2015

Features for Ranking Tweets Based on Credibility and Newsworthiness

Jacob W. Ross
Wright State University

Features for Ranking Tweets Based on Credibility and Newsworthiness

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

by

Jacob W. Ross
B.S.C.S., Wright State University, 2013

2015
Wright State University

Wright State University
GRADUATE SCHOOL

May 7, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Jacob W. Ross ENTITLED Features for Ranking Tweets Based on Credibility and Newsworthiness BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Krishnaprasad Thirunarayan, Ph.D.
Thesis Director

Mateen Rizki, , Ph.D.
Chair, Department of Computer Science and Engineering

Committee on
Final Examination

Krishnaprasad Thirunarayan, Ph.D.

Keke Chen, Ph.D.

Derek Doran, Ph.D.

Robert E.W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

ABSTRACT

Ross, Jacob. M.S., Department of Computer Science, Wright State University, 2015. *Features for Ranking Tweets Based on Credibility and Newsworthiness.*

We create a robust and general feature set for learning to rank algorithms that rank tweets based on credibility and newsworthiness. In previous works, it has been demonstrated that when the training and testing data are from two distinct time periods, the ranker performs poorly. We improve upon previous work by creating a feature set that does not over fit a particular year or set of topics. This is critical given how people utilize social media changes as time progresses, and the topics discussed vary. In addition, we are constantly gaining new tweet data. Thus, it is important to be able to have a set of features that can perform well across many different topics, and across different years. In our approach, we present a methodology for selecting features based on how they can capture credibility and newsworthiness regardless of year and topic. In order to derive such features, we use the studies done on credibility perception of social media as well as the clues provided in past works in this domain. We also present new features that, to our knowledge, have not been used in previous works in this domain.

Contents

List of Figures

List of Tables

Acknowledgment

First and foremost, I would like to thank my thesis advisor, Dr. Krishnaprasad Thirunarayan (T.K. Prasad) for his unwavering patience and support throughout this project. He kept me on the right track in terms of my thesis and also gave me valuable research and teaching experience. I also would like to thank Dr. Keke Chen and Dr. Derek Doran for taking time out of their schedules to be on my thesis committee.

I owe Aditi Gupta and Carlos Castillo a huge thanks for generously sharing their annotated tweet datasets. Without their generosity and support this thesis would not have been possible. I would like to thank Hemant Purohit, Wenbo Wang, and Pramod Anantharam from Kno-e-sis for taking the time for answering my many questions and for helping me get my feet wet with my own project.

I would also like to take this time to thank my friends and family for their support throughout my graduate studies. They have all done way more for me than they realize; I cannot thank them enough.

I dedicate this thesis to my parents Timothy and Susanna Ross, my grandmothers Gloria Darleen Ross and Mary Virginia Ehrgott, and to the loving memory of my grandfathers Steward Ross and William Harry Ehrgott.

Introduction and Motivation

We discuss past works that study how social media is used as a news source, how social media is utilized by users during high impact and crisis events, how social media can be leveraged for solving real world problems, and how false information on social media can spread. We show that there are many research areas involving social media that can benefit from improved methods for determing the credibility and newsworthiness of tweets. We then state our research goals and outline the organization of the rest of this document.

1.1 Introduction

There is no denying that the popularity of social media has risen greatly over the past few years. Currently, there are 288 million monthly active users on the micro-blogging site, Twitter[1]. Users on Twitter write and share short messages, called Tweets, that are limited to 140 character. Tweets are shared with the author's followers, which in turn can share the tweet, or "re-tweet" with their followers. An average of 500 million tweets are sent per day. Twitter is also a global phenomenon, 77% of Twitter accounts are outside of the United States, and Twitter supports 33 languages.

Due to the popularity of Twitter, there has been a push to conduct research that can utilize the vast amounts of data collected by Twitter. The topics discussed on Twitter range from the mundane to events that have made it into international news. Thus, there is a natu-

[1] about.twitter.com/company

ral inclination harness tweets in order to gain knowledge about world events, and to help us create methods to react to these events. Here, we discuss past works that demonstrate social media can act as a news source, and how false information is spread on Twitter during disaster scenarios. We also discuss previous works that aim to solve real world problems by leveraging social media.

1.1.1 Social Media as a News Source

We discuss how Twitter and other online social media are often used as a news source, and how the users will often use online social media as a means of gaining knowledge about real world events. Twitter and other social media sites often blur the line between social media and news source. Kwak et al. [30] demonstrate that a large portion of the conversations on Twitter are directly related headline news topics. At the time of their study, they reveal that 85% of the topics discussed on Twitter are related to current headline news reports, and that 1 in 5 Twitter users regularly tweet about such topics. They also provide insight into the power of the retweet. On average, whenever a tweet becomes retweeted by a follower, 1000 more twitter users are exposed to the tweet. Kwak et al. also reveal that people typically will learn of a news headline from a news source (i.e., CNN) before they see it on Twitter, but, occasionally tweets about a potential news headline will show up as a trending topic on Twitter before it has been reported on by a major news source. This can be attributed to the fact that Twitter users can tweet in real time as an event unfolds. Lehmann et al. [32] aimed to create a system to automatically find news curators on Twitter. They cite a study on news management[2] that reports that out of 613 journalists contacted, 54% of them use online social media as a means of gaining information about a certain topic. During their research, they discovered that Twitter users will engage with journalists directly to discuss news topics.

Perhaps the most eye-opening research on Twitter as a news source was conducted by

[2]http://www.oriellaprnetwork.com/sites/default/files/research/OriellaDigitalJournalismStudy2012FinalUS.pdf

Hu et al. [22]. They studied the nature of Tweets propagated that were related to the death of Osama Bin Laden. In their study, they reveal that tweets claiming Bin Laden's death surfaced before major news sites reported on it, and before the official statement was made by the White House. Hu et al. also reveal that the first wave of tweets were accepted with low confidence; only 50% of the users they analyzed fully believed Bin Laden had died based off of tweets. However, as more information spread on Twitter, confidence grew to 80% and rose steadily afterward as news sources and official statements confirmed the early tweets.

1.1.2 Social Media and Disaster Scenarios

We discuss past works that analyze the affect of Twitter during disaster scenarios. These works reveal that people who are affected by a disaster scenario will use Twitter in order to share information or gain information about the disaster event.

Acar et al. [2] conducted a study on how people affected by the 9.00 scale earthquake that hit Japan in the March of 2011 were utilizing online social media. They aimed to answer these four questions:

- What kinds of messages did Twitter users post immediately after the earthquake?
- How do messages from people directly affected by the earthquake differ from the messages from people indirectly affected by the earthquake?
- What problems did the Twitter users experience when using the service after the earthquake?
- What recommendations do Twitter users make to improve communication during disaster scenarios?

Acar et al. categorize tweets propagated during the disaster scenarios as warnings, help requests, and reports about the environment. Based on their observations, they noticed

that users often had difficulties determining whether or not a tweet contained valid and accurate information or if it contained false information. As a result, users did not feel comfortable retweeting messages, even if the message was about a request for help. Many of the users they surveyed suggested that there should be a system implemented to keep track of users who tend to spread false information.

Kongthon et al. [29] conducted a similar study to Acar et al. [2] except the tweets they gathered were propagated during the 2011 Thai floods. They report that the people affected by the flood act as citizen reporters, meaning, people will tweet what their current environment is like. By using the content of the tweets collected and the geolocation information that comes along with the tweets, they demonstrate that there is potential to utilize this information to aid disaster relief efforts. Also, similar to Acar et al. [2], they express the need for a tool to verify information being propagated.

Mendoza et al. [35] analyzed how rumors can spread on Twitter during disaster scenarios. They showed that the users on Twitter will question and doubt rumors or information that is false. This leads to the notion that the Twitter has the potential to be self filtering. This paper shows that the keywords for a disaster topic change in correlation to how the disaster scenario itself evolves. Mendoza et al. also show that the most influential users for a disaster topic are people directly related to the event itself, news organizations, and relief organizations.

The common theme in these and other works [36, 47, 11, 33, 4] that analyze how information spreads on Twitter during disaster events is that they express a need for the verification of the tweets. As revealed in [2], people are hesistant to retweet a message they cannot confirm. Typically, there are tweets that are requests for help. These works also express the potential to use Twitter data to aid disaster relief efforts.

1.1.3 Leveraging Social Media to Solve Real World Problems

Due to the sheer amount of data available from Twitter, there has been a push in research to try and solve real world problems by leveraging online social media. We will discuss previous works that leverage social media in order to discover correlations between Twitter activity and real world events, track and analyze disease outbreaks, predict the location of Twitter users, and create systems to aid in disaster relief.

In [52], Zhang et al. aimed to predict stock market indicators by analyzing Twitter posts. In order to do this, they measure the emotion of users by tracking words that indicate fear. They show that there is a correlation between how reluctant users are to invest and trade and how well the various stocks perform that day. They show that when the fear of trading and investing is low, stocks tends to rise, and vice versa. Tumasjan et al. [46] and Rao et al. [41] conduct similar studies, and show similar results. Along the same lines as prediciting financial success, Asur et al. [5] try to predict how well a movie will perform at the box office by analyzing Tweets that discuss the movie. They show that there is a correlation between how well a movie performs at the box office, and how much discussion takes place about the movie on Twitter.

Researchers have also leveraged online social media to help understand disease breakouts. Signoroni et al. [44] hypothesized that due the the sheer volume of tweets available, you can accurately track diseases such as H1N1 and swine flu. The correlations they show are intuitive, such as when user interest in antiviral drugs drop, official disease reports claim that the cases reported are mostly mild in nature. They also show that people who are affected will tend to tweet about their discomfort levels. Based on their results, they show that, in general, tweet content tends to correlate with actual disease activity. Bodnar et al. [6] aimed to diagnose Twitter users based on their tweets. They reveal that half of the users who have the disease will tweet about it explicitly. By combining text analysis and anomaly detection, they are able to "diagnose" Twitter users with high accuracy. They note that they could extend their approach in hopes of identifying those who have stigmatized

diseases, or diseases that may be hard to detect through tradtional means.

