Predicting Large-Scale Internet Censorship

- A Machine Learning Approach

A thesis presented to the faculty of

the School of Engineering and Applied Science

University of Virginia

in partial fulfillment of

the requirements for the degree of

Master of Science

Systems and Information Engineering

by

Jin Li

August 2015

APPROVAL SHEET

The thesis ▼

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science ▼

AUTHOR

The thesis has been read and approved by the examining committee: ▼

Dr. Gerard P. Learmonth

Advisor

Dr. Matthew Gerber

Dr. Brantly Womack

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, Dean, School of Engineering and Applied Science

August ▼

2015

ABSTRACT

Social media have an increasingly penetrating effect on our daily lives and entire society. Reviewing on social media research conducted in the past, one important aspect, content deletion due to Internet censorship, has received little direct attention in light of the ongoing media censorship in China. Exposing this aspect of censorship allows citizens to better understand the mechanism of Internet censorship, to help them make informed decisions on how to efficiently participate in society events and in the larger context to maintain a free and open Internet. Our research aims to facilitate a better understanding of social media censorship, and to provide means to automatically detect and predict future content deletion. In this research, a machine learning approach is introduced and applied for this effort. Our research results have revealed vital correlations between the occurrence of real-world political events and online censorship activities as well as public opinion and sentiment expressed; a framework is proposed to predict which microblog will be more likely to be deleted under Internet censorship; and first results are produced. Furthermore, we evaluate model performance by incorporating public sentiment as an aggregate feature in model construction and test the feasibility. As a result, we achieve 95.6% AUC score using naïve Bayes algorithm with social features. To our knowledge, this is the first analysis results ever reported in such task.

ACKNOWLEDGEMENTS

Foremost, I would like to express my deepest appreciation to my advisor Dr. Gerard P. Learmonth for his continuous support; for his guidance for me to develop interests and pursue my passion on this research topic; for his patience, encouragement, knowledge and experience. This thesis would not have been possible without him.

Besides my advisor, I want to give thanks to the members of my thesis committee: Dr. Matthew Gerber and Dr. Brantly Womack for their encouragement, insightful comments and feedback.

To my fellow colleagues in UVA Center for Large-Scale Computational Modeling (CLSCM): Chee Chun “Johnny” Gan, Michael Purvis, Samantha Vacik and Osama Eshera, I also want to say thanks for the stimulating discussion, helpful feedback, and for the time we spent together eating all kinds of bagels.

Last but not least, I want to give my thanks to my parents Zhang You Li and Xia Wang for their unconditional love and support throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 – INTRODUCTION

Social media, as an ever-expanding platform for the public to express opinions, have had an increasingly penetrating effect on our daily lives and entire society. Many of the issues, such as social equity, ecology, and national security, have led to massive online social discussions and responses. Rich information that has been created and shared by users on social media platform is relatively easy to access via open APIs.

The power to provide a different perspective reflecting on real-world events has led to social media being used to study many real-world phenomena, such as predicting presidential elections [1], associating stock movement with online sentiment variations [2], predicting earthquake [3], flu outbreaks [4], networking and formulating political uprisings and social protests [5] [6, 7, 8], predicting user re-tweet behavior [9], and automatically detecting violent extremists' cyber-recruitment [10].

Recently, machine learning techniques have been applied on many of the tasks in social media research, such as sentiment classification [11], topic detection [12], spam email classification [13], re-tweet behavior prediction [9], and have obtained good performance. After reviewing the social media research that conducted in the past, we found there is an important aspect that has not yet received much direct attention, which is content deletion due to censorship. However, due to the rising interest in studies of Internet security and media censorship, studies of online deletion behavior began to draw research interests.

In the past, studies that investigate the issue of Internet censorship are primarily focused on developing systems to detect censorship and to provide descriptive statistics on its mechanism, such as identifying keywords, influential or controversial users who are more likely to be censored, and the aggregate properties such as time span, speed and patterns evolved [14, 15].

Given the fact that detecting censorship is possible [16] and extensive empirical analysis of the censored content has been conducted in the past, we now focus on further investigating the possibility of predicting Internet censorship via machine learning approach. Meanwhile, we provide descriptive statistics on aggregate properties as well as the correlation between public opinions on social media and real-world phenomenon by a case study to facilitate a better understanding of Internet censorship and its effect on political uprisings in heavy-censored environment.

