. Because the different types of explicit questions that users ask on social media reflect their information needs, these are reactions to different types of social media events, for example, rumors.

Among users who have information needs, many will choose to read posts by others, discuss the details, and search with search engines. However, a few users will post questions directly asking for the needed piece of information. By training a classifier to automatically detect posts on Twitter that are questions, we extracted a large collection of question posts. We grouped these questions according to different keywords and checked their correlations with different types of social media events. As questions are explicit expressions of emerging information needs, we found a high correlation between the question topics and the topics of trending and bursting social media events.

Based on our analysis, we found that a specific type of questions, namely, confirmation questions or enquiries, is related to social media rumors. As shown in previous examples, we use these enquiry activities as signals to build our rumor detection system. Each rumor is detected in the form of a group/cluster of posts that are related

to a disputed factual claim. Again, the key insight is that when there is a rumor, even though most posts will not raise questions about it, there will likely be a few that do. By conducting content analysis on a labeled rumor tweet dataset, we found some signature text phrases that are used by some individuals to express skepticism about whether a claim is factual and that are less likely to be used to express anything else. These text phrases include: "Is this true?", "Really?", and "What?". Although relatively few individual posts related to any particular rumor use any of these enquiry phrases, many rumor clusters include some posts that do use these phrases, and such posts appear quite early in the diffusion process.

We then applied this finding in our general event detection framework. We first search for posts with the enquiry phrases to identify signals, then we cluster them by comparing content similarity after removing the enquiry phrases and collecting related nonsignal posts for each cluster, and finally, we use a classifier to rank the clusters by the likelihood that they truly contain a disputed factual claim. Our experiment shows that, by searching for the very rare but very informative signal phrases and using clustering and a classifier on the clusters, our rumor detector has a surprisingly good performance.

The enquiry signals that we used for rumor detection have unique language patterns and phrases that are easy to identify. However, signals for detecting persuasion campaigns and other targeted events in general may not contain such unique and simple patterns. Possible signals might contain many patterns and combinations of these. It is important to have an automatic or semiautomatic approach to find ways to identify good signals, as good signals can significantly improve the effectiveness and efficiency of detection while improper signals can make a detection system worse by decreasing both precision and recall. Therefore, in our next study, we first developed an automatic way to identify potential patterns that could be used to filter out nonsignal posts in our persuasion campaign detection system.

Above, we have generally discussed the characteristics of signals in targeted event detection and have shown how enquiry signals are used in rumor detection, as an example. By quantifying some key characteristics of signals, such as their uniqueness in targeted events, we built metrics to evaluate how a given signal identification algorithm performs with respect to each characteristic. By jointly maximizing scores on these metrics on labeled targeted events, we could automatically learn a signal identifier in the form of a binary post-classifier using a set of patterns as features. We trained this classifier on labeled persuasion campaigns and used it in our persuasion campaign detection system.

Persuasion campaigns are series of activities that attempt to gain public support for an opinion or a course of action [83]. The motivations behind persuasion campaigns in social media can be either legitimate or unethical and manipulative—illustrated, for example, by Michelle Obama's campaign to gain support in opposition to Boko Haram and ISIS's recruiting campaign, respectively (shown in Figure 1.2). To detect persuasion campaigns in a large number of posts on social media, we first developed a definition and a codebook for social media persuasion campaigns. Then we manually labeled a set of persuasion campaigns in the form of clusters of posts. We used this labeled dataset to train a signal identification algorithm. Finally, we applied the algorithm to filter out nonsignal posts and build the persuasion campaign detection system. Experimental results based on Twitter show that our detection system is both effective and efficient compared to baseline methods.

The rest of this dissertation is organized as follows. We begin with a brief review of existing approaches to event detection and detection systems built for detecting different types of targeted events in Chapter II. We summarize these approaches based on their methods and discuss whether and how they can be generalized as a framework for detecting different types of targeted events. In Chapter III, we introduce and discuss our general framework for detecting social media rumors, persuasion

campaigns, and other targeted events where signals can be identified.

In Chapter IV, we conduct content analysis on different types of social media posts and user activities, especially question-asking activities. We demonstrate that question posts are indicators for identifying bursting events and emerging information needs in general and that some of them can be used as signals for rumor detection. In Chapter V, we discuss application of our general framework to the problem of detecting social media rumors, using enquiries as signals. In Chapter VI, we quantify the characteristics of signals as objectives for a good signal identification learning algorithm. Then we apply this algorithm in our general framework to detect social media persuasion campaigns. Finally, we summarize our findings and discuss future research directions in Chapter VII.

CHAPTER II

A Review of Social Media Event Detection

Social media data mining tasks target at extracting meaningful information from large scale of social media content. As a sub-category of these tasks, social media event detection targets at detecting collective information from social media users. We list some of the event detection tasks and show their common characteristics despite of specific event type each of them want to detection.

From the characteristics, many existing approaches adopts similar general frameworks. These inspires us to develop a detection framework for detecting social media rumors and persuasion campaigns. We briefly discuss the frameworks and provide a high level summary in this chapter. We will review existing work in rumor detection and persuasion campaign detection respectively in Chapter V and Chapter VI.

2.1 Event Detection Tasks in Social Media

Over the past ten years, social media research has expanded to make use of the massive amounts of data generated by online social media and the concomitant activity by online users. It is now possible for researchers to make use of the large amounts of user activities and detect online events/phenomena, such as online communities [80], spams [48] [108] and trending topics [71]. The close relationship between users' online activities and their real world activities also allows researchers to use detected

online phenomena in detecting or predicting real world events, e.g., trend in stock market [11], happening of earthquakes [92] and epidemics [38] [57], etc.

2.1.1 Common Characteristics of Social Media Event Detection Systems

Event detection in online social media has been studied for many years [4]. Most studies focus on methods to extract social media events from clustering user activities or posts. Each social media event is represented by a cluster of activities or posts. The grouping criteria or similarity measure between activities or posts used to develop a clustering algorithm correspond to the definitions of social media events [93]. Commonly used criteria include similarity of keywords [71], [53], and memes [61].

Following this formulation of social media event detection, various types of clustering algorithms and topical modeling analyses have been developed [51]. Stream clustering algorithms can be used to cluster input social media post streams or user activity streams [93] [2]. After the clustering algorithm, user activities or posts with similar content and topic are grouped into one cluster. Each cluster corresponds to the emerging and spreading of a particular social media event.

Many studies concentrate on a particular type of online social media event. Research topics include ongoing bursting events [9], trending social media memes [31] [61], potential bursting events [53] [54], breaking news [84], social media rumors and misinformations [100] [101], and persuasion campaigns [59]. Event detection systems treat specific type of event as targeted event usually apply classifiers to identify them from other events.

2.1.2 Types of Targeted Events

Trending event detection is one of the most representative social media event detection tasks. The relevant studies aim to detect events that are popular or will be popular, on social media platforms such as Twitter [71] [10]. The research groups

social media posts into clusters and uses features of the clusters such as the cluster size or the trend to identify a subset of popular clusters. Clustering large amounts of online posts and online activities is both time- and space-consuming, however. Thus, detecting trends in real time depends on developing efficient algorithms and sufficient computational resources [107].

Besides trending events, event detection systems target at detecting various types of events. Community detection studies look at the clusters of users having direct or close relationships [80]. Spam detection algorithms identify various unwanted and malicious spammer or hacker activities [48] [108]. Other topics include detecting real world events based on users' online social media activities. The sentiments expressed in social media posts have proven useful for detecting stock market trends [11]. For one event, a study finds that users' online social media posts on an earthquake are faster than official sources [92]. Another study suggests that the aggregate features of social media content can be used to detect epidemic outbursts [57].

2.2 General Framework for Detecting Targeted Events

Most of the above mentioned existing works treat detecting specific types of social media events as a classification problem and adopt classifier on clusters of posts or activities. Some of them treat clusters as inputs for training and evaluating their classifiers without discussing how the clusters are generated. Others build end to end detection systems by first applying clustering algorithms on posts or user activities before the classification task. In general, they follows a clustering plus classification framework for event detection.

Based on a predefined similarity measurement, the clustering plus classification framework first groups (clusters) similar online users' activities (posts). Each cluster represents the collective activities corresponding to a social media event. Then they classify each cluster as positive (targeted event) or negative (other event) [4] [99] [88]

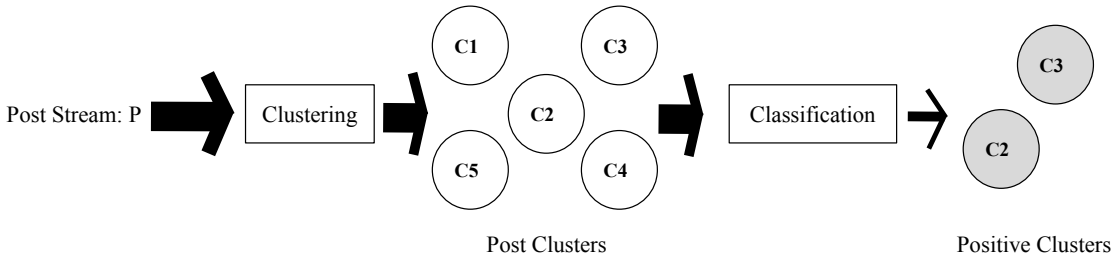


Figure 2.1: Clustering + Classification Framework

[89]. For example, posts can be grouped by hashtags and the classifier uses features such as the times series of the cluster size to detect and predict any trending hashtags [61] [17]. And by clustering posts using content similarity, trending statements can be detected [107]. Figure 2.1 shows the workflow of this framework.

By adopting this framework, different types of targeted events, such as rumors or controversial statements [101] has been studied. In existing rumor detection systems [56] [43], classifiers are trained using different types of collective features extracted from post clusters.

The classifiers can be trained to identify targeted events and work well on a balanced dataset where targeted events are common, such as memes, or bursting events, etc. When used for detecting relatively rare events, such as rumors and persuasion campaigns, however, the framework can be inadequate. This is because the framework clusters all posts. When targeted events are relatively rare, it is inefficient to cluster all of them when only a few clusters are the detection targets. Moreover, as there are more negative clusters than positive ones, a classifier trained on a limited and unbalanced dataset with a different distribution of positives and negatives can overfit and predict inaccurate labels.

Is it possible to flip the classification and clustering process, so that we first classify posts to decide whether it belongs to a target cluster or not, then cluster only the positive individual posts into clusters? By doing this, we can filter out many negative

activities and improve the efficiency in clustering.

Because social media events correspond to clusters of posts, the label is assigned to a cluster instead of a individual post. The classification task on individual post cannot yield a good performance by simply applying the cluster label to its posts. For example, it is unclear and hard to train a classifier on a single post to decide whether it is a general persuasion campaign or not. Moreover, most important features reported by existing event detection systems are collective features that can only be extracted at cluster level, such as network features, aggregated content features and statistics [27].

A post level classifier is only sufficient to detect whether or not a post is related to a specific targeted event such as a specific social media rumor, due to the limitation of post level feature [86]. A recent work studying general rumor detection trains classifier to directly identify whether a individual post belongs to a specific type of rumors or not [46] . A tweet level rumor classifier is trained and tested on a labeled dataset. However, it is not scalable to identify general social media rumors on real social media post stream, where rumors have various topics. In general, classifying whether a post belongs to any rumor cluster will need more features than features from the post itself.

Although detecting targeted events with a post level classifier is hard, it inspires us to discover properties of individual post that are easy to identify and related to target clusters. In our framework, we use posts that are signals to generate clusters and then apply cluster level classifier to detect targeted events. Our approach is a combination of the above mentioned two lines of existing works.

As discussed in Chapter I, our general framework for rumor and persuasion campaign detection extends the clustering + classification approach by adding a signal filtering step. It makes use of some indicating features from individual posts, i.s., signals, before clustering. A proper selected signal identification algorithm can improve

both the efficiency and effectiveness of targeted event detection. In the next chapter, we will provide high level description of our proposed framework. And we will apply this framework to build our rumor detection system and persuasion detection system respectively.

CHAPTER III

A General Framework to Detect Targeted Events

In this chapter, we introduce the general framework for building rumor and persuasion campaign detection systems. We discuss our system's input and output and its three steps for detection: signal filtering, clustering, and classification. As signals for rumors are different from signals for persuasion campaigns, this general framework does not specify how to identify signals for the specific types of targeted social media events.

3.1 Problem Definition

In this dissertation, we study the problem of social media event detection. In particular, we detect two specific types of events: rumors and persuasion campaigns. Our detection systems take a stream of social media posts as input and produce social media events as output. A social media post contains user-generated content and is the result of online user activity. A social media event is comprised of users' collective activities in the form of a group of posts with similar content. For example, a social media event might be the trending of a social media meme, which is a group of posts including the origin meme and posts that share and discuss it. Our targeted events, rumors and persuasion campaigns, are groups of posts that spread and discuss either a rumor or a persuasion campaign. The next two sections discuss different types

of online activities and how collective activities can be reflected by groups of social media posts.

3.1.1 User activities and posts in social media

Social media platforms such as Twitter and Facebook offer their users various types of functions for interacting. Users can subscribe and consume information posted by other users. They can also post and share information on their own page. Users can have conversations about a post by commenting and replying. They can also communicate via direct messages or other supported functions.

The functions offered by social media allow users to engage in many different types of activities. In this dissertation, we focus on users' public activities that can be seen by everyone, usually provided by functions such as creating posts, resharing or retweeting, and commenting. They are listed as follows:

- Posting. The user creates a new post on his or her wall or timeline. The post content can be originally generated or come from sources outside the social media platform.

- Resharing. The user reshares a post from another user on his or her wall or timeline. In Twitter, this behavior is called retweeting, and the shared posts will begin with "RT". There are two types of retweets. One type directly shares the original content. The other type shares by adding a short comment along with the original content.

- Commenting. The user comments on a post by another user. In Twitter, comments become tweets posted by the user who owns the comments. In Facebook, comments can also be found on the user's page.

These functions can result in the creation of a social media post in the user's timeline or public profile. For the purposes of this dissertation, we do not consider

other activities such as adding friends, clicking and reading, and more.

3.1.2 Post clusters as social media events

A social media event consists of collective online activities of many users that have a common topic or a common objective, such as questioning a piece of information or spreading one. These posts share similarities, such as their content, and can be grouped into a cluster that corresponds to the social media event. There are many types of social media events, such as breaking news, trending memes, rumors, and persuasion campaigns. Posts for a given type of event can be clustered based on a specific similarity measure, and detection of social media events can then be modeled as detection of post clusters from the post stream. For example, in rumor detection, we cluster posts by their content and identify clusters with rumorous ones. In this dissertation, we use the terms "social media events", "collective activities", and "groups of posts" to refer to post clusters.

3.1.3 Target clusters

We use the term "target clusters" to refer to the post clusters we want our detection systems to detect. These can be clusters corresponding to social media rumors, persuasion campaigns, or other types of targeted events. Although rumors and persuasion campaigns are defined differently and correspond to different types of target clusters, they share some characteristics that make their detection more meaningful than detection of other general social media events. Rumors ad persuasion campaigns are:

- Influential. The spreading and outbursting of rumors and persuasion campaigns are usually more influential than other general social media events, since most of them target and can result in changing people's opinions and real-world behaviors.

- Relatively rare. Compared to other post clusters, such target clusters are relatively rare. They sometimes correspond to events that are unusual or abnormal.
- Time sensitive. Considering the temporal dynamics, target clusters will emerge, grow, and disappear or transform to normal clusters. For example, a rumor will be verified or debunked, and a persuasion campaign can emerge and quickly draw people's attention and then end when people stop discussing it.

3.1.4 System Input and Output

In this section we provide a formal definition of our detection system's input and output.

The input to our detection system is a stream of social media posts

$$\{P_1 = (p_1, t_1), P_2 = (p_2, t_2), \ldots, P_k = (p_k, t_k), \ldots\},$$

where each $P = (p, t)$ is a tuple representing a post and its time stamp.

The output from our detection system at an arbitrary point in time is a set of target clusters

$$\{C_1 = \{P_{c_1}^1, P_{c_1}^2, \ldots, P_{c_1}^{k_1}\}, C_2 = \{P_{c_2}^1, P_{c_2}^2, \ldots, P_{c_2}^{k_2}\}, \ldots C_n = \{P_{c_n}^1, P_{c_n}^2, \ldots, P_{c_n}^{k_n}\}\},$$

where each element C denotes a cluster containing post and time stamp tuples. $P_{c_i}^j$ represents the j-th post for cluster C_i.

3.2 A general framework for detecting target clusters

We propose a framework that can be used to detect different types of target clusters. Our rumor detection system and persuasion campaign detection system are built using this framework. In contrast to existing social media event detection

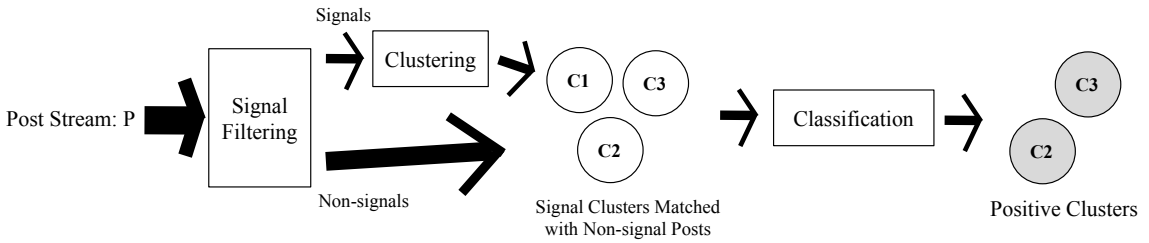


Figure 3.1: Workflow of Our Proposed General Framework

systems that first cluster all posts and then identify target clusters, our proposed framework introduces a filtering step to filter out the majority of the posts before clustering.

As discussed in Chapter I, the emergence and spreading of social media rumors or persuasion campaigns can result in common user reactions. Some of these can be found explicitly in social media posts, which then serve as signals of the existence of target clusters. In any target cluster, only a small proportion of the posts are signals. But since these reactions are common to all target clusters, every target cluster will have some. They are like the tip of the iceberg, and just as we can spot icebergs by their tips, we can detect target clusters by their signals.

Based on this intuition, we propose a general framework to detect target clusters in a social media post stream. Figure 1.4 of Chapter I shows this conceptually. Given the formal definition of our detection problem, we show the workflow of our framework in Figure 3.1. In addition to the input post stream, our framework assumes that a signal identification algorithm and a measure of similarity between posts are given.

In our framework, we first use the signal identification algorithm to separate posts into two groups, signals and non-signal posts. Then we cluster only the signals using the given similarity measure. Each cluster of signals is a subset of a potential target cluster.

We then match non-signal posts to these clusters using the same similarity mea-

sure, and when there is a match, the non-signal post is added to the cluster. The clusters that now contain both signals and non-signal posts are candidate target clusters. Finally, we identify the actual target clusters among these candidates using a binary classifier trained on their collective features. In the next sections, we will discuss each of the steps in the framework.

3.2.1 Signal filtering

Prior to this step, we analyze a specific type of target cluster, such as social media rumors, and develop a signal identification algorithm for this type of cluster. The identification algorithm can combine automatic feature selection and analyses by domain experts. We discuss how to identify signals for social media rumors in the next chapter which describes building a rumor detection system. When we turn to building a persuasion campaign detection system, we will also introduce an automatic algorithm that learns identifiers for signals.

In the signal filtering step for a specific type of target cluster, the input post stream is run through its signal identification algorithm. This filtering step outputs two streams, one consisting of signals and the other of non-signal posts. Only the signals will be used to generate new clusters in the next step.

3.2.2 Clustering

This step begins by clustering the signals using the given similarity measure. Each cluster that is obtained is a subset of a potential target cluster. Then non-signal posts are matched against these clusters, and matching non-signal posts are added to the clusters. We now have clusters that contain both signals and non-signal posts. We output these as candidate target clusters for the next step. Unlike other existing frameworks, which cluster all posts, our clustering step does not generate clusters unless they contain signals. In this way our framework avoids clustering all posts,

which is both time consuming and unnecessary since target clusters are relatively rare.

There are two reasons why we need to add non-signals to the signal clusters: (1) Every target cluster contains both signals and non-signal posts. By matching and adding non-signal posts to the clusters, we can gain a complete understanding of a target cluster, which is beneficial for later analysis. (2) Signals exist not only in target clusters but also in some non-target clusters. Therefore, we need to extract collective features from clusters with both signals and non-signal posts in order to identify the true positives in the next step.

We note that our framework does not require a specific clustering algorithm. Any generic stream-clustering algorithm, such as [6] or [2], can be used. We can also use a batch-based clustering algorithm such as k-means by running it once for a given period of time and updating the results.

3.2.3 Classification

As discussed above, not all of the candidate clusters generated by the previous step will be target clusters, since signals can also occur in non-target clusters. Therefore, we use a binary classifier to identify target clusters from the candidate clusters. The classifier is trained using a labeled dataset with target and non-target clusters. Different types of collective features can be used to represent each cluster, such as content features, statistics, temporal features, and so forth. This step is similar to that found in existing event detection systems; however, the number of our candidate clusters is much smaller, with relatively more target clusters.

We adopt this general framework to build our rumor detection system in Chapter V and our persuasion campaign detection system in Chapter VI. For these two applications, the signal identifier, the post similarity measurement, and the features for classification are designed differently and individually for each application.

3.3 Discussion

As discussed in Chapter II, a common framework for building a social media event detection system for a specific type of event is to first cluster all posts and then classify the clusters. In contrast, our framework introduces signal filtering at the beginning and modifies the clustering step so that it only generates clusters that have at least a few signals. As a consequence, in the end we apply classification to only a small subset of all clusters of posts.

Our framework can efficiently filter out the majority of non-signal posts before clustering. This improves the detection efficiency. To identify target clusters, we only need to classify a set of candidate clusters that contains a higher proportion of target clusters than the set of all post clusters. This aslo improves the effectiveness of the detection. It must be noted that our framework relies on spotting signals in the post stream in order to detect candidate target clusters. Signal identification is thus critical for building our detection systems. They will not work well if signals are hard to identify or there are no signals that can be used.

To build our rumor detection system using this framework, we need to first discover signals and develop a signal identification algorithm. One property that social media rumors have is that, regardless of their specific content and topic, the spreading of the rumor will result in some users' skepticism about whether they are true. The common user reaction of asking questions can provide a clue to help us find proper signals. To understand how to identify such a reaction in user's posts, we conducted a content analysis of different types of user activities, particularly question-asking activities and question posts. We found that some of these are reactions to different types of social media events, including rumors; we will discuss this in the next chapter.

Based on our analysis, we found that confirmation questions or enquiries can be used as signals for rumor detection. We manually selected a set of enquiry patterns, such as "Is this true?" or "What?", based on further content analysis. We use these

patterns to identify posts that are signals of rumors. We then apply our framework to detect rumors on Twitter; we will talk about this in Chapter V.

For social media events like persuasion campaigns, the potential signals may contain complicated patterns that would be hard to identify manually. Thus, it is important to have an automatic approach to building a signal identifier. In Chapter VI, we will discuss learning a signal identifier by optimizing combined objectives. Each objective quantifies a property that we propose good signals should have. We use a signal identifier learned from a set of labeled persuasion campaigns to build a detection system with our framework.

Although in this dissertation we only build rumor and persuasion detection systems using the framework, our framework can be widely adapted for many other types of targeted social media events. One condition for using our framework is that the targeted events should exhibit some commonalities that can be captured from posts. For example, the emergence and spreading of social media rumors, despite the variety of topics, result in user reactions that question their truth. Similarly, common reactions can also be found in the spreading of persuasion campaigns.

CHAPTER IV

Analyzing User Asking Question Activities in Social Media

Users can have different types of activities, such as posting, sharing or commenting. And each of these types can have their unique content and topic distribution [121]. In this chapter, we conduct content analysis on social media user activities. We study most of them by understanding the content of corresponding social media posts.

By conducting this analysis, we want to learn how individual user activities become collective activities and social media events. We also want to understand the correlation between activity types and event types. To be more specific, one of our goal is to discover what posts are highly correlated with specific types of social media events.

We focus our research on user asking question activities and the corresponding question posts. We discuss how to extract and analyze user asking question activities from billions of online conversations collected from Twitter [122]. With an automatic text classifier, we can accurately detect real questions in tweets (i.e., tweets conveying real information needs). We then present a comprehensive analysis of the large-scale collection of questions we extracted. We found that questions being asked on Twitter are substantially different from the topics being tweeted in general. These questions

detected on Twitter have a considerable power of predicting social media trending events, and the trends of Google queries. Many interesting signals emerge through longitudinal analysis of the volume, spikes, and entropy of questions on Twitter, which provide insights to the understanding of user behavioral patterns and the impact of different types of events and in social platforms. Some of the insights found in this study help us find patterns to identify signals for detecting social media rumors.

4.1 Overview

Recent years have witnessed an explosion of user-generated content in social media. Online social platforms such as Facebook, Twitter, Google+, and YouTube have been complementing and replacing traditional platforms in many daily tasks of Web users, including the creation, seeking, diffusion, and consumption of information. On these platforms, user can share their experience by creating posts, commenting on friends' posts and other activities, resharing information from inside or outside one platform, and can do many other different types of activities. In this dissertation, we study clusters that are groups of different user activities with similar topic or content, and we want to find target clusters from indications of individual user activities. In this chapter, we first want to study and analyze different types of user activities, understand why and how are they different, and why some of them can indicate the occurrence of a specific type of target clusters and some of them cannot. By conducting content analysis on four major options users have on Google Plus, i.e., creating post, resharing, commenting and plusing one (like), we found out that user activities under these four different options are considerably different.

Next we want to study a specific type of user activities and have a in-depth understanding of why it is different compared to other types of activities and what this specific type of user activities can indicate. User asking question activities in social media, is the type of user activities that we are interested in. A recent research inter-

est has been shown in understanding how people seek for information through online social networks [78, 26, 81, 79, 67], how this "social information seeking" behavior differs from that through traditional channels such as search engines or online question answering (Q&A) sites, and how the social channel complements these channels [77, 47, 75]. Based on a survey conducted by Morris et al. in 2010, 50.6% of the respondents[1] reported having asked questions through their status updates on social networking sites [78]. The questions they ask involve various needs of recommendations, opinions, factual knowledge, invitations and favor, social connections, and offers. They covered many topics such as technology, entertainment, shopping, and professional affairs [78]. An analysis later by Efron and Winget suggested that 13% of a random sample of tweets (microblogs posted on Twitter.com) were questions [26].

Why is it compelling to understand the questions asked on social platforms? This emerging research interest largely attributes to the importance of understanding the information needs of Web users. Indeed, as the core problem in information retrieval, a correct interpretation of the information needs of the users is the premise of any automatic system that delivers and disseminates relevant information to the users. It is the common belief that the analysis of users' information needs has played a crucial role behind the success of all major Web search engines and other modern information retrieval systems. Better understanding and prediction of users' information needs also provides great opportunities to business providers and advertisers, leading to effective recommender systems and online advertising systems.

Long have Web search engines been the dominating channel of information seeking on the Web. According to recent statistics [2], 4 billion of search queries are submitted to Google every day. The rest of the territory is shared by other channels such as online question answering (Q&A) sites such as Yahoo! Answers. A statistic in 2010

[1] Note for the selection bias as all were Microsoft employees.

[2] `http://www.comscore.com/Insights/Press_Releases/2012/4/comScore_Releases_March_2012_U.S._Search_Engine_Rankings`

reported a daily volume of 823,966 *questions and answers* [3], in which each question on average earned five to six answers according to [96]. This is much smaller than the number of information needs asked through search engines.