Location prediction of Twitter users is another domain in which researchers leverage online social media. Davis et al. [13] aimed to predict the location of messages on Twitter by analyzing user relationships. This proved to be a difficult task because only 22% of Twitter relationships are not mutual, and most users do not provide their location in their profiles. Mahmud et al. [34] rely on the content of tweets in order to make predictions. They make two important claims in order to help them derive features for their classifiers: people will tweet more about where they live than other places, and people will visit places around where they live more so than other locations. By combining these two heuristics and content based features, they show promising results for predicting the location of users when geo-location data is not available.

Purohit et al. [39] devised a system in order to identify those who need help (seekers) and those who can provide help (suppliers) during disaster scenarios. As a part of their system, they utilize lexical heuristics (word based) and syntatctic heuristics (syntax based). The lexical heuristic encompasses what words seekers tend to use versus the words suppliers tend to use. However, lexical heuristics alone are not enough, so they define syntactic heuristics based on the order of words and parts of speech. They derive rules to identify seekers and suppliers based on these two heuristics, and show promising results.

1.2 Motivation

Ultimately, our goal is to improve the state of the art for ranking tweets based on credibility and newsworthiness. We are motivated by the fact that false information on Twitter can spread rapidly, and can have negative impact on people involved in real world events. We also recognize, in past work, rankers typically perform poorly when trained on a dataset from a different time period than the testing data. This is problematic, because we are constantly gaining new and unseen data that needs to be verified.

1.2.1 False Information Propagation On Twitter

We discuss previous works that show how false information can spread on Twitter, and how it can have a negative impact on society. We discuss three papers by Gupta et al. that demonstrate how false information can spread and create problems during crisis events. We also analyze Mendoza et al. [35] that explains how false information can spread on Twitter in the general sense.

In [35], Mendoza et al. conduct a case study to test the rate at which false rumors spread on Twitter. By collecting tweets that are known truths and tweets that are known false rumors, they analyze how differently Twitter users interact with truthful tweets in comparison to false rumors. They show that for any given tweet that contains truthful information, 95% of the associated tweets possess qualities that confirm the truth. However, 50% of tweets associated with a known false rumor will dispute and deny claims made. They show that in general, when a tweet contains false information, other users will question the tweet.

In [20], Gupta et al. reported on how fake images were spread on Twitter during Hurricane Sandy. They claim that, in general, online social media has potential benefits for aiding in disaster relief efforts. However, there are also malicious users who aim to use the popularity of a disaster topic in order to spread false rumors that incite panic and derail relief efforts. In the case of Hurricane Sandy, there were several fake images that became viral that created panic for those users affected by the hurricane. Gupta et al.

Figure 1.1: Fake Images Shared During Hurricane Sandy

discovered that 86% of the tweets containing fake images were retweets. This implies that whenever a fake image popped up, it was not the original post, rather, users were unknowingly propagating false information. Due to this, Gupta et al. claim that only a small portion of users were responsible for creating false content, and the power of the retweet is what is responsible for spreading the fake images. They also discovered that when a user affected by a crisis event retweets a message, a majority of the time the original message is not from a user who the retweeter is following or friends with. This implies that during crisis events, people who are affected are willing to share information even if it is from an unknown source.

In [19], Gupta et al. conduct another study on how false information can spread on Twitter during a high impact event. This time, they analyze data from a terrorist attack, the 2013 Boston Marathon Bombings. Their findings reveal that 29% of the viral content generated during the Boston Bombings were false rumors and fake content. They also show that a large number of accounts responsible for spreading false information, were accounts that have high social reputation and verified by Twitter. Over 6000 accounts were created during this event in hopes of exploiting the event's popularity in order to spread spam, phishing attacks, and rumors. Some of these accounts selected usernames in a fashion that would lead a user to believe they were not malicious (e.g. the fake username "bostonmarathons" versus the real account "bostonmarathon"). The paper states that malicious content posted to online social media during crisis events can result in damage, chaos, and

monetary losses in the real world. This claim is supported in a technical report also by Gupta et al.[16] . Here, they show how two false events from the Mumbai bombings in 2011 affected real world events. The first false event was a report about a fourth explosion. There were roughly 500 retweets supporting this claim. The second, and perhaps the more problematic false rumor, was that hospitals had a shortage of blood. There were 2000 tweets and retweets claiming this, and people who wished to help out arrived at hospitals to donate blood, even though the hospitals had adequate blood supplies.

As we can now see, false information spreading on online social media is problematic in the real world. Thus, we are motivated to improve on the state of the art for techniques that aim to detect credible and newsworthy tweets.

1.2.2 Ranker Performance on New Data

In Gupta et al. [18], the authors demonstrate a key problem for learning to rank tweets based on credibility and newsworthiness. In their approach, the ranking SVM algorithm [25] performs well when the training and testing data are from the same year. However, the ranker performs poorly when the training data is from 2011 and the testing data is from 2013. The last row in Table 1.1 shows the portability problem. This is a crucial problem

Training	**NDCG@25**	**NDCG@50**	**NDCG@100**	**Testing**
2011 events	0.4765	0.5966	0.7359	2011 events
2013 events	0.3951	0.4919	0.7219	2013 events
2011 events	*0.3743*	*0.3693*	*0.3783*	*2013 events*

Table 1.1: Results from Gupta et al. [18]

to try an solve simply because we are constatnly gaining new tweet data, and how people utilize and share information on online social media is constantly changing. Thus, we must be able to rank new and unseen data with the data we already have access to and is well known.

1.3 Research Goals

We aim to improve on the state of the art that aims to rank tweets based on credibility and newsworthiness. Specifically, we create a set of features that will perform well on training and testing data that are from two distinct time periods. This is an important goal because the landscape of online social media is constantly changing, as well as how people utilize online social media in their everyday lives. Our 2011 and 2013 data sets were provided by Gupta et al. from [17] and [18] respectively. In [18], they achieve an $NDCG@100$ score of 0.3783 when training on the 2011 dataset and testing on the 2013 dataset. With our new approach to selecting and generating new features, we achieve an $NDCG@100$ score of .6998 when testing on the 2011 dataset and testing on the 2013 dataset.

1.4 Outline

Here we explain the organization of the remainder of this thesis document. In Chapter 2, we discuss previous works that research credibility perceptions of online social media. We will also discuss previous works that aim to classify or rank tweets based on credibility and newsworthiness. In Chapter 3, we explain how our datasets were collected and annotated. We also discuss our datasets properties such as topics and class ditributions. In Chapter 4, we explain our approach for selecting and generating new features to achieve our research goal as well as show ranker performance with our features. In Chapter 5 we compare our results to Gupta et al. [18] and discuss future work.

Previous Work

We discuss previous works on credibility perceptions, as well as previous works that aim to automatically determine the credibility and newsworthiness.The approaches to automatically determinine credibility and newsworthiness of tweets either utilize classifiers or learning to rank algorithms. At the end of this chapter we will summarize the most common features that appear in comtemporary works.

2.1 Credibility Perceptions

Morris et al. conducted a survey in order to understand the user's perception of credibility on Twitter [37]. They focused on users searching for content on Twitter based on topic and not on a specific author. Typically, a user will accept a known author's claims as fact. However, now that users tend to search based on topic, they cannot be guranteed that the authors of the tweets for a particular topic are the ones they are familiar with. They make some interesting observations, such as, users do not have enough clues to accurately assess credibility on content alone; users will use other clues such as profile image and user name in order to help them assess credibility. In their experiments, Morris et al. aim to uncover what features of a tweet or tweet source users utilize in order to judge its credibility. Table 2.1 lists the top 10 features users in their experiments agreed were high indicators of credibility.

Feature	Average Credibility Impact
is a RT from someone you trust	4.08
verified author topic expertise	4.04
author is someone you follow	4.00
contains URL you clicked thru to	3.93
author is someone youve heard of	3.93
account has verification seal	3.92
author often tweets on topic	3.74
many tweets w/ similar content	3.71
personal photo as user image	3.70
author often mentioned/retweeted	3.69

Table 2.1: Top 10 Credibility Indicators from Morris et al. in [37] based on the scores given by users in their experiment.

A majority of the features that enabled the users to determine credibility were associated with the author of the tweet. The author based features can be grouped into three categories: *influence, topical expertise, and reputation*. Influence based features include follower, retweet, and mention counts. Topical expertise features are features an author has that indicate they are an expert on the topic the viewer is interested in. We can glean such indicators through the author's homepage, the author's tweet history, outside webpages that are on topic that mention the author, and the author being in a location relevant to the topic. Reputation based features help indicate the familiarity a user has for the author of a tweet. Such features can be whether the author is followed by the user, the author is someone the user has heard of before, or if the author's account has been verified by Twitter.

Content based features also proved to be beneficial in Morris et al.'s paper. The content based features users felt revealed the most about a tweet's credibility were if the tweet contains a URL that leads to a reputable webpage, if there are multiple tweets that make the same claim as the tweet in question, and the use of standard grammar. There are other features the users in the survey reported on as being good indicators for credibility. Users noted that the image an author uses as their profile picture affects how they judge tweet credibility. Users were more willing to trust an author that had an image of themselves or of an image related to the topic they are interested in. Other profile images, such as

cartoons or the default Twitter picture, indicated decreased percieved credibility. Similarily, the strucutre of the author's username seemed to impact the user's percieved credibility of the author's tweet.

Yang et al. [51] conducted research on how credibility indicators for online social media differ from American users and Chinese users. They hypothesize that indicators will differ between American users on Twitter, and Chinese users on Sina Weibo[1] based on cultural differences.