This thesis contributes to the study of social media and understanding the effect of Internet censorship in the following ways:

- We identify patterns and trends emerged from Internet censorship on microblogs and revealed correlation between occurrence of real-world political events and online censored volume and public sentiment;
- We identify a new machine learning problem - predicting which microblog will be more likely to be deleted in the future under the impact of Internet censorship;
- We propose a framework to solve the above machine learning problem and perform the first results; and
- We present a novel approach by extracting and incorporating public sentiment as an aggregate feature in model construction and evaluate model performance to test its feasibility;

The paper is organized as follows: firstly, in beginning two chapters, we present an overview of the Chinese Internet, Internet censorship, Chinese microblogging, as well as Sina Weibo to prepare readers to get familiar with the background, terminology and significance in using the data from this platform in our approach. Then, in chapter 2, we highlight related works that examine microblog deletion and censorship via different approaches. In Chapter 3 we present our methodology and outline data collection, feature extraction, sampling and classification setup. Chapter 4 presents sample datasets and our results both from real-world correlation and sensitive words detection via descriptive statistical analysis and from predictive modeling along with results discussion. Finally, chapter 5 concludes the research and discusses future work.

Internet Censorship in China

As an ever-expanding platform for public discourse, social media increasingly affect our lives especially our decisions and behavior on how to participate in society events. In China, social media have mushroomed to a gargantuan scale. The Pew Research Center claims “China has more Internet users than nearly other countries have people” [17]. According to Statistical Report on Internet Development in China 2013, among the 591 million Chinese Internet users, 91% use social networking sites, compared to 67% in the United States [18]. As a result, the one-party state is increasingly recognizing and embracing the power of social media. In a national anti-corruption campaign launched in January 2013, Chinese President Xi Jinping assumed the role of social media as “a sharp weapon in the fight against corruption.” The state media also live tweeted one of Xi’s routine visits to rural Hebei province [19]. However, Chinese government has not loosened, and to some degree has even tightened its control over the Internet via censorship on sensitive content. According to King, approximately 13% of all social media posts are censored in China [20]. A law has passed to require social media users to provide real names, and instructs service providers to engage in first-line censorship.

Internet censorship has been known to the public both from individuals commenting on their own message disappearances, and from allegedly leaked memos from the Chinese government instructing media to remove all content relating to some specific keyword or event. When the microblog deletion is detected, reasons behind can vary from maintain individual privacy and social image, in which case it is the posters themselves deleted the content they have previously posted. On the other hand, when systematic large-scale content deletion occurs, justifications may range from maintaining public order and safety to protection of morality from obscenity to the protection of intellectual property or copyright [20]. However, in most cases, large-scale content deletion has been viewed as a hindrance to a free and transparent society, which as past research revealed, intends to suppress collective activities, such as social protests that may arise and as a result of online discussions [21]. In light of the ongoing censorship of media in China, exposing censorship and the methods used to achieve it allows citizens to make informed decisions about how they participate in society to ensure freedom of speech and access to information, which in a larger context to maintain a free and open Internet.

Our approach aims to facilitate a better understanding of social media censorship as a large-scale systematic deletion by presenting descriptive statistics on public opinion and sentiment and its correlation with real-world events as well as providing means to automatically monitor and predict future message deletion on the Internet.

Sina Weibo

The Chinese Internet has reached an extraordinary speed and breath of individual connectivity in just a decade. By end of 2013, China has reached about 591 million Internet users penetrating 40%

of country's population. There is no other country with more citizens, in absolute numbers, using the Internet.

Microblog ("weibo" in Chinese) refers to mini-blogging services in China, including social chat sites and platform sharing. Weibo uses a format similar to its U.S. counterpart Twitter. Weibo and Twitter both allow users to post 140-character long messages; a user can follow other users to easily receive their tweets in an aggregated news feed. By default, all messages are public and accessible by anyone who browses their website. Users can engage in a conversation by replying to a message posted by another user or by mentioning a user; both methods use the convention of including “@username” in the reply tweet or mentioning tweet.