The emergence of social platforms seems to be a game-changer. If the ratio reported by Efron and Winget [26] still holds today, there will be over 50 million questions asked through Twitter according to a recent statistic of 400 million tweets posted per day [4]. This number, although still far behind the number of search queries, has already overwhelmed the number of questions in traditional Q&A sites. Moreover, it has been found that people tend to ask different questions to their friends rather than to search engines or to strangers on Q&A sites. In Figure 4.1, we can see people asking questions in their tweets by either broadcasting so that any of their followers can respond to them, or by targeting the question to particular friends. The results of the survey by Morris et al. suggested that respondents especially prefer social sites over search engines when asking for opinions and recommendations, and they tend to trust the opinions of their friends rather than strangers on Q&A sites [78]. It is reported in [77] that users enjoy the benefits of asking their social networks when they need personalized answers and relevant information that unlikely exists publicly on the Web. It is also reported that information needs through social platforms present a higher coverage of topics related to human interest, entertainment, and technology, compared to search engine queries [81].

All evidence suggests that the questions being asked through social networks present a completely new perspective of online information seeking behaviors. By analyzing this emerging type of behavior, one anticipates to help users effectively fulfill their information needs, to develop a new paradigm of search service that bridges search engines and social networks (e.g., social search [29, 75]), and to predict what

[3] `http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/`

[4] http://news.cnet.com/8301-1023 3-57448388-93/twitter- hits-400-million-tweets-per-day-mostly-mobile/

Tweets Conveying Information Need	Tweets not Conveying Information Need
Do you know whether there is a roadwork on I94	Man so everybody a frank ocean fan now? Idc I was an original…
Which restaurant nearby has a discount?	Why do I always do this? #hesatool #fml
@someuser u work today???	@someuser how are you?
Can anyone suggest some local restaurants in Beijing?	They're still together, why haven't they broken up yet?!?!
@someuser, do you what I am doing is good?	Umm what? It's already August? Hey Summer, #wheredygo?
What's your favorite summer album to throw on a car stereo?	Im still gone smile! What are you thanking?! Em not
Is my avi cute?	Why won't people understand that?!

Figure 4.1: Instances of tweets conveying an information need from asking questions, and those which don't.

the users need in order to strategize information service provision, persuasion campaign, and Internet monetization.

The availability of large scale user-generated content in social network sites has provided a decent platform for this kind of analysis. This revives our memory about the early explorations of analyzing search engine query logs (e.g., [97, 104, 72]). Indeed, the analysis of information needs with large-scale query logs has provided tremendous insights and features to researchers and practitioners, and it has led to a large number of novel and improved tasks including search result ranking [5], query recommendation [8], personalization [105], advertising [14], and various prediction tasks [39, 49]. We believe that a large-scale analysis of information needs on online social platforms will reproduce and complement the success of query log analysis, the results of which will provide valuable insights to the design of novel and better social search and other online information systems.

In this chapter, we take the initiative and present the first very large scale and longitudinal study of information needs in Twitter, the leading microblogging site. Questions that convey information needs are extracted from a collection of **billions** of microblogs (i.e., tweets). This is achieved by an automatic text classifier that distinguishes real questions (i.e., tweets conveying real information needs) from tweets

with question marks. With these detected questions as signal activities of emerging information needs, we are able to present a comprehensive description of the information needs with both the perspectives of content analysis and trend analysis. We find that questions being asked on Twitter are substantially different from the content being tweeted in general. We prove that information needs detected on Twitter have a considerable power of predicting the trends of search engine queries. Through the in-depth analysis of various types of time series, we find many interesting patterns related to the entropy of language and bursts of information needs. These patterns provide valuable insights to the understanding of the impact of bursting events and behavioral patterns in social information seeking.

4.2 Analyzing Different Types of User Activities

In this section, we analyze user's online activities by their types and measure the content differences among them. The findings we have here will help us understand how to identify different types of activities from features of posts, and why some activities with unique features can be used as signals to detect targeted events.

Here we analyze users' different types of online activities in Google+ using an anonymized data set. We first extract the topic entities in posts as our features to construct feature vectors for each post. For each users' behavioral actions on posts, we aggregate the corresponding post feature vectors to build an entity vector for each user-behavior combination. Then we coarsely measure the differences of topical interests represented by these user-behavior entity vectors. We show that there exists significant differences between these vectors, which motivates our approach to utilize a small proportion of special user activities as indications to detect target clusters.

4.2.1 Dataset Description

We use anonymized Google+ users' public behaviors in May 2014 to conduct our analysis. We analyzed all user actions on all public posts, and each record is represented as a tuple: (u, b, E), where a user u (with an anonymized id) used behavior b to engage with a post containing E set of entities. There are four types of activites in our data: Create Post, Reshare, Comment, and +1.

Instead of using low-level features such as word tokens, we extract higher-level semantic concept from the post in the form of entities using Google's Knowledge Graph [98], which contains entities that represent concepts such as computer algorithms, landmarks, celebrities, cities, or movies. It currently contains more than 500 million entities, which provides both wide and deep coverage of topics.

Entity extraction is an open research problem and not a focus of our work here, but in a nutshell, we utilized an entity extractor based on standard entity recognition approaches that utilize prior co-occurrences between entities, likelihood of relatedness between entities, entities' positions within the text, and then finally ranking the topicality of the entity for the text.

Given a post, we use its corresponding Knowledge Graph entities as features to represent its topics. Therefore, each E in input tuple (u, b, E) is a set of Knowledge Graph entities. For example, if a user u_1 created a post with his dog's picture, this behavior might correspond to $(u_1, CreatePost, \{\text{“}Dog'', \text{“}Pet'', \dots\})$. If another user u_2 commented on a post with a YouTube video about Minecraft on Xbox, this behavior might correspond to the tuple $(u_2, Comment, \{\text{“}Minecraft'', \text{“}Xbox'', \dots\})$.

4.2.2 Measuring differences among activities

For each user, we aggregate the entities from the posts she interacted with using a particular type of activity. In the end, for each user, we obtain four sets of topic entities corresponding to the four activity types mentioned above.

Behavior	Comment	Plus One	Reshare	Create Post
Comment	1	0.092	0.050	0.102
+1	0.092	1	0.048	0.071
Reshare	0.050	0.048	1	0.012
Create Post	0.102	0.071	0.012	1

Table 4.1: Average Jaccard similarity between pairs of behavior types

We then use the Jaccard similarity index to measure the differences between the sets. Jaccard similarity index is a common metric for measuring the similarity between two sets and is calculated as follows given sets A and B: $J(A, B) = \frac{A \cap B}{A \cup B}$.

After we calculate Jaccard similarity scores of different activities for each user, we then average the scores across all users. We filter out users who have less than 10 entities as non-active. Table 4.1 shows the results of the average Jaccard similarities. We can see that the Average Jaccard Index between any two types of activities is low. Take user's commenting and +1 behaviors as an example, only 9% of the topics overlapped between these two behaviors. We also measure the difference between user's publishing and consuming behaviors. We combine the entities of user commenting and +1 behaviors as a set of entities of consuming, and we combine the entities of user creating post and resharing behaviors as a set of entities of publishing. The average Jaccard Index is 0.122. The low overlap rate of these Jaccard scores suggests that user acts differently in different behaviors.

4.2.3 Discussion

The results of the analysis show that, for each user, she typically have different topic interests with each different type of activities. That is, she will often create posts on topics that are different from the topics she comments on. The results suggest that different types of user activities may have different properties and play different roles in events in social media.

This corresponds to the recent efforts on building recommender systems by sepa-

rating users' activities types and provide recommendation results for different context. Our finding is also related to findings in other areas such as sociology. Sociologists have shown that people present different images to others in their everyday life and their everyday conversations engage in different topics with different audiences [40]. The emergence of social media has also drawn sociologists' interests to study this phenomenon in online communities. For instance, sociologists theorize that, because users do not have a clear idea of the exact audiences in public social media, they end up with blurred context boundaries [70]. However, because different types of behaviors, such as posting or commenting, affect very different audiences, our analysis below suggest that users still show different 'identities', exhibiting different types of behaviors around different topics in social media. By conducting qualitative study, Zhao et al. point out that users experience social media platforms such as Facebook as multiple different functional regions, similar to their multiple identities in real life [120].

Understanding the content and topic differences among user activity types provides us insights on detecting targeted events in social media. The uniqueness of one type of activities can be indicating to specific types of targeted events. Our general detection framework is to leverage such uniqueness from signals to identify target clusters with high accuracy.

After we have shown the differences among the four common types of activities, in the next section, we continue our content analysis on user asking questions activities and have an in-depth understanding of its characteristics. They are interesting to study because questions are related to what people want to know, and it could be potential signals for many different types of targeted events. We will show how user asking question activities are different from other activities, and how their uniqueness can be indicating to bursting events and emerging information needs. In next chapter, we will show that enquiries, a specific type of questions, can be used as signals to

detect social media rumors.

4.3 Identify Questions Asked in Twitter

Although a simple way to identify questions is to find question mark, not all tweets with question marks are real questions. In order to detect signal activities of emerging information needs from tweets, we need to distinguish tweets that convey a real information need from many false positives such as rhetorical questions, expressions of sentiments/mood, and many other instances. Figure 4.1 presents examples of tweets that convey real information needs and those which don't.

In this section, we present the task of detecting question asking activities from tweets which is casted as a text classification problem. Given a tweet that contains one or two question marks, the task is to determine whether it expects an *informational* answer or not. In this section, we first describe our dataset, then give a formal definition of this problem and rubrics based on which human annotators can accurately classify a tweet. A qualitative content analysis is conducted in order to develop a codebook of classification and generate a set of labeled tweets as training/testing examples. We then introduce a classifier trained with these examples, using the state-of-the-art machine learning algorithms and a comprehensive collection of features. The performance of the text classifier is evaluated and presented in the last subsection.

4.3.1 Dataset Description

We analyze a longitudinal collection of microblogs (tweets) collected through the Twitter stream API with Gardenhose access, which collects roughly 10% of all public statuses on Twitter. The collection covers a period of 358 days, from July 10th, 2011 to June 31st 2012. A total number of 4,580,153,001 (12.8 million tweets per day) tweets are included in this collection, all of which are self-reported as tweets in

English. Every tweet contains a short textual message constrained by 140 characters, based on which we determine whether it conveys an information need. For every tweet, we keep the complete metadata such as the user who posted the tweet, the time stamp at which it was posted, and geographical locations of the user if provided. In our analysis we adopt only the time and user information but leave the richer metadata for future analysis.

Note that a tweet may be a retweet of an existing tweet, may mention one or more users by "@" their usernames, and may contain one or more hashtags (user-defined keywords starting with an "#"). In our analysis, we keep the original form of all hashtags, but de-identify all usernames mentioned in the tweets (e.g., substituting all of them with a token "@someuser").

To analyze information needs in these tweets, we choose tweets that appear to be questions as signals. Specifically, we focus on tweets that contain at least one question mark. Note that this treatment could potentially miss information needs that are presented as statements. According to statistics in [78], 81.5% of information needs asked through social platforms were explicitly phrased as questions and included a question mark. Questions phrased as statements were often preceded by inquisitive phrases like "I wonder," or "I need" [78]. Because there is little foreseeable selection bias, we choose to focus on questions with explicit question marks instead of enumerating these arbitrary patterns in an ad hoc manner. In our collection of tweets, 10.45% of tweets contain explicit appearance of question mark(s).

4.3.2 Training Question Detector

4.3.2.1 Definition and Rubrics

Given a tweet with question marks, our task is to determine whether this tweet conveys a real information need or not (i.e., real questions). A formal definition is needed to describe what we mean by "a real information need." Inspired by the liter-

ature of how people ask questions on Twitter and Facebook [81, 77], we provide the following definition and rubrics of "real questions:"

A tweet conveys an information need, or is a real question, if it expects an informational answer from either the general audience or particular recipients.

Therefore, a tweet conveys an information need if

- **it requests for a piece of factual knowledge, or a confirmation of a piece of factual knowledge**. A piece of factual knowledge can be phrased as a claim that is objective and fact-checkable (e.g., "Barack Obama is the 44th president of the United States").

- **it requests for an opinion, idea, preference, recommendation, or personal plan of the recipient(s), as well as a confirmation of such information.** Here the information been requested is subjective, which is not fact checkable at the present.

A tweet does not convey an information need if it doesn't expect an informational answer. This includes rhetorical questions, expressions of greeting, summary of the content (eye attractors), imperial requests (to be distinguished from invitations), sarcasm, humor, expressions of emotion (complaints, regrets, anger, etc), or conversation starters.

Figure 4.1 shows some examples of tweets conveying information need and tweets which don't. Using the description we proposed above, a human annotator can easily classify a tweet. In the following subsections, we introduce how we extract features from the tweets, how we select features using the state-of-the-art feature selection techniques, and how we train classifiers using a single type of feature and then combine them using boosting.

4.3.2.2 Human Annotation

Based on the rubrics, we developed a codebook[5] and recruited two human annotators to label a random sample of tweets. We sampled 5,000 tweets randomly from our collection, each of which contains at least one question mark and self-reported as English. Finally, 3,119 tweets are labeled as real tweets in English and have same labels by the two coders. Among the 3,119 tweets, 1,595 are labeled as conveying an information need and 1,524 are labeled not conveying an information need. The inter-rater reliability measured by Cohen's kappa score is 0.8350, the proportion of the agreements in all the results is 91.5%. The 3,119 labeled tweets will be used to train and evaluate the classifier of information needs.

4.3.2.3 Text Classification

Feature Extraction The classification of tweets is a particularly challenging because of the extremely short length of content (i.e., a tweet has a limited length of 140 characters). This makes the textual features in an individual tweet extremely sparse. To overcome this challenge, we not only utilize lexical features from the content of the tweets, but also generalize them using the semantic knowledge base WordNet [74, 30]. It is also our intent to include syntactical features as well as metadata features. We extracted four different types of feature from each tweet, i.e., lexical ngrams, synonyms and hypernyms of words (obtained from the WordNet), ngrams of the part-of-speech (POS) tags, and light metadata and statistical features such as the length of the tweet and coverage of vocabulary(i.e., number of different words used in a tweet divided by the number of different words in the whole dataset), etc..

Lexical Features

We included unigrams, bigrams, as well as trigrams. The start and end of a

[5]The codebook is made available at `http://www-personal.umich.edu/~zhezhao/projects/IN/codebook.html`

tweet are also considered in the ngrams. This gives us great flexibility to capture features that reflects the intuitions from qualitative analysis. For example tweets beginning with the 5Ws (who, when, what, where, and why) are more likely to be real questions. All lexical features are lowercased and stemmed using the Krovetz Stemmer [55]. Hashtags are treated as unique keywords. To eliminate the noise of low frequent words, a feature is dropped if it appears less than 5 times. This resulted in 44,121 lexical features.

WordNet Features

To deal with the problem of data sparsity, we attempt to generalize the lexical features using the synonyms and the hypernyms of the words in tweets. We hope this approach would connect different features sharing relevant semantics in different tweets. By doing this, our algorithm can also handle words that haven't been seen in the training data, thus is anticipated to achieve a higher performance with limited training data.

In [69], the authors studied how different types of relevant words from WordNet influence the results of text classification. In most cases, using only synonyms and hypernyms can improve classifiers such as Support Vector Machine (SVM) the most. We explored different WordNet features in our task and drew the same conclusion. We therefore adopt only synonyms and hypernyms of words in a tweet as additional features. Note here we actually excluded this semantic generalization for nouns in a tweet. This is because our task is to discover patterns of how people ask questions, instead of what they ask. 23,277 WordNet features are extracted.

Part-of-Speech Features

Compared to a statement, questions present special patterns of syntactic structure. Therefore we attempt to include syntactic features into consideration. Syntactic parsing of billions of tweets appears to be costly and probably unnecessary, since the quality of parsing is compromised given the inaccurate use of language in social media.

We thus seek for features that capture light syntactic information. We first obtain part-of-speech of the words in a tweet, and then extract ngrams of these part-of-speech tags. That is, given a tweet with n words, $w_1, w_2, \ldots, w_n$, we extract grams from the part-of-speech sequence of the tweet, is $t_1, t_2, \ldots, t_n$, and then extract unigrams, bigrams and trigrams from this part-of-speech sequence as additional features of the tweet. 3,902 POS features are extracted in total.

Meta Features

We also include 6 metadata features and simple statistical features of the tweet such as the length of the tweets, the number of words, the coverage of vocabulary, the number of capitalized words, whether or not the tweet contains a URL, and whether or not it mentions other users. We believe these features are possibly indicative of questions.

Feature Selection The four types of extracted features represent each tweet as a vector with a very large number of dimensions. This is not surprising given the huge and open vocabulary in Twitter. Even though we can reduce the number of features by various heuristics of post-processing, the number of features remaining is still far larger than the number of training examples. Therefore, it is essential to conduct feature selection and further reduce the dimensionality of the data.

In this work, we adopt the state-of-the-art feature selection method named Bi-Normal Separation (BNS) proposed in [32]. In this work, the author proved that the proposed metric for feature selection outperformed other well-known metric such as Information Gain and Chi-distance. Specifically, let tp and tn be the number of positive cases with and without a given feature, fp and fn be the number of negative cases with and without the feature. Let tpr be the sample true positive ratio (i.e., $tpr = tp/(tp + fn)$) and fpr be the sample false positive ratio (i.e., $fpr = fp/(fp + tn)$).

The BNS metric of a given feature can be calculated by

$$\|F^{-1}(tpr) - F^{-1}(fpr)\|, \tag{4.1}$$

where F is the Normal cumulative distribution function.

4.3.2.4 Training Classifier

After feature selection, we move forward and train four independent classifiers using the Support Vector Machine (SVM) [22], based on each of the four types of features. then combine the four classifiers that represent four types of features into one stronger classifier using boosting. This is done through the Adaptive Boosting method called Adaboost [34].

Adaboost is an effective algorithm that trains a strong classifier based on several groups of weak classifiers. Usually Adaboost can obtain one classifier better than any of the weak classifiers. However, when the performances of the weak classifiers are higher than a certain level, it is hard to use this algorithm to generate a better classifier. This situation seems to apply to our scenario, since the SVM classifiers are sufficiently strong. In [66], the authors indicated that the reason why this problem occurs is that after several iterations, when the combination of weak classifiers starts to achieve a higher performance, the diversity inside the combination is getting lower. That says, new weak classifiers are likely to make same predictions as the old ones. To solve this problem, they add a parameter to control for the diversity of the weak learners in each iteration. We aslso adopt this technique to combine the four SVM classifiers.

We define parameter *div* as the threshold of a minimum diversity of a new weak classifier to be added in each iteration in the Adaboost. The diversity that a new classifier could add in iteration t is defined as follows:

$$div_t = \frac{1}{N} \sum_{i=1}^{N} d_t(x_i) \tag{4.2}$$

$$d_t(x_i) = \begin{cases} 0 & \exists k, f_k(x_i) = f_t(x_i) \\ 1 & \forall k, f_k(x_i) \neq f_t(x_i) \end{cases} \tag{4.3}$$

Here $d_t(x_i)$ is the diversity of classifier to be added in iteration t to data point x_i. N is the size of the training set. $f_k(x_i)$ is the predicted result of the classifier in iteration k for data point x_i. Our information need detection algorithm uses this modified Adaboost named AdaboostDIV. The diversity of a classifier represents how much new information it could provide to a group of classifiers that have already been trained in Adaboost. This value will be smaller and smaller when there are more classifiers adopted. In each iteration of AdaboostDIV, we examine the diversity of a new classifier. If the diversity of this classifier is higher than minimal threshold div, we accept this classifier into the group of classifiers. Otherwise we terminate the algorithm.

4.3.3 Evaluation

We train and evaluate our algorithm using the manually labeled set of 3,119 tweets. 10-fold cross validation and the metric of classification accuracy are adopted to evaluate each candidate classifier.

Before feature selection, there are 44,121 ngram lexical features, 23,277 WordNet features, 3,902 Part-of-Speech features, and 6 meta features. In Table 4.2, we compare the performance of the four SVM classifiers using each of the four types of features and various feature selection algorithms. The findings are consistent with the conclusions in [32]. Feature selection using the BNS metric outperformed two other metrics, namely accuracy (ACCU) and Information Gain, both of which improved over the classifiers without feature selection. Among the four types of features alone, ngram

Feature Type	Lexical	WordNet	POS	Meta
Raw	0.745	0.610	0.668	**0.634**
ACCU	0.790	0.673	0.718	/
Information Gain	0.804	0.676	0.723	/
BNS	**0.856**	**0.702**	**0.745**	/

Table 4.2: Results of SVM classifiers. Lexical features performed the best. Feature selection improved classification accuracy.

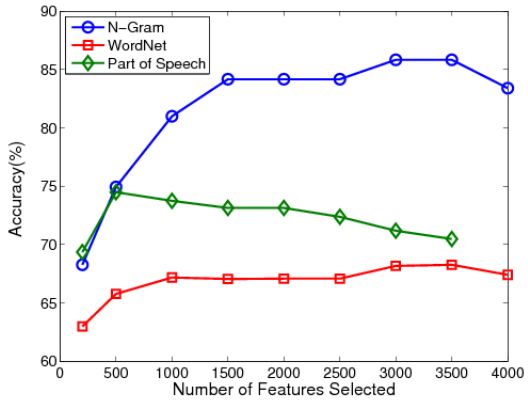


Figure 4.2: Feature selection using BNS

lexical features appear to provide the best performance, while the six meta features provide the weakest result which is also far better than random.

Figure 4.2 shows a fine tuning of the number of features selected using BNS. Clearly, when too few or too many features are selected, the classification performance drops because of insufficient discriminative power and overfitting, respectively. Based on our experiment results, we select 3,795 top ranked lexical features, 3,119 top WordNet features, as well as 505 top Part-of-Speech features.

At last, we combined the four SVM classifiers, representing four types of features, using AdaboostDIV. The accuracy of the classifier (with 10-fold cross validation) improved from 85.6% to 86.6%. The small margin suggests that the lexical features are strong enough in detecting information needs, while other types of features add little to the success. Using Adaboost instead of AdaboostDIV compromised the performance, which is consistent to the findings in [66].

Finally, the best performing classifier (four SVM classifiers combined with AdaboostDIV, with feature selection with BNS) is adopted to classify all the tweets in our collection. In our evaluation, the improvements made by feature selection and AdaboostDIV passed the paired-sample t-test at the 5% significance level.

4.4 Analyzing the Characteristics of Question Asking Activity

User asking question activities are related to the emerging information needs. Although many social media users won't ask questions when they have information needs, when such signal behaviors appears, just like the tip of the iceberg, it will indicate that there is an emerging information needs. In this application, we cannot extract non-signal behaviors such as searching in Twitter or search engines and reading posts. But with the large amount of questions detected in our dataset, we can still capture many meaningful emerging information needs.

After applying the text classifier above to the entire collection of tweets, we detected 136,841,672 tweets conveying information need between July 10th 2011 to June 31st 2012. This is roughly a proportion of 3% of all tweets, and 28.6% of tweets with question marks. With this large scale collection of real questions on Twitter, we are able to conduct a comprehensive descriptive analysis of user's information needs. Without ambiguity, we call all the tweets collected as the Background tweets, whether they are questions or not. We call tweets that convey information needs as Information Needs (or short as IN), or simply Questions.

4.4.1 General Trend Analysis

Once we are able to accurately identify real questions (or information needs), the first thing to look at is how many questions are being asked and how they are

distributed. Below we present the general trend of the volumes of questions being asked comparing to the total number of tweets in the background. For plotting purposes, we choose to show the trend of the first 5 months from this entire time scope, from July 10th 2011 to November 30th, 2011. Most of the events occurred during this period of time, so plotting the whole year's time series would take more space and cannot be shown distinctly. These 5 months contain a collection of 1,640,850,528 tweets, in which 51,263,378 conveyed an information need. We use this time period for all visualization and time-series analysis below.

Since there is a huge difference between the raw numbers of information needs and the background tweets, we normalize the time series so that the two curves are easier to be aligned on the plot. Specifically, we normalized all the time series using the Z-normalization. That is, for the i^{th} data point valued x_i in the time series, we transform the value by following equation:

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{4.4}$$

Where μ and σ are the mean and standard deviation of all data points in this time series. This simple normalization doesn't change the trend of the time-series, but allows two series of arbitrary values being aligned to the same range. In the plot, a positive value means the daily count of IN/background tweets is above the average count over time, and a negative value means the count is below the mean. An actual value x on one day indicates that the count of that day is x standard deviations away from the average.

From Figure 4.4.1, we observe that both the number of tweets and the number of questions are increasing over time. There are observable but weak days-of-week patterns, which differ search engine logs which present significant weekly patterns (more queries on weekdays than weekends) [72]. The trend is much more sensitive

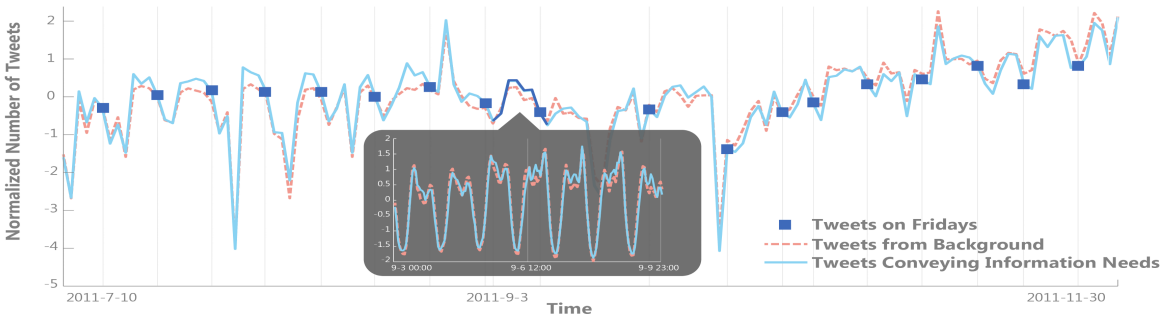
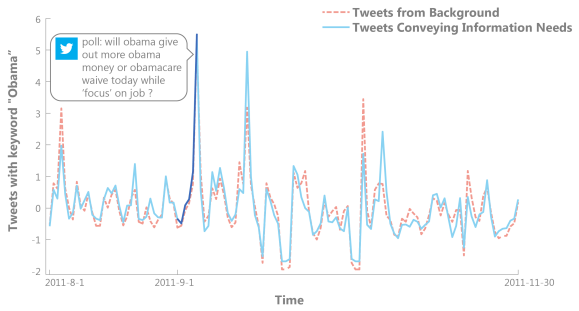


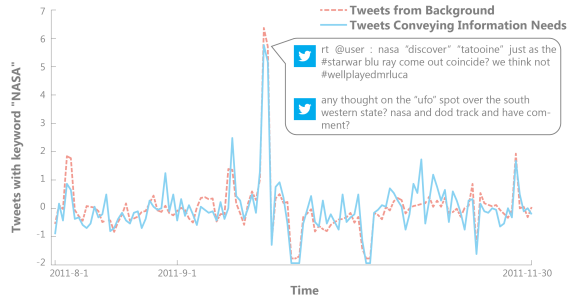
Figure 4.3: Questions and background tweets over time.

than that of query logs [72], with obvious and irregular spikes and valleys scattered along the time line. This implies that user's information seeking behaviors on Twitter are more sensitive to particular events than the behaviors on search engines. The subfigure presents a strong daily pattern, where both the total number of tweets and information needs peak in late morning and early evening, leaves a valley after noon, and sinks soon after midnight.