Hypothesis	**Description**	**Result**
H1	Overall, people will find tweets from men more credible	Supported
H1a	H1 will be more prominent in China than in the U.S.	Not Supported
H1b	H1 will be more prominent for political tweets	Supported
H2	Users will find tweets from users with topical user names more credible	Supported
H3	Users will find tweets from users with a photo as their profile image more credible	Supported
H3a	H3 will be less prominent in Chinese users	Supported
H4	Users will find tweets from users from liberal areas more credible	Supported
H4a	H4 will be less prominent in U.S. users	Supported
H4b	H4 will be more prominent in Chinese users	Supported
H5	People will find tweets from their friends more credible	Supported
H5a	H5 will be more prominent in Chinese users	Not Supported
H6	Chinese users will find microblog updates more credible that U.S. users	Supported

Table 2.2: Hypotheses of Yang et al in [51]

We summarize Yang et al.'s hypotheses in Table 2.2. They show that credibility perceptions can be quite different across two different cultures. They form their hypotheses based on tendencies of each culture. For example, in Hypothesis H6, they reason that Chinese users will find microblog updates far more credible than American users. They form this hypothesis becuase the Chinese government censors traditional media, and that Chinese culture, in general, greatly values social connections.

[1] http://www.weibo.com/login.php?lang=en-us

Shariff et al. [43] conducted a study in order to analyze how online social media users tend to judge or misjudge tweet credibility. They reveal that topics involving politics have the largest number of misjudged tweets. This can be attributed to the fact that a majority of the tweets involving politics are often questions and opinions. They note that the political tweets that are labaled correctly are usually linked to a known and reputable news source. Shariff et al. also did an indepth analysis on the tweets most users misjudged. 95% of those tweets were breaking news and political news. They show that tweets that lack a link to outside resources, such as a URL, are often difficult for users to judge.

2.2 Predicting Tweet Credibility via Classification

Castillo et al. [8] produced one of the earliest works on automatically predicting tweet newsworthiness and tweet credibility. Their data collection consisted of two phases. First, label and keep tweets that are deemed newsworthy. Secondly, label the newsworthy with a credibility score. In order to obtain these annotations, Castillo et al. utilized Amazon Mechanical Turk[2] to post the tweets in question. Mechanical Turk users label tweets based on newsworthiness and credibility. They keep labeled tweets where five out of seven users agree on the score. For the credibility labeling, users were asked to assign one of four scores:*almost certainly true, likely to be false, almost certainly false, and cannot be decided*. Annotators were also asked to provide reasoning as to why they gave a tweet a certain score. By reading these comments, Castillo et al. learned the key factors that annotators used to make their judgments. They note that some topics will evoke emotion out of the users posting about that topic. Typically, the overall emotion of the topic will be caputred in the sentiment on the tweets about that topic. Annotators commented on how you can estimate the certainty that someone has while sharing information. If the propagator has low confidence, they will typically question the information they are sharing. Annota-

[2]https://www.mturk.com/mturk/welcome

tors typically labeled tweets to have high credibility if the tweet cites an external link to a known source. The link can act as proof for the claim made in the tweet. Annotators also commented on characteristics of the tweet's author; annotators took things such as screen name, profile description, and user picture into account. The dataset for training and testing have features that fall into one of four categories:

- Message Based Features
- User Based Features
- Topic Based Features
- Propagation Based Features

For the task of predicting the newsworthiness of a tweet, Castillo et al. train a cost sensitive classifier. In this case, the punishment for misclassifying a tweet that is labeled as "chat" or "unsure" is 0.5, while misclassifying tweets labeled as `NEWS` were weighted at 1.0. The J48 classifier yielded the best results with 89% accuracy. For the credibility classification portion, they yielded 86% accuracy, recall, and F1 score. They then used the GINI split criteria in order to reveal the best features for the J48 decision tree. Tweets that contain a URL tend to be credible, tweets with negative sentiment tend to be credible, and tweets with many retweets tend to be credible. A URL essentially acts as proof for a claim made in the tweet. If a tweet is retweeted many times, this indicates many users found the tweet useful. They also reveal the tendencies of non-credible tweets. Information deemed non-credible tends to be created by people who have a low tweet counts, and a small fraction of tweets with positive sentiment were deemed to be credible.

Yang et al. [50] conduct a similiar study to Castillo et al. in [8], except they focus on Sina Weibo rather than Twitter. Sina Weibo is the most popular online social media outlet in China, and has nearly 8 times as many users as Twitter. Sina Weibo has a built in rumor detection tool, so they were able to collect confirmed false rumors from Sina Weibo

directly, and there is no need for a data annotation step. Although Twitter and Sina Weibo differ, there are many text and author based features that are relevant to both. Yang et al. define five categories for tweet features:

- Content Based Features
- Client Based Features
- Account Based Features
- Propagation Based Features
- Location Based Features

Content based features are very similar to those in Castillo et al. [8]. Examples of content based features include whether the message contains pictures, number of positive and negative emoticons, and whether the tweet contains a URL or not. The client based features describe what medium the user used in order to post their message, for example, if the user posted from a mobile client or from a web browser. Examples of account based features are if the user has a verified account, whether or not the user has a description, gender, and number of friends. The gender feature is an example of a feature that is not readily available from the Twitter API, but is available from Sina Weibo. Location based features capture where the event being discussed takes place. Propagation based features include features such as if the message was re-tweeted and the number of comments for the message. In addition to these features, Yang et al. proposed two new features: location of the event and the client program used to post the message. The effect the new features had are in Table 2.3. Overall, these new features were helpful, and goes to show that putting

Feature Set	**Accuracy with SVM**	**Accuracy with New Features**
Content Based Features	.7258	.78
Account Based Features	.7263	.7736
Propagation Based Features	.7234	.7866

Table 2.3: Results of Yang et al. [50]

effort into creating new features can help increase classifier accuracy.

Kang et al. [28] propose three ways to model credibility on Twitter: *a social based model, a content based model, and a hybrid based mode*l that contains ideas from the previous two models. The social based models aim to leverage details from the underlying social network in Twitter. This model assumes a retweet is a sign of credibility. The content based model aims to analyze the features that make up the strucutre of the text in a tweet. Such features include patterns of speech, number of positive and negative sentiments words, and number of intensifier words. Kang et al. discovered numerous indicators they claim can help determine credibility. They show how the number of followers an author has can help us determine the credibility of their tweets. They show there is a significant correlation between credibility and the number of followers the author has. However, when the number of followers a person has exceeds 1500, this correlation is no longer present. Furthermore, if the number of followers a user has is exceptionally large, this tends to correlate to non-credible tweets. This could be due to the fact that many users with an extremely high number of followers tend to have fake accounts follow them that can be paid for. The number of URLs is a sign of credibility for both authors and tweets. Credible authors tend to post tweets with URLs in them, and credible tweets tend to have URLs in them. Retweets tended to be credible, and longer tweets tended to be retweeted more than short tweets.

Bobidou et al. [7] aim to classify tweets based on credibility using a variety of classifiers. They form their dataset by labeling tweets that are confirmed by outside sources as credible, and labeling tweets that have known fake images as non-credible. They collect tweets from two topics, Hurricane Sandy and the Boston Marathon Bombing. Using the same features as Gupta et al. in [20], they show that, on the Hurricane Sandy dataset, they achieve 81.38% accuracy using the KStar classifier, and they achieve 81.25% accuracy on the Boston Marathon dataset using J48. However, they also show a great decrease in accuracy when training on the Hurricane Sandy dataset and testing on the Boston Marathon

dataset. The best accuracy they achieve is 58% using Random Forests.

2.3 Learning to Rank Tweets Based on Credibility and Newsworthiness

We discuss previous works that aim to rank tweets based on newsworthiness and credibility by using learning to rank algorithms. Duan et al. [14] aims to automatically rank tweets based on relevance, not neccesarily credibility which appears in later work. In order to rank tweets based on relevance, they use features that capture the account authority of the author as well as features that describe the content of the tweet in question, with ranking SVM. Additionally,they categorize their features as content relevance features, twitter specific features, and account authority features. In order to capture account authority, they make these assumptions: users who have more followers have been mentioned in more tweets, listed in more tweets, and are retweeted by other authority accounts. They decribe four scores that can be calculated in order to quantify account authority: *follower score, mention score, list score, and popularity score*. *Follower score* is the number of followers the account in question has. *Mention score* is the number of times the author is mentioned in other tweets. *List score* is the total number of lists the author appears in other users. *Popularity score* is an adapted version of PageRank [38] that is calculated based on retweets. They make some key observations summarized below:

Firstly, the account authority features are important for automatically ranking tweets based on relevance. This shows that users take into account the source of the tweet when judging its relevance. Length of the tweet was deemed an important feature, as well as whether or not the tweet contains a URL. They note that removing the URL feature causes a significant decrease in ranker performance. With their best feature set, they achieve an $NDCG@10$ score of .55 on their tweet corpus.

In [17], Gupta et al. used ranking SVM and pseudo relevance feedback (PRF) in order to rank tweets based on their credibility score. They address the problem that even though a topic in general can be credible, tweets on that topic can be non-credible and can potentially contain false information. A prime example of this are the false images that spread through Twitter during Hurricane Sandy. Hurricane Sandy was not a rumored event, however false information pertaining to it spread on Twitter rapidly and from otherwise credible sources [20]. Tweets were labeled with the following annotations:

- Definitely Credible
- Seems Credible
- Definitely Non-Credible
- Related to a Topic, No Information
- Tweet is Unrelated to Topic

Features are categorized as either content based features or source based features. They make some key observations about what features a tweet has that are correlated with credibility and newsworthiness. Tweets with a large number of unique characters tend to be credible. They attribute this to the fact that tweets with mentions, hashtags, and URLs will contain more unique characters. A tweet containing swear words tends to be relevant to a topic, but it is often a reaction to the topic and contains no information. As has been revealed in numerous previous works, they also discover that a tweet with a URL tends to be credible. With their approach, they achieve a $NDCG@50$ score of 0.73 after applying PRF.