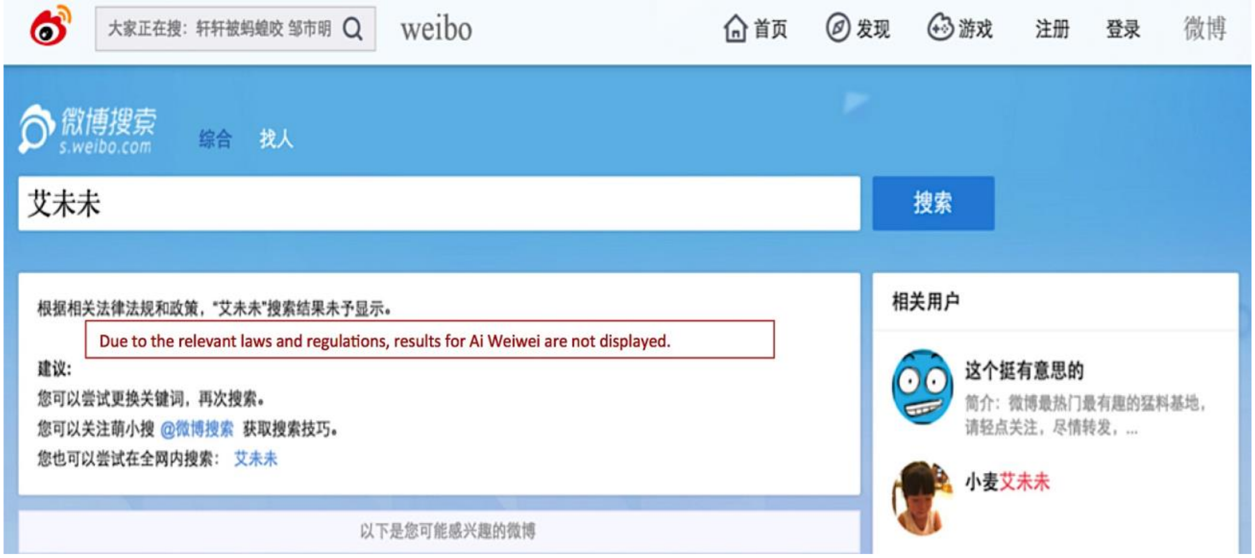
The key difference of Weibo and Twitter are that Weibo is used almost exclusively by Chinese language speakers while Twitter is used globally in different languages. Additionally, hashtags were employed by a double-hashtag "#Hashtag#" method, since the lack of spacing between Chinese characters necessitates a closing tag.

Sina Weibo[1] is the most visited microblog site in China. Because of the site's popularity, "weibo" is often used generically to refer to Sina Weibo. Since 2009, social network sites have taken off in China, and in 2012, more than 300 million Chinese now use microblog to communicate. Sina Weibo, in particular, is growing faster than other social platform over the years. Despite that it was founded three years after Twitter, it has grown to 324 million users in China compared to 564 million users worldwide for Twitter, and was reported as the fastest growing top-tier social network worldwide.

Prior technical reports reveal that the mechanism behind the censorship is likely to be using a porous network of Internet routers used to filter the worst of blacklisted keywords, but

[1] http://weibo.com/

the censorship regime relies more heavily on domestic companies, such as Sina Weibo, to police their own content under governmental regulations with penalty of fines, shutdown and criminal liability [16]. The CEO of Sina Weibo, reports that the company employs at least 100 censors, though the figure is considered to be a low estimate [16]. Figure 1 show the result for search a sensitive term Ai Weiwei[2] on Sina Weibo, which returns a message “Due to the relevant laws and regulations, results for [search team] are not displayed” without any relevant microblogs.



大家正在搜：轩轩被蚂蝗咬 邹市明

艾未未

根据相关法律法规和政策，“艾未未”搜索结果未予显示。

Due to the relevant laws and regulations, results for Ai Weiwei are not displayed.

建议:
您可以尝试更换关键词，再次搜索。
您可以关注萌小搜 @微博搜索 获取搜索技巧。
您也可以尝试在全网内搜索： 艾未未

以下是您可能感兴趣的微博

这个挺有意思的
简介：微博最热门最有趣的猛料基地，请轻点关注，尽情转发，...