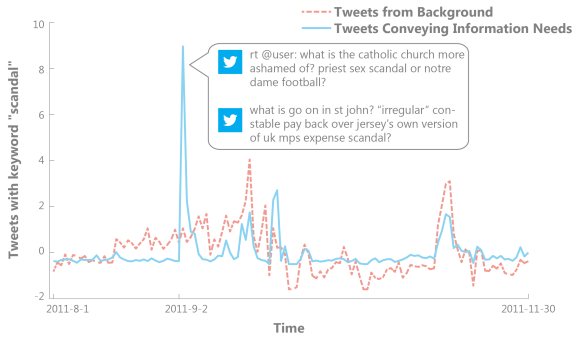
In general, the trend of information needs correlates with the trend of the background, which means the information needs on Twitter are likely to be social driven but not information driven. This is not surprising since real world events are likely to stimulate both the demand and supply of information. The more interesting signals in the plot are the noticeable differences between the two curves. On some days there is a significantly overrepresented "demand" of information (i.e., questions) than the "supply" (i.e., background), where there appears a noticeable gap between the two curves. This offers opportunities to analyze what people want, provide better recommendations, and develop propaganda. It is also an interesting observation from the hours-of-day trend that information needs are always overrepresented between the two peaks, before and after noon.



(a) Trend of tweets conveying information need with keyword "obama"



(b) Trend of tweets conveying information need with keyword "nasa"



(c) Trend of tweets conveying information need with keyword "scandal"

Figure 4.4: Trend of tweets conveying information need with different keywords

4.4.2 Keyword Analysis

With a sense of the general trend of how people ask, the next question is what people ask. Previous literature has provided insights on the categorization and distribution of topics of questions [78, 81]. Here we are not repeating their efforts, but to provide a finer granularity analysis of the keywords. After removing stopwords from the tweets, we extracted all unigrams and bigrams from tweets classified as conveying information needs. We trim this list of keywords by keeping those appeared every day of our time frame. We believe these keywords that the most representative of the *everyday* information needs of users in Twitter instead of information needs only triggered by particular events. For each of these keywords, we keep the daily count of the number of background tweets containing the keyword and the number of information needs containing the keyword.

With these counts, we can distinguish keywords that appeared frequently in information needs and those appeared frequently in the background. Table 4.3 lists a subset of keywords that are significantly overrepresented in information needs (i.e., have a much larger frequency in IN than in Background tweets, normalized by the maximum of the two frequencies), compared to the keywords significantly overrepresented in the background. One can observe from the table that keywords about technology (e.g., "noyoutube," "pocket camera," "skype," "waterproof phone") and recommendation seeking (e.g., "any suggestion," "any recommend") have a high presence in questions while URLs (e.g., "http"), greetings (e.g., "good night," "god bless") and requests (e.g., "follow back") are more frequent in the background. This finding is consistent with the quantitative analysis in literature [78, 81].

We further dropped the keywords that appeared less than 10 times a day in average, from which we obtained 11,813 keywords. For these keywords, we generated time series that represent the demand of information about these keywords, by counting the number of questions and general tweets containing particular keywords everyday.

Frequent in IN	Frequent in BACKGROUND
noyoutube	http
butterfly fall	user video
pocket camera	follow back
Monday	retweet
skype	beautiful
any suggestion	photo
waterproof phone	good night
any recommend	god bless

Table 4.3: Overrepresented keywords in information needs and background

Figure 4.4 presents the trends of information needs and background tweets containing three particular keywords, namely "Obama," "NASA," and "scandal." In Fig. 4.4(a), we can see that the trend of information needs closely correlates with the background, with several noticeable bursting patterns. These patterns generally correspond to real world events. For example, the largest spike around September 8th was correlated with President Obama's speech about the $450 billion plan to boost jobs. Such types of major events are likely to trigger both questions and discussions in online communities, thus have caused a correlated spike of both information needs and the background.

The trends of the keyword "NASA" present a different pattern. The questions and the background align well around the big spike, but disjoin in other time periods. In general, the trend of information needs is more sensitive than the background discussions, presenting more fluctuation. These smallish spikes are not triggered by major events, but rather reflecting the regular demands of information. The trends of the keyword "scandal" is even more interesting. Even the major spikes don't correlate with questions and with the background. For example, the big spike in information needs was triggered by a widespread cascade of tweets that connects "Priest sex scandal" with "Notre Dame football," which is more like a cascade of persuasion, rumor, or propaganda instead of an real event.

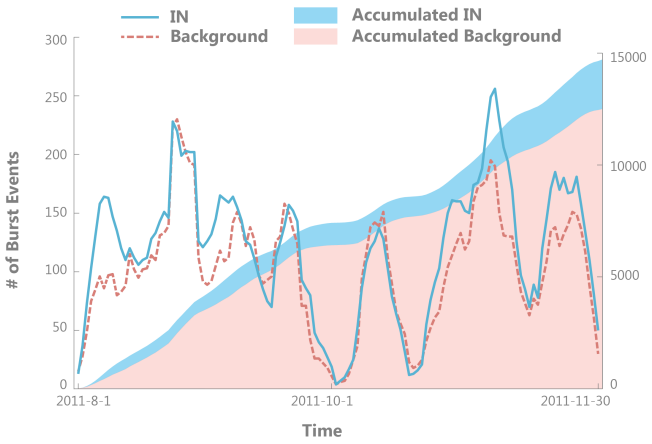


Figure 4.5: Bursts detected from IN and background

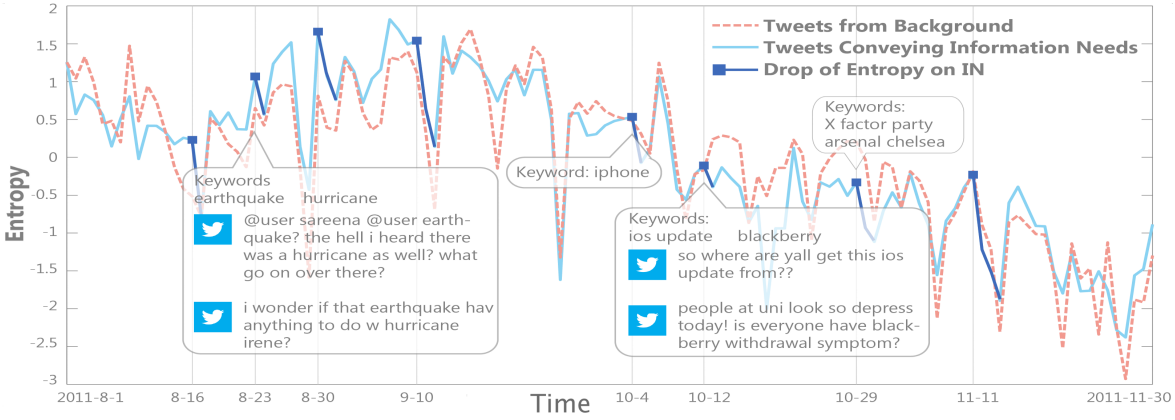


Figure 4.6: Entropy of word distributions in questions and background

4.4.3 Burstiness Analysis

The anecdotal examples above presented interesting insights in understanding the different roles of bursting patterns in the time series. This is done by comparing individual spikes in information needs with the pattern in the background in the same time period. A different perspective of investigating such bursting patterns is to compare them longitudinally. How many spikes are like the spike caused by Obama's job speech? If a similar bursting pattern can be found among the information needs of a different keyword, that means there is an event that have made a similar impact with the president's speech in terms of triggering the users' behaviors of information

seeking.

Literature has thrown light on how to detect real events based on burst detection in social media [115, 114]. In our analysis, we adopt a straightforward solution to detect **similar** burst events in the time series of information needs and the background. Specifically, we select a signature bursting pattern of a real event as a query (e.g., the spike corresponding to Obama's job speech in Figure 4.4(a)) and retrieve all similar spikes in the time series of other keywords. The similarity measurement is the Euclidean distance between Z-normalized time series. By doing this, we found 14,640 burst patterns in the time series of information needs and 12,456 burst patterns in the background of all keywords. Figure 5 plots the number of burst events that have a similar impact as the Obama speech, aggregated from the time series of all different keywords. Apparently, there are more such spikes in the time series of information needs rather than in the background, which reassures our finding that the behavior of question asking is more sensitive than the narrative discussions of events. The number of bursting patterns tops in late August and the month of October, which coincides with the two series of events related to "Hurricane Irene" and "Occupy D.C."

4.4.4 Entropy Analysis

The investigation of bursting patterns provides insights about understanding the impact of real events on Twitter users' information seeking behaviors. The impact is featured by the sudden increase of information needs (or background tweets, or both) containing certain keyword. Another way to measure the impact of an event is to look at how it influences the content of information people are tweeting about and asking for. Shannon's Entropy [36] is a powerful tool to measure the level of uncertainty, or unpredictability of a distribution. It is well suited for sizing challenges, compression tasks, as well as the measure of diversity of information. We apply

Shannon's entropy to the information needs detected, by measuring the entropy of the word distribution (a.k.a., the language model) in all background tweets and in all questions every day. Clearly, a lower entropy indicates a concentration of discussions on certain topics/keywords, and a higher entropy indicates a spread of discussions on different topics, or a diversified conversation.

Our intuition is that if a major event influences the discussion and information seeking behaviors, the topics in the background or in the questions on that day will concentrate on the topics about that event. Thus we are likely to observe a decreased entropy. Figure 4.6 plots the entropy of the language models of all information needs, and of all tweets in the background over time. We mark several points in the time series where we observe a sudden drop of entropy on the next day, which indicates a concentration of topics being discussed/asked. We selected these points by the significance of the entropy drop and the differences between the entropy of IN and the entropy of background. We then extract the keywords that are significantly overrepresented in the day after each marked point, which give us a basic idea about the topics that have triggered this concentration. These keywords are good indicators of the actual events that have triggered the concentration (e.g., "the hurricane Irene," "arsenal chelsea" and "the rumor about the release date of iphone 5").

It is especially interesting to notice that on some particular days, entropy drops in information needs but increases in the background. We believe these are very indicative signals for monitoring what the public needs. For example, on October 12th, 2011, there was a sudden drop of entropy in information needs which didn't occur in the background tweets. The discussions concentrated on keywords like "ios," "update," and "blackberry." Indeed, on that day Apple released the new operation system iOS 5, which triggered massive questions about how to get the updates. During the same time, there was a series of outages which caused a shutdown of the Blackberry Internet Service. Such an event has contributed in the concentrations of questions about

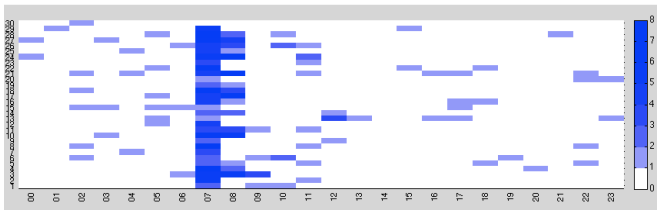


Figure 4.7: Questions of a user of low entropy.

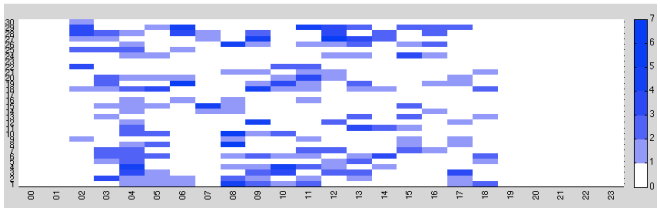


Figure 4.8: Questions from a user of high entropy.

Blackberry. It is interesting to see that these events about technology indeed had a larger impact in questions instead of the background tweets, which is again consistent with the statistics in literature [78, 81]. Clearly, analyzing the entropy of information needs provides insights on detecting events that have triggered the concentration of information needs. Such discoveries indicate compelling opportunities for search and recommender services, advertising, and rumor detection.

Interestingly, we found entropy analysis not only a powerful tool for macro-level analysis of the impact of events, but also effective in micro-level analysis of the information seeking behaviors of individual users. Indeed, we can also compute the entropy of the distribution of the number of questions that a user asks among different hours of a day. Behaviors of users with a low entropy are more predictable than behaviors of users with a high entropy. Below we show the two behavior patterns from two specific users. One is with high entropy and the other is with low entropy in Figure 4.7 and 4.8 respectively. In these two figures, the x-axes represent the 30 days in September, 2011, and the y-axes represent the 24 hours in each day. The different colors in these two figures represent different numbers of posts (the legends are shown on the right side of the figures).

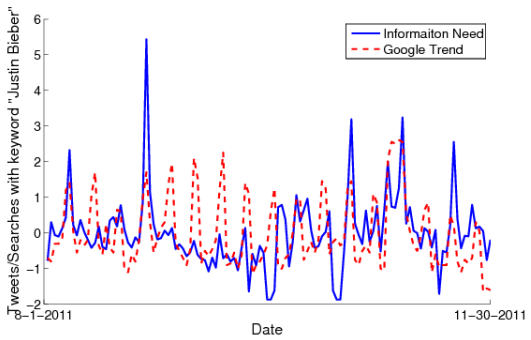
Clearly, the user with low entropy is fairly predictable: he always asks questions at 7am. By looking into his tweets, we found that this is an automatic account that retweets open questions from Yahoo! Answers. The second user is much less predictable, who seemed to be asking questions all over the hours of a day except for the bed time. By looking into his tweets, we found that this is a user who uses Twitter as an instant message platform, who chats with friends whenever he is awake. This user-level analysis on entropy of information needs presents insights on characterizing different individual behaviors.

4.4.5 Prediction of Trending Events

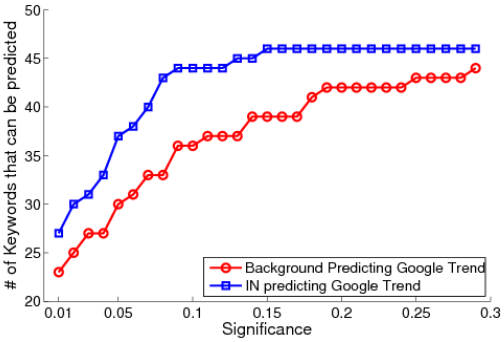
Up to now, we have presented many interesting types of analysis, mostly on the longitudinal patterns of information needs. We see various insights about how to make use of the analysis of information needs in Twitter. Previous literature has visioned the different and complementary roles of social networks and search engines in information seeking. What if we compare the information needs (questions) posted on Twitter and the information needs (queries) submitted to search engines? Is one different from the other? Can one predict the other? If interesting conclusions can be drawn, it will provide insight to the search engine business.

To do this, we compare the trends of information need in Twitter with the trends of Google search queries. Figure 4.9(a) shows the time series of the Twitter questions containing the keyword "Justin Bieber" and the Google trend of the query "Justin Bieber". We use this query as an example because it is one of the most frequent search queries in Google 2011 and is also contained in a large number of Twitter questions. We can see that information needs in Twitter is more sensitive to bursting events, while the same queries in Google presents a more periodic pattern (e.g., days-of-week pattern).

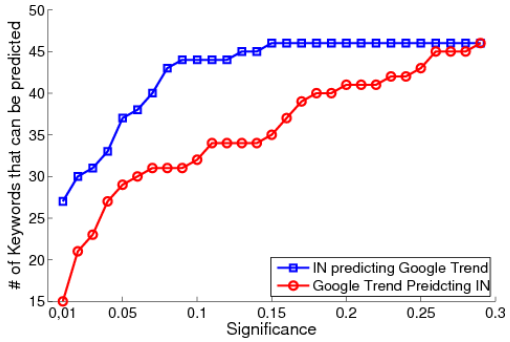
We then move forward to test whether the information needs from one platform



(a) Keyword: Justin Bieber



(b) Background v.s. information needs in predicting Google trends. The higher the better



(c) Information need v.s. Google trends in predicting each other. The higher the better

Figure 4.9: Twitter information needs can predict search queries.

can predict those in the other, using the Granger causality test. The Granger causality test is a statistical hypothesis test for determining whether a time series is useful in forecasting another [41]. In [11], it is used to test whether the sentiment of Twitter users can predict stock market.

Specifically, we selected a subset of keywords and manually downloaded the trends of these keywords as queries submitted to Google [6]. The subset of keywords contains twenty keywords that have a high frequency in background tweets, twenty keywords that have a high frequency in the questions, and twenty keywords from the most popular search queries in Google. To select this subset, we sorted all the keywords by frequency from the three different sources and select the top 20 named entities and nouns. If there is an overlapping keyword from multiple sources, we simply add a new keyword from the source with lower frequency of the overlapping keyword[7]. We then use the Granger causality test to test whether the three trends (Twitter background, Twitter information needs, and Google queries) of each keyword can predict each other. By changing the parameters in Granger causality test, we can test the prediction power of one time series to the other for different lags of time. Here, we only show the results with lag of 5 days. We obtained similar results with other different lags.

Results show that the trends of information needs in Twitter have a good predictive power in predicting trends of Google queries and are less likely to be predicted by the Google trends. This is measured by "*of how many keywords, one type of time series can predict another type of time series, given certain significance level.*" From Figure 4.9(b), we see that the information needs in Twitter have a better predictive power than the background in predicting Google trends. From Figure 4.9(c), we see that the information needs in Twitter have a better predictive power in predicting

[6]Since we don't have access to real search logs, we used the Google trend: `http://www.google.com/trends/`

[7]The List of the keywords can be found at `http://www-personal.umich.edu/~zhezhao/projects/IN/wlist`

Google trends rather than the other way around. Between information needs in Twitter and Google trends, the questions in Twitter have a stronger predictive power of Google queries, which successfully predicts the Google trends of more than 60% of the keywords with a significance level of 0.05. Among these keywords, 9 of them are from the popular Google queries. This is a promising insight for search engine practitioners to closely watch the questions in Twitter and improve the search results of targeted queries whenever a bursting pattern is observed.

4.5 Related Work

To the best of our knowledge, this is the first work to detect and analyze information needs from billion level, longitudinal collection of tweets. Our work is generally related to the qualitative and quantitative analysis of information seeking through social platforms (e.g., [81, 19, 77]) and temporal analysis of online user behaviors (e.g., [11, 39]).

4.5.1 Questions on Social Platforms

As described in Section 4.1, there is a very recent interest in understanding how people ask questions in social networks such as Facebook and Twitter [19, 78, 26, 81, 79, 67]. This body of work, although generally based on surveys or small scale data analysis, provides insights to our large-scale analysis of information needs in Twitter. For example, In [81], the authors labeled 4,140 tweets using Mechanical Turks and analyzed 1,351 of them which were labeled as real questions. They presented a rich characterization of the types and topics in these questions, the responses to these questions, and the effects of the underlying social network. In [78, 77, 113], Morris et al. surveyed whether and how people ask questions through social networks, the differences between these questions and questions asked through search engines, and how different cultures influence the behaviors. Efron and Winget further confirmed

this difference with a study of 375,509 tweets. Using a few simple rules, they identified 13% of these tweets as questions. They also provided preliminary findings on how people react to questions.

More sophisticated methods have been proposed to detect questions in online forums and Q&A sites [21, 109]. A recent work [62] studied the same problem in the context of Twitter, which presented a classifier that achieved 77.5% of accuracy in detecting questions from tweets. A much more accurate classifier is needed, however, to analyze information needs at a very large scale.

It is interesting to see the effort of making use of the understandings of social information seeking. In [47], Morris et al. proposed SearchBuddies, an automatic content recommendation for information seeking behavior. The proposed work finds relevant content based on the content and social context of Facebook status asking for information. Such effort can also be found in work like [29, 77, 106], where a new paradigm of search service, social search, is discussed. These explorations provided good motivations to our effort of large-scale analysis of information needs on social platforms.

4.5.2 Temporal Analysis of User Activities

The techniques of analysis used in our work is related to the existing work of analyzing user behaviors in general. For example, in [11], the authors proved that sentiment trends in Twitter has a power of predicting the Daw Jones Industrial Average. In their approach, the Granger Causality Test is used to test this predictive power. In [39], the authors used the Google trend related to influenza spread worldwide to detect which stage the flu was at and to predict the trend of the flu. Our analysis provides another important application of these methods. Note that our analysis is also related to the analysis of large scale search engine logs (e.g., [97, 104, 72]). Indeed, we do anticipate the analysis of information needs in social platforms to complement the

analysis of information needs through search engines, and provide a totally different perspective and insights to search engine practitioners.

4.6 Summary

In this chapter, we studied the differences between different types of user activities and focused on analyzing one specific type of activities, user asking question activities. This type of activities through social platforms attracted much interest because of its unique properties and complementary role to Web search. In this work, we present the first large-scale analysis of user asking question activities in Twitter. We proposed an automatic classification algorithm that distinguishes real questions from tweets with question marks with an accuracy as high as 86.6%. Our classifier makes use of different types of features with the state-of-the-art feature selection and boosting methods.

We then present a comprehensive analysis of the large-scale collection of questions we extracted. We found that questions being asked on Twitter are substantially different from the topics being generated by other types of activities. Question asking behavior are highly related to bursting events and have a considerable power of predicting emerging information needs shown in Google trend. Many interesting signals emerge through longitudinal analysis of the volume, spikes, and entropy of questions on Twitter, which provide valuable insights to the understanding of the impact of real world events in user's information seeking behaviors, as well as the understanding of individual behavioral patterns in social platforms.

Based on the insights from this analysis, we foresee many potential applications that utilizes the better understanding of what people want to know on Twitter. One possible future work is to develop an effective algorithm to detect and predict what individual users want to know in the future. By doing this one may be able to develop better recommender systems on social network platforms. With the presumption of

accessing large scale search query logs, a promising opportunity lies in a large-scale comparison of social and search behaviors in information seeking. On the other hand, improving the classifier to detect tweets with implicit information need such as tweets that is not an explicit question or without a question mark is also a potential future work. Furthermore, it is interesting to do some user-level analysis, such as studying the predictive power of different groups of users to see whether there exists a specific group of users that contributes to predicting the trend most.

Some findings in this analysis help us understand how users' posts can reflect the intention or reactions of their activities. The content and topic differences among different types of activities suggest that, some specific type can be unique and indicative to some social media events. Question asking activities, and question posts are the explicit expressions of users' information needs of various topics. As questions in general are highly correlated with bursting events, some questions, such as enquiries, are highly correlated with unconfirmed events, i.e., rumors. This correlations initiates our study on detecting social media rumors. In the next chapter, we will introduce our rumor detection system that uses enquiry posts as signals.

CHAPTER V

Early Detection of Social Media Rumors

In this chapter, we build a social media rumor detection system. Many previous event detection techniques identify trending topics in social media, even topics that are not pre-defined. We use our proposed general framework in this dissertation to identify trending rumors, which we define as topics that include disputed factual claims. Putting aside any attempt to assess whether the rumors are true or false, it is valuable to identify trending rumors as early as possible.

It is extremely difficult to accurately classify whether every individual post is or is not making a disputed factual claim. We are able to identify trending rumors by recasting the problem as finding entire clusters of posts whose topic is a disputed factual claim.

Inspired by our content analysis in last chapter, the key insight is that when there is a rumor, even though *most* posts do not raise questions about it, there may be a *few* that do. If we can find signature text phrases that are used by a few people to express skepticism about factual claims and are rarely used to express anything else, we can use those as detectors for rumor clusters. Indeed, we have found a few phrases that seem to be used exactly that way, including: "Is this true?", "Really?", and "What?". Relatively few posts related to any particular rumor use any of these enquiry phrases, but lots of rumor diffusion processes have **some** posts that do and

have them quite **early** in the diffusion.

We have developed a technique based on searching for the enquiry phrases, clustering similar posts together, and then collecting related posts that do not contain these simple phrases. We then rank the clusters by their likelihood of really containing a disputed factual claim. The detector, which searches for the very rare but very informative phrases, combined with clustering and a classifier on the clusters, yields surprisingly good performance. On a typical day of Twitter, about a third of the top 50 clusters were judged to be rumors, a high enough precision that human analysts might be willing to sift through them.

5.1 Overview

On April 15th of 2013, two explosions at the Boston Marathon finish line shocked the entire United States. The event dominated news channels for the next several days, and there were millions of tweets about it. Many of the tweets contained rumors and misinformation, including fake stories, hoaxes, and conspiracy theories.

Within a couple of days, multiple pieces of misinformation that went viral on social media were identified by professional analysts and debunked by the mainstream media.[1] These reports typically appeared several hours to a few days after the rumor became popular and only the most widely spread rumors attracted the attention of the mainstream media.

Beyond the mainstream media, rumor debunking Websites such as Snopes.com and PolitiFact.org check the credibility of controversial statements.[2] Such Websites heavily rely on social media observers to nominate potential rumors which are then fact-checked by analysts employed by the site. They are able to check rumors that

[1]Source: `http://www.cnn.com/2013/04/16/tech/social-media/social-media-boston-fakes/` and `http://www.scpr.org/blogs/news/2013/04/16/13322/boston-marathon-bombings-rumor-control-man-on-the/`

[2]Source: `http://www.snopes.com/politics/conspiracy/boston.asp`

are somewhat less popular than those covered by mainstream media, but still have limited coverage and even longer delays.

One week after the Boston bombing, the official Twitter account of the Associated Press (AP) was hacked. The hacked account sent out a tweet about two explosions in the White House and the President being injured. Even though the account was quickly suspended, this rumor spread to millions of users. In such a special context, the rumor raised an immediate panic, which resulted in a dramatic, though brief, crash of the stock market [25].

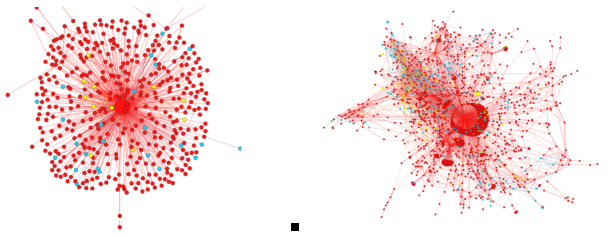
The broad success of online social media has created fertile soil for the emergence and fast spread of rumors. According to a report of the development of new media in China, rumors were detected in more than 1/3 of the trending events on microblog media in 2012.[3]

Rather than relying solely on human observers to identify trending rumors, it would be helpful to have an automated tool to identify potential rumors. The goal of such a tool would not be to assess the veracity of claims made in the rumors, merely to identify when claims were being spread that some people were questioning or disputing. If such a tool can identify rumors **earlier** and with sufficiently high precision, human analysts such as journalists might be willing to sift through all the top candidates to find those that were worth further investigation. They would then assess the veracity of the factual claims. Important rumors might be responded to earlier, limiting their damage. In addition, such a tool could help to develop a large collection of rumors. Previous research on rumor diffusion has included case studies of individual rumors that spread widely (e.g., [44]), but a fuller understanding of the nature of rumor diffusion will require study of much larger collections, including those that reach only modest audiences, so that commonalities and differences between

[3]Ironically, this report was misinterpreted by a major news media source, which coined a new rumor that "more than 1/3 of trending topics on Weibo are rumors."
Source: `http://truth.cntv.cn/erjiye/20/`

diffusion patterns can be assessed.

We propose a new way to detect rumors **as early as possible** in their life cycle. The new method utilizes the enquiry behavior of social media users as sensors. The key insight is that some people who are exposed to a rumor, before deciding whether to believe it or not, will take a step of information enquiry to seek more information or to express skepticism without asserting specifically that it is false. Some of them will make their enquiries by tweeting. For example, within 60 seconds after the hacked AP account sent out the rumor about explosions in the White House, there were already multiple users enquiring about the truth of the rumor (Figure 5.1). Table 5.1 shows some examples of these enquiry tweets.