In [18], Gupta et al. extend their work from [17] in order implement a real time browser based system for ranking tweets based on credibility and newsworthiness called TweetCred[3].They gather tweets from six topics that occured in 2013. Tweet annotation is

[3]https://chrome.google.com/webstore/detail/tweetcred/fbokljinlogeihdnkikeeneiankdgikg?hl=en

similar to their earlier paper [17]. First, tweets are labeled as either containing information about the event, the tweet is related to the event but has no information, or the tweet is not related to the event. Then, the tweets deemed newsworthy were labeled as either definitely credible, seems credible, definitely incredible, or cannot be determined. This implies a newsworthy tweet contains information about the event, but could have non-credible elements in it. In their ranking scheme, a tweet that is newsworthy but non-credible will rank higher than a tweet that is an opinion and contains no information about the event itself. Gupta et al. define five categories for tweet features:

- Tweet Meta-Data
- Tweet Content Features
- User Based Features
- Linguistic Based Features
- External Resource Features

Tweet meta-data features include features such as number of seconds since the tweet was posted and the source of the tweet. Tweet content features include number of characters, number of words,and number of URLs. Linguistic based features include features such as presence of swear words, negative emotion words, and number of positive emotion words. External based features include the Web of Trust[4] score for provided URLs and ratio of likes to dislikes to an attached Youtube video. For their purposes, the response time of their system was important, so they sacrificed ranker accuracy for faster training and testing times. Coordinate Ascent yielded the best NDCG@100 score of 0.7607. For their system Gupta et al. chose SVMRank since it performed only slightly worse in terms on NDCG ($NDCG@100 = 0.719$) but was much faster than the Coordinate Ascent approach. They observed that both tweet and author based features were key in the ranking task. However,

[4]https://www.mywot.com/

as mentioned in Section 1.4, training on the 2011 dataset from [17] and testing on the 2013 data from this paper yielded poor results. This can be attributed to the fact that the most important features for each dataset are often different. For example, whether or not a stock symbol is present in the tweet was deemed important for the 2011 dataset, but it was not important for the 2013 dataset. They suggested in future work, their model will need to be updated and re-trained in order to deal with the change of importance of features.

2.4 Popular Features from Previous Work

We summarize the most popular features used in previous works. In total, we gather features from 17 different papers, some of which we have already discussed. We gather features from works that use classifiers to automatically predict credibility [8, 50, 21, 28, 49, 9, 16, 15, 7], features from works that use learning to rank algorithms[18, 17], and features gleaned from works that take hybrid or other appraoches to quantify and model credibility [40, 24, 45, 1, 42, 3]. In Table 2.4 we list the Tweet based features that appear in at least

Tweet Features
isRetweet(9)
tweetLength(9)
numWords(8)
numMentions(7)
numHashtags(7)
numURLS
hasURL(7)
numRetweets(7)
hasHappyEmoticon(6)
hasSadEmoticon(6)
sentimentScore (6)

Table 2.4: Popular Tweet Based Features

Author Features
numFollowers(9)
numFriends(9)
numTweets(8)
hasDescription(7)
isVerified(6)
ratioFriendsFollowers(6)

Table 2.5: Popular Author Based Features

6 of the previously mentioned papers, and in Table 2.5 we similarily list the most popular author based features. We use these features as a starting point for our own feature set.

Datasets

Here we discuss the datasets used to carry out our experiments. We have two distinct sets of annotated tweets from 2011 and 2013. We obtained these datasets from Gupta et al. to enable comparisons with their work. The 2011 dataset has been discussed in their first paper on ranking tweets [17] and the 2013 dataset has been discussed their TweetCred paper [18]. Each dataset set contains tweets that pertain to a *topic*. Each topic is a newsworthy event that took place in either either 2011 or 2013. Each dataset is a list of `<TweetID, Annotation>` pairs. Annotations are discussed below. In order to obtain tweets in the JSON format, we utilize the Twitter RESTful API[1] and write Python scripts to pull the data. We will discuss how we convert the JSON files to the LibSVM [10] format. Details of this code will be discussed in this chapter. We will also discuss the topics covered in each dataset, as well as class distributions for each dataset. Each dataset was labeled by human annotators as described in [17] and [18]. Each tweet was labeled with one of the following annotations in order of most relevant to least relevant:

5 Tweet contains information and is credible

4 Tweet contains information and seems credible

3 Tweet contains information and is non-credible

2 Tweet is relevant to the topic but contains no information

[1] http://dev.twitter.com/rest/public

1 Tweet is spam

Here we show examples of tweets from each category. Figure 3.1 is an example of a credible tweet and has the common elements of a credible tweet as discussed previously. Figure 3.2 is an example of a newsworthy but non-credible tweet as it lacks key elements credible tweets tend to have. Figures 3.3 and 3.4 are examples of tweets that contain no relevant news information, but pertain to their topics. Figure 3.5 is an example of a tweet that was labeled as spam. It contains a hashtag relevant to a topic, but none of the text of the tweet is relevant to the topic. Figure 3.6 is a tweet that originated from a fake account that mimics the official Boston Marathon Twitter account.

Figure 3.1: An example of a credible and newsworthy tweet. Notice how this tweet contains an outside link to a well known and credible news site: the BBC.

Figure 3.2: An example of a noncredible and newsworthy tweet. This tweet contains information about the topic it pertains to, but, has no links to external sources, and has no replies or retweets.

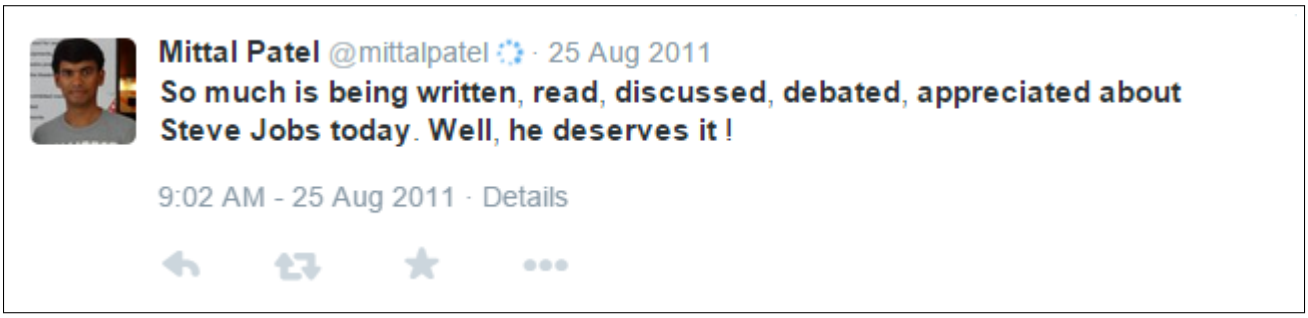

Figure 3.3: An example of a tweet that stays on topic, but does not contain any information that will allow the reader to gain any knowledge about the topic.

Figure 3.4: A tweet the pertains to the Boston Marathon, but has no newsworthy content.

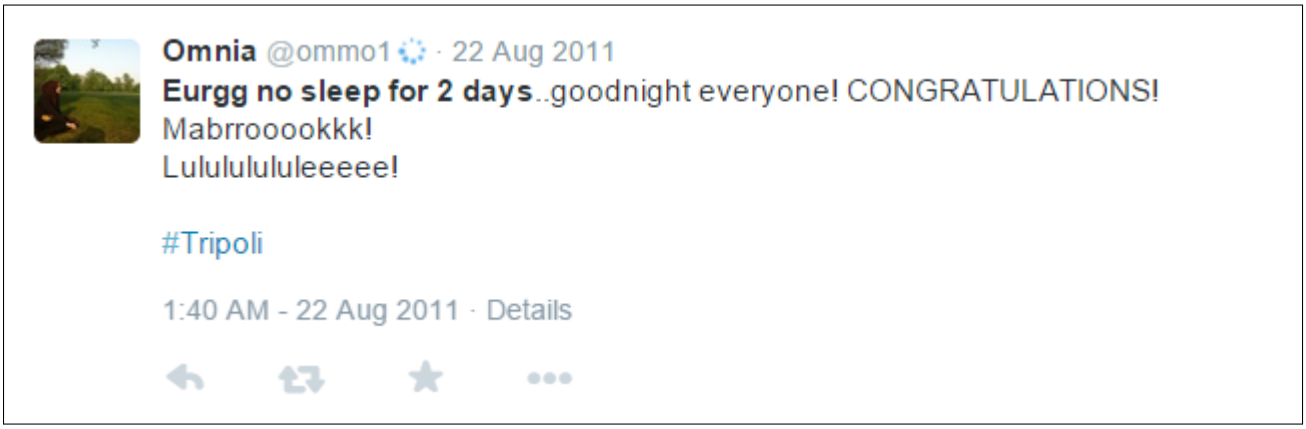

Figure 3.5: An example of a spam tweet. This tweet contains the hashtag #tripoli which corresponds to the Libya Rebels topic. However, this tweet contains nothing relevant to that topic.

Figure 3.6: A well known false tweet the surfaced during the Boston Marathon Bombings. The source of this tweet is not the official Boston Marathon Twitter account.

3.1 2011 Dataset

The 2011 Dataset contains 4028 tweets that spans 14 different topics. We explain these topics in Table 3.1. In Table 3.2 we show the class distributions for the 2011 Dataset. The topics vary from disaster scenarios (i.e., the stage collapse at the Indiana State fair and the London Riots), to politics (i.e., the Anna Hazare protests in India), and topics involving entertainment and technology (i.e., Steve Jobs resigning and the Facebook messenger app.)