小麦艾未未

Figure 1: Search results for censored microblogs on Sina Weibo[3]

[2] A Chinese contemporary artist and political activist: https://en.wikipedia.org/wiki/Ai_Weiwei
[3] http://s.weibo.com/

CHAPTER 2 - RELATED WORK

A large volume of research has been conducted on social media in general, including social network topology and properties [22], online user classifications [9], relations to real-life phenomena [23, 24], etc. The study of microblogs has been a particular research focus, involving predicting online user re-tweeting behaviors [9], association of the use of microblogs to real-world political and financial events [2], etc. Among them, one important aspect of microblogs - content censorship - has yet received little direct attention in previous studies due to the rising Internet security and media censorship. Studies of online deletion behavior began to be drawn research attention.

Social Media Content Deletion

Content posted on social media sometimes disappears from users' timelines. To investigate this issue, many researchers in the past have focused on developing systems to detect and uncover the deleted content for the public [20, 15, 21]. To get access to and restore previously deleted information, King deployed a system to collect online posts before they got deleted from multiple sources of social media platforms in China and showed the kinds of content that censors primarily focus on [20]. Fu applied a discriminatory keyword analysis approach and collected deleted posts from Chinese microblogging platform Sina Weibo following all users with a high number of followers and developed a system to make deleted content publicly accessible [16]. Zhu employed a contrasting strategy to follow a core set of users who have a high rate of post deletions to provide a high-fidelity view of what is being deleted by the censor and when [15].

To understand deleted content behavior, researchers have compared the differences between deleted and undeleted content through empirical quantitative analysis by providing aggregate properties such as sensitive words, deletion speed and frequency [15]. Researchers have also found that even if users have deleted the content, data is not necessarily removed immediately or completely. On Facebook, deleted photos were still accessible after users request a content removal for almost a year later. Twitter deletes photos and posts instantaneously, but it allows third party applications to access users' deleted tweets [25].

Chinese Social Media and Censorship

In terms of reasons behind some content being deleted, prior studies have demonstrated that users of online social networks delete their own posts to manage their social consequences and maintain privacy [15, 25]. Other than concerns of personal social consequence, one of the primary causes for systematic large-scale third-party deletion is the on-going media censorship aiming to maintain national security and social stability. King found that roughly 13% of all blog posts in China were deleted due to Internet censorship [20].

The use of social media has been part of a number of prominent events in China, including the protests of Wukan, the Shifang protest, and anti-Japanese protests [14]. Social media have not only been used for online communication during the events, but have been a way to force the government to address issues directly, such as Beijing rainstorms. The correlation between the use of social media and the recent rise of prominent political events around the world has also been studied and validated [26, 27]. Among the prominent events, researchers have found the presence of some sensitive terms indicating a higher probability of the deletion of a post, and that the posts that contain political sensitive content within a hot online topic are more likely to be censored. By statistically comparing the difference between appearance of certain political terms on Twitter and on Sina Weibo, Bamman showed that the presence of some sensitive terms indicated a higher probability of the deletion of a post, and that geographic differences can also lead to varied deletion outcomes [14]. Zhu found that most deletions occur within 5-30 minutes, and nearly 90% of the deletions happen within the first 24 hours of the post, and the topics of which related posts were mass removed fastest are those containing sensitive words and are about a hot online topic [15].

Prediction of Microblog Deletion

Apart from recovering deleted social media posts to the public and empirical studies on deleted content, social media can also be used to detect deletion events and predict their future outcomes. In the past, social media have been used for a number of different event-detection and prediction problems [3, 4, 1]. Yet, little direct attention has been received on the study to automatically detect and predict deleted online content, especially the content that has been deleted primarily due to censorship.

Not many research studies we found have worked directly on the automatic detection and prediction of the deleted microblogs. Among these researches, Morrison examined the feasibility of automatically detecting censorship of microblogs based on topological features through network growing model and compared two censorship strategies - a uniform strategy and a strategy based on removing entire repost cascades - to simulate varying levels of message deletion [28]. The research provided insights on the feasibility of automatically classifying censored and uncensored networks, and demonstrated that among the proposed strategies, deletion of repost cascades result in higher classification accuracy. However, the research has ignored the problem of data sampling in an online social network. In reality, it is not feasible to collect the complete communication graph due to the scale of the data, which is expected to negatively impact classifier accuracy. The major drawback is that the feasibility of this method has not been validated on real data and has neglected the variation between different online social network sources.