(a) 60 seconds after the hacked twitter account sent out the White House rumor there were already sufficient enquiry tweets (blue nodes).

(b) Two seconds after the first denial from an AP employee and two minutes before the official denial from AP, the rumor had already gone viral.

Figure 5.1: Snapshots of the diffusion of the White House rumor. Red, yellow and blue nodes: Twitters spreading, correcting, and questioning the rumor.

Of course, not all tweets about a rumor will be such skeptical enquiries. As features for classifying *individual tweets*, enquiry signals are insufficient. Even if they yielded high precision, the recall would be far too low. As features for classifying *tweet clusters*, however, they provide surprisingly good coverage. Our technique for automatically detecting rumors is built around this signal.

We make three contributions in this work. First, we develop an algorithm for identifying *newly emerging, controversial* topics that is scalable to massive stream of tweets. It is scalable because it clusters only *signal tweets* rather than all tweets,

Table 5.1: Examples of enquiry tweets about the rumor of explosions in the White House

Oh my god is this real? RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
Is this true? Or hacked account? RT @AP Breaking: Two Explosions in the White House and Barack Obama is injured
Is this real or hacked? RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
How does this happen? #hackers RT @user: RT @AP: Breaking: Two Explosions in the White House and Barack Obama is injured
Is this legit? RT @AP Breaking: Two Explosions in the White House and Barack Obama is injured

and then assigns the rest of the tweets only if they match one of the signal clusters. Second, we identify a set of regular expressions that define the set of signal tweets. This crude classifier of signal tweets based on regular expression matching turns out to be sufficient. Third, we identify features of signal clusters that are independent of any particular topic and that can be used to effectively rank the clusters by their likelihood of containing a disputed factual claim.[4]

The algorithm is evaluated using the Twitter Gardenhose (a 10% sample of the overall tweet stream) to identify rumors on regular uneventful days. We also evaluate it using a large collection of tweets related to the Boston Marathon bombing event. We compare the algorithm to various baselines. It is more scalable and has higher precision and recall than techniques that try to find all trending topics. It detects more rumors, and detects them much earlier than a related technique that treats only debunks or corrections as signals of rumors as well as techniques that repy on tracking all trending topics or popular memes. The performance is also satisfactory in

[4]At the risk of redundancy, we emphasize that our technique does not make any attempt to assess whether rumors are true or not, or classify or rank them based on the probability that they are true. We rank the clusters based on the probability that they contain a *disputed* claim, not that they contain a *false* claim.

an absolute sense. It successfully detects 110 rumors from the stream of tweets about the Boston Marathon bombing event, with an average precision above 50% among the top-ranked candidates. It also achieves a precision of 33% when outputting 50 candidate rumors per day from analysis of the Gardenhose data.

5.2 Related Work

5.2.1 Detection Problems in Social Media

Although rumors have long been a hot subject in multiple disciplines (e.g., [24, 91, 76]), research on identifying rumors from online social media through computational methods has only begun in recent years. Our previous work has shown that particular known rumors can be retrieved with a high accuracy by training a machine learning classifier for each rumor [86]. Here we seek to identify new rumors, not necessarily retrieve all the tweets related to them.

Much previous research has tried to develop classifiers for a more challenging problem than ours, automatically determining whether a meme that is spreading is true or false ([112, 17, 42, 56]). Application domains have included "event rumors" in Sun et al. [99], and fake images on Twitter during Hurricane Sandy [44]. The "Truthy" system attempts a related classification problem, whether a spreading meme is spreading "organically" or whether it is being spread by an "astroturf" campaign controlled by a single person or organization [88, 89].

Identifying the truth value of an arbitrary statement is very difficult, probably as difficult as any natural language processing problems. Even if one knows what the truth is, the problem is related to textual entailment (recognizing whether the meaning of one given statement can be inferred from another given statement), the accuracy of the art of which is lower than 70% on balanced lab data sets [23]. This is even harder for short posts in social media.

Thus, most existing approaches that attempt to classify the truthfulness of spreading memes utilize information beyond the content of the posts, usually by analyzing the collective behavior of how users respond to the target post. For example, many studies identified the popularity of a post (e.g., number of posts that retweeted or replied to the post) as a significant signal. This information is used either directly as features of the "rumor" classifier (e.g., [17, 42, 112, 56, 99, 101]), or as filters to prescreen candidate topics (e.g., to only consider the most popular posts [43] or "trending topics" [17, 42]), or both [17, 42]. Other work identified burstiness [101], temporal patterns [56, 43], or the network structure of the diffusion of a post/topic [89, 17, 95, 56] as important signals.

Most of these features of the tweet collection can only be collected after the rumor has circulated for a while. In other words, these features only become meaningful when the rumor has already reached and been responded to by many users. Once these features become available, we also make use of them in our classifier that ranks candidate rumor clusters. However, since we have set ourselves the easier task of detecting controversial fact-checkable claims, rather than detecting false claims, we are able to rely for initial detection on content features that are available much earlier in a meme's diffusion.

Some existing work uses corrections made by authoritative sources or social media users as a signal. For example, Takahashi and Igata tracked the clue keyword "false rumor" [101]. Both Kwon et al. [56] and Friggeri et al. [35] tracked the judgments made by rumor debunking websites such as Snopes.com. Studies of rumors on Weibo.com also tracked official corrections made by the site [112, 99]. These correction signals are closer in spirit to those we employ. They suffer, however, from limited coverage and delays, only working after a rumor has attracted the attention of authoritative sources. In our experiments we will compare the recall and earliness of rumor detection through our system using both correction and enquiry signals to

a more limited version of our system that uses only correction signals.

Another related problem is detecting and tracking trending topics [71] or popular memes [61]. Even if they are effective at picking up newly popular topics, they are not sufficiently precise to serve as trending rumor detectors, as most topics and memes in social media are not rumors. As an example, Sun et al. collected 104 rumors and over 26,000 non-rumor posts in their experiment [99]. Later in this work, we will compare the precision of the candidate rumors filtered using our method and those filtered through trending topics and meme tracking.

5.2.2 Question Asking in Social Media

Another detection feature used in related work is question asking. Mendoza et al. found on a small set of cases that false tweets were questioned much more than confirmed truths [73]. Castillo et al. therefore used the number (and ratio) of question marks as a feature to classify the credibility of a group of tweets. The same feature is adopted by a few follow-up studies [42, 44].

In fact, the behavior of information seeking by asking questions on online social media has drawn interest from researchers in both social sciences and computer science (e.g., [19, 78, 81, 122]). Paul et al. analyzed a random sample of 4,140 tweets with question marks [81]. Among the set of tweets, 1,351 were labeled as questions by Amazon Mechanical Turkers. Morris et al. conducted surveys on if and how people ask questions through social networks. They analyzed the survey responses and presented findings such as how differently users ask questions via social media and via search engines, and how different cultures influence the behaviors [78, 77, 113]. These studies have proved that question asking is a common behavior in social media and provided general understanding of the types of questions people ask.

To study question asking behavior at scale, our previous work detected and analyzed questions from billions of tweets [122]. The analysis pointed out that the ques-

tions asked by Twitter users are tied to real world events including rumors. These findings inspired us to make use of question asking behavior as the signal for detecting rumors once they emerge.

Though inspired by the value of question marks as features for classifying the truth value of a post, for our purposes we need a more specific signal. Previous work has shown that only one third of tweets with question marks are real questions, and not all questions are related to rumors [81, 122]. In our work, we carefully select a set of regular expressions to identify enquiry tweets that are indicative of rumors.

5.3 Problem Definition

5.3.1 Defining a Rumor

Many variations of the definition of rumors have been proposed in the literature of sociology and communication studies [82]. These different definitions generally share a few insights about the nature of rumors. First, rumors usually arise in the context of ambiguity, and therefore the truth value of a rumor appears to be uncertain to its audience. Second, although the truth value is uncertain, a rumor does not necessarily imply *false* information. Instead, the term "false rumor" is usually used in these definitions to refer to rumors that are eventually found to be false. Indeed, many pieces of *truthful* information spread as rumors because most people don't have first-hand knowledge to assess them and no trusted authorities have fact-checked them yet. Having such intuitions and following the famous work of DiFonzo and Bordia in social psychology [24], we propose a practical definition:

"A rumor is a controversial and fact-checkable statement."

We make the following remarks to further clarify this definition:

- "Fact-checkable": In principle, the statement has a truth value that could be determined right now by an observer who had access to all relevant evidence. This excludes statements that cannot be fact-checked or those whose truth value will only be determined by future events (e.g., "Chelsea Clinton will run for president in 2040.").

- "Controversial (or Disputed)": At some point in the life cycle of the statement, some people express skepticism (e.g., verifications, corrections, statements of disbelief or questions). This excludes statements that are fact-checkable but not disputed (e.g., "Bill Clinton tried marijuana," as Clinton himself has admitted it.).

- Any statement referring to a statement meeting the criteria above is also classified as a rumor. This includes statements that point to several other rumors (e.g., "Click the link `http://...` to see the latest rumors about Boston Bombing.").

The above definition of rumor is effective in practice. As we describe below, human raters were able to achieve high inter-rater reliability labeling statements as rumors or not.

5.3.2 The Computational Problem

Based on the conceptual definition, we can formally define the computational problem of real-time detection of rumors.

Definition V.1. (Rumor Cluster). We define a rumor cluster R as a group of social media posts that are either declaring, questioning, or denying the same fact claim, s, which may be true or false. Let S be the set of posts declaring s, E be the set of posts questioning s, and C be the set of tweets denying s, then $R = S \cup E \cup C$. We say s is a candidate rumor if $S \neq \emptyset$ and $E \cup C \neq \emptyset$.

Naturally, posts belonging to the same rumor cluster can either be identical to each other (e.g., retweets) or paraphrase the same fact claim. Posts that are enquiring about the truth value of the fact claim are referred to as *enquiry posts* (E) and those that deny the fact claim are referred to as *correction posts* (C).

Definition V.2. (Real-time Rumor Detection). Consider the input of a stream of posts in social media, $\mathcal{D} = \langle (d_1, t_1), (d_2, t_2) \dots \rangle$, where $d_i, i \in [1, 2, \cdots]$ is a document posted at time t_i. The task of real-time rumor detection is to output a set of clusters $\mathcal{R}_t = \langle R_{t,1}, R_{t,2}, \dots, R_{t,l} \rangle$ at time t after every time interval Δt, where the fact claim $s_{t,j}$ of each cluster $R_{t,j} \in \mathcal{R}_t$ is a candidate rumor.

Given any time point t where a new set of clusters are output, the clusters must satisfy that

$$\forall R_{t,j} \in \mathcal{R}_t, \exists (d', t') \in R_{t,j} \; s.t. \; t - \Delta t < t' \leq t$$

This means that the output rumor clusters at time t must contain at least one tweet posted in the past time interval Δt. Clearly, a cluster about a fact claim s can accumulate more documents over time, such that $R_{t_1,j} \subseteq R_{t_2,j}$ if $t_1 < t_2$ and $s_{t_1,j} = s_{t_2,j} = s$. Therefore, we can naturally define the first time (t_1 in the previous example) where a rumor cluster about a fact claim s is output as the *detection time* of the candidate rumor s. Our aim is to minimize the delay from the time when the first tweet about the rumor is posted to the detection time.

5.4 Early Detection of Rumors

We propose a real-time rumor detection procedure that has the following five steps.

1. **Identify Signal Tweets.** Using a set of regular expressions, the system selects only those tweets that contain skeptical enquiries: verification questions and corrections. These are the signal tweets.

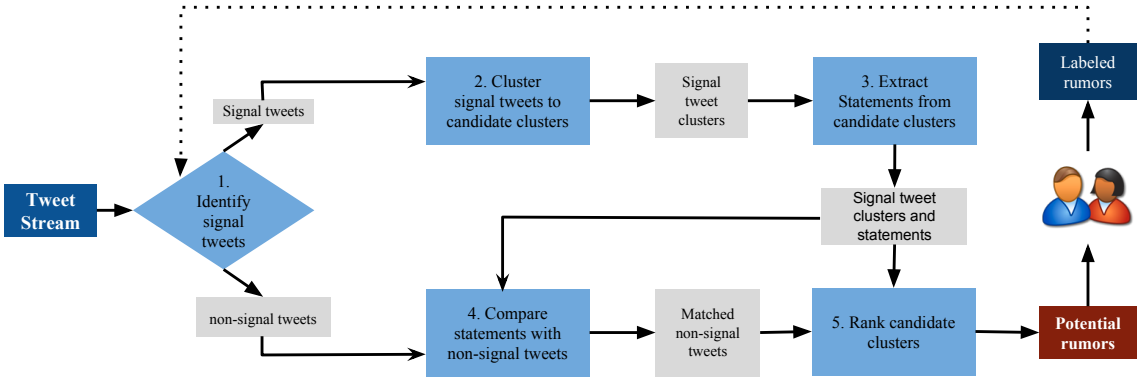


Figure 5.2: The procedure of real-time rumor detection.

2. **Identify Signal Clusters.** The system clusters the signal tweets based on overlapping content in the tweets.

3. **Detect Statements.** The system analyzes the content of each signal cluster to determine a single statement that defines the common text of the cluster.

4. **Capture Non-signal Tweets.** The system captures all non-signal tweets that match any cluster's summary statement, turning a signal cluster into a full candidate rumor cluster.

5. **Rank Candidate Rumor Clusters.** Using statistical features of the clusters that are independent of the statements' content, rank the candidate clusters in order of likelihood that their statements are rumors (i.e., controversial and fact-checkable).

The algorithm operates on a real-time tweet stream, where tweets arrive continuously. It outputs a ranked list of candidate rumor clusters at every time interval Δt, where Δt could be as small as the interval when the next tweet arrives. In practice, it will be easier to think of the time interval as, for example, an hour or a day, with many tweets arriving during that interval.

The system first matches every new tweet posted in that interval to rumor clusters detected in the past, using the same method of capturing non-signal tweets (component 4). Tweets that do not match to any existing rumors will go through the complete set of five components listed above, with a procedure described in Figure 5.2. If very short time intervals are used,

5.4.1 Identify Signal Tweets

The first module of our algorithm extracts enquiry tweets. Not all enquiries are related to rumors [122]. A tweet conveying an information need can be either of the following cases:

- It requests a piece of factual knowledge, or a verification of a piece of factual knowledge. Factual knowledge is objective and fact-checkable. For example: "According to the Mayan Calendar, does the world end on Dec 16th, 2013?"
- It requests an opinion, idea, preference, recommendation, or personal plan of the recipient(s), as well as a confirmation of such information. This type of information is subjective and not fact-checkable.

We hypothesize that only verification/confirmation questions are good signals for rumors. In addition, we expect that corrections (or debunks) are also good signals. To extract patterns to identify these good signals, we conducted an analysis on a labeled rumor dataset.

Discover patterns of signal tweets We analyzed 10,417 tweets related to five rumors published in [86], with 3,423 tweets labeled as either verifications or corrections. All tweets are lowercased and processed with the Porter Stemmer [85]. We extracted lexical features from the tweets: unigrams, bigrams and trigrams. Then we calculated the Chi-Square score for each feature in the data set. Chi-Squared test is a

Pattern Regular Expression	Type
is (that — this — it) true	Verification
wh[a]*t[?!][?1]*	Verification
(real? — really ? — unconfirmed)	Verification
(rumor — debunk)	Correction
(that — this — it) is not true	Correction

Table 5.2: Patterns used to filter Enquiries and Corrections

classical statistical test of independence and the score measures the divergence from the expected distribution if one assumes a feature is independent of the class label [32]. Features with high Chi-Square scores are more likely to appear only in tweets of a particular class. Patterns which appear excessively in verification and correction tweets but are underrepresented in other tweets were selected to detect signal tweets. From the patterns with high Chi-Square scores, human experts further selected those which are independent of any particular rumor. The patterns we selected are listed in Table 5.2.

As a way to identify all the tweets containing rumors, this set of regular expressions has relatively low recall. Even on the 3,423 tweets labeled as either verifications or corrections in our training data, only 572 match these regular expressions. In the signal tweet identification stage, however, it is far more important to have a high precision. Low recall of *signal tweets* may still be sufficient to get high recall of *signal clusters*. By identifying patterns that are more likely to appear only in signal clusters, even though these patterns only appear a few times inside each rumor cluster, our framework can make use of them to detect many rumors. Note that although current patterns are discovered from a data set of five rumors, we could in principle rerun this process after more rumors are labeled the detection framework, shown as the dotted line in Figure 5.2.

5.4.2 Identify Signal Clusters

When a tweet containing a rumor emerges, many people either explicitly retweet it, or create a new tweet containing much of the original text. Therefore, tweets spreading a rumor are mostly near duplicates, as illustrated in Table 5.1. By clustering, we aim to group all the near duplicates, which are either retweets or tweets containing the original rumor content.

There are many different clustering algorithms such as the K-Means [68]. Many have a high computational cost and/or need to keep an $N \times N$ similarity matrix in memory. Given that we expect tweets about the same rumor to share a lot of text, we can trade off some accuracy for efficiency. In contrast to exploratory clustering tasks where documents may be merely similar, we want to cluster tweets that are near duplicates. Therefore, using an algorithm such as connected component clustering can be efficient and effective enough for our purposes. A connected component in an undirected graph is a group of vertices, every pair of which are reachable from each other through paths. An undirected graph of tweets is built by including an edge joining any tweet pair with a high similarity.

We use the Jaccard coefficient to measure similarity between tweets. Given two tweets d_a and d_b, the similarity between d_a and d_b can be calculated as:

$$J(d_a, d_b) = \frac{|Ngram(d_a) \cap Ngram(d_b)|}{|Ngram(d_a) \cup Ngram(d_b)|}$$

Here $Ngram(d_a)$ and $Ngram(d_b)$ are the 3-grams of tweets d_a and d_b. Jaccard distance is a commonly used indicator of the similarity between two sets. The similarity values from 0 to 1 and a higher value means a higher similarity.

To further improve efficiency, we use the Minhash algorithm [15] to reduce the dimensionality of the Ngram vector space, which makes calculating Jaccard similarity much faster. The Minhash algorithm is used for dimensionality reduction and fast

estimation of Jaccard similarities. In our approach, we randomly generate 50 hash functions based on the md5 hash function. Then we use the 50 corresponding Minhash values to represent each tweet. In our implementation of the connected component clustering algorithm, we set the threshold for adding an edge at 0.6 (60% of the hashed dimensions for the two tweets are equal).

The connected components in this graph are the clusters. We create a cluster for each group of three or more tweets connected together. Connected components can be found by either breadth-first search or depth-first search, which has a linear time complexity $O(E)$, where E is the number of edges. Since we want to cluster tweets that are near duplicates, setting a high similarity threshold (0.6) yields a relatively small number of edges. At this point, the procedure will have obtained a set of candidate rumor clusters $\mathcal{R}$. The next stage extracts, for each cluster R_i, the statement s_i that the tweets in the cluster promote, question, or attempt to correct. In our approach, for each rumor cluster, we extract the most frequent and continuous substrings (3-grams that appear in more than 80% of the tweets) and output them in order as the summarized statement. We keep the summarization component simple and efficient in this study, though algorithms such as LexRank [28] may improve the performance of text summarization.

5.4.3 Capture Non-signal Tweets

After the statement that summarizes the tweets in a signal cluster is extracted, we use that statement as a query to match similar non-signal tweets from the tweet stream, tweets that are related to the cluster but do not contain enquiry patterns. To be consistent, we still use the Jaccard similarity and select tweets whose similarity score with the statement is higher than a threshold (0.6 in our implementation). This step partially recovers from the low recall of signal tweet detection using limited signal patterns.

Note the efficiency gain that comes from matching the non-signal tweets only with the statements summarizing signal tweets. In particular, it is not necessary to compare each of the non-signal tweets with each other non-signal tweet. Non-signal tweets may form other clusters but we do not need to detect those clusters: they do not contain nuclei of three connected signal tweets and thus are unlikely to be rumor clusters.

5.4.4 Score Candidate Rumor Clusters

After we have complete candidate rumor clusters, including both signal and non-signal tweets, we score them. A simple way to output clusters is to rank them by popularity. The number of tweets in a cluster measures the statement's popularity. The most popular candidates, however, may not be the most likely to be rumors. There may be statistical properties of the candidate rumor clusters that are better correlated with whether they really contain disputed factual claims.

We extracted 13 statistical features of candidate clusters that are independent of any particular substantive content. We then trained classifiers using these features, to obtain a better ranking function. The features are listed as follows.

- Percentage of signal tweets (1 feature): the ratio of signal tweets to all tweets in the cluster.
- Entropy ratio (1 feature): the ratio of the entropy of the word frequency distribution in the set of signal tweets to that in the set of all tweets in the cluster.
- Tweet lengths (3 features): (1) the average number of words per signal tweet; (2) the average number of words per any tweet in the cluster; and (3) the ratio of (1) to (2).
- Retweets (2 features): the percentage of retweets among the signal tweets and the percentage of retweets among all tweets in the cluster.

- URLs (2 features): the average number of URLs per signal tweet and the average number per any tweet in the cluster.
- Hashtags (2 features): the average number of hashtags per signal tweet and the number per any tweet in the cluster.
- @Mentions (2 features): the average number of usernames mentioned per signal tweet and the number per any tweet in the cluster.

We use rumor clusters labeled by human annotators to train a classifier that ranks the candidate clusters by their likelihood of being rumors (detail described in section 5.3). We select two commonly used classifiers, the Support Vector Machine (SVM [22]) with the LIBSVM package [18], and the Decision Trees [13] with a Matlab implementation. [5] The detailed results are shown in Section 5.

5.5 Evaluation

5.5.1 Experiment Setup

In this section, we present empirical experiments to evaluate the proposed method of early detection of rumors.

5.5.1.1 Data Sets

We first selected two different collections of tweets. One focuses on a specific high-profile event, i.e. the Boston Marathon bombing in April 2013. The other consists of a random sample of tweets from a month that was not unusually eventful.

Boston Marathon Bombing (Boston). Two bombs exploded at the finish line of the annual Boston Marathon competition on April 15th, 2013.[6] We chose this context as a typical example of unpredictable real-world events.

[5] `http://www.mathworks.com/help/stats/classificationtree-class.html`

[6] `http://en.wikipedia.org/wiki/Boston_Marathon_bombings`, retrieved on March 7, 2015.

To obtain a complete set of tweets related to this event, we collected tweets containing keywords such as "Boston," "marathon," and "explosion" and their variations, starting several hours after the explosion, using the official tracking API of Twitter's. The tracking API returned all tweets containing those keywords after 23:29 GMT. We used the Twitter search API to collect tweets posted before this time point which contained the same set of keywords. In summary, we collected 10,240,066 tweets through the search API (13:30 GMT, April 14 to 23:29 GMT, April 15) and 23,001,329 tweets through the tracking API (23:29 GMT, April 15 to May 10, 2013), adding up to 30,340,218 unique tweets in the entire data set.

GARDENHOSE. Besides the stream related to a major event, we are also interested in the performance of the proposed method in detecting rumors from everyday tweets. We thus collected a tweet stream in a random month of the year 2013 (November 1 to November 30, 2013), through the official stream API with Gardenhose access (10% sample of the real-time stream of all tweets). This data set contains 1,242,186,946 tweets. Although the size is forty times larger than the BOSTON set, we anticipated that the density of rumors in this everyday tweet stream may be lower.

To process such data sets of over a billion records, we implemented our methods in MapReduce and conducted the experiments on a 72 core Hadoop cluster (version 0.20.2). The main components of our framework, including filtering, clustering and retrieval algorithm are implemented using Apache Pig (version 0.11.1).

5.5.1.2 Baselines and Variants of Methods

To obtain a comprehensive understanding of the effectiveness of the overall method, the identifiers of signal tweets, and the algorithms used to rank statements, we tested six variants of the method. The first four variants all rank candidate rumors purely

by popularity, the number of tweets in the cluster. They vary in the algorithm used to identify signal tweets. The last three variants all use both enquiry and correction tweets as signals. They vary in the algorithm used to rank the candidate rumor clusters.

Variant 1 (baseline 1): Trending Topics. This straightforward baseline method directly clusters all the input tweets. It treats all the tweets as signal tweets rather than selecting only a subset. This method echoes the common approaches to detecting trending topics and then identifying rumors among them [17, 42]. After tweets are clustered and statements are extracted, this baseline method simply outputs the top candidate clusters with the largest number of tweets.

Variant 2 (baseline 2): Hashtag Tracking. Hashtags are well recognized signals for detecting trending and popular topics and have been used previously in rumor detection [89, 101]. As a second baseline, popular hashtags (i.e., those that appear more than 10 times) are used to filter the signal tweets. Tweets containing these hashtags are clustered and statements are extracted from these clusters. Clusters with the largest number of tweets are presented to the user.

Variant 3 (baseline 3): Corrections Only. One novel contribution of our approach is the utilization of enquiry tweets as early signals of rumors. To test the performance of these signals, we downgraded our identifier of signal tweets by only using correction tweets as filtering signals, i.e., tweets containing the correction patterns in Table 5.2, including "rumor," "debunk," or "(this—that—it) is not true." Certain correction patterns such as the phrase "false rumor" have been used previously in the literature to identify rumors [101].

Variant 4: Enquiries and Corrections. Besides the baseline methods, we included

three variants that treat both enquiries and corrections as signal tweets, using the complete set of patterns from Table 5.2. To enable comparison with the baselines, variant 4 still ranks the candidate rumor clusters purely by popularity.

Variant 5: SVM ranking. This version ranks the candidate rumor clusters based on their scores using the trained *SVM* classifier. Like Variant 4, it treats both enquiries and corrections as signal tweets.

Variant 6: Decision tree ranking. This version ranks the candidate rumor clusters based on their scores using the trained *Decision Tree* classifier. Like Variants 4 and 5, it treats both enquiries and corrections as signal tweets.

5.5.1.3 Ground Truth

We recruited two human annotators to manually label candidate rumors (i.e. rumor clusters as defined in Section 5.3) as either a real rumor or not. The annotators made decisions based on the statement extracted from the cluster, actual tweets in the cluster, and other useful information about the statement through Web search. To train the annotators, we developed a codebook according to the definition of rumors discussed in Section 5.3 which includes both the definition and examples of rumor and non-rumor statements. After being trained, both annotators labeled all the top-ranked candidate rumor clusters extracted by either the *Popularity* method, the *Decision Tree* method, or the *Correction Signal* method, from the first week of the GARDENHOSE data set and the first two days and the eighth day of the BOSTON data set. At most 10 clusters per hour per method were annotated for the BOSTON data set and at most 50 clusters per day per method were annotated for the GARDENHOSE data set. These added up to 639 candidate rumor clusters. The inter-rater reliability was satisfactory, achieving a Cohen's Kappa score of 0.76. Such a high agreement also

demonstrates the coherence of our definition of rumors. For the statements the two annotators did not agree on, an expert labeled them and broke the tie. Another 1,440 clusters generated by the first two baseline methods (*Trending Topics* and *Hashtags*) on the same 72 hours of the BOSTON data set were then labeled by one of the two annotators after they were well trained. It took an average about 80 hours for each annotator to finish the labeling task including training.