Topic	Description
UK Riots	Riots take place in London; 5 people died and many more injured
Libya Rebels	Rebels take control on city in Libya fighting Qadaafi's forces
Virginia Earthquake	5.8 magnitude earthquake hits Virginia
Stocks Downgrade	S&P downgrades from AAA to AA-plus
Hurricane Irene	Hurricane causes $10.1 billion in damages, 55 deaths
Indiana Fair Stage Collapse	Stage collapses during performance at the Indiana State Fair, 5 dead and 40 injured
Mumbai Bombings	Three bombings take place in Mumbai. 26 people dead and 130 injured.
Anna Hazare Anti-Corruption	Anna Hazare's anti-corruption protests against the Government of India
Steve Jobs Resigns	Steve Jobs resigns as Apple's CEO.
Google Purchases Motorola	Google buys Motorola for $12.5 billion.
Rupert Murdoch Scandal	Phone hacking scandal involving Rupert Murdoch.
The Situation and Abercrombie and Fitch	Abercrombie and Fitch asks "The Situation" to stop wearing their clothing.
Bert and Ernie Gay Marriage	Rumors of Bert and Ernie from Sesame Street being a gay couple.
Facebook Messenger	Facebook launches their new, independent messenger app.

Table 3.1: Topics and Descriptions of the 2011 Dataset, 4028 total Tweets

Class	Number of Tweets	Percentage Newsworthy/Not Newsworthy
5 - Definitely Credible	656	34% Newsworthy
4 - Seems Credible	602	
3 - Definitely Non-Credible	113	
2 - Relevant but not Newsworthy	2352	66% Not Newsworthy
1 - Spam	305	

Table 3.2: Class Distributions for 2011 Dataset

3.2 2013 Dataset

Topic	Description
Boston Marathon Bombings	Bombs explode near the finish line of the 2013 Boston Marathon. 3 people are killed and 260 are injured.
Typhoon Haiyan	Record breaking typhoon near the Philippines that claimed 6,000 lives.
Cyclone Phailin	550,000 people evacuated due to tropical cyclone near India.
Washington Navy Yard Shooting	12 people shot and killed by gunman inside the Naval Sea Systems Command.
Polar Vortex	Mid-western United States hit with record low temperatures in the winter of 2013.
Oklahoma Tornadoes	Tornado hits Moore, Oklahoma. 24 people dead and 377 injured.

Table 3.3: Topics and Descriptions of the 2013 Dataset, 2198 total Tweets

Class	Number of Tweets	Percentage Newsworthy/Not Newsworthy
5 - Definitely Credible	555	46% Newsworthy
4 - Seems Credible	371	
3 - Definitely Non-Credible	95	
2 - Relevant but not Newsworthy	845	54% Not Newsworthy
1 - Spam	332	

Table 3.4: Class Distributions for 2013 Dataset

The 2013 contains 2198 Tweets that spans 6 different topics. We explain the 2013 topics and class distributions in Tables 3.3 and 3.4 respectively.

3.3 Obtaining Twitter JSON Files

We use the Tweepy[2] API and write a Python script to pull the JSON objects of the tweets. Given a list of tweet ID's, we can make API requests to pull the JSON object for that tweet. The Twitter APR limits users to 180 requests every 15 minutes. A tweet JSON file contains

[2]http://www.tweepy.org/

information such as the body of the tweet, the author name, the author ID, the date the tweet was created, and other information. In Figure A.1 we show an example of a tweet JSON object. In Figure A.2 we show the Python code for obtaining the tweet JSON objects. JSON object are stored as Python dictionaries, and from these we can form the instances in the training and testing data.

3.4 Creating Training and Testing Data Files

The first step for converting tweet JSONs into feature vectors is to load each JSON pulled earlier into a collection of dictionaries. Figure 3.7 shows the Python code for storing each JSON as a dictionary.

```
for row in tweet_csv_reader:
        tweetid_credibility_dict[row[0]] = row[1]


# import all JSONs from directory into array
dict_array = []
for filename in os.listdir(tweet_jsons_dir):
            #argv[2] is the directory of JSON files
            json_file_reader = open(sys.argv[2]+'/'+filename,'r')

           #load a JSON object as a Python Dictionary
            tweet_dict = json.load(json_file_reader)

            dict_array.append(tweet_dict)
```

Figure 3.7: Creates a map of `<TweetID, Annotation>` pairs and loads each JSON into an array of Python `dict` objects.

In order to create the LibSVM [10] file used for testing and training, we need to glean the information from the JSON objects that are now Python dictionaries. A simple example is obtaining the number of URLs a tweet has. This, is shown in Figure 3.8.

```
#compute numURLS
numURLS = len(tweet_dict['entities']['urls'])

#compute hasStockSymbol
hasStockSymbol = 0 #False
    if '$' in tweet_dict['text'].encode('utf-8'):
                hasStockSymbol = 1 #True

#compute numQuestionMarks
numQuestionMarks = (tweet_dict['text'].encode('utf-8')).count('?')

#compute numExclamationMarks
numExclamationMarks = (tweet_dict['text'].encode('utf-8')).count('!')

#many more features calculated here in full code

credibilityScore = tweetid_credibility_dict[tweet_dict['id_str']]
instance = str(credibilityScore)+' qid:1'
                  + ' 1:'+str(numURLS)
                  +'  2:'+str(hasStockSymbol)
                  +'  3:'+str(numQuestionMarks)
                  + ' 4:'+str(numExclamationMarks)
```

Figure 3.8: Some examples of converting JSON data to features for an instance in the dataset.

```
<line> .=. <target> qid:<qid> <feature>:<value> <feature>:<value> ...
    ... <feature>:<value> # <info>
<target> .=. <positive integer>
<qid> .=. <positive integer>
<feature> .=. <positive integer>
<value> .=. <float>
<info> .=. <string>
```

Figure 3.9: Each instance in the training and testing data must have this form. The `<target>` field is the class of the instance. Each feature is listed as a positive integer, followed by its value as a float.

```
5 qid:1 1:1 2:0 3:1 4:0
```

Figure 3.10: An instance in the LibSVM format. Here, we show how the tweet in Figure A.1 is mapped to the features in Figure 3.8. The class of is this instance is ”5”, meaning it is definitely credible and newsworthy.

Each feature we want to compute is stored in a variable, and then concatenated together as a string. The first element in the string is the class the instance belongs to. In Figure 3.9 we show the grammar of an instance in the LibSVM format. Following the class are feature and value pairs, each feature has an integer ID and each value must be a floating point number. In order to label the instance with the proper class, we look up the annotation based on the tweet ID from the collection of `<TweetID, Annotation>` pairs. The full code for creating the LibSVM file and pulling the tweet JSONs is at www.wright.edu/~ross.138/TwitterCode.zip.

Approach

We describe our approach for improving ranker performance when the testing and training datasets are from two distinct years. We use the implementation of ranking SVM [26] by Chang et al [10] as our ranker. We describe our feature set, describe our new features, and show the ranking of features for each dataset based on their F-Score as described by Chen et al [12]. Here we list the feature set we used for our datasets. A majority of the features are popular features used in previous works. We categorize features as either being author based or tweet based. In total, we have 42 features, 11 of which are new. Table 4.1 lists and describes the tweet based features. Table 4.2 lists and describes the author based features.

The lexicon we used for the positive and negative words come from Hu et al. from their work on customer reviews [23]. Our curse word lexicon is the top ten most frequently used curse words as determined by Wang et al. [48].

Tweet Based Features	**Description**
numPositiveWords	Number of positive sentiment words in the tweet.
numNegativeWords	Number of negative sentiment words in the tweet.
numCurseWords	Number of curse words in the tweet
numSelfWords	Number of self words in the tweet. Self words are ”I”, ”Me”, ”My”, ”Mine”, ”I’m”, ”Myself”
tweetLength	The number of characters in the tweet total.
*differenceFromMeanPositive**	Given the average number of positive words per tweet for a topic; subtract from this the number of positive words in the tweet.
*differenceFromMeanNegative**	Given the average number of positive words per tweet for a topic; subtract from this the number of negative words in the tweet.
retweetCount	The number of times this tweet has been retweeted.
isReply	Whether or not this tweet is a reply to another tweet.
numURLS	The number of URLS this tweet contains
numMentions	The number of users mentioned in this tweet.
numHashtags	The number of hashtags this tweet contains.
numWords	The length of the tweet in terms of how many words it contains.
numPunctuation	The number of punctuation marks in the tweet. {. ! ? ; : , ” & ( ) - /}
*ratioPunctuationNumWords**	The number of punctuation marks in a tweet divided by the number of words in the tweet.
*ratioPunctuationTweetLength**	The number of punctuation marks in a tweet divided by the length of the tweet in terms of number of characters.
hasStockSymbol	Whether or not the tweet contains the stock symbol character $.
hasColonSymbol	Whether or not the tweet contains the colon symbol :.
hasGeoCoordinates	Whether or not the the post has a location associated with it.
hasVia	Whether or not the tweet contains the string ’via’, implying a retweet.
numUniqueCharacters	The number of distinct characters the body of the tweet contains.
numQuestionMarks	The number of question mark symbols in the body of the tweet.
numExclamationMarks	The number of exclamation mark symbols in the body of the tweet.
hasHappyEmoticon	Whether or not the tweet contains a positive emoticon, such as :) : D =)
hasSadEmoticon	Whether or not the tweet contains a negative emoticon, such as : (=(
*hasPray**	Whether or not the tweet contains the sub string ” pray”

Table 4.1: Tweet based features. Features with an asterisk denote our new features.