Another research study we found that has attempted to classify and predict deleted tweets was from Petrovic, who applied supervised learning algorithms to predict deletion on Twitter with a mix of social, author and text features [25]. A Support Vector Machine (SVM) algorithm trained by a mixture of all features achieved relatively high F1 score of 27.0. Within the feature types, user ID outperformed social and lexical features, which suggests that tweet deletion varies to a great extent from user to user. Further investigation on social features has shown that the number of tweets and the re-tweet status have more impact than other social features. The research demonstrated that tweet deletion can be automatically predicted ahead of time, and concluded that most of the deletion was done by users themselves primarily due to the swear words they contain through content analysis. The methodology is similar to the one we adopted, which is to apply supervised learning algorithms to predict future deletion outcomes. However, on a different scenario under active Internet censorship of which is expected to have major impact on the deletion outcomes, the model may not perform well and thus the conclusions may vary. Furthermore, previous researches have also neglected to address the influence of sentiment polarity and various emotions expressed, of which previous studies have demonstrated its predictive power on real-world events.

To the best of our knowledge, this research presents the first predictive modeling results applying machine-learning techniques on microblog deletion in heavy-censored environment. Our approach extracts sentiment features and demonstrates that public sentiment feature is a beneficial addition to text and social features in the construction of classification models.

CHAPTER 3 – METHODOLOGY

In this chapter, we show the extent to which microblog deletion can be automatically detected, and future deletion outcomes can be predicted by applying predictive modeling algorithms. We assign deletion labels to each of the microblogs indicating whether a microblog is deleted or not. We apply binary classifiers to train and evaluate predictive models and predict future output labels that determine the occurrences of a deletion event. We extract both content-based features and relevant social features as input to predictive models. In addition, we apply sentiment analysis techniques and incorporate sentiment features as an addition to social and text features to train the classification models.

This chapter starts from an overview of the field of machine learning and framework of our methodology and data collected. Then, we move on to explain the methods adopted to extract input features, feature selection method, classification algorithms and model evaluation method in detail.

Machine Learning

Machine learning is "a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence" [29]. Arthur Samuel in 1959 defines machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed" [30]. Tom Mitchell later in 1997 provides the commonly cited definition and states machine learning as a field "concerned with the question of how to construct computer programs that automatically improve with experience." The common task is to construct a computer program that "is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E* " [31]. In our case, the task *T* is to detect and predict Internet censorship on microblog as a document binary classification problem. The experience *E* that computer program learns from is the training datasets with censorship labels, which account for 80% of total data we collected. To evaluate the performance of our classification models trained by different classification algorithms and combination of feature sets, we use Area under Curve (AUC) score as performance measure *P*. Thus, we formulate a machine learning problem and can apply machine learning techniques on this problem.

System Framework Overview

To automatically detect and predict outcomes of future microblog deletion, we started by collecting raw data samples that are the ones expected to be representative to the whole Chinese social media data. Then, we pre-process the raw datasets by applying Natural Language Processing (NLP) techniques on the content of the microblog: word segmentation, stop word removal (words that do not have meaning in terms of the context), URL removal. Next, we extract the features that will be used in training the model and apply feature selection techniques to select the most contributing features to reduce dimensions of training datasets. After splitting the whole samples into training set and testing set, we build supervised-learning classification models on different sets of features trained on training set and evaluated on testing set to choose the best-performing model. We can predict outcomes of the new data sets by using the best classifier. An overview of our system framework is showed in Figure 2. After pre-processing the raw datasets, a set of features is extracted by feature selection method. The samples are spilt into training (80%) and testing sets (20%) and are used to evaluate each classification algorithm via

AUC score, and to choose the best model to be applied on unseen data to predict labels.