5.5.1.4 Evaluation Metrics

We selected several quantitative metrics to evaluate the effectiveness and efficiency of our proposed method. We calculated precision@N, which is the percentage of real rumors among the top N candidate rumor clusters output by the a method. Since it is not practical in general to manually label a complete data set with tens of millions of tweets and hundreds of thousands of clusters, we cannot directly evaluate the actual recall of a rumor detection method. However, the number of rumors each method returns can be an indirect way to understand whether one method can detect more rumors than another. Another important dimension of the effectiveness of a rumor detection system is how early a rumor can be detected. We calculated the detection time, the time when the algorithm was first able to identify it as a candidate rumor. Finally, we also evaluated the scalability of the proposed method by plotting the scale of the data against the running time of the algorithm.

5.5.2 Effectiveness of Behavioral Signals

We evaluated the effectiveness of rumor detection algorithms using different signals. We first compared the precision of the top-ranked candidate rumor clusters output by different methods. For a fair comparison, all methods ranked the candidate rumor clusters simply using the *popularity* (i.e. number of tweets in each cluster).

5.5.2.1 Precision of Candidate Rumor Clusters

We compared the precision of our proposed methods using both enquiry and correction signals with all three baseline methods on the BOSTON data set. Note that both the baselines 1 and 2 (*Trending Topics* and *Meme Tracking*) have to cluster a huge number of tweets, nearly all the incoming tweets of every time period, hence they cannot handle the scale of all tweets in the GARDENHOSE data set. Therefore, we compare the proposed methods with only Baseline 3 (*Correction Signals*) on the GARDENHOSE data set. These results are summarized in Table 5.3. Clearly, the use of both enquiry and correction signals typically detects more rumors than using no signal (trending topics) or using memes (meme tracking), and the top-ranked rumor clusters are much more precise. Using both enquiry and correction signals, our method detected 110 rumors from the stream of tweets related to the Boston Marathon bombing, with an average precision@10 above 50% (half of the top 10 candidate clusters output by the system are real rumors). On the stream of everyday tweets, this method detected 92 rumors from a random month of 2013, with the average precision@50 above 26% (one fourth of the top 50 clusters output by the system are real rumors).

Some interesting observations can be made from these results. Detecting trending topics or tracking popular memes can reveal some rumors, but they both suffer from a low precision among the candidate clusters (lower than 10%), and thus miss many rumors if the user can only check a certain number of candidates (i.e., 10 per hour or 50 per day). This is because both methods inevitably introduce many false positives, popular statements that are not disputed. Detection using correction signals only also misses half of the rumors in the Boston event, probably because the behavior of debunking rumors is less common than enquiries in social media, as it certainly requires more effort of the users.

Using correction signals achieves a high precision among detected candidates. This

Table 5.3: Precision of rumor detection using different signals. Candidate rumors ranked by popularity only. Maximum number of output rumor clusters: 10 per hour for BOSTON and 50 per day for GARDENHOSE.

Method	Data Set	Candidates Detected	Real Rumors	Precision
Trending Topics	BOSTON	720	71	0.099
Hashtag Tracking	BOSTON	720	35	0.049
Corrections only	BOSTON	109	52	0.466
Enquiries+ Corrections	BOSTON	194	**110**	**0.521**
Corrections only	GARDENHOSE	312	87	**0.279**
Enquiries+ Corrections	GARDENHOSE	350	**92**	0.263

is not surprising as statements already explicitly corrected or referred in tweets as "rumors" are likely to in fact be disputed factual claims. Interestingly, using enquiry as well as correction signals yields a similar precision.

5.5.2.2 Earliness of Detection

One of the most important objectives of our study is to detect emerging rumors as early as possible so that interventions can be made in time. Correction signals may appear only in a later stage of a rumor's diffusion. If this is the case, detecting rumors using such signals may have less practical value, as the rumors may have already spread widely. To verify this and further understand the usefulness of enquiry signals, we measured the earliness of detection. We computed the difference between the time points at which the same rumor was first detected by different methods, assuming that the algorithms are run in batch mode to output results only once per hour. The results are summarized in Table 5.4.

We first compare the method which uses both enquiry and correction signals with Baseline 3 (correction signals only). Since different methods may yield different

Table 5.4: Earliness of detection comparing to Enquiries+ Corrections: enquiry signals help to detect rumors hours earlier.

Method	Data Set	Rumors detected	Rumors matched	Average delay
Corrections only	BOSTON	52	46	+4.3h
Trending Topics	BOSTON	71	53	+3.6h
Hashtag Tracking	BOSTON	35	31	+2.8h

clustering results and/or statements for the same rumor, we manually matched the 52 rumors detected by *correction only* from the BOSTON data set with the 110 rumors detected by both *enquiries and corrections*. We obtained 46 rumors detected in the top 10 results per hour by both methods.

There are 27 rumors that *enquiries and corrections* detected at least one hour earlier than *correction only*. The two detected the rest of the 19 rumors in the same hour. The detection of a rumor using *enquiries and corrections* is on average 4.3 hours earlier.

In Figure 5.3, we plot the detection time of each matched rumor using the two different methods. The x axis represents time from April 15th 17:00 (GMT) to April 22nd 17:00 (GMT). Each row in the plot presents a rumor. A dark blue bar marks the hour when the rumor is first detected by our method, and a red bar marks the hour when the rumor is first detected by Baseline 3 (*correction signals*). Some example rumors are annotated in the figure.

For example, at 20:00 (GMT) April 15th the *enquiries and corrections* algorithm would have output the popular rumor that the police identified a Saudi national as the suspect. This was one hour earlier than people started to realize it was false and tweet corrections. For another widespread rumor about an 8-year-old girl who died in the explosion, *enquiries and corrections* identified it almost one day earlier than tracking correction signals only.

In theory, how early can rumors be detected through enquiry signals, if candidate rumors were output continuously rather than hourly? We marked the time points when the system captures **at least three** signal tweets. On average, a real-time system that tracks enquiry signals can *hypothetically* detect a rumor after its first appearance in 9.6 minutes. To collect at least three correction tweets, a method has to wait for 236.7 more minutes on average. Not all candidate rumor clusters are actually output by our algorithms, so precision would have to be sacrificed to detect all rumors that quickly.

We also compare *enquiries and corrections* with Baseline 1 (trending topics), and Baseline 2 (meme tracking), on the earliness of detection. For Baseline 1, we matched the 71 rumors detected by *trending topics* with 110 rumors detected by our method. We obtained 53 common rumors detected by both methods. On average these rumors were detected as trending topics 3.6 hours later than using enquiry+correction signals. For Baseline 2, we matched the 35 rumors detected by *meme tracking* with rumors detected by our method. 31 of them are matched. On average these rumors were detected as trending memes 2.8 hours later than using enquiry+correction signals. The earliness of our detection method compared to other methods passed paired-sample t-test at significance level of 0.01.

In brief, we see that the use of enquiry tweets as signals not only detects more rumors, but also detects them hours faster than tracking trending topics or popular memes. Tracking correction signals, although it yields high precision, is the latest among all methods.

5.5.3 Ranking Candidate Rumor Clusters

We assessed the benefits produced by ranking the candidate clusters, using the 13 statistical features described in the previous section. We tested the performance of ranking functions based on Support Vector Machines and Decision Trees compared

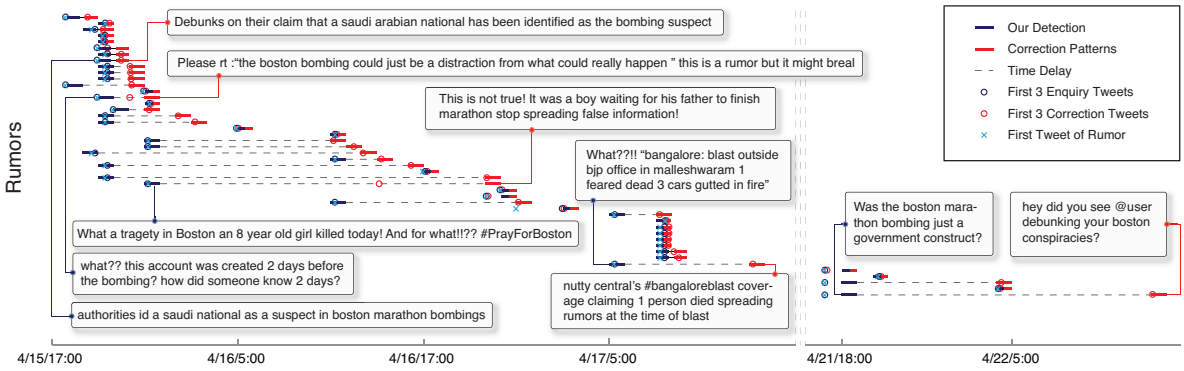


Figure 5.3: The earliness of rumor detection using enquiry signals or correction signals: enquiry tweets detect rumors much earlier.

with two other baseline methods. The first baseline ranks the clusters by the number of tweets inside each cluster, referred to as *Popularity*. The second baseline ranks the clusters based on the retweet ratio in the cluster of tweets, which was reported as an indicative feature of rumors [101]. The second baseline is referred to as *Retweet Ratio*.

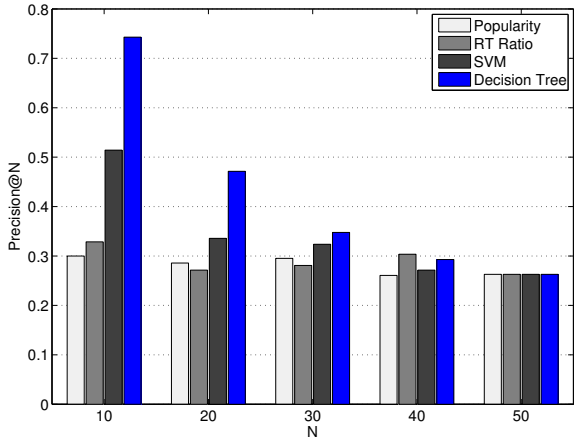


Figure 5.4: Precision@N of different ranking methods

We applied the ranking algorithms to the 350 candidate rumor clusters labeled by human annotators (the 50 most popular clusters for each of the seven days from 2013-11-1 to 2013-11-7 in GARDENHOSE data set). We used 6 days of labeled results to train the classifiers and the remaining day's results to test the algorithms. We

did a 7-fold cross validation, holding out each of the seven days and computing the average performance. Figure 5.4 shows the results. We used Precision@N to evaluate different ranking algorithms. In this figure, we can see that *retweet ratio* has comparable performance to *popularity*, which remains about 0.3 no matter what N is. Our ranking algorithm significantly improves the Precision@N when N is small. The precision of the top 10 statements per day is above 0.7 using a Decision Tree, which outperforms SVM and the baseline methods. Of course, as N approaches 50, the precisions equalize since all the algorithms are essentially re-ranking the 50 most popular items. Note we are dealing with the classification task on only hundreds of examples and the 13 features we extracted are in different scales. In such case a decision tree is easier to tune than the more sophisticated SVM [16]; this may explain why the Decision Tree algorithm achieved a better performance than SVM. The improvement of Precision@10 and Precision@20 made by the decision tree compared to other methods passed the paired-sample t-test at significance level of 0.01.

Next, we tested whether the decision tree algorithm would find more rumors if it was able to suggest its own 50 top-ranked candidate clusters among all the candidates instead of reranking the most popular 50. In the previous figure, it was restricted to re-ranking the 50 most popular ones. We evaluated the performance using Precision@N. Figure 5.5 shows the results for the GARDENHOSE data set. We used tweets from 2013-11-1 to 2013-11-3 in the GARDENHOSE data set to train the decision tree and then used tweets from 2013-11-4 to 2013-11-7 to test, with popularity ranking as the baseline. Results show that we can not only improve Precision@N when N is small, but also find more rumors in 50 output statements. 33% of our output statements are rumors. The improvement of Precision@N when $N \leq 40$ made by our ranking algorithm passed the paired-sample t-test at significance level of 0.01.

In order to verify that the ranking algorithm is not overfitting only one data set, We also applied the decision tree trained using 7 days of labeled results in GARDEN-

HOSE data set to rank rumor clusters detected hourly from BOSTON data set. We got similar results as in Figure 5.5. The average precision at 2, 4, 6, 8 an 10 in an hour is improved compared to *popularity* based ranking. The features used at the top levels of the decision tree include percentage of signal tweets and the average numbers of words, URLs and @mentions per any tweet in the cluster.

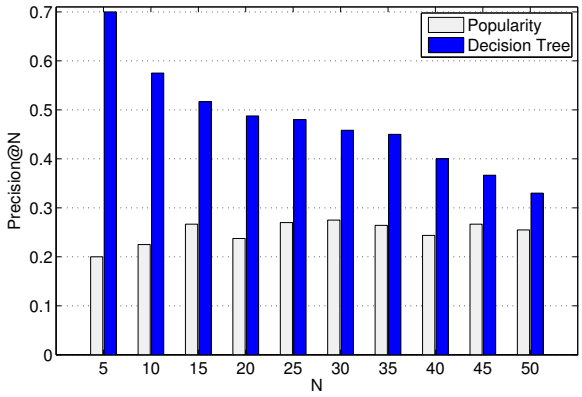


Figure 5.5: Precision@N if rumor clusters are ranked by the Decision Tree. One third of top 50 clusters are real rumors.

5.5.4 Efficiency of Our Framework

We have shown that our rumor detection algorithm is effective in detecting rumors in their early stage with reasonable precision. We now show that our framework is computationally efficient. Our framework first filters tweets with specific signals, then uses clustering to detect statements in this smaller group of tweets and at last outputs potential rumor statements. Compared to approaches that first generate trending topics and then identify rumors, we reduce the cost significantly in the detection process.

We first tested the time cost of our algorithm (*decision tree ranking* in Section 5.2 which uses both enquiry and correction signals) compared to baseline methods of running the algorithm on one batch of tweets from one time interval. We started from 1,000 tweet batches randomly sampled from tweets in our data set, then increased

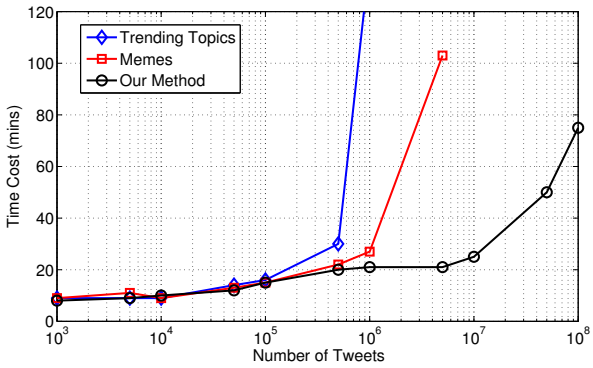


Figure 5.6: Running time vs. batch size.

the number of tweets to 100,000,000 exponentially. Figure 5.6 shows the results. The x-axis shows the number of tweets to be processed in log scale. The baseline methods here are the Baseline 1 and 2 from Section 5.2, which try to detect trending topics or popular memes (hashtags) first. For baseline methods, we used the same clustering and ranking implementations as our method except they don't filter tweets with enquiry or correction signals and they don't have to retrieve tweets back after clustering. When the scale reaches one million tweets, Baseline 1 cannot finish in hours. Our method performs consistently and does not take much longer even at the 10 million scale: it can process 100 million tweets in about 80 minutes. It is intuitive that *Meme Tracking* achieves an intermediate performance. It clusters only those tweets that contain popular and trending hashtags, and thus scales somewhat better than clustering all tweets to find trending topics, but is still not as efficient as our method, which clusters a much smaller number of signal tweets.

We also tested our algorithm on one month's tweets from the GARDENHOSE data set, collected at November 2013. We set the time interval to be a day. The average number of tweets every day in the GARDENHOSE data set was about 40 million. As we would expect, experiment results indicate that the time cost does not increase significantly after processing several days, even with the accumulation of older rumor clusters. On average it took 28.77 minutes for our algorithm to finish detecting rumors

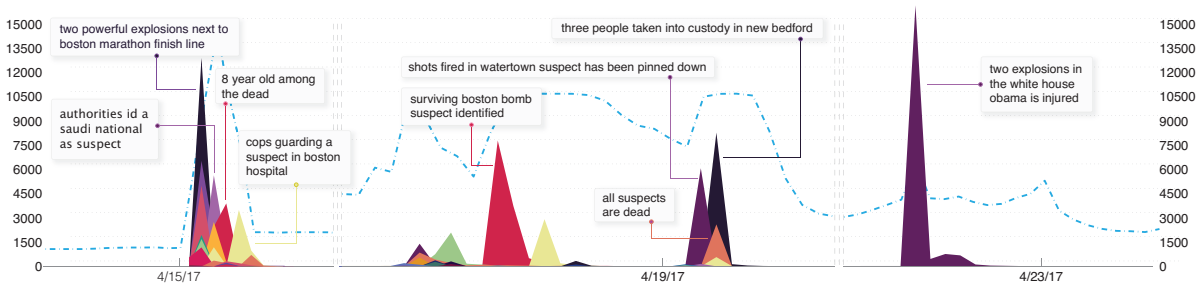


Figure 5.7: Tracking detected rumors about Boston Marathon bombing

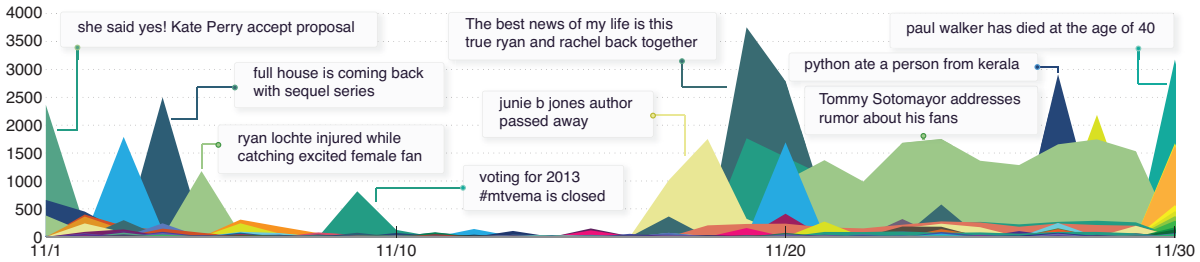


Figure 5.8: Tracking detected rumors in November 2013

each day and took 14.38 hours in total to process the 1.2 billion tweets in the entire month.

5.5.5 Discussion

We have shown in the experiments that tweets asking verification questions or making corrections to controversial statements are very important signals of rumors early in their life cycle. Some users who have an information need of evaluating a rumor will post tweets either asking verification questions to express suspicions, or will correct the rumor after their investigation. Verification questions are particularly useful because they appear much sooner and thus facilitate earlier detection of rumors.

Not all clusters that include tweets that ask verification questions or use correction phrases are actually rumors. We identified 13 different statistical features of clusters of tweets, such as average tweet length and percentage of signal tweets, etc. By training

a decision tree model, we built a powerful ranking algorithm that ranks tweet clusters by how likely they are to be rumors (i.e., controversial, fact-checkable claims). The precision we can achieve is much higher than without the ranking algorithm.

We demonstrated that our proposed framework can scale up well. By clustering only the small set of signal tweets, we avoided the computational cost of detecting popular statements or trending topics from the entire corpus. Our proposed framework is robust even if the number of tweets to be processed exceeds 100 million.

To give the readers a sense of the end-to-end operation of our system, we present the rumors detected in the two data sets. For the BOSTON data set, we identified the top 5 candidate rumor clusters each hour. Figure 5.7 plots the number of tweets hourly of each identified rumor statement. The dotted blue line in the background shows the number of tweets arriving in that hour.

Although the small set of regular expressions may yield a low recall of enquiry signals, when the candidate rumor clusters detected by our system are labeled by human experts, they can be used to enrich the set of signals. Indeed, using statistical feature selection techniques [32], one can extract features that are highly representative in the rumor clusters and underrepresented in the non-rumor clusters. These patterns can be approved by human experts and added into the pipeline to identify more signal tweets, thus improving precision and recall of rumor detection. By using rumors of our two data sets labeled by annotators, we have already discovered a few additional promising patterns such as "scandal?", or "fact check." We leave the iterative improvement of the signal patterns to future work.

One may be curious about what types of rumors are being circulated in a random week of tweets. From the GARDENHOSE data set, we output the 10 top-ranked rumors for each day and tracked them (Figure 5.8). We also show example statements extracted from the most popular rumor clusters. Most everyday rumors turn out to be gossip about celebrities, with occasionally emerging anecdotes like "python ate

person."

5.6 Summary

One post of a rumor in social media can sometimes spread beyond anyone's control. A rumor about two explosions in the White House is a perfect example of how a single tweet out of more than 9,000 tweeted in the same second spreads and causes real damage.

In social media, users share information based on different types of needs, including the need to verify controversial information. We point out that such information needs can not only help spread rumors, but also provide the first clue for detecting them.

Based on this important observation, we applied our proposed general framework in this dissertation and designed a rumor detection system. We cluster only those tweets that contain enquiry patterns (the signal tweets), extract the statement that each cluster is making, and use that statement to pull back in the rest of the non-signal tweets that discuss that same statement. We then rank the clusters based on statistical features that compare properties of the signal tweets within the cluster to properties of the whole cluster.

Extensive experiments show that our proposed method can detect rumors effectively and efficiently in their early stage. With a small Hadoop cluster, in about half an hour we process 10% of all the tweets posted in one day on Twitter. If we output 50 candidate statements, about one third of them are real rumors, and about 70% of the top ranked 10 clusters are rumors.

There is still considerable room to improve the effectiveness of the rumor detection method. We can improve the filtering of enquiry and correction signals by training a classifier rather than relying on manually selected regular expressions. We can further develop a method to automatically update the filtering patterns in real time to prevent potential spamming of the detection system. We can also explore more

features for each statement and train a better ranking algorithm for candidate rumor clusters. Another direction is to adopt this method to detect rumors automatically and generate a large data set of rumors, which can benefit many potential analyses such as finding features that are potentially correlated to the truth value of a rumor, or analyzing general diffusion patterns or the life cycle of rumors.

CHAPTER VI

Detecting Persuasion Campaigns in Social Media

> "Non-Islamic, non-foreign-motivated terrorist actions have killed at least as many Americans on American soil as those who were promoted by jihadists. But what we have also seen is ISIL evolve, because of the sophistication of their social media, to a point where they may be inspiring more attacks - even if they're self-initiated, even if they don't involve complex planning - than we would have seen some time ago."
>
> Barack Obama

Persuasion campaigns consist of a series of activities that attempt to gain public support for an opinion or course of action [83]. The motivation can be legitimate or unethical and manipulative such as ISIS's (or ISIL) recruiting campaign on Twitter. As the use of social media websites increases, it becomes crucial to distinguish the wide variety of persuasion campaigns from the massive amounts of social media posts.

In this chapter, we describe the construction of an event detection system for detecting social media persuasion campaigns. The objective is to identify as many persuasion campaigns as possible, so that users, the media, etc., can analyze, understand, and possibly prevent potential damage.

As discussed in previous chapters, existing approaches for detecting social media events such as persuasion campaigns usually consist of two steps. First, they cluster

the posts corresponding to the social media event. Second, they apply a classifier to identify persuasion campaigns from other events. To improve the effectiveness and efficiency of this process, we adopt the general framework introduced in Chapter III that utilizes a filtering step to filter irrelevant content by the identifying signals of targeted events.

In Chapter V, we showed that enquiry posts are the signals to detect rumors. Conducting content analysis on posts related to rumors allowed us to manually extract a set of unique language patterns and phrases such as "is it true??" in order to identify enquiry posts. Unlike rumors, however, signals for detecting persuasion campaigns do not necessarily contain unique patterns and simple phrases. In fact, potential signals may contain many patterns and their combinations. Therefore, we need an algorithm that automatically finds good signal patterns from noisy data.

In this chapter, we quantify the key characteristics of the signals in detecting persuasion campaigns and build metrics to evaluate a given signal identifier. We propose to learn a binary classifier for identifying signals by jointly maximizing the metrics. We train the classifier on labeled (known) persuasion campaigns and use it in our persuasion campaign detection system. We use Twitter data to validate the detection framework. The results show that our proposed method effectively detects persuasion campaigns in the form of post clusters compared to baseline methods.

6.1 Overview

By March 21, 2016, presidential candidates from both parties had raised $619 million according to the Center for Responsive Politics. Most of the money was spent to operate and support various types of campaigns. It was estimated that the amount spent on political advertising for the 2016 presidential election was 20% more than the amount spent for the 2012 election. More than half of an estimated $1 billion budget

Figure 6.1: Word Cloud of a Subset of Hashtags for the Presidential Campaign in Jan 2016

for digital media was spent on social media persuasion campaigns[1]. Clearly, social media websites such as Twitter and Facebook, had become marketing battlefields for online persuasion campaigns.

According to a study on Facebook [12], Facebook feeds had a significant impact on increasing turnout by a total of 340,000 votes in the 2012 presidential election. Another study [20] found that people, especially young people, were likely to participate political discussion or activity online. As an example, Figure 6.1 shows a word cloud we generated by tracking hashtags related to the 2016 presidential campaigns on Twitter in January 2016. In addition to candidates and politicians representing both parties, celebrities, activists, companies and organizations, had built and operated social media profiles as a way to promote their own images, music albums, movies, products, or social activities.

Absent well-established quality control mechanisms such as those used by traditional news media, online marketing and social media persuasion campaigns has proven risky. Using social media, the Islamic State of Iraq and Syria (ISIS) successfully recruited as many as 3,000 individuals from outside the region join to join as jihadists[2]. Early detection of ISIS?s online campaigns has become a national security concern for many countries including the U.S.

[1] http://www.huffingtonpost.com/r-kay-green/the-game-changer-social-m_b_8568432.html

[2] http://www.cbsnews.com/news/isis-uses-social-media-to-recruit-western-allies/

Table 6.1: Group of Tweets Related to a Persuasion Campaign on Twitter

@GregoryMeeks,In Nigeria, human life and human right has lost its meaning say NO to #TyrantBuhari Free Nnamdi Kanu #FreeBiafra
The truth will set you free, but first it will make you miserable. #FreeBiafra #FreeNnamdikanu Biafra referendum not war
These children all died of acute starvation, over 2 million of them. Yet nobody talks about #Biafra. #FreeBiafra now
Why must Nigeria be on every bad news in the world.#FreeBiafra from Nigeria.

In general, researchers in many fields will benefit from an automatic social media persuasion campaign detection system. Assembling massive datasets of online persuasion campaigns leads to deeper analyses of patterns, strategies, and diffusions. Any persuasion campaign can be viewed from multiple perspectives by studying the related and oppositional campaigns. Research outcomes help the media, which is not a monolith, distinguish between real and so-called fake news and present unbiased coverage.

In this chapter, we build a social media event detection system to detect online persuasion campaigns. Social media persuasion campaign corresponds to collective activities with the attempt to gain public support for an opinion or course of action. It can be identified in the form of social media post cluster. Posts inside each post cluster are related to the same persuasion campaign. Table 6.1 shows an example of a group of tweets in an persuasion campaign of freeing Biafra from Nigeria.

Detecting persuasion campaigns in social media by language and content automatically is challenging because campaign organizers and supporters employ different strategies to disseminate posts and enlarge a campaign. This significantly adds the difficulty for automatic detection systems to understand and discover these campaigns by their language and content. For example, ISIS?s recruiting campaigns intend to

use language that computer algorithms and human analyses fail to detect.

Existing approaches to detect persuasion campaigns usually apply classifiers to groups of posts, i.e., post clusters. This follows the clustering and classification pipeline shown in Figure 2.1 and discussed in Chapter II. Post clusters corresponding to social media events can be generated by clustering posts given a post similarity measurement [4]. The classifiers are trained to identify persuasion campaigns from the post clusters. Various types of collective features, i.e., content features, network features, statistics, and temporal features are used to train the classifiers. The collective features can help identify persuasion campaigns with uncertain language patterns.