Author Based Features	**Description**
isVerified	Whether or not the author's account has been verified by Twitter.
numFollowers	The number of other Twitter users that follow the author's account.
numFriends	The number of other Twitter users the author follows.
ratioNumFollowersNumFriends	The number of followers the author has divided by the number of friends the user has.
userAge	In years, how long the author has been active on Twitter.
numStatuses	The number of original tweets the author has produced.
screenNameLength	The number of characters in the author's screen name.
numDescriptionURLS	The number of URLS in the author's biography.
hasDescription	Whether or not the author filled out the optional biography for their account.
descriptionLength	The number of total characters in the author's description.
*ratioNumFollowersUserAge**	The number of followers the user has divided by how long they have been on Twitter in years.
*ratioNumFriendsUserAge**	The number of friends the user has divided by how long they have been on Twitter in years.
*ratioNumStatusesUserAge**	The number of tweets a user has made divided by how long they have been on Twitter in years.
*numPositiveWordsDescription**	The number of punctuation marks in the tweet. {. ! ? ; : , " & ( ) - /}
*numNegativeWordsDescription**	The number of punctuation marks in a tweet divided by the number of words in the tweet.
*numCurseWordsDescription**	The number of punctuation marks in a tweet divided by the length of the tweet in terms of number of characters.

Table 4.2: Author based features. Our new features are denoted with an asterisk.

4.1 New Features

We describe the motivation behind our new features, as well as how these features are calculated.

differenceFromMeanPositive/Negative: Castillo et al. noted in their paper that tweets that have negative sentiment tend to be credible [8]. We create two features that aim to capture when the sentiment of a tweet matches the overall sentiment of the topic it is in. We hypothesize that tweets that have similiar sentiment to the rest of the tweets in the topic will be credible. However, if a tweet does not have the same sentiment of the topic overall, this could mean the tweet in non-credible or not newsworthy. First, we calculate the average number of positive and negative words for all tweets in the topic. For our purposes, we use the tweets in our dataset and do not use tweets outside of our annotated tweet set to calculate the average number of positive and negative words. Then, for each tweet, we calculate the number of positive and negative words in the tweet, and find the difference between this number and the mean number of positive and negative words for the tweets of that topic. In Table 4.3 we show the average number of positive and negative words per tweet for each topic. In general, the high impact and negative (i.e., The Boston Marathon Bombings, Mumbai Blasts) topics tend to have more negative words per tweet than positive words. Topics that are less negative (i.e., Bert and Ernie Gay Marriage, the Facebook Messenger) tend to have more positive words than serious and high impact topics. Figure 4.2 shows how this feature is calculated.

Topic	**Average Number of Positive Words Per Tweet**	**Average Number of Negative Words Per Tweet**
Anna Hazare Protests (2011)	.3714	.3
Virginia Earthquake(2011)	.0857	.2
Facebook Messenger (2011)	.3584	.0189
Google Buys Motorola (2011)	.1119	.0299
Hurricane Irene (2011)	.0945	.1417
State Fair Stage Collapse (2011)	.0833	.2121
Libya Rebels (2011)	.1075	.3118
London Riots (2011)	.16	.38
Mumbai Blasts (2011)	.1795	.8461
Rupert Murdoch Scandal (2011)	.0571	.3333
Stock Downgrade (2011)	.1034	.5747
Bert and Ernie Gay Marriage (2011)	.1379	.2241
Steve Jobs Resigns (2011)	.2439	.1382
The Situation and A&F (2011)	.2214	.1526
Boston Marathon Bombings (2013)	.1357	.5
Cyclone Phailin (2013)	.1899	.6783
Navy Yard Shooting (2013)	.0867	.5667
Oklahoma Tornadoes (2013)	.1912	.4645
Philippine Typhoon (2013)	.352	.504
Polar Vortex (2013)	.1939	.6182

Table 4.3: Average sentiment per Tweet for each topic

```
1
2 For each tweet (t)  in topic:
3             numPositiveWords = 0
4             numNegativeWords = 0
5     For each word in t:
6         if word in positiveLexicon: numPositiveWords++
7                         if word in negativeLexicon: numNegativeWords++
8           distanceFromMeanPositive(t) = ...
                meanNumberOfTopicPositiveWords - numPositiveWords
9           distanceFromMeanNegative(t) = ...
                meanNumberOTopicfNegativeWords - numNegativeWords
```

Figure 4.1: Pseudocode for calculating the `differenceFromMeanNegative` and `differenceFromMeanPositive` for each tweet, given its topic.

ratioPunctuationNumWords and ratioPunctuationNumCharacters: Morris et al. [37] discovered in their study on credibility perceptions that users perceive irregular grammar as a sign of non-credibility [37]. In order to capture this, we create two new features, `ratioPunctuationNumWords` and `ratioPunctuationNumCharacters`. It is normal for users to not adhere to standard grammar rules when composing tweets due to the character limit on Twitter. Thus, we believe one way to capture anamolous grammar is if the user uses too many punctuation marks, or if they use too few punctuation marks. How these features are calculated are shown in Equations 4.1 and 4.2, where t is the tweet in question.

$$ratioPunctuationWords\,(t) = \frac{numPunctuation\,(t)}{numWords\,(t)} \tag{4.1}$$

$$ratioPunctuationNumCharacters\,(t) = \frac{numPunctuation\,(t)}{numCharacters\,(t)} \tag{4.2}$$

ratioNumFollowersUserAge, ratioNumFriendsUserAge, ratioNumStatusesUserAge: Here we describe three author based features that are the ratios of features of an author and the amount of time the author has been active on Twitter. We hypothesize that people who have been on Twitter for a longer period of time will tend to have more followers, more friends, and have produced more tweets. If people have an unusually high amount of either of these, then this could influence perceived credibility of this author. If a user has a high number of followers for the amount of time they have been active on Twitter, this could imply that many of their followers are bots. If a user produces a large number of tweets for the amount of time they have been active on Twitter, this can capture whether or not the user tends to tweet many spam or non-newsworthy tweets. In Equations 4.3, 4.4, and 4.5 we show how each of these features are computed for a given user, u.

$$ratioNumFollowersUserAge\,(u) = \frac{numFollowers\,(u)}{accountAge\,(u)} \tag{4.3}$$

$$ratioNumFriendsUserAge\,(u) = \frac{numFriends\,(u)}{accountAge\,(u)} \tag{4.4}$$

$$ratioNumStatusesUserAge\,(u) = \frac{numStatuses\,(u)}{accountAge\,(u)} \tag{4.5}$$

Author description based features: We also create three author based features that are based on the sentiment of the description of the author: `numNegativeWordsDescription`, `numPositiveWordsDescription`, and `numCurseWordsDescription`. Sentiment based features for the text of tweet has been well explored in previous works. We simply map this feature to the optional descriptions each user on Twitter has. If the author does not have a description, each of these features are set to 0.

```
1
2  For each tweet (t) in topic:
3              a = author of t
4              numPositiveWordsDescription = 0
5              numNegativeWordsDescription = 0
6      numCurseWordsDescription = 0
7
8             if a.description is null:
9               #nothing
10           else
11      for each word in a.description:
12          if word in positiveLexicon: numPositiveWordsDescription(u)++
13          if word in negativeLexicon: numNegativeWordsDescription(u)++
14          if word in curseWordLexicon: numCurseWordsDescription(u)++
```

Figure 4.2: Pseudocode for calculating the `numPositiveWordsDescription`, `numNegativeWordsDescription`, and `numCurseWordsDescription` of a user, u, for the given tweet, t.

hasPray: We create a feature, `hasPray` to detect whether or not the tweet has the sub string " pray". We pick this word because it appears in tweets that are relevant to a topic, but do not contain any newsworthy information. Tweets that contain the " pray" substring are often people offering emotional support, and do not contribute to helping the reader understand the event.

4.2 Feature Ranking

We rank our entire feature set for each dataset with the LibSVM extention developed by Chen et al. that uses F-Score to rank the features [12].

Feature Ranking for 2011 Dataset	**Feature Ranking for 2013 Dataset**
numURLs (.3215)	numURLs (.3215)
hasStockSymbol(.2529)	hasColonSymbol(.2779)
hasColonSymbol(.2445)	numUniqueCharacters(.2034)
numUniqueCharacters(.1464)	numPunctuation(.1910)
numPunctuation(.1348)	**hasPray*(.1496)**
ratioPunctuationTweetLength*(.1006)	**ratioPunctuationTweetLength*(.1188)**
ratioPunctuationNumWords*(.0712)	**ratioPunctuationNumWords*(.0819)**
numSelfWords(.06)	numSelfWords(.0817)
hasVia(.031471)	tweetLength(.069)
isReply(.0237)	**numNegativeWordsDescription*(.045)**
numHashtags(.0237)	numNegativeWords(.0437)
numQuestionMarks(.0227)	**differenceFromMeanNegative*(.0391)**
hasPray*(.0177)	numStatuses(.0369)
numPositiveWords(.0158)	**ratioNumStatusesUserAge*(.0342)**
differenceFromMeanPositive*(.0141)	hasVia(.0304)
numPositiveWordsDescription*(.0135)	numExclamationMarks(.0245)
numStatuses(.0111)	numHashtags(.0226)
ratioNumStatusesUserAge*(.0099)	numQuestionMarks(.0214)
numCurseWords(.00838)	numWords(.0172)
tweetLength(.0078)	isReply(.0161)
numCurseWordsDescription*(.0075)	**differenceFromMeanPositive*(.0143)**
numExclamationMarks(.0073)	retweetCount(.0129)
numWords(.0073)	numPositiveWords(.0129)
numNegativeWords(.0042)	descriptionLength(.0118)
differenceFromMeanNegative*(.0036)	**numPositiveWordsDescription*(.0114)**
numFriends(.0025)	numDescriptionURLS (.0083)
numNegativeWordsDescription*(.0025)	hasGeoCoordinates(.0075)
hasHappyEmoticon(.0025)	hasDescription(.0075)
ratioNumFollowersNumFriends(.0024)	ratioNumFollowersNumFriends(.0074)
ratioNumFriendsUserAge*(.0021)	**ratioNumFollowersUserAge*(.0052)**
descriptionLength(.002)	hasHappyEmoticon(.0049)
screenNameLength(.002)	numFollowers(.0048)
numFollowers(.002)	numCurseWords(.0047)
hasSadEmoticon(.0019)	**numCurseWordsDescription*(.0047)**
numDescriptionURLS(.0017)	screenNameLength(.0046)
hasDescription(.0014)	hasSadEmoticon(.0046)
hasGeoCoordinates(.0014)	**ratioNumFriendsUserAge*(.0041)**
retweetCount(.001)	numMentions(.0036)
ratioNumFollowersUserAge*(.001)	numFriends(.0035)
numMentions(.0008)	userAge(.0027)
userAge(.0006)	hasStockSymbol(.0014)
isVerified(.0000)	isVerified(.0000)

Table 4.4: The ranking of our feature sets based on F-Score. Our new features are denoted with an asterisk.