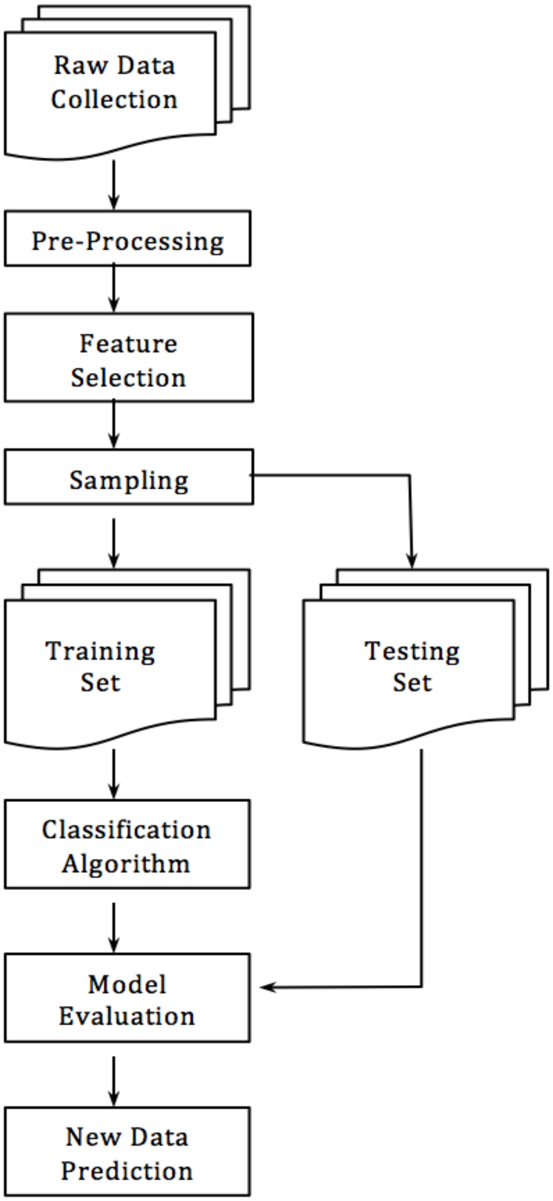


Figure 2: System Framework

Data Collection

The decision to use Sina Weibo data was made due to the public nature of the service and the amount of data available from public data stream we had access to. Raw data are the microblogs and their associated social characteristics. They are collected from August to October 2012. The data is over 219,835 microblogs primarily in Chinese languages. Each of the microblogs is assigned with a label, of which 1 indicates that a deletion due to censorship has been spotted at any time before the end of December 2012. Otherwise, the microblog is assigned with a label 0 indicating the microblog has not been deleted before the end of December 2012. In total, 9% of the microblogs within these four months have been labeled deletion due to censorship.

In addition to the raw datasets we collect and use to train predictive models, we also extract two additional datasets of which are subsets to our full datasets. One is all censored datasets consisting of the information on14203 microblogs, which is used to provide comparative descriptive statistics between censored and uncensored datasets and to detect frequently censored keywords. Another one is the data extracted by matching the key term "Senkaku Islands Disputes" with microblog content. This datasets are used for calculating the public sensitive scores to observe its relevant effect, reflection and correlation with real-world political uprising in our case.

The raw datasets are collected from Weiboscope, which is a data collection and visualization project developed by the research team at the Journalism and Media Studies Centre at the University of Hong Kong (JMSC)[4]. The datasets were collected using a random sampling approach. Deploying Sina Weibo Open API, a random sample of Sina Weibo user accounts was constructed; then, the sample's user information and the most recent posts were fetched and stored into datasets. This sampling approach is reported in a PLOS ONE article [32]. We use the

[4] Open Weibosocpe Data Access: http://weiboscope.jmsc.hku.hk/datazip/

datasets because one of the objectives for Weiboscope project is to make censored Sina Weibo posts of a representative group of Chinese microbloggers publicly accessible, and thus expected to contain all important features for investigating on the issue of content deletion and censorship. The project also enables academic use of the data for better understanding of the social media in China and making the Chinese media system more transparent.

Features

In this section, we describe the microblog and user features and the method to extract additional features to train predictive models. We categorize all features into three groups: text, social and sentiment.