Similar to social media rumors and other targeted events discussed in the previous chapters, detecting persuasion campaign from a large collection of posts or post stream by firstly generating post clusters is time-consuming. Detection accuracy is also affected by classifiers trained on and applied to imbalanced dataset, because relatively rare persuasion campaigns are distributed in large quantities of irrelevant social media posts, and most post clusters are not persuasion campaigns.

One way to overcome these challenges is to filter out the majority of irrelevant content before clustering the posts into groups and extracting the collective features. In Chapter III, we proposed a general framework to detect targeted social media events by identifying signals and filtering out non-signal posts. Signals are posts that appear in every target clusters usually not in non-target clusters. Because every target cluster will have a few signals, by clustering only signals and matching non-signal posts into signal clusters, our detection framework generates fewer non-target clusters before classification. Therefore both efficiency and effectiveness for event detection are improved.

The reason why signals exist in certain types of targeted events is that, the emergence and spreading of the events can lead to common user reactions despite the content of an event. For example, in Chapter V, to detect social media rumors, we

found that several unique user activities, such as posting enquiry posts, has been found mostly in rumor clusters than in non-rumor clusters. The enquiry posts are highly accurate signals to detect rumors.

Similarly, as persuasion campaigns are targeted to grain supporters, despite actual content and topic, common user reactions such as supporting or opposing exist. The signals reflecting the reactions cannot be easily controlled by persuasion campaign organizers trying to avoid detection. If we can identify signals for persuasion campaigns, we can build persuasion campaign detection systems using our general framework. Unlike social media rumors, however, persuasion campaigns express opinions instead of factual claims and the resulting expressions of common reactions can be more diverse. The signals for detecting persuasion campaigns may not contain certain simple and unique language patterns.

To apply our detection framework to persuasion campaigns and other targeted events, we need to identify their signals, even if it is difficult to do so from intuition and domain knowledge. Therefore, our goal is to automatically learn a signal identification algorithm from a labeled dataset of persuasion campaigns. We model the signal identification process as a binary classification problem, i.e., given a post, predicting whether or not it is a signal. Although a standard way to learn this binary classification algorithm is to optimize the objectives calculated on labeled signals and non-signal posts, we don't have the labels since we do not know what exactly are the signals for detecting persuasion campaigns. To learn a signal classifier directly using labeled persuasion campaigns instead of labeled signals, we need to define the objective and loss function for classifier training based on the labeled events, i.e., target clusters. To achieve this, we first need to understand the characteristics of the signals, so that we can quantify them and use them as objectives.

As discussed above, to detect persuasion campaign post clusters from a post stream using signals, the signals should exist only in the target clusters. Every target

cluster should contain a few signals. The efficiency improvement of our detection framework depends on the frequency of signals in a post stream, and the effectiveness improvement depends on the accuracy of the signals in the target clusters and not in other clusters. Based on these understandings, we define the objectives that can be calculated on labeled target clusters to learn a signal classifier.

After we learn the signal classifier and use it to identify signals, we apply our general framework to build the persuasion campaign detection system. We apply a clustering algorithm on identified signals, match non-signals to signal clusters, and classify the candidate clusters. To evaluate our approach, we apply our detection system to Twitter, using the hashtags of each tweet, i.e., post, as features for clustering. We generate a labeled persuasion campaign dataset to train the signal classifier and evaluate its properties. Then we compare the learned signal classifier with manually crafted signal identifiers by conducting a content analysis similar to the method described in Chapter V. Finally, we evaluate the effectiveness and efficiency of our persuasion campaign detection system by comparing with several baseline approaches.

The contributions of Chapter VI are as follow:

- We apply our framework for detecting targeted events to detect social media persuasion campaigns. We provide a formal definition of persuasion campaigns and build a system to detect them from Tweet stream.

- We discuss and quantify the characteristics of signals for detecting targeted events. We define multiple objectives to evaluate any given signal identification algorithm on labeled targeted events.

- We build a signal classifier to identify the signals for persuasion campaign detection. We learn the classifier by jointly optimizing our defined objectives such as precision, recall, and filter rate.

- We generate a labeled dataset of online persuasion campaigns containing prod-

uct promotion, political campaigns, social activities and spams for our detection system. And we evaluate it on real Twitter data.

The rest of this chapter is organized as follows. In section 6.2, we discuss related work on social media event detection, especially existing detection frameworks for persuasion campaign detection . In section 6.3, we formally define our detection problem. After that we will define our signal identification algorithm as a binary classifier or a signal function and propose a method to learn it automatically. We then describe our system implementation in section 6.6. In section 6.7 we describe how we generate our dataset. In section 6.8 we give the evaluation results. We conclude and discuss future work in the last section.

6.2 Related Work

In this section, we summarize existing research on persuasion campaign detection, including specific problem definitions, features studied and adopted, and how we use these existing approaches to build helped our proposed framework. We also discuss existing work on multiple instance learning and anomaly detection, which inspire us on learning signal classifier from labels of target clusters.

6.2.1 Social Media Persuasion Campaign Detection

There are many studies of the characteristics of different online campaigns and the detection problem [1] [64] [59] [58] [27] [119] [118] [87] [52] [63] [65]. While most work focuses on detecting persuasion campaigns using the clustering and classification framework discussed in Chapter II, many studies devise clustering strategies with meaningful post similarity measurements and effective collective features for classification.

Lee et al. study campaign detection by building message graphs based on different constraints on content similarity between social media posts/messages [59] [58]. The

campaigns have a broader definition than the persuasion campaigns we use. Lee et al. generate campaigns as post clusters based on different level of similarity and network structures. They provide us with insights into adopting a proper similarity measurement or grouping criterion for clustering posts in building our persuasion campaign detection system.

Besides clustering strategies, other studies explore and discover important features of persuasion campaigns. Tan et al. study the lanuange patterns of bursting events [102] and persuasive statements [103] on social media platforms such as Twitter and Reddit. Their most recent study [103], finds out that language patterns play an important role in predicting persuasiveness, especially the patterns existing between the interactions of opinion holders and users holding opposing opinions. This finding aligns with our hypothesis in our worker that signals of detecting persuasion campaigns exist because of users' pro and con common reactions.

Another work shows that many spams and promoting campaigns in social media contain URL, and it studies how to detect spam and promoting campaigns grouped by similarity measurement defined based on the URLs [119] [118]. A classifier is trained by using a campaign's statistical features and using a included URL to identify whether the clusters or candidate campaigns are positive or negative.

As language and content features extracted from individual post or group of posts are important in detecting persuasion campaigns, many approaches also study how other types of collective features help in training classifiers to identify persuasion campaigns. In a recent paper of detecting promoted social media campaign [27], multiple types of features are used to train a classifier to identify whether posts clusters grouped by hashtags are promoted by Twitter or not. They compared and analyzed the importance of each type of feature, including network features extracted from mentioning and retweeting networks, user account features, temporal features, and content features. The authors find that network features and content features

are the most important features in detecting promoted campaigns.

Some approaches focus on spatial and temporal features extracted from clusters to detect different types of persuasion campaigns such as opinion spams [63], campaign promoters [64], and promoted hashtags [1]. By detecting internet advocates [87] and cyber armies [52] in political campaigns, studies reveal the importance of the statistical features from users and posts collections, such as message and propagation patterns, network structures, posting times, etc..

All the above mentioned approaches study different aspects of persuasion campaign, e.g., opinion spams, promoting hashtags, advocates and promoters etc. Therefore, the features they used in training their specific classifiers are meaningful to be adopted in training our classifier to detect persuasion campaigns after filtering and clustering. In our work, our classifier combines content and many of the features cited above, such as statistics about URL, retweets and mentioned users.

6.2.2 Anomaly Detection and Multiple Instance Learning

Unlike existing clustering and classification approaches, our detection system discovers signals and filters non-signal posts before generating post clusters. In order to learn a binary classifier on post as signal identification algorithm without labels on posts, we directly use labels of post clusters, i.e., whether a given post cluster is a persuasion campaign or not. In general, our problem is similar to general machine learning techniques such as multiple instance learning [7] and anomaly detection [117], where each item's label is related to the group's label.

Graph anomaly detection or collective anomaly detection, proposes to detect groups of anomaly activities by analyzing both individual activities and groups. Generative probabilistic models, e.g., the Multinomial Genre Model [111], Flexible Genre Model [110], and GLAD model [116], have been proposed to model the relationship between individual activity and the group of activities.

Multiple instance learning studies the problem of classification where class labels are assigned to bags of items instead of individual items. In a binary classification scenario, the law of inheritance in this problem defines the label of a set as positive if at least one item in the set is positive [7]. Similarly, a item level classifier is learned to classify sets. Multiple instance learning has been applied to detecting social media events. For example, Wang et al. develop a event detection framework to detect posts related to civil unrest events using multiple instance learning. In their study, they train a sentence level classifier with labels of news posts and adopt the classifier to identify sentences related to civil unrest in each post and classify the posts as civil unrest related.

We adopt the key concept of multiple instance learning and extend it for detecting social media persuasion campaigns. Specifically, we design our specific objectives for learning a signal classifier considering both effectiveness and efficiency, because we use the signal classifier as a filtering step and later apply a cluster level classifier to post clusters.

6.3 Problem Definition

In this section, we modify and extend the definition, given in Chapter III, for detecting social media persuasion campaigns. Our detection system takes post streams as the inputs, and output detected target post clusters. We use the following notations shown in table 6.3.

6.3.1 Detecting Target Clusters in Social Media

Most social media event detection problems treats each event as a cluster of social media posts and/or other types of user activities. To reduce ambiguity, our notion of event is equivalent to a post cluster [4].

The problem of detecting persuasion campaigns from social media post stream can

$p =< f, t >$	A post in social media, corresponding to a tuple of content feature f and time stamp t.
$P = \{p_1, p_2, \dots\}$	A steam of social media posts.
$G : P \times P- > \{0, 1\}$	Similarity measurement or grouping criterion for posts. A function that returns 1 if two input posts belong to one cluster/group, and 0 if not.
$c = \{p_1, p_2, \dots, p_k\}$	A cluster of social media posts. These posts are grouped using specific grouping criterion.
$C = \{c_1, c_2, \dots\}$	A stream of clusters of social media posts.
$T : C- > \{0, 1\}$	A function that returns the label of any given cluster c. T(c) = 1 if c is a target cluster and 0 otherwise. T can be a codebook defining the target cluster or a golden standard.

Table 6.2: Notation used in problem definition

be modeled as a general social media event detection problem for a targeted event type. As discussed in Chapter III, we use clustering algorithm to group posts into post clusters and adopt supervised learning algorithms to identify target clusters. Therefore, the inputs of our system contain a social media post stream, such as a tweet stream; a grouping criterion for the clustering algorithm, e.g., a similarity measurement for posts such as the cosine similarity on text; and a set of labeled persuasion campaigns for supervised learning. Given the three inputs, the detection algorithm should output a stream of target clusters corresponding to persuasion campaigns.

Problem Definition 1. Social Media Persuasion Campaign Detection.

INPUTS:

- A stream of social media posts: $P = \{p_1, p_2, \dots\}$,
- A grouping criterion: $G : P \times P -> \{0, 1\}$.
- A labeled dataset of persuasion campaigns:

$$L = \{< c_1, T(c_1) >, < c_2, T(c_2) >, \dots, < c_k, T(c_k) >\}$$

OUTPUTS:

- A binary classifier: $\hat{T} : C -> \{0, 1\}$, where

$$\hat{T} = arg\min_{T'}(\sum_{<c,T(c)>\in L} Loss(T'(c), T(c)))$$

  $Loss(label1, label2)$ is a loss function that returns a loss if $label1 \neq label2$.

- A stream of target clusters: $C = \{c_1, c_2, \dots\}$, where
  $\forall c \in C : c \subseteq P \wedge \hat{T}(c) = 1$, and $\forall p_i, p_j \in c \in C : G(p_i, p_j) = 1$

The problem definition extends the general targeted event detection definition in Chapter III for persuasion campaign detection. To obtain a labeled dataset of persuasion campaigns in the form of post clusters, we need a codebook of definitions to decide whether a given post cluster corresponds to a persuasion campaign or not. We then use it decide whether or not a given post cluster corresponds to a persuasion campaign. The next section describes persuasion campaigns on social media platforms such as Twitter and the codebook of definitions.

6.3.2 Understanding Persuasion Campaigns

Our definition is summarized from existing definitions of persuasion campaigns in the field of phycology, communication and politics [90] [37], and modified to fit the

context of social media. Unlike some existing discussions of social media persuasion campaigns in the literature and from some approaches for detecting different types of campaigns [50] [60], we study general social media persuasion campaigns without specific topics.

A persuasion campaign in social media can be a political campaign, propaganda, spam or opinion spam, a promotion, etc. We summarizes different types of persuasion campaigns studied in related works and propose our definition of persuasion campaign. A persuasion campaign is a movement targeted to change public opinion and gain support for a specific topic. It can be represented by and extracted from a collection/cluster of social media posts, i.e., a post cluster.

Definition VI.1. Persuasion Campaign Post Cluster (Persuasion Campaign): Given a cluster of social media posts c, if most of the posts in c relate to a specific topic designed to change public opinion or gain support for an event or an entity, then c represents a persuasion campaign.

If it satisfies the definition, a subset of the persuasion campaign cluster can also be a persuasion campaign. A single post designed to change public opinion or gain support for an event or an entity, is a persuasive post. There are two major characteristics of a persuasion campaign.

Attitudes/Opinions: Persuasion campaigns are related to opinions, not fact-checkable events. Persuasion campaigns can be reduced to statements that express attitudes on events and/or entities, but not the events themselves. Attitudes/opinions can also urge an action/event/entity.

Examples: A collection of posts on the topic "Recreational marijuana is legalized in Colorado" is a fact/event, whereas a collection of posts on the topic "Recreational marijuana should be legalized in Michigan" is a persuasion campaign. A collection of posts on the topic "Check this out! http://www.fake-product.com" is a persuasion

campaign. Although it expresses no clear opinion if the underlying goal is to urge users to learn about a product or event (e.g, the url promotes a product), it is still a persuasion campaign. If a celebrity tweets ?Everyone please follow me!? the original tweet and the retweets are a persuasion campaign.

Controversy: Opinions that do not need additional support and opinions that most users agree with are not persuasion campaigns.

Examples: A collection of posts on the topic "We should sleep at night every day", or "The president of the U.S. should work for people in the U.S." are not persuasion campaigns, because these opinions are held by most people, whereas "President Obama should put more effort into improving the education system for the American people" is a persuasion campaign.

Common categories of persuasion campaigns in social media include but are not limited to the following.

(1) Promotion of an entity, such as products or a company. Example: #EatASnicker hashtag.

(2) Political persuasion, such as asking for support for a candidate or opposing other candidates, or asking for support for a political issue. Examples: #MakeAmericaGreatAgain to support Republican presidential candidate Donald Trump and #FeeltheBern to support Democratic candidate Bernie Sanders.

(3) Social issues and protests. Example: www.worldcancerday.org/about/2016-2018-world-cancer-day-campaign

Although not every persuasion campaign can be identified as one of these categories, many persuasion campaigns belong to these categories are very influential. Usually in spreading these campaigns, many social media users who support them help in the spreading and many other strategies such as automatic spammers and promoters are used.

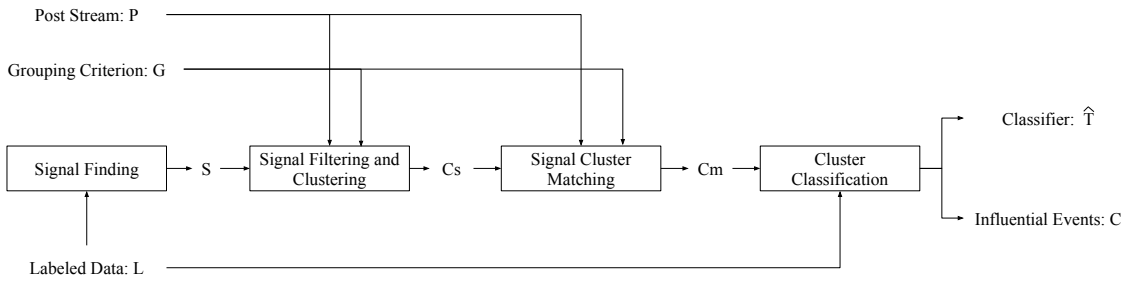


Figure 6.2: Our system framework

Social media users, automatic spammers, and promotional strategies disseminate the majority of persuasion campaigns via hashtags, which are words or phrases preceded by a ?#? sign. Many persuasion campaign detection systems rely on hashtags to extract post clusters [27]. We consider hashtags as features that help us group posts into post clusters. We use our definition of persuasion campaigns to label the post clusters and their corresponding hashtags in order to obtain a labeled dataset.

6.4 Framework Overview

In this section, we introduce our framework for detecting social media persuasion campaigns. This is an extension of the general framework in III. We include a signal finding step in which we learn a signal classifier to identify signals. Also different from our framework in rumor detection, where we process post stream batch by batch, we develop our framework to handle stream data, so that real time detection can be performed. Figure 6.2 illustrates the framework.

6.4.1 Signal Finding

In the signal finding step, we will learn a signal classifier, i.e., a function S, from a labeled dataset of cluster L. In L, clusters have either 1 (positive) or 0 (negative) labels. Below is the definition of this step, specifying the input and output.

Definition VI.2. Signal Finding.

INPUTS:

- A labeled dataset generated by a set of clusters and a codebook:

$$L = \{< c_1, T(c_1) >, < c_2, T(c_2) >, \ldots, < c_k, T(c_k) >\}$$

- Σ: parameters used to define the objective of a good signal function.

OUTPUT:

- Signal classifier $S : P- > \{0, 1\}$, for a given post p, S(p) = 1 if p is a signal post, S(p) is 0 otherwise.

Signal classifier S is a function that maps any social media post p to 1 or 0. If $S(p) = 1$, p is a signal, and $S(p) = 0$ means p is a non-signal post. In the previous rumor detection system [123], S can be formulated as a regular expression matching algorithm that matches a set of manually selected language patterns corresponding to an enquiry post. To identify the signals for persuasion campaigns, we model S as a weighted linear combination of content features. This function captures more complex patterns for identifying signals than regular expressions, and is efficient in filtering post stream. We will discuss objectives used to learn it from labeled dataset of clusters in Section 6.5.

6.4.2 Signal Filtering and Clustering

After learning a signal classifier S, we use it to filter input post stream P. Then we cluster the signals into candidate clusters C_s using a given grouping criterion or similarity measurement method G. The definition are as follows:

Definition VI.3. Signal Filtering and Clustering.

INPUTS:

- A stream of social media posts: $P = \{p_1, p_2, \dots\}$.
- A grouping criterion: $G : P \times P -> \{0, 1\}$.
- Signal classifier: $S : P -> \{0, 1\}$.
- Threshold λ on the number of signal posts contained in a cluster.

OUTPUT:

- A stream of clusters: $C_s = \{c_{s_1}, c_{s_2}, \dots\}$, where $\forall p_i, p_j \in c \in C_s : G(p_i, p_j) = 1$, $\forall p \in c \in C_s : S(p) = 1$, and $\forall c \in C_s : c \subseteq P \wedge |c| \geq \lambda$.

We note that the stream of candidate clusters C_s are clusters of signals. A candidate cluster is a cluster that has more than a pre-defined number (λ) of signals. We set λ to three, which is the same as the number of signals set in our rumor detection system.

6.4.3 Signal Cluster Matching

Since a target cluster contains both signals and non-signal posts. After clustering signals, we need to match new posts from post stream P to candidate clusters C_s. The new posts can be either signals or non-signal posts. Thus, we generate clusters that contain all relevant posts under group criterion G. Because our system is built for stream data and every post is processed only once for each step, the non-signal posts processed before a candidate cluster is created will not be matched. The definitions of input and output are as follows:

Definition VI.4. Signal Cluster Matching.

INPUTS:

- A stream of social media posts: $P = \{p_1, p_2, \dots, \}$.

- A stream of clusters: $C_s = \{c_{s_1}, c_{s_2}, \dots\}$.
- A grouping criterion: $G : P \times P -> \{0, 1\}$.

OUTPUT:

- A stream of clusters: $C_m = \{c_{m_1}, c_{m_2}, \dots\}$, where
  $\forall c_{m_i} \in C_m, \exists c_{s_j} \in C_s : c_{s_j} \subseteq c_{m_i} \wedge c_{m_i} - c_{s_j} \subseteq P$,
  and $\forall p_i, p_j \in c \in C_m : G(p_i, p_j) = 1$.

6.4.4 Cluster Classification

This final step is to classify candidate clusters and identify target clusters. It combines two steps, i.e., the training of a classifier from the labeled dataset of target clusters, and applying the classifier to identify the target clusters from candidate cluster stream C_m. The definitions of input and output are as follows:

Definition VI.5. Cluster Classification.

INPUTS:

- A stream of clusters: $C_m = \{c_{m_1}, c_{m_2}, \dots\}$.
- A labeled dataset generated by a set of clusters and a codebook:

$$L = \{< c_1, T(c_1) >, < c_2, T(c_2) >, \dots, < c_k, T(c_k) >\}$$

OUTPUTS:

- An binary classifier: $\hat{T} : C -> \{0, 1\}$, where
  $\hat{T} = arg\min_{T'}(\sum_{<c,T(c)>\in L} Loss(T'(c), T(c)))$
- A stream of clusters: $C = \{c_1, c_2, \dots, \}$, $C \subseteq C_m$,
  where $\forall c \in C : \hat{T}(c) = 1$, and $\forall c_{m_i} \in (C_m - C) : \hat{T}(c_{m_i}) = 0$.

We output the stream of target clusters C identified by a learned classifier $\hat{T}$.

In this section, we have introduced our framework and define the inputs and outputs of each step. The next sections explain how we implement our persuasion campaign detection system, beginning with the design of objective functions for training the signal classifier.

6.5 Training Classifier for Signal Identification

In this section, we introduce our method of learning a signal classifier automatically from labeled dataset of target clusters. We discuss the characteristics of the signals in detecting targeted events. We formalize and build an objective function with labeled clusters as input for each of the characteristics. By joint optimizing the objectives, we learn a binary classifier for signal identification.

6.5.1 Objectives of Signal Identification

Signals are an social media posts that are the indications of target clusters. Signals exist in every target cluster and every target cluster contains a few signals. This is the uniqueness of signals for targeted event detection. Our previous work on rumor identified enquiry signals by language patterns manually selected from feature analysis conducted on labeled rumors [123]. To generalize the the process, we propose to learn a signal classifier from the labeled target clusters, by understanding and quantifying the key properties of the signals.

In our detection framework, a candidate cluster is defined as one having more than a pre-defined number (λ) of signals. For any post cluster c, whether or not it is a candidate cluster is represented by a function F of c.

$$F(c) = \delta(\sum_{p \in c} S(p) - \lambda) \tag{6.1}$$

Where

$$\delta(v) = \begin{cases} 1, v \geq 0 \\ 0, v < 0 \end{cases}$$

In general, we want to find a set of candidate clusters containing as many positive target clusters and as few negative target clusters as possible. Therefore, our signal identification algorithm, i.e., the signal function S defined in section 6.4, need perform well in the following three aspects: (1) High Recall: To preserve enough relevant posts for generating target clusters. (2) High Precision: To preserve posts only from target clusters, so that clusters of the preserved posts contain the fewest non-target clusters. (3) High filter rate: To filter most irrelevant posts before clustering.

We represent and evaluate the three aspects for any given signal function S by using function F.

High Recall : For any c that is a target cluster $T(c) = 1$, F(c) should be 1 in most cases. Given a labeled dataset L of target clusters, we evaluate signal function S by the recall of its corresponding classification criterion F by calculating the percentage of true positive cases in all positive clusters, i.e., True Positive ((TP)) rate:

$$TP(S, L) = \frac{\sum_{<c,T(c)>\in L}(F(c) \times T(c))}{\sum_{<c,T(c)>\in L} T(c)} \tag{6.2}$$

High Precision For any c that is not a target cluster $T(c) = 0$, F(c) should be 0 in most cases. We evaluate signal function S by the precision of its corresponding classification criterion F, by calculating the percentage of true negative cases in all negative cases, i.e., True Negative (**TN**) rate:

$$TN(S, L) = \frac{\sum_{<c,T(c)>\in L}((1 - F(c)) \times (1 - T(c)))}{\sum_{<c,T(c)>\in L}(1 - T(c))} \tag{6.3}$$

High Filter Rate The ratio of signal posts to all posts should be low, so that most posts can be filtered. We evaluate signal function S by calculating the number of signal posts in all labeled dataset and compared it to the number of all posts, i.e., Filter Rate (**FR**):

$$FR(S, L) = 1 - \frac{\sum_{p \in c \wedge <c,T(c)> \in L} S(p)}{\sum_{<c,T(c)> \in L} |c|} \tag{6.4}$$

Ideally, if a signal function S and its corresponding classification criterion F can obtain both high recall and high precision, the final post cluster classification step is unnecessary. Signal identification algorithms, however, need to filter most of the tweets before clustering to improve efficiency, i.e., high filter rate. Therefore, all three aspects are needed to force signal function S to look for posts with uncommon features that distribute mostly in target clusters. This is a much harder goal to achieve than the objectives for the final post cluster classification algorithms, which intuitively look for features that are common in target clusters but not in other clusters. Therefore, a signal function needs to find a good tradeoff among these three properties. Combining with a final post cluster classification algorithm in our detection framework, we can detect targeted events both efficiently and effectively.

A good signal identification algorithm needs to balance the effectiveness with efficiency. It first needs to have a high recall or high True Positive rate ((TP)) described in Equation 6.2, so that after signal filtering, most of the positive target clusters can be preserved. Since we have a final cluster classification step, signal filtering can look for tradeoff between a promising precision or True Negative rate (**TN**) described in Equation 6.3 and high Filter Rate (**FR**) described in Equation 6.4.

Meanwhile, since a signal function learns the features and properties of individual posts instead of the entire cluster, it uses of different set of features than the third classification step. Because of the constrains on the filter rate, these features are

usually uncommon in general but related to a small proportion of users' common reactions to the targeted events. Usually, the reactions are both content irrelevant and more robust to over-fitting. Applying a good signal identification algorithm allows us to eliminate many potential false positive cases. In our experiments, we will evaluate this by comparing the final accuracy of detection systems using different signals and not using signals.

In summary, to learn a good signal classifier and signal function S, three objectives need to be considered. A high **TP** is required for not filtering out positive target clusters before classification step. A high **TN** increases the final accuracy after the classification. And a high **FR** increases the efficiency of detection system.

6.5.2 Learning Signal Classifier

We model signal function S(p) using a linear combination of features from post p:

$$S(p) = \delta(\mathbf{W}\mathbf{\Phi}(\mathbf{p})) = \delta(\sum_{\phi(p)^{(i)} in \mathbf{\Phi}(\mathbf{p})} (w_i \times \phi(p)^{(i)})), \tag{6.5}$$

where $\mathbf{\Phi}(\mathbf{p})$ is the feature vector of post p, $\phi(p)^{(i)}$ is the ith dimension of the features of post p, $\mathbf{W}$ is a vector of the feature weights we need to learn from labeled dataset, and δ is a sign function for classification. Usually it can be replaced by exponential function such as in Logistic Regression.