In Table 4.4, we list the ranked features in order from best to worst. There is significant overlap in the top 10 features across each dataset. One feature in particular, `hasStockSymbol` ranks very highly for the 2011 dataset, but is ranked extremely low in the 2013 dataset. This was also the case in the TweetCred paper by Gupta et al. [18]. This is a sign that this particular feature is overfit for the 2011 dataset, and has no bearing for the 2013 dataset. In the next section, we will show the affects of removing this particular feature and how it affects ranker performance.

The top features for each dataset come as no suprise. The feature `numURLS` ranks as the best feature for each dataset. The feature `hasColonSymbol` is the text based feature where it indicates whether or not the tweet has a colon in it. Colons appear in tweets that re-tweets, and colons often precede a URL. The feature `numUniqueCharacters` correlates with high credibility because tweets that have hashtags and URLs are likely to have more unusual characters than just plain text. Two of our new features, `ratioPunctuationTweetLength` and `ratioPunctuationNumWords` appear in the top 10 features for each data set. The `hasPray` feature ranks 13 out of 42 for the 2011 dataset and 5th for the 2013 dataset. The worst features are features that are not present in a majority of tweets or users, or, these features appear equally for all users and for all tweets. The feature `isVerified` ranks as the worst feature because most Twitter users do not have verified accounts.

4.3 Results

We show how the ranking SVM algorithm performs using our feature set. Ranking SVM is described by Joachims in [27] and we use the implementation that is an extension of LibSVM [10] by Lee et al. [31]. We use Normalized Discounted Cumulative Gain (NDCG) as our performance metric in order to compare results with the previous work. As described in Equation 4.6, we select a stopping point, p, and yield the DCG score based on how well the ranker orders the first p documents (in our case tweets) based on their relevance score. If tweets with lower relevance are placed before documents with high relevance, the DCG score in penalized. The Ideal Discounted Cumulative Gain, $IDCG$, is the DCG score of a ranker the ranks the documents based on relevance perfectly. The $NDCG$ of a ranker is the actual DCG score of the ranker divided by the $IDCG$ that the ranker could theoretically achieve if it is perfect. The $NDCG$ score will always be between 0 and 1. If the ranker correctly ranks the first p documents, then its $NDCG_p$ score would be 1.0. For our purposes, the possible rel scores are 5 (definitely credible and newsworthy), 4 (seems credible and newsworthy), 3(non-credible and newsworthy), 2 (relevant to a topic, but contains no information), and 1 (spam).

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2\,(i+1)} \tag{4.6}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{4.7}$$

We apply linear scaling to each feature so that each value falls between -1 and 1. We also randomly shuffle the data instances so that when we use cross validation, tweets from similar topics are not grouped together.

In Table 4.5 we show how well the ranking SVM algorithm performs on the 2011 dataset with 4 cross-fold validation. For the 2011 dataset, ranker performance decreases when removing the `hasStockSymbol` feature. Earlier we discussed how this feature was

deemed important for the 2011 dataset, but was deemed unimportant for the 2013 dataset. This is also reflected in Tables 4.6 and 4.7. Table 4.6 shows the results when training and testing with the 2013 dataset. Ranker performance increases slightly after removing the `hasStockSymbol` feature. Table 4.7 shows that ranker performance improves slightly after removing the `hasStockSymbol` feature that seems to overfit the 2011 dataset.

2011 Dataset with stock symbol feature		***2011 Dataset without stock symbol feature***	
NDCG@25	.9698	**NDCG@25**	.6313
NDCG@50	.9337	**NDCG@50**	.6433
NDCG@75	.855	**NDCG@75**	.6404
NDCG@100	.8082	**NDCG@100**	.6492

Table 4.5: Ranker performance on the 2011 Dataset with 4 Cross-fold validation. Ranker performance drops when the stock symbol feature is removed.

2013 Dataset with stock symbol feature		***2013 Dataset without stock symbol feature***	
NDCG@25	.7770	**NDCG@25**	.7976
NDCG@50	.7391	**NDCG@50**	.7489
NDCG@75	.7287	**NDCG@75**	.7359
NDCG@100	.7147	**NDCG@100**	.7271

Table 4.6: Ranker performance on the 2013 Dataset with 4 Cross-fold validation. Ranker performance improves slightly when the stock symbol feature is removed.

Train 2011 + Test 2013 with all features		***Train 2011 + Test 2013 after removing the stock symbol feature***	
NDCG@25	.5784	**NDCG@25**	.6286
NDCG@50	.6407	**NDCG@50**	.6637
NDCG@75	.6342	**NDCG@75**	.7033
NDCG@100	.6641	**NDCG@100**	.6998

Table 4.7: Ranker performance when training on the 2011 dataset and testing on the 2013 dataset. Overall, the NDCG score improves after removing the problematic stock symbol feature that seems to overfit the 2011 dataset.

Conclusion and Future Work

5.1 Conclusion

We summarize our results and compare them with the results reported in the TweetCred paper by Gupta et al. [18]. Specifically, we used the same ranking algorithm (ranking SVM) and the same datasets for meaningful comparison. Essentially, we show that feature selection you greatly impacts ranker performance when the training and testing data come from two distinct years. This is an important result because we are constantly gaining new and unseen data that needs to be verified in terms of credibility and newsworthiness.

TweetCred - Train 2011 + Test 2013		***Our approach with new features - Train 2011 + Test 2013***	
NDCG@25	.3743	**NDCG@25**	.6286
NDCG@50	.3693	**NDCG@50**	.6637
NDCG@100	.3783	**NDCG@100**	.6998

Table 5.1: The results in the left column are the results reported by Gupta et al. [18]. The results on the right are the NDCG scores we achieved with our feature set.

5.2 Future Work

One of the challenges of doing research in this domain is that there is no concrete and set definition for "credibility". In the previous works, each approach had their own definition of credibility the human annotators use. This means that people who use a different definition

for credibility cannot meaningfully compare results with one another. We need to develop a common and concrete definition for credibility so that we can meaningfully compare results. A common hub for human annotators could help alleviate this problem. This could lead to building one shared database amongst many research groups so that everyone is working under the same pretenses, and results can be meaningfully compared. The process of annotating tweets based on credibility is a long process; if there is a common database with annotated tweets this could save a lot of time for data creation and evaluation, and can provide a common baseline for comparison.

Bibliography

[1] Mohammad-Ali Abbasi and Huan Liu. Measuring User Credibility in Social Media. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 441–448. Springer, 2013.

[2] Adam Acar and Yuya Muraki. Twitter for crisis communication: lessons learned from Japan's tsunami disaster. *International Journal of Web Based Communities*, 7(3):392–402, 2011.

[3] RMB Al-Eidan, HS Al-Khalifa, and AS Al-Salman. Measuring the credibility of Arabic text content in Twitter. In *2010 Fifth International Conference on Digital Information Management (ICDIM)*, pages 285–291. IEEE, 2010.

[4] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. Tweedr: Mining twitter to inform disaster response. *Proc. of ISCRAM*, 2014.

[5] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 492–499. IEEE, 2010.

[6] Todd Bodnar, Victoria C Barclay, Nilam Ram, Conrad S Tucker, and Marcel Salathé. On the ground validation of online diagnosis with Twitter and medical records. In *Proceedings of the Companion Publication of the 23rd International Conference on*

World Wide Web Companion, pages 651–656. International World Wide Web Conferences Steering Committee, 2014.

[7] Christina Boididou, Symeon Papadopoulos, Yiannis Kompatsiaris, Steve Schifferes, and Nic Newman. Challenges of computational verification in social multimedia. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pages 743–748. International World Wide Web Conferences Steering Committee, 2014.

[8] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information Credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 675–684. ACM, 2011.

[9] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Predicting information credibility in time-sensitive social media. *Internet Research*, 23(5):560–588, 2013.

[10] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] Akemi Takeoka Chatfield, Hans J Jochen Scholl, and Uuf Brajawidagda. Tsunami early warnings via Twitter in government: Net-savvy citizens' co-production of time-critical public information services. *Government Information Quarterly*, 30(4):377–386, 2013.

[12] Yi-Wei Chen and Chih-Jen Lin. Combining SVMs with various feature selection strategies. In *Feature Extraction*, pages 315–324. Springer, 2006.

[13] Clodoveu A Davis Jr, Gisele L Pappa, Diogo Rennó Rocha de Oliveira, and Filipe de L Arcanjo. Inferring the location of Twitter messages based on user relationships. *Transactions in GIS*, 15(6):735–751, 2011.

[14] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

[15] Alper Gün and Pınar Karagöz. A Hybrid Approach for Credibility Detection in Twitter. In *Hybrid Artificial Intelligence Systems*, pages 515–526. Springer, 2014.

[16] Aditi Gupta and Ponnurangam Kumaraguru. Twitter explodes with activity in mumbai blasts! a lifeline or an unmonitored daemon in the lurking? IIIT. Technical report, Delhi, Technical report, IIITD-TR-2011-005, 2011.

[17] Aditi Gupta and Ponnurangam 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.

[18] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. TweetCred: A Real-time Web-based System for Assessing Credibility of Content on Twitter. *CoRR*, abs/1405.5490, 2014. Accessed May 2014.