A. **Text Features**

Content of a microblog is important especially in the prediction of content deletion task because previous researches have demonstrated that posts with certain sensitive keywords have higher rate of being deleted [15]. Lexical content thus provide a large amount of useful information for model training and content deletion classification. Lexicon tokens are extracted from the content of microblog texual messages and are used as predictors.

We extracted lexicon tokens and developed the corpus by first segmenting the words on the content using Chinese lexicon analysis system ICTCLAS, which was developed by Institute of Computing Technology of Chinese Academy of Sciences based on multi-layer Hidden Markov Model[5]. Then, we use regular expression to remove non-Chinese lexicons, remove non-expressive characters, including URLs, punctuation marks, emoticons, and weibo special characters, such as hashtags and direct reply. Furthermore, we remove lexicons from microblogs

[5] Institute of Computing Technology, Chinese Lexical Analysis System: http://ictclas.org/

that match with a list of stop words we developed, that is, high-frequency words usually having little lexical content, and their presence in a text fails to distinguish it from other texts. Filtering out these stopwords out of a document before further processing increased processing efficiency by reducing dimension and resulted in a more accurate classification. Finally, we performed frequency analysis on single tokens to identify popular words and remove the rare or short tokens, that it, less than 2 appearances in the whole corpus.

B. **Social Features**

Previous empirical studies have revealed the differences on some of the social features, such as geographical location, date that the microblog has been posted etc., between deleted microblogs and the non-deleted ones. We assume some of the social information associated a microblog can be important in content deletion classification and incorporate social features into our model development. We categorize social features into two groups: features associated with message content, and features associated with users who posted the microblog.

User features are gender, province and whether the user is a verified user or not. For each of the microblog, some of the features are directly associated with each of the microblogs, such as re-tweet status, message source, image presence, geographical location presence.

C. Sentiment Features

Sentiment has been shown to have predictive power on real-life events such as predicting stock movement [2]. With respect to content deletion, previous studies have shown that there is a difference in public sentiment between censored and uncensored micrblogs [16]. We assume sentiment features can be a beneficial addition to our feature sets in this task, and we adopt lexicon-based sentiment extraction approach to calculate sentiment scores for each of the microblogs.

To construct Chinese polarity opinion lexicon dictionaries, we firstly need sentiment dictionaries that we can compare our datasets to. We install and combine HowNet-Vocabulary for Sentiment Analysis[6] and National Taiwan University Sentiment Dictionary (NTUSD)[7] to serve as our polarity dictionary. After word segmentation via Chinese lexicon analysis system ICTCLAS, we use the following approach to calculate polarity scores:

$$\text{Positive Scores} = \frac{\text{\# of positive words matches in a given message}}{\text{\# of polarity words matches in a given message}}$$

$$\text{Negative Scores} = \frac{\text{\# of negative words matches in a given message}}{\text{\# of polarity words matches in a given message}}$$

Positive and negative scores that have been calculated in this way can be viewed as the ratio of matched positive and negative words out of all the words in each microblog. The sentiment category of each microblog can be determined by comparing the calculated positive score to the negative one. If the positive score is greater than the negative, then that microblog is categorized as positive sentiment; if the positive is smaller, then it is negative; if there is no difference, then it is categorized as neutral and is excluded from further analysis.

In summary, three groups of features are extracted and used to train the predictive models. Table 3.1 details all the social features we extracted. Features are categorized into three groups: Content Text Features, Social Features and Sentiment Features with their subgroups. In Figure 3, we visualize our feature vector space and censorship labels to illustrate our feature representation approach and to facilitate a better understanding on how we formulate this research task as a

[6] Hownet: http://www.keenage.com/

[7] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, “Opinion extraction, summarization and tracking in news and blog corpora,” in Proceedings of the 2006 AAAI Spring Symposium Series on Computational Approaches to Analyzing Weblogs, 2006.

machine learning binary classification problem.

Feature Group	Subgroup	Feature Details
Content	Text	
Social	Microblog	Microblog ID, Retweet Status, Image Presence, Geographical Location Presence, Deletion Indicator, Censorship Indicator, Microblog Source, Microblog Created Time
	User	User ID, Verification Status, Gender, Province
Sentiment	Polarity Score	Positive Score
		Negative Score

Table 1: Feature Summary for Microblog Deletion Classification

Feature Selection

Overall, final feature set is fairly large due to the features such as textual features from microblog text, which is a type of data where word attributes are sparse and high dimensional with low frequencies on most of the words. In total, we extracted 67,415 textual features, which consist of 11 billion highly sparse entries versus 1 million non-sparse.

To reduce the size of feature set used for data representation and optimize the use of computing resources, and to remove the noise from the data in order to optimize the classification performance, we apply feature selection techniques in our classification setup. In the text pre-processing, we applied a typical feature selection technique by removing stop words to reduce the feature space, memory consumption, and processing time. We further apply more feature selection method by removing the features with less contributing value. We achieve this by first scoring the features in accordance with a weighting scheme designed to rank the importance of the feature for a given classification task and reduce document term matrix sparsity by 2%.

We further adopt linear regression analysis as part of our feature selection strategy by calculating the coefficient p-values to determine which terms to keep in the logistic regression model. P-value for each term is used to test the null hypothesis that the coefficient is equal to zero, which indicates that removing the term does not change model's performance. A low p-value (< 0.05) indicates that rejecting the null hypothesis is feasible; thus, a relatively low p-value indicates that the associated term is likely to be a meaningful addition to model construction.

Classification

We formulate our task of detecting and predicting Internet censorship on microblog as a document binary classification problem. To predict deletion labels of a microblog given the relevant information of a microblog x, our goal is to learn a function *f* that maps a microblog x to a binary value $y \in \{1, 0\}$, where y indicates if x is deleted or not.

The classifiers we are going to build needs features to use for classifying documents. A feature is the items that can determine the classification of the document, which in our case is as being either censored or not censored. The features we use fall into three categories: words, social, and our extracted aggregate feature sentiment. Our task has challenges as the main variables are of large categorical domain, which is sparse and high dimensional. Thus, it is critical to design classifiers to overcome the challenges.

The classifiers learn how to classify a document by being trained with the training datasets. In general, the more examples of documents and correct classifications the classifiers see, the better the classifier will become at making future predictions. We adopt three classification algorithms, which have been popular to use in the past on similar tasks, to build classifiers for our task: Naïve Bayes, L1 and L2 regularized logistic regression. We will present in details about the algorithms we apply and how we use them to build classifiers to classify microblogs in our task.

A. Naïve Bayes Classifier

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with independence assumptions between the features. This classification method is called naïve because it assumes that the probabilities being combined are independent of each other; thus, the probability of one feature in the document being in a specific category is unrelated to the probability of the other words being in that category. A naive Bayes classifier considers each of these features to contribute independently to the probability of a category regardless of any possible correlations between other features. In many classification tasks in machine learning, naïve Bayes has proven to be an effective method.

Naïve Bayes classifier is based on applying Bayes theorem represented by a conditional probability model: given a problem instance to be classified represented by a vector $X = (x_1, \dots, x_n)$ representing n features, it assigns to this instance probabilities $p(C_k|x_1, \dots, x_n)$ for each of k possible outcomes or classes [33].

Bayes' Theorem is usually written as: $p(A|B) = \frac{p(B|A)\, p(A)}{p(B)}$, which is a way to calculate posterior probabilities of an instance based on prior probabilities: $posterior = \frac{prior * likelihood}{evidence}$. Using Bayes' theorem, the conditional probability can be decomposed as $p(C_k|X) = \frac{p(X|C_k)\, p(C_k)}{p(X)}$. In our case, we are given a microblog to be classified represented by a vector $X = (x_1, \dots, x_n)$

with n features and to be assigned to a binary classification where $y \in \{1, 0\}$, where y indicates if a microblog is deleted or not. Our application on Bayes' Theorem becomes: $p(y|X) = \frac{p(X|y)\, p(y)}{p(X)}$. In plain English, the equation becomes: $(Censorship\ Label|Microblog) = \frac{p(Microblog\ |\ Censorship\ Label)\ p(Censorship\ Label)}{p(Microblog)}$.

We use package *e1071*[8] to construct naïve Bayes classifier by computing the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.

B. Regularized Logistic