To learn a good signal function S, we need to consider **TP**, **TN** and **FR**. Our objective function combines the three properties as:

$$Obj(S, L) = r_{TP}TP(S, L) + r_{TN}TN(S, L) + r_{FR}FR(S, L), \tag{6.6}$$

where r_{TP}, r_{TN}, and r_{FR} are the weights of the three properties. We need to make r_{TP} higher than r_{TN} and r_{FR} as discussed above, because high recall is more important than precision and filter rate in the signal function. Therefore, given a

labeled dataset L of target clusters, we can learn the feature weights $\mathbf{W}$ in signal function S by maximizing the objective $Obj(S, L)$.

We use gradient descent to learn signal function S:

$$\mathbf{W_{k+1}} := \mathbf{W_k} + \eta \frac{\partial Obj(S, L)}{\partial \mathbf{W}}, \tag{6.7}$$

where η is the learning rate and k is the number of iterations in the gradient descent. To calculate the gradient, we use sigmoid function to replace function $\delta(v)$:

$$\delta_S(v) = 1/(1 + \exp^{-kv}),$$

where k is a hyper parameter determining the steepness of the sigmoid function. In this work, we set k = 10.

6.6 Framework Implementation

In this section, we describe how we implement our detection framework to detect persuasion campaigns in social media. We first build a signal identification algorithm by learning the signal function S. Given a set of labeled persuasion campaigns L, we learn S by maximizing the objective function described in Equation 6.6. Using the same labeled dataset L, we also train the classifier $\hat{T}$ for classifying post clusters. With learned S, we apply our framework to a stream data in Twitter.

Given input social media post stream P, we filter non-signal posts and cluster signals into candidate clusters and then match every new incoming post from stream P to the candidate clusters. Next, we run post cluster classifier $\hat{T}$ on the candidate clusters in every short time interval and output the detected target clusters.

6.6.1 Signal Learning

To learn signal function S, we first extract features from posts. We use ngrams, i.e., uni-gram, bi-gram, and tri-gram as features. We filter low frequent ngrams which appear less than five times and remove high frequent ngrams which appear more than 100,000 times in the labeled dataset.

In our optimization process with gradient descent, due to non-convexity from joint maximizing three objectives, the initialization of the variables is very important. To obtain good initial states, we randomly generate initial weights 100 times, directly calculate their objective values, and select 10 with the highest value. We start training with these 10 sets of initial weights separately. It takes 100 iterations before we observe convergence on the objective value. We end by selecting the result with the highest objective value as the final output of signal function S.

6.6.2 Signal Filtering, Clustering and Matching

Having learned the signal function S, we now apply it to filter the input post stream P. We cluster the signals and clusters with more than λ signal posts become candidate clusters. And we merge every incoming post in the post stream that can be matched with candidate clusters into the matched clusters. We set λ to 3, which we also use in our rumor detection system (see Chapter V for details).

The process of filtering, clustering and matching is not a typical stream clustering problem. Unlike existing stream clustering approaches, our system does not cluster all posts, i.e., we only need to output and update clusters having more than λ signals. We extend some of the existing stream clustering approaches [94] [3] to develop our own stream signal filtering, clustering and matching algorithm.

Algorithm 1 shows the pseudo code of our signal filtering, clustering and matching algorithm. We output the updated information of candidate clusters C_m whenever a new post matches with a candidate cluster or we detect a new candidate. In our

Data: Social Media Post Stream P, Signal Function S
Result: A Stream of Matched Candidate Clusters: C_m
initialization ;
$C_s = \{\}$; /* signal clusters */
$C_m = \{\}$; /* Stream of Candidate clusters */
while *post p in P* **do**
  matched = MatchToCluster(p, C_m) ; /* Check if post p can be matched a cluster in C_m */
  if *matched* **then**
    Output C_m[matched].insert(p) ; /* Insert post into matched candidate cluster */
    continue;
  end
  if *S(p)* **then**
    matched = MatchToCluster(p, C_s);
    if *matched* **then**
      C_s[matched].insert(p);
      if *size(C_s[matched] ¿= λ)* **then**
        Output C_m.insert(C_s.pop(matched)) ; /* Move signal clusters with more than λ signal posts to C_m */
      end
    end
    else
      C_s.insert(CreateCluster(p)) ; /* Create a new signal cluster for non-matched signal post */
    end
  end
end

Algorithm 1: Signal Filtering, Clustering and Matching

implementation, we use and inverted index on cluster centers to reduce the time required to match post with clusters.

Our system has the flexibility of using different post features and similarity measurements for clustering. For example, if tweets are clustered by Jaccard similarity on ngrams, we can extract clusters of near duplicate statements. Since the posts that disseminate one persuasion campaign can have totally different ngram content, we use hashtags as features and measure their cosine similarity between tweets. To match post into candidate cluster, we measure the similarity between the post and cluster center. We set a high threshold (0.99) of the cosine similarity so that the majority of tweets inside one cluster will contain same hashtag or set of hashtags.

6.6.3 Classifying a Group of Tweets

To train the classifier to identify whether a candidate cluster is a persuasion campaign or not, we extract both the content features and other statistical features from the clusters.

For the content features, we extract uni-grams, bi-grams and tri-grams from the labeled dataset and apply stemming and filtering to both low-frequent (less than 5 times) and high frequent (more than 100,000 times) terms.

For the statistical features, we use features similar to previous studies of rumor detection and persuasion campaign detection [123] [119] [27]. These features include: percentage of retweets in culture, percentage of tweets with URL in cluster, percentage of tweets with mentioned users in cluster, percentage of tweets with hashtags in cluster, average length of tweets, content entropy, and average number of words.

We choose linear kernel SVM as our classification algorithm. By running a five-fold cross-validation on our labeled persuasion campaign dataset, the average accuracy of using content feature only is 0.819 and the average accuracy of using both content feature and statistics is 0.853. Therefore, we train classifier with the two types of

features combined, and use the trained classifier to identify persuasion campaigns from the candidate cluster stream C_m.

6.7 Experiment Setup

In this section, we describe how to generate the labeled dataset and the dataset for evaluating our persuasion campaign detection system. We also list baseline methods used in evaluating detection performance, including signal learning.

6.7.1 Datasets

We use post stream of social media platform Twitter to conduct our experiment. We have been tracking tweets using Twitter's gardenhose API that provides 10% of random sampled tweets. Our archive contains tweets from Jan 31st 2015 to Feb 7th 2016. Different proportion of these tweest are used to for either generating labeled dataset or evaluating detection performance.

6.7.1.1 Labeling Persuasion Campaigns

To generate a labeled dataset for training classifiers, we conduct an online search of the most popular hashtags associated with persuasion campaigns on Twitter. We identify 80 hashtags, covering different categories discussed in Section 6.3, such as politics, spams, and commercials. We track them on Twitter and generate post clusters for each hashtag. We also randomly select 80 hashtags from all the hashtags appearing in January 2016 as potential negative examples in the labeled dataset.

Our positive clusters and their corresponding hashtags derive from five sources:

- Hashtags of supporting and opposing 2016 U.S presidential campaign candidates: We select 21 popular pro or con hashtags relating with presidential candidates that appear in January 2016. Each cluster contains tweets having the corresponding hashtag.

- Promoting tweets and hashtags. We identify 303,618 promoted english tweets in January 2016 from our Gardenhose collection. We group them by hashtag. We manually select the most popular 25 hashtags that are relevant with promotion content. Finaly, we retrieve all tweets containing one of the 25 hashtags to create tweet clusters.

- Superbowl commercials. We identify a total of 17 official commercial hashtags for both the 2015 and the 2016 Superbowl football game. For 2015 commercials, we create clusters by searching tweets from Jan 31st to Feb 2nd 2015. And for 2016 we use tweets for the month of January for 2016.

- ISIS propaganda. Since ISIS uses Twitter to post propaganda content, we find some of its supporting or opposing campaigns. We manually identify and track 7 relevant hashtags from Nov 27th 2015 to Jan 31st 2016.

- Urging support for social issues or actions. We identify 10 popular hashtags in this category. They are mostly related to public reactions to specific controversial events, e.g., hashtag #RefugeesNotWelcome during the Syria migrant crisis in Europe in January 2016, and #EndYulinFestival for protesting a festival that celebrating dog eating.

We develop a codebook based on our definition of social media persuasion campaigns in Section 6.3. We employ two human annotators to identify real persuasion campaigns from the 160 hashtags. We retain only the post clusters having at least 50 tweets and having same label by both annotators. The totals are 66 positive clusters and 76 negative clusters. Our labeled positive target clusters represented by their hashtags can be found online [3].

Besides model training, we also need to label results of different detection systems for evaluation purposes. We ask the annotators to manually label the output post

[3] `http://www-personal.umich.edu/~zhezhao/image/labels`

clusters from our system and baseline systems using the codebook. The inter-rater readability measured by the kappa score [45] between the two annotators is 0.767.

6.7.1.2 Testing Dataset

We use tweets from Feb 1 to Feb 7 2016 as the input post stream. There are about 36 million tweets daily and 255,050,763 tweets in total for the study period. We run our detection system and baseline detection systems and output detected persuasion campaigns for each day of our dataset. We evaluate the detection performance by averaging the daily performance.

6.7.2 Evaluation Methods

We extend our general detection framework for targeted social media events shown as in Figure 3.1 for persuasion campaign detection and include a learning process to learn a signal classifier for signal filtering. To evaluate our work, we need to evaluate both signal learning, and the end-to-end persuasion campaign detection performance.

6.7.2.1 Signal Learning

To evaluate our proposed signal function learning approach, we need to compare our method with other ways of finding signal identification algorithms such as the content analysis used in our previous rumor detection system. In that method, the content analysis uses feature selection methods such as Chi-Square to generate a list of language patterns. We manually delete patterns that are relevant to the content of specific rumors and select a few that reflect common user reactions to rumors, e.g., enquiry patterns.

We also study how different weighted combinations of the three objectives (recall, precision and filter rate) in Equation 6.6, can be used to learn a good signal function. The three baseline methods and our method are as follows.

Baseline 1 Chi-Square and Manually Selection. We conduct Chi-Square feature selection method on our labeled datasets [33]. We compare the ngram features in all tweets that are relevant to at least one persuasion campaign to tweets that are irrelevant to any persuasion campaigns. To obtain performance, we remove the content of hashtags and only keep the hashtag sign representing a hashtag in all tweets during preprocessing. We manually filter out the features relevant to specific persuasion campaigns and select 12 features that are content-irrelevant. The top features ranked by Chi-Sqaure are: "url #hashtag #hashtag", "#hashtag #hashtag #hashtag", and "please retweet" etc.

Baseline 2 Optimize Accuracy. In this method, we learn signal function S using our approach but adjust the weights of the three objectives. We set the weight on Filter Rate (r_{FR}) to 0, and set the weight on True Positive Rate (r_{TP}) and True Negative Rate (r_{TN}) to 1. The learning process is similar to maximizing only the accuracy of the classification criterion F of signal function S in Equation 6.1.

Baseline 3 Equally optimize TP, TN and Ratio. We set all three weights in the objective function in Equation 6.6 to 1. Therefore, a signal function will be learned by treating **TP**, **TN** and **FR** equally important.

Our Method As described in Section 6.5, a good signal identification algorithm need to have a high **TP**, so that positive target clusters will not be filtered in the signal filtering step. sTherefore, we set weight on **TP**, (r_{TP}) to 2 and set r_{TN} and r_{FR} to be 1. So during optimization, True Positive rate have higher importance than True Negative rate and Filter Rate.

To evaluate the four methods, we randomly separate 20% of the labeled dataset into a testing set, and use the remaining 80% to find the signal functions of the different methods. To tune parameters such as learning rate η, we conduct four-fold

cross validation in the training set. To evaluate different signal functions, we calculate the **TP** rate, **TN** rate and **FR** in the testing set.

6.7.2.2 Detection Framework

After signal function S has been learned, we apply the signal function learned to detect persuasion campaigns from the social media post stream. Our objectives are to compare our method with existing clustering and classification frameworks, and ascertain whether our learned signal function outperforms other signal identification algorithms, e.g., Baseline 1 method in signal learning of using Chi-Square and manually selected patterns.

Baseline 1 Clustering and Classification. This method do not use any signal filtering techniques. Instead, it clusters all posts and generates candidate clusters for classification.

Baseline 2 Manually Selected Signal + Clustering + Classification. In this method, we use the Baseline 1 method of signal learning to identify signals, and then filter the posts and cluster signals to generate candidate clusters. We classify the candidate clusters matching both signal and non-signal posts by using the learned cluster classifier. This method can be viewed as directly applying and extending the rumor detection framework [123].

Our Method Automatically Learned Signal + Clustering + Classification. In our method, we learn signal function by maximizing the objective function defined in Equation 6.6 and apply the learned signal function to detect persuasion campaigns using the framework in Section 6.4.

To evaluate the accuracy detection and report the average accuracy of the three methods, we sort the positive clusters identified by each method by cluster size for

each day from February 1 to February 7 2016, and output the 50 most popular clusters. We evaluate the relevant recall by merging the clusters detected by all methods into a single persuasion campaign set. We also assess the run times and computational costs of the three methods.

6.8 Experiment Results

In this section, we show our experimental results. We discuss the performance of learned signal classifier using our method compared to the baseline methods. Then we show the effectiveness of detecting social media persuasions campaign using our framework compared to baseline approaches. We also discuss the efficiency of our detection system. We also give our analysis on the earliness of signals identified by the learned signal classifier, and discuss how to learn early signals using our method. At last we summarize and discuss our insights from the results.

6.8.1 Signal Learning

As discussed in previous section, to learn the feature weights in the signal function S using labeled persuasion campaign dataset L, we maintain multiple random initialized weights and train them separately. Our experiment results show that the objective value usually converges to a local maximum and stops within 100 iterations. Therefore, for each set of initialized weights, we stop training at the 100th iteration and select the set with the highest objective value.

For Baseline method 1, i.e., selecting patterns manually from Chi-square feature selection analysis, we first calculate Chi-square score on the training set, which is 80% of the labeled dataset. Then we manually select 12 pattens as the signal patterns. Any tweet having at least one of these patterns is a signal. We use these patterns as a baseline signal identification algorithm.

Note that Baseline method 1 for signal learning will also be used in building

baseline detection system in later experiments, where we should use the entire labeled dataset to conduct this analysis. We compare the Chi-Square scores calculated on the training set with the scores calculated on the entire labeled dataset. We find the same 12 patterns on the entire labeled dataset as in the training set. So in later experiments of evaluating detection systems, we use these 12 patterns for identifying signals for the baseline detection system.

For Baseline methods 2 and 3 and our method, we conduct a four-fold cross-validation in the training dataset to select the best learning rate and regularization factor for each method and then use the tuned parameters to learn the weights in the complete training set. We evaluate these methods using three objectives, i.e., TP, TN, and Filter Rate, on testing set. To eliminate the randomness of performance caused by achieving different local maximums, we run the evaluation process five times and report the mean of each metric. Table 6.3 lists the results and Figure 6.3 lists the results of plotting in a figure with an error bar.

Although there is no single numeric value indicates which signal function is better, we know that a good signal function should first have a high TP rate, then a high TN rate and a high FR rate. Comparing with Baseline 2 and Baseline 3, our method learns a better signal.

Comparing our method with Baseline 2, our method outperforms Baseline 2 for the TN and FR rates. Baseline 2 optimizes TN and TP equally and does not optimize FR. It is expected that the filter rate is lower than our method. But ideally Baseline 2 should have a higher TN rate than our method. The reason why Baseline 2 has a lower TN rate than our method is due to the non-convexity of our optimization process. When training a traditional binary classifier such as logistic regression, the convexity of its optimization on accuracy makes the learned classifier balance with TN and TP. However, since we don't have labels on posts, the definition of our objective is not directly on the signal function S but on an gated function F, which makes the

optimization no longer convex.

Because Baseline 2 does not optimize filter rate, the training easily falls into saddle points where many tweets are signals. This causes the TP close to 1 and the TN close to 0, because the classification criterion, gated function F, in Equation 6.1 identifies a positive cluster by checking whether clusters have no less than λ signals. Since λ is not only small but irrelevant to cluster size, when many tweest are signals, more positive clusters, both true positives and false positives, will be identified. In this case the sigmoid function used to replace the sign function in F will generate 0 gradients, which makes the optimization stuck there. On the other hand, adding Filter rate as another objective prevents predicting too many signals and balancing the performance of TP and TN.

Comparing with Baseline 3, setting a higher weight on True Positive rate in our method returns the required high TP rate. Because Baseline 3 sets equal weights on TP, TN, and FR, the results learned balance TP and TN with a high FR. Noting that a good signal identification algorithm should have high a TP so that we can reserve as many true target clusters as possible before classification, by setting weight on TP to be 2 and weights on TN and FR to 1, we can learn a signal function with high TP and medium TN. Also, since Baseline 3 treats TP and TN equally, the standard deviation shown in Figure 6.3 is much higher than the other methods including ours. This is because different local maximums achieve either high TP or high TN.

Baseline 1 and our method perform similarly for TP and TN, although Baseline 1 has a lower FR. This shows by manually select content irrelevant features from a rank-list generated be feature selection method such as Chi-Square, we can get good patterns for high TP and medium TN for persuasion campaign detection. Lacking a constraint on FR, the signal identification algorithm learned using this method could show low efficiency in filtering irrelevant posts before clustering.

6.8.1.1 Examples of Top-ranked Features for Learned Signal Function

Baseline 1 The 12 patterns extracted using feature analysis in Baseline 1 mostly relate with the use of multiple hashtags, and asking for retweeting. This reflects the common use reaction of asking for support. For example, it can capture tweets such as "Lets Vote against, Disparity, Discrimination and Suppression of Mamata.. Lets Slap Mamata Legally #Vote4GJM #VoteForBJP #Gorkhaland.". The 12 patterns extracted are: "#hashtag #hashtag #hashtag", "url #hashtag #hashtag", "#hashtag #hashtag url", "chance to win", "@user will", "please retweet", "retweet please", "you retweet ", "let's get", "retweet this @user", "you retweet", "retweet this". These patterns are indicative to persuasion campaigns, altough some of them, such as multiple hashtags, or asking for retweet, are relatively common in post stream. Commonality makes signal identification and filtering less efficient.

Our method Since our method learns a binary classifier that contains a complex combination of a large number of patterns, we give a few examples of the top patterns ranked by learned weights. Compared to Baseline 1 which uses multiple hashtags, some of the top-ranked patterns in our method combine the use of hashtags with persuasive pro or con expressions. For example, the pattern "fantastic #hashtag" captures this tweet related to a persuasion campaign. "I've already voted. Have you? Pls #vote and #RT for this fantastic #Plymouth initiative: vmbvoom.com/pitches/cre8t...". This pattern express the common user reactions of supporting. We also have patterns such as "time to banish" expressing common reactions of opposing. Some other top-ranked patterns similar to asking for retweets, although less common, including "amp to follow","rt amp follow", "shout outs required", etc.

Table 6.3: Performance of Different Signal Learning Methods

Method	True Positive rate	True Negative rate	Filter Rate
Baseline 1	0.9333	0.1667	0.6418
Baseline 2	1	0.0111	0.5858
Baseline 3	0.3067	0.8556	0.9986
Our Method	0.9333	0.2	0.9435

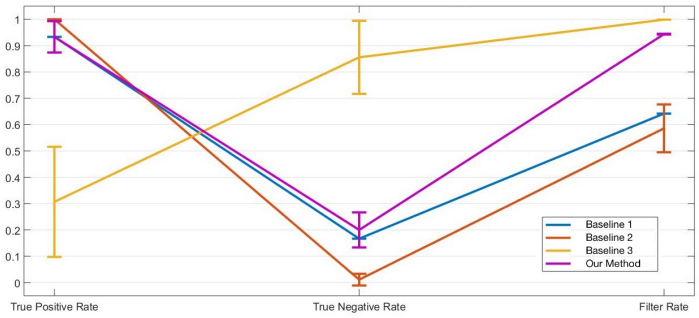


Figure 6.3: Performance of Signal Function Learning

6.8.2 Persuasion Campaign Detection

In this subsection we evaluate the effectiveness of detecting persuasion campaigns from tweet stream. We compare our framework with the clustering + classification framework without signal filtering. We also compare our learned signal classifier with the signal patterns manually selected from Baseline 1. The setup of our detection system and the two baseline approaches in this evaluation are introduced in Section 6.7.

To compare the effectiveness of different frameworks, we run the three detection systems on 10% of the daily tweets from February 1-7. For each day, each method outputs all positive clusters classified by the same classifier trained on the labeled dataset. The implementation of the classifier is discussed in Section 6.6. We rank the output positive clusters by size, select the top 50 largest clusters, and ask our two human annotators to label whether they are persuasion campaigns or not. We use average accuracy, which is the average of the correct prediction percentage in the 50

clusters over the seven days of the study period.

We also want to understand the recall of each framework. Since the cost of labeling all post clusters outputted from the three detection systems is high, we calculate relevant recall on the set of the top 50 largest clusters from the three systems that have already been labeled as persuasion campaigns by the annotators. To identify whether a cluster returned from one method contains the same persuasion campaign as a cluster returned from another method, we compare the hashtag(s) identified as a cluster center. When two clusters' centers contain the same hashtag(s), it means they refer to the same persuasion campaign. In our experiment, after merging all labeled persuasion campaign clusters from all methods, we have a total of 260 unique persuasion campaigns.

Table 6.4 lists the performance results. We can see that our method of using automatically learned signal classifier outperforms other two methods. Using manually selected signals have higher precision and recall than baseline method 1 (no signal filtering). This is because applying signal functions can filter negative cases and keep most of the positive cases, which further increase accuracy of detection.

Applying signal function and a signal filtering step cannot change the prediction results of the final step of classification, but it can select the candidate clusters to be classified by the classifier. Proper selection, such as filtering many potential false negative clusters, greatly improves final detection accuracy.

For event detection scenarios such as persuasion campaign detection, positive example are rare. Thus, a cluster level classifier trained on limited labeled dataset is likely to overfit and perform less well on a larger and imbalanced set of post clusters in a real tweet stream. Adding a signal filtering step changes the set of candidate post clusters for classification by providing another perspective on post level thus can improve the situation of overfitting.

Unlike the use Chi-Square for manual pattern selection, our method targets at

Table 6.4: Detection Performance of Different Detection Method

Method	Average Accuracy	Relevant Recall
Baseline 1	0.463	0.415
Baseline 2	0.569	0.45
Our Method	**0.646**	**0.569**

joint optimizing for a high true positive rate, low false negative rate, and high filter rate. Our method not only looks for content-irrelevant post level features as in the manual selection, but also for uncommon features, i.e., it handles both data imbalance and overfiiting.

6.8.3 Earliness of the Signal

Although we are not looking for signals that appears at the early stage of persuasion campaign in our signal learning process, it is important to study the earliness of detection. Because it will be more meaningful if we can detect persuasion campaigns at their earlier stages.

Since we apply the same classifiers at the classification step, if we output all positive clusters as detection results, the baseline method 1 of not using signals will always be the method that detects a given persuasion campaign earliest. However, since there will be a large number of false positive post clusters outputted at the same time, it may not be practical for us to locate meaningful persuasion campaigns.

Instead of checking all results, it is more practical to check the most popular positive clusters returned by the detection system. Therefore, we use the largest 50 positive clusters detected by each method to evaluate earliness.

For each method, we compare the first day it outputs a positive persuasion campaign to the earliest day the same persuasion campaign has been detected by any methods. For the baseline method 1, in seven days, 108 unique clusters has been detected, 99 of them are detected on the same day as the earliest detection and 9 clusters are detected later. The average detection delay compared to the earliest de-

tection is 0.18 day. For the baseline method 2, 117 unique clusters has been detected and 114 of them are detected on the same day as the earliest detection, the average detection delay compared to the earliest detection is 0.05 day. And for our method, 148 unique clusters are detected and 140 of them are detected on the same day as the earliest detection. The average detection delay compared to the earliest detection is 0.12 day.

In general, using signal filtering detects persuasion campaigns earlier than baseline method without signal filtering when only showing the most popular results. Here, since manually selected signal doesn't filter as many posts as our signal function, but still have similar TP and TN rate, some persuasion campaigns can be detected earlier than our method. Our method filters most of the tweets and still has a relatively early detection result compared to Baseline 1.

Next we also want to understand how constraining the size and time stamp of the training data in learning the signal classifier assistis in identifying early signals. If our labeled dataset only contains the early posts of each persuasion campaigns and we can still learn a good signal, this signal can by applied and may achieve a better earliness in detection. We sort the tweets in each labeled post cluster by time, and limit each cluster to only contain the earliest k tweets. We compare the signal function learned on the training set with different k and show the results in Figure 6.4. From the result we can see, when k is small, we can still learn the signal function with high TP and TN, but the filter rate is relatively low. When k is too large, performance becomes unstable and decreases. This result suggests that choosing a proper training size is important for learning a good signal and possibly an early signal.

6.8.4 Detection Efficiency

We also evaluate the efficiency of each detection system. We run them using our stream based clustering algorithm discussed in Algorithm 1. For Baseline 1, we

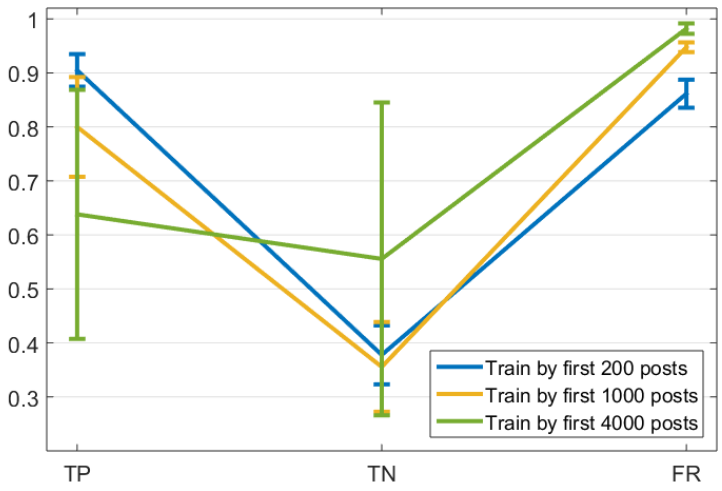


Figure 6.4: Learning Signal Function using Different Size of Labeled Dataset

treat every tweet as a signal and any cluster has more than 3 tweets as a candidate cluster. For Baseline 2, we use manually selected patterns as the signal identification algorithm. For our method, we use signal function S learned automatically as the signal identification algorithm. All detection systems are run on a single core 2.3GHz CPU.

We set a threshold limit on the size of candidate clusters. If a candidate cluster size is larger than 1,000,000, we replace the oldest ones with the newest ones. Without this constraint, the number of candidate clusters will grow as the posts from the input stream are processed, and Baseline 1 and 2 cannot finish our scalability test within a day. We also implement an inverted index for matching tweets with clusters to further improve efficiency. Figure 6.5 shows the time cost of each method by scale of input stream.