[19] Aditi Gupta, Hemank Lamba, and Ponnurangam Kumaraguru. $1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on Twitter. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–12. IEEE, 2013.

[20] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam 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.

[21] Manish Gupta, Peixiang Zhao, and Jiawei Han. Evaluating Event Credibility on Twitter. In *SDM*, pages 153–164. SIAM, 2012.

[22] Mengdie Hu, Shixia Liu, Furu Wei, Yingcai Wu, John Stasko, and Kwan-Liu Ma. Breaking news on Twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2751–2754. ACM, 2012.

[23] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[24] Yukino Ikegami, Kenta Kawai, Yoshimi Namihira, and Setsuo Tsuruta. Topic and Opinion Classification Based Information Credibility Analysis on Twitter. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4676–4681. IEEE, 2013.

[25] Thorsten Joachims. Making large scale SVM learning practical. Technical report, Universität Dortmund, 1999.

[26] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM, 2002.

[27] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226. ACM, 2006.

[28] Byungkyu Kang, John O'Donovan, and Tobias Höllerer. Modeling topic specific credibility on Twitter. In *Proceedings of the 2012 ACM international Conference on Intelligent User Interfaces*, pages 179–188. ACM, 2012.

[29] Alisa Kongthon, Choochart Haruechaiyasak, Jaruwat Pailai, and Sarawoot Kongyoung. The role of Twitter during a natural disaster: Case study of 2011 Thai Flood. In *2012 Proceedings of PICMET'12: Technology Management for Emerging Technologies (PICMET)*, pages 2227–2232. IEEE, 2012.

[30] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600. ACM, 2010.

[31] Ching-Pei Lee and Chih-Jen Lin. Large-scale linear ranksvm. *Neural Computation*, 26(4):781–817, 2014.

[32] Janette Lehmann, Carlos Castillo, Mounia Lalmas, and Ethan Zuckerman. Finding news curators in twitter. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pages 863–870. International World Wide Web Conferences Steering Committee, 2013.

[33] Xin Lu and Christa Brelsford. Network Structure and Community Evolution on Twitter: Human behavior change in response to the 2011 Japanese Earthquake and Tsunami. *Scientific reports*, 4, 2014.

[34] Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. Where Is This Tweet From? Inferring Home Locations of Twitter Users. In *ICWSM*, 2012.

[35] Marcelo Mendoza, Barbara Poblete, and Carlos 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.

[36] Alexander Mills, Rui Chen, JinKyu Lee, and H Raghav Rao. Web 2.0 emergency applications: How useful can Twitter be for emergency response? *Journal of Information Privacy and Security*, 5(3):3–26, 2009.

[37] Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia 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.

[38] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, Stanford, CA, 1998.

[39] Hemant Purohit, Andrew Hampton, Shreyansh Bhatt, Valerie L Shalin, Amit P Sheth, and John M Flach. Identifying Seekers and Suppliers in Social Media Communities to Support Crisis Coordination. *Computer Supported Cooperative Work (CSCW)*, 23(4-6):513–545, 2014.

[40] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu 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.

[41] Tushar Rao and Saket Srivastava. Analyzing stock market movements using Twitter sentiment analysis. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 119–123. IEEE Computer Society, 2012.

[42] Srijith Ravikumar, Raju Balakrishnan, and Subbarao Kambhampati. Ranking tweets considering trust and relevance. In *Proceedings of the Ninth International Workshop on Information Integration on the Web*, page 4. ACM, 2012.

[43] S. Shariff, X. Zhang, and M. Sanderson. User Perception of Information Credibility of News on Twitter. In *Advances in Information Retrieval*, pages 513–518. Springer, 2014.

[44] Alessio Signorini, Alberto Maria Segre, and Philip M Polgreen. The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic. *PloS one*, 6(5):e19467, 2011.

[45] Yu Suzuki. A credibility assessment for message streams on microblogs. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on*, pages 527–530. IEEE, 2010.

[46] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. volume 10, pages 178–185, 2010.

[47] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what Twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1079–1088. ACM, 2010.

[48] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. Cursing in english on twitter. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, pages 415–425, New York, NY, USA, 2014. ACM.

[49] Xin Xia, Xiaohu Yang, Chao Wu, Shanping Li, and Linfeng Bao. Information credibility on Twitter in emergency situation. In *Intelligence and Security Informatics*, pages 45–59. Springer, 2012.

[50] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM, 2012.

[51] Jiang Yang, Scott Counts, Meredith Ringel Morris, and Aaron Hoff. Microblog credibility perceptions: Comparing the USA and China. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 575–586. ACM, 2013.

[52] Xue Zhang, Hauke Fuehres, and Peter A Gloor. Predicting stock market indicators through Twitter; I hope it is not as bad as I fear. *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.

Appendix A

Source Code

```
{
    "contributors": null,
    "truncated": false,
    "text": "What is the polar vortex and what causes it? http://t.co/nnhLfRQqD8",
    "in_reply_to_status_id": null,
    "id": 420296382629421060,
    "favorite_count": 0,
    "source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>",
    "retweeted": false,
    "coordinates": null,
    "entities": {
        "symbols": [],
        "user_mentions": [],
        "hashtags": [],
        "urls": [
            {
                "url": "http://t.co/nnhLfRQqD8",
                "indices": [
                    45,
                    67
                ],
                "expanded_url": "http://www.cbsnews.com/news/what-is-the-polar-vortex-and-what-causes-it",
                "display_url": "cbsnews.com/news/what-is-t   "
            }
        ]
    },
    "in_reply_to_screen_name": null,
    "id_str": "420296382629421056",
    "retweet_count": 0,
    "in_reply_to_user_id": null,
    "favorited": false,
    "user": {
        "follow_request_sent": false,
        "profile_use_background_image": true,
        "profile_text_color": "634047",
        "default_profile_image": false,
        "id": 143935499,
        "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme3/bg.gif",
        "verified": false,
        "profile_location": null,
        "profile_image_url_https": "https://pbs.twimg.com/profile_images/425000239389364224/6MXj0yGg_normal.jpeg",
        "profile_sidebar_fill_color": "E3E2DE",
        "entities": {
            "description": {
                "urls": []
            }
        },
        "followers_count": 26,
        "profile_sidebar_border_color": "D3D2CF",
        "id_str": "143935499",
        "profile_background_color": "EDECE9",
        "listed_count": 0,
        "is_translation_enabled": false,
        "utc_offset": -21600,
        "statuses_count": 691,
        "description": "Soy Mexicano. Me Encanta Viajar por Carretera!!!",
        "friends_count": 127,
        "location": "Por ahora vivo en Texas",
        "profile_link_color": "088253",
        "profile_image_url": "http://pbs.twimg.com/profile_images/425000239389364224/6MXj0yGg_normal.jpeg",
        "following": false,
        "geo_enabled": true,
        "profile_banner_url": "https://pbs.twimg.com/profile_banners/143935499/1403267290",
        "profile_background_image_url": "http://abs.twimg.com/images/themes/theme3/bg.gif",
        "name": "Otilio Vallejo",
        "lang": "en",
        "profile_background_tile": false,
        "favourites_count": 129,
        "screen_name": "OtilioVallejo",
        "notifications": false,
        "url": null,
        "created_at": "Fri May 14 21:00:46 +0000 2010",
        "contributors_enabled": false,
        "time_zone": "Central Time (US & Canada)",
        "protected": false,
        "default_profile": false,
        "is_translator": false
    },
    "geo": null,
    "in_reply_to_user_id_str": null,
    "possibly_sensitive": false,
    "lang": "en",
    "created_at": "Mon Jan 06 20:50:41 +0000 2014",
    "in_reply_to_status_id_str": null,
    "place": null
}
```


Figure A.1: An example of a tweet JSON object

```
# argv[1] = list of tweet ids
# argv[2] = file to store missing ids
# argv[3] = file to store error log
# argv[4] = dir to store jsons

import json
import os
import tweepy
import sys
import time
from tweepy.parsers import *

# Put in your own Twitter App credentials here
consumer_key = 'CONSUMER_KEY_HERE'
consumer_secret = 'CONSUMER_SECRET_HERE'
access_token = 'ACCESS_TOKEN_HERE'
access_secret = 'ACCESS_SECRET_HERE'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

#Pull tweets as a JSON
api = tweepy.API(auth, parser = JSONParser())

# Input files
ids_dir = sys.argv[1]
error_ids_dir = sys.argv[2]
error_log_dir = sys.argv[3]

#file writers
ids_fr = open(ids_dir, 'r')
error_ids_fa = open(error_ids_dir, 'a+')
error_log_fa = open (error_log_dir, 'a+')

# Pull 150 tweets at a time. Really you get 180, but I would occasioanlly
# get errors when trying to pull in 180 at a time. 150 is a safer underestimate
i = 0
for tweet_id in ids_fr:
    if i%151 == 1:
        print "Rate limit reset:" + api.rate_limit_status().__str__()



    try:
        # create the file to store JSON
        fw = open (sys.argv[4]+'/json'+str(i)+'.json', 'w+')

        #get tweet as dict, this is the request to the API
        tweet = api.get_status(tweet_id)

        #dict -> JSON
        tweet_json = json.dumps(tweet)

        fw.write(tweet_json +'\n \n')
        time.sleep(1)
    except TweepError, e:
        # could be caused by a Tweet no longer existing or other
        # log the error and the id that caused it
        error_log_fa = open (error_log_dir, 'a+')
        error_log_fa.write(e.__str__())
        error_log_fa.write(tweet_id)
        error_log_fa.close()
        error_ids_fa = open(error_ids_dir, 'a+')
        error_ids_fa.write(tweet_id)
        error_ids_fa.close()

    i = i + 1
    if i % 151 == 0:
        print "Rate limit hit:" + api.rate_limit_status().__str__()
        # sleep for 15 minutes
        time.sleep(910)
```

Figure A.2: Code to pull Tweets via the Tweepy API