We can see that when the number of tweets processed is small, all the three methods cost about the same amount of time and have comparable efficiency. When the number of tweets is small, feature extraction and other processes, such as signal filtering, cluster matching, and outputting, cost more time than clustering. And Baseline 1 does not have signal filtering function to match incoming post with signal

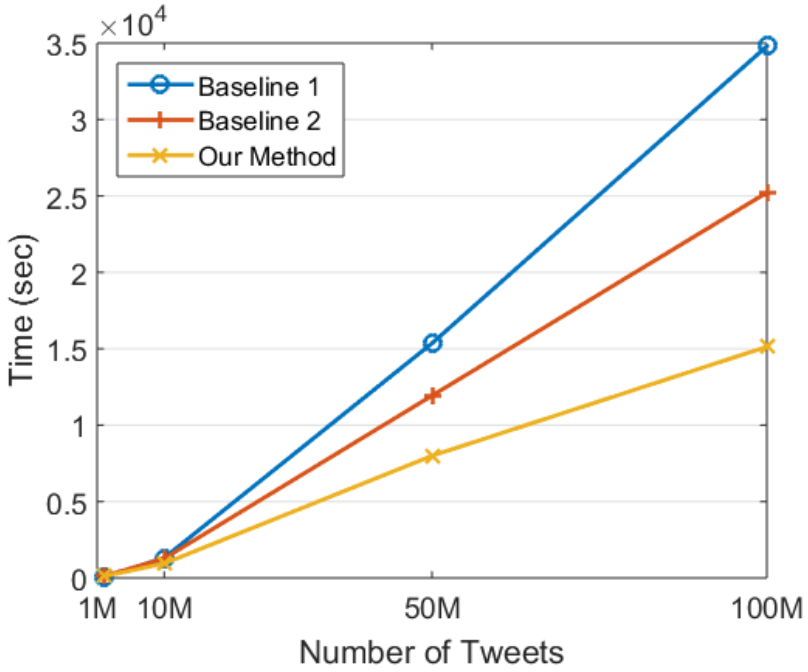


Figure 6.5: Time Consumed (seconds) by Method

patterns. When scalability increases, the cost of Baseline 1 increases significantly. Baseline 2 is more efficient than baseline 1 because it filters more than 50% of the tweets. Our method has the shortest run time, i.e., it process 100M posts in about four hours.

Since the run time may be influenced by many uncontrollable factors such as other running processes, and hard drive efficiency. To get a better idea of the computational resources signal functions can save, we count the number of function calls for similarity measurement in our framework. In our detection task, other than measuring the similarity, most functions such as feature extraction and signal filtering are called only once for each incoming post. For the clustering algorithm, comparing pair-wise similarity between posts or between post and cluster center consumes the most time. Figure 6.6 shows the number of similarity comparison functions (in log scale) called by method. By applying our learned signal, the number of function calls is significantly less than both baseline methods. Note that our detection system users an inverted

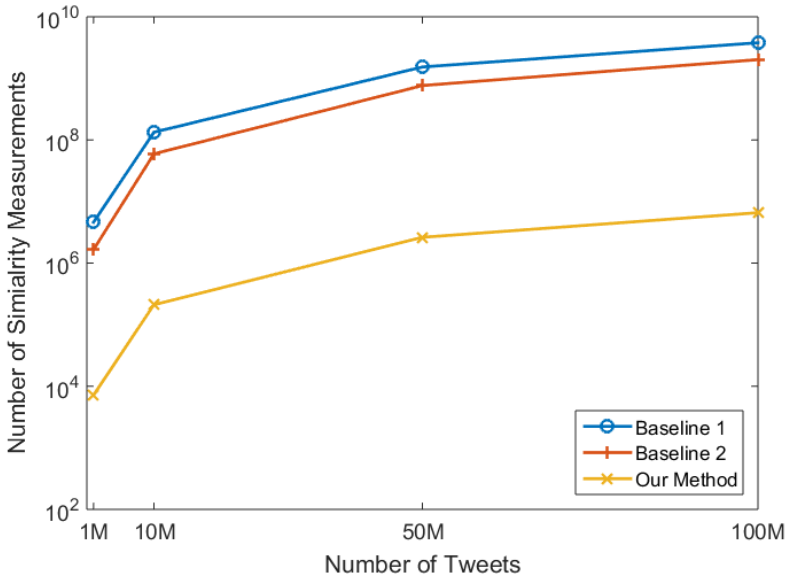


Figure 6.6: Number of Similarity Measurements by Method

index to match every new post in a stream with candidate clusters. The matching and retrieval calculations are not counted as similarity measurements. Applying state-of-the-art retrieval approaches can further increase the efficiency of our system.

To sum up, in this section, we conduct experiments on our social media persuasion campaign detection system. We test our method of learning signal classifier for signal filtering from labeled persuasion campaigns. We also compare our detection system with other existing detection frameworks and signal identification algorithms on a real world Twitter dataset. For signal learning, our method can learn better signal identification algorithm evaluated by the key properties of signal filtering. And we apply the learned signal identification algorithm in detecting persuasion campaigns and achieve better performance than baseline methods.

6.9 Summary

In this chapter we studied the problem of detecting persuasion campaigns from social media post stream. We revisited the definition and general framework of de-

tecting target clusters in social media and applied them in detecting persuasion campaigns. To solve the detection problem, we extended our proposed framework by automatically learning a signal identification algorithm. We quantified the properties of signals for filtering irrelevant posts, including precision, recall, and filter rate, and proposed to learn a signal classifier to identify signals from a labeled persuasion campaign dataset. We applied our learned signal classifier to our detection framework. Experiments using tweet stream from Twitter showed that our proposed method outperformed detection systems without signal filtering and detection systems that used manually selected patterns for filtering signals.

As the experiment has demonstrated, adopting signal filtering can achieve some level of earliness in detecting persuasion campaigns compared to detection systems without signal filtering. Future research, therefore, should study how to learn a signal not only have high true positive rate, true negative rate and filter rate, but also appears at the early stage of a target cluster.

Having generated additional persuasion campaigns by using our detection system, future research should consider how different signals perform differently in detecting subsets of persuasion campaigns such as political campaigns, promotions, and spams.

CHAPTER VII

Conclusion

This dissertation proposes a general framework to detect online social media rumors and persuasion campaigns effectively and efficiently. In this chapter we summarize the development and application of the two detection systems. We note the contributions of our research to the published literature on social media and the implications for commonly used data mining applications. Chapter 7 ends with our suggestions for future research.

7.1 Summary

Today, Facebook, Twitter, Google Plus, and other social media platforms as yet unknown play an outsized role in modern communications. Increasingly, people depend on them to interact with other users, to broadcast (disseminate) their opinions, and to get information. The advent of so-called fake news is one of several reasons that computer scientists need to identify and closely examine the dissemination of rumors and the persuasion campaigns associated with social media events.

Existing methods used by researchers to detect social media events first apply clustering techniques to cluster posts. Each post cluster corresponds to a social media event. Reachers apply a predefined set of rules or a classifier trained on a labeled dataset to identify whether a post cluster corresponds to a targeted social

media event. The detection methods, however have proven ineffective and inefficient when applied to detect events involving only a small subset of all post clusters such as rumors and persuasion campaigns, because of the imbalanced situation of target clusters in all clusters.

In this dissertation we improve the detection methods by adding a signal identification and filtering step. Signals, which are the posts that indicate the existence of targeted events, reflect the common user reactions during an event?s emergence and dissemination.

Instead of clustering all posts, we only cluster the signals identified from a social media post stream. We match the nonsignal posts with the signal clusters to obtain a subset of the post clusters that contain signals. We apply a classifier to identify the target clusters. We use the general framework to build rumor and persuasion detection systems. The framework can also be adapted to detect other types of social media events with similar properties.

The use of a signal filtering step before clustering filters out a large percentage of the posts from the post stream. In other words, only the signals that are likely to be related to targeted events are used to generate candidate clusters, and the resulting candidate clusters contain fewer irrelevant events than the clusters generated from the entire post stream. This general framework for rumor or persuasion campaign detection significantly reduces the computational cost as well as the difficulty of event classification.

Do signals exist at all? How to identify them? And what make them different from other posts? As the most critical component of our framework is the identification of good signals, we start by conducting a content analysis of posts on social media, in this case, Twitter, to understand users? common reactions to different types of events and how users? individual activities or posts explicitly reflect such reactions. We analyze the difference between user activity types according to the the content of

their corresponding posts.

Specifically, we track and analyze users? question-asking behaviors on Twitter. Because the types of explicit questions asked reflect users? information needs, they also represent reactions to different types of social media events, e.g., rumors. Training a classifier to automatically detect the posts on Twitter that are questions allows us to extract a large collection of question posts. By grouping these questions by keywords, our analysis found popular topics in questions are different from the topics being generated by other posts. Question topics have high correlation with social media bursting events and have a considerable power of predicting trend of Google search. Our analysis provides insights in detecting social media rumors. A specific type of questions, namely, confirmation questions or enquiries, are related to social media rumors.

What are signals for social media rumors? And how can be build effective and efficient system to make use of them to detect social media rumors? In order to build social media rumor detection system, we need to answer these two research questions. Because rumors are factual claims with disputes, when a rumor occurs, it results in common user reactions such as enquiring and fact-checking. There will be a small number of posts enquiring the truth value of the rumor statement in a rumor cluster, although most of them are not. We continue our content analysis by understanding enquiries and other common user reactions of social media rumors. By comparing feature differences between posts in rumor clusters and posts in non-rumor clusters, we discover a set of unique language patterns that related to enquiring and fact-checking posts. These patterns are used to identify signals of social media rumors.

We apply the proposed general framework to detect rumors on Twitter. We cluster only the tweets containing enquiring and fact-checking patterns. We extract the statements made by each cluster and match them with nonsignal tweets that

discuss the same statement. The clusters with both signal and nonsignal tweets become candidate clusters. We train a classifier to rank the candidate clusters by the probability that they are rumor clusters, based on their statistical features. Our experiment shows that we can detect rumors with high accuracy more efficiently than when we do not use signals. Observing that enquiry posts can appear in very early stages of a rumor infers that our detection system can detect rumor clusters far earlier than the existing detection methods.

The results inspire us to apply our framework to another type of targeted event, social media persuasion campaigns. Unlike rumors, the signals for detecting persuasion campaigns may not contain unique and simple language patterns. In fact, the signals may contain many patterns and combinations.

What are the properties of signals for detecting target clusters? Can we automatically learn a post classifier for signal identification using those properties? In our work of building social media persuasion campaign detection system, we identify three key properties of signals for targeted event detection.

- High recall: All or most target clusters have these signals. Therefore, we do not filter target clusters by filtering nonsignal posts.
- High precision: Signals exist mostly or only in target clusters. Therefore, we can filter out many nontarget clusters.
- High filter rate: Signals are relatively uncommon compared to other posts. Therefore, only clustering signals increases detection efficiency.

A good signal identification algorithm needs to have all three properties, so our detection framework using it is more effective and efficient compared to frameworks without it. And when these three properties cannot be satisfied at the same time, high recall is the first priority. This is because a cluster classification step after clustering signals and generating candidate clusters can identify positive clusters. So

it is more important that most potential positive clusters are preserved after signal filtering process. Quantifying the key properties into learning objectives enables us to develop a joint optimization learning algorithm to learn a signal classifier from labeled persuasion campaigns.

Once the signal classifier is learned, it can be used to identify signals from social media post stream. We apply it in our general detection framework to detect persuasion campaigns. Our detection system takes input of social media post stream, filters non-signal posts and clusters signals into signal clusters. Then by matching non-signals to signal clusters, it gets a stream of candidate clusters. Then we identify persuasion campaigns from these candidate clusters using a binary classifier. In our evaluation, we show that by using signals identified from a learned signal classifier, we can detect persuasion campaigns with higher precision and recall compared to framework using signals identified from manually selected language patterns and framework without using signal filtering.

Once the signal classifier is learned, we use it to identify the signals in a social media post stream and incorporate the learned signal classifier into our general detection framework to detect persuasion campaigns. Our detection system uses the social media post stream as input, identifies and clusters the signals into signal clusters. By matching the non-signals with the signal clusters, we obtain the stream of candidate clusters. Then we identify persuasion campaigns by using a binary classifier. In our evaluation, we show that by using the signals identified from a learned signal classifier, we can detect persuasion campaigns with higher precision and recall than frameworks using manually selected signals and frameworks without signal identification and filtering.

7.2 Contributions and Limitations

General detection framework for rumors and persuasion campaigns We develop a general framework for detecting social media rumors and persuasion campaigns. Our framework is both efficient and effective. It yields better detection performance than existing approaches that do not identify and use signals. It can also be applied to other types of social media events, if the emerging and spreading of such events can result in common user reactions .

Content analysis of online user activities The content analysis we conduct to understand the different types of user activities and their corresponding posts can be used for other social media mining applications. We can leverage the content and topic differences between the posts generated by type of user activities to build user profiles for context-aware recommendation tasks. Our analysis of user-asking question activities shows how a bursting events correlates with the trend of question posts, and provide insights into understanding the potentially popular memes in social media.

Social media rumor detection system We build a system to detect social media rumors at early stage. By monitoring social media posts of enquiring, fact-checking and correcting activities, we can identify the early signs of occurrences of rumors. Our system can be applied to real world social media website and perform detection task in real time. The ability to detect rumors at the earliest stages can help computer scientists, the media, and political campaigns to quickly analyze and evaluate detected rumors. Most importantly, early rumor detection can also draw attention to false or fraudulent claims before dissemination occurs.

A leaning algorithm to discover signals We propose a learning algorithm that can directly learn a signal classifier from labeled target clusters. Absent our automatic learning algorithm, researchers must rely on domain knowledge to manually

select the unique language patterns and features to identify the signals of a targeted event. Manual selection, however, is ineffective when no simple rules and patterns can be found to identify the signals for events such as persuasion campaigns. The automatic learning algorithm allows researchers to identify signals having complex combinations of patterns and features. The the weights of a signal classifier learned for a specific detection task can provide insights into discovering a problem?s unknown characteristics.

Social media persuasion campaign detection system We build a social media persuasion campaigns detection system. Social media persuasion campaigns have great power in affecting people's everyday life. Our system of detecting persuasion campaigns are built and tested on Twitter stream, it achieves high accuracy and is robust to deal with the large scale of social media post stream in real time.

Our general detection framework also has some limitations. We can only detect some types of social media events where signals can be found. For events where no common user reactions can be found, we cannot identify their signals.

The framework is most efficient when the signals and targeted events are relatively uncommon. Otherwise, it is essential to cluster most posts before classification. Although proper filtering could improve performance as a preprocessing step of the classification, the improvement will not be as significant as the detection of uncommon target clusters.

7.3 Future Research

Our general detection framework, rumor detection system, and persuasion campaign detection system suggest the following research streams.

Earliness of signals In rumor detection, enquiries are signals that appear at an early stage of rumor clusters. In a persuasion campaign, both manually selected patterns and learned signal classifiers show some level of earliness in the detection process. The next step should develop a numeric metric to evaluate and estimate the earliness of signals. Adding earliness to the learning objective could automatically learn the early signals for detecting targeted events.

Tracking detected events Our work studies the detection of social media events. After events are detected, it is often necessary to track and monitor them, in an effort to understand their diffusion, patterns of spreading, and how they influence and are influenced by other events.

Applying our framework in network analysis Our framework detects social media events by detecting target clusters of social media posts. It can also be used to detect clusters of items other than posts. One direct application we want to study is clusters of users. There are many existing works in detecting different types of communities in social media. We are interested in extending our framework to detect different types of social user communities and understand the relevant signals.

Detecting target clusters beyond social media We also want to apply and extend our framework to the outside of social media, where clustering applications take variety of forms. For example, we want to examine how to apply our framework to detect diseases from medical images such as MRI or CT scans. Identifying potential cancers as a target cluster could help researchers to learn the potential signals for this type of target cluster.

Detecting terrorist groups and propaganda Although in our work of persuasion campaign detection, we have discussed the detection of terrorist group's pro-

paganda as detecting persuasion campaigns in social media. Critical study of this subcategory of persuasion campaigns is urgently needed. There are two challenges to overcome: the lack of a labeled dataset, and the complex strategies employed by terrorist groups to evade detection by governments and social media hosts. Developing a detection system for propaganda could prevent dissemination.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Detection of persuasion campaigns on twitter? by sax-vsm technology.

[2] C. C. Aggarwal. A survey of stream clustering algorithms., 2013.

[3] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment, 2003.

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

[5] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006.

[6] N. Ailon, R. Jaiswal, and C. Monteleoni. Streaming k-means approximation. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 10–18. Curran Associates, Inc., 2009.

[7] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, pages 577–584, 2003.

[8] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query recommendation using query logs in search engines. In *Current Trends in Database Technology-EDBT 2004 Workshops*, pages 395–397. Springer, 2005.

[9] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. 2011.

[10] J. Benhardus and J. Kalita. Streaming trend detection in twitter. *International Journal of Web Based Communities*, 9(1):122–139, 2013.

[11] J. Bollen, H. Mao, and X.-J. Zeng. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003, 2010.

[12] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.

[13] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[14] A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, D. Metzler, L. Riedel, and J. Yuan. Online expansion of rare queries for sponsored search. In *Proceedings of the 18th international conference on World wide web*, pages 511–520. ACM, 2009.

[15] A. Z. Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE, 1997.

[16] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.

[17] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM, 2011.

[18] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[19] E. H. Chi. Information seeking can be social. *IEEE Computer*, 42(3):42–46, 2009.

[20] C. J. Cohen and J. Kahne. Participatory politics. new media and youth political action. 2011.

[21] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. pages 467–474, 2008.

[22] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[23] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006.

[24] N. DiFonzo and P. Bordia. *Rumor psychology: Social and organizational approaches.* American Psychological Association, 2007.

[25] P. Domm. False rumor of explosion at white house causes stocks to briefly plunge; ap confirms its twitter feed was hacked., April 2013.

[26] M. Efron and M. Winget. Questions are content: a taxonomy of questions in a microblogging environment. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–10, 2010.

[27] F. M. A. F. Emilio Ferrara, Onur Varol. Detection of promoted social media campaigns.

[28] G. Erkan and D. R. Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[29] B. Evans and E. Chi. Towards a model of understanding social search. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 485–494. ACM, 2008.

[30] C. Fellbaum. Wordnet: An electronic lexical database. *Cambridge, MA: MIT Press*, 38(11):39–41, 1998.

[31] E. Ferrara, M. JafariAsbagh, O. Varol, V. Qazvinian, F. Menczer, and A. Flammini. Clustering memes in social media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 548–555. IEEE, 2013.

[32] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[33] G. Forman. Feature selection for text classification. *Computational methods of feature selection*, 1944355797, 2008.

[34] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1995.

[35] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[36] R. Gallager. Claude e. shannon: A retrospective on his life, work, and impact. *Information Theory, IEEE Transactions on*, 47(7):2681–2695, 2001.

[37] D. A. Garvin and M. A. Roberto. Change through persuasion. *If you read nothing else on change, read thesebest-selling articles.*, page 26, 2005.

[38] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

[39] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, February 2009.

[40] E. Goffman. The presentation of self in everyday life. 1959.

[41] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, July 1969.

[42] A. Gupta and P. Kumaraguru. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, page 2. ACM, 2012.

[43] A. Gupta, H. Lamba, and P. Kumaraguru. $1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–12. IEEE, 2013.

[44] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 729–736. International World Wide Web Conferences Steering Committee, 2013.

[45] K. L. Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC, 2014.

[46] S. Hamidian and M. Diab. Rumor detection and classification for twitter data. In *SOTICS 2015*, pages 71–77.

[47] B. Hecht, J. Teevan, M. R. Morris, and D. Liebling. Searchbuddies: Bringing search engines into the conversation. *ICWSM*, pages 138–145, 2012.

[48] X. Jin, C. Lin, J. Luo, and J. Han. A data mining-based spam detection system for social media networks. *Proceedings of the VLDB Endowment*, 4(12), 2011.

[49] R. Jones, R. Kumar, B. Pang, and A. Tomkins. I know what you did last summer: query logs and user privacy. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914. ACM, 2007.

[50] G. S. Jowett and V. O'donnell. *Propaganda & persuasion*. Sage Publications, 2014.

[51] A. Karandikar. *Clustering short status messages: A topic model based approach*. PhD thesis, University of Maryland, 2010.

[52] M.-C. Ko and H.-H. Chen. Analysis of cyber army's behaviours on web forum for elect campaign. In *Information Retrieval Technology*, pages 394–399. Springer, 2015.

[53] S. Kong, Q. Mei, L. Feng, F. Ye, and Z. Zhao. Predicting bursts and popularity of hashtags in real-time. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 927–930. ACM, 2014.

[54] S. Kong, Q. Mei, L. Feng, and Z. Zhao. Real-time predicting bursting hashtags on twitter. In *Web-Age Information Management*, pages 268–271. Springer, 2014.

[55] R. Krovetz. Viewing morphology as an inference process. *16th ACM SIGIR Conference*, pages 191–202, 1993.

[56] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang. Prominent features of rumor propagation in online social media. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1103–1108. IEEE, 2013.

[57] V. Lampos, T. De Bie, and N. Cristianini. Flu detector-tracking epidemics on twitter. In *Machine Learning and Knowledge Discovery in Databases*, pages 599–602. Springer, 2010.

[58] K. Lee, J. Caverlee, Z. Cheng, and D. Z. Sui. Content-driven detection of campaigns in social media. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 551–556. ACM, 2011.

[59] K. Lee, J. Caverlee, Z. Cheng, and D. Z. Sui. Campaign extraction from social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):9, 2013.

[60] K. Lee, P. Tamilarasan, and J. Caverlee. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. 2013.

[61] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM, 2009.

[62] B. Li, X. Si, M. R. Lyu, I. King, and E. Y. Chang. Question identification on twitter. pages 2477–2480, 2011.

[63] H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[64] H. Li, A. Mukherjee, B. Liu, R. Kornfield, and S. Emery. Detecting campaign promoters on twitter using markov random fields. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 290–299. IEEE, 2014.

[65] X. Li, Y. Liu, M. Zhang, S. Ma, X. Zhu, and J. Sun. Detecting promotion campaigns in community question answering. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2348–2354. AAAI Press, 2015.

[66] X. Li, L. Wang, and E. Sung. Adaboost with svm-based component classifiers. *Eng. Appl. Artif. Intell.*, 21(5):785–795, 2008.

[67] Z. Liu and B. Jansen. Almighty twitter, what are people asking for? *ASIST*, 2012.

[68] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[69] T. N. Mansuy and R. J. Hilderman. Evaluating wordnet features in text classification models. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, pages 568–573. AAAI Press, 2006.

[70] A. E. Marwick et al. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New media & society*, 13(1):114–133, 2011.

[71] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.

[72] Q. Mei and K. Church. Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the international conference on Web search and web data mining*, pages 45–54. ACM, 2008.

[73] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.

[74] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[75] M. Morris and J. Teevan. Exploring the complementary roles of social networks and search engines. *Human-Computer Interaction Consortium Workshop(HCIC)*, 2012.

[76] M. R. Morris, S. Counts, A. Roseway, A. Hoff, and J. Schwarz. Tweeting is believing?: understanding microblog credibility perceptions. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 441–450. ACM, 2012.

[77] M. R. Morris, J. Teevan, and K. Panovich. A comparison of information seeking using search engines and social networks. *ICWSM*, 10:23–26, 2010.

[78] M. R. Morris, J. Teevan, and K. Panovich. What do people ask their social networks, and why?: a survey study of status message q&a behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1739–1748. ACM, 2010.

[79] J. Nichols and J. Kang. Asking questions of targeted strangers on social networks. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 999–1002. ACM, 2012.

[80] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[81] S. A. Paul, L. Hong, and E. H. Chi. Is twitter a good place for asking questions? a characterization study. In *ICWSM*, 2011.

[82] S. C. Pendleton. Rumor research revisited and expanded. *Language & Communication*, 18(1):69–86, 1998.

[83] M. Pfau and R. Parrott. *Persuasive communication campaigns*. Pearson College Division, 1992.

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

[85] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[86] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011.

[87] S. Ranganath, X. Hu, J. Tang, and H. Liu. Understanding and identifying advocates for political campaigns on social media, 2016.

[88] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In *ICWSM*, 2011.

[89] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web*, pages 249–252. ACM, 2011.

[90] R. E. Rice and C. K. Atkin. *Public communication campaigns*. Sage, 2012.

[91] R. L. Rosnow. Inside rumor: A personal journey. *American Psychologist*, 46(5):484, 1991.

[92] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860, 2010.

[93] H. Sayyadi, M. Hurst, and A. Maykov. Event detection and tracking in social streams. In *ICWSM*, 2009.

[94] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[95] E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*, pages 83891I–83891I. International Society for Optics and Photonics, 2012.

[96] C. Shah. Measuring effectiveness and user satisfaction in Yahoo! answers. *First Monday*, 16(2-7), 2011.

[97] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM, 1999.

[98] A. Singhal. Google blog: Introducing the knowledge graph: Things, not strings, May 2012.

[99] S. Sun, H. Liu, J. He, and X. Du. Detecting event rumors on sina weibo automatically. In *Web Technologies and Applications*, pages 120–131. Springer, 2013.

[100] T. Takahashi and N. Igata. Rumor detection on twitter. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 452–457. IEEE, 2012.

[101] T. Takahashi and N. Igata. Rumor detection on twitter. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 452–457. IEEE, 2012.

[102] C. Tan, L. Lee, and B. Pang. The effect of wording on message propagation: Topic-and author-controlled natural experiments on twitter. In *Proceedings of the ACL*, pages 175–185, 2014.

[103] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of WWW*, pages 613–624, 2016.

[104] J. Teevan, E. Adar, R. Jones, and M. Potts. Information re-retrieval: repeat queries in yahoo's logs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 151–158. ACM, 2007.

[105] J. Teevan, S. Dumais, and D. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 163–170. ACM, 2008.

[106] J. Teevan, D. Ramage, and M. Morris. # twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international Conference on Web search and Data Mining*, pages 35–44. ACM, 2011.

[107] O. Tsur, A. Littman, and A. Rappoport. Efficient clustering of short messages into general domains. In *ICWSM*, 2013.

[108] A. H. 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.

[109] K. Wang and T.-S. Chua. Exploiting salient patterns for question detection and question retrieval in community-based question answering. pages 1155–1163, 2010.

[110] L. Xiong, B. Póczos, and J. G. Schneider. Group anomaly detection using flexible genre models. In *Advances in neural information processing systems*, pages 1071–1079, 2011.

[111] L. Xiong, B. Póczos, J. G. Schneider, A. J. Connolly, and J. VanderPlas. Hierarchical probabilistic models for group anomaly detection. In *AISTATS*, pages 789–797, 2011.

[112] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM, 2012.

[113] J. Yang, M. R. Morris, J. Teevan, L. A. Adamic, and M. S. Ackerman. Culture matters: A survey study of social q&a behavior. *ICWSM*, 11:409–416, 2011.

[114] J. Yao, B. Cui, Y. Huang, and X. Jin. Temporal and social context based burst detection from folksonomies. In *AAAI*. AAAI Press, 2010.

[115] J. Yao, B. Cui, Y. Huang, and Y. Zhou. Bursty event detection from collaborative tags. *World Wide Web*, 15(2):171–195, 2012.

[116] R. Yu, X. He, and Y. Liu. Glad: group anomaly detection in social media analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(2):18, 2015.

[117] R. Yu, H. Qiu, Z. Wen, C. Lin, and Y. Liu. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 18(1):1–14, 2016.

[118] X. Zhang, Z. Li, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in twitter. *ACM Transactions on the Web (TWEB)*, 10(1):4, 2016.

[119] X. Zhang, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in the twitter social network. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1194–1199. IEEE, 2012.

[120] X. Zhao, N. Salehi, S. Naranjit, S. Alwaalan, S. Voida, and D. Cosley. The many faces of facebook: Experiencing social media as performance, exhibition, and personal archive. In *SIGCHI 2013*, pages 1–10.

[121] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416. International World Wide Web Conferences Steering Committee, 2015.

[122] Z. Zhao and Q. Mei. Questions about questions: An empirical analysis of information needs on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1545–1556. International World Wide Web Conferences Steering Committee, 2013.

[123] Z. Zhao, P. Resnick, and Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405. International World Wide Web Conferences Steering Committee, 2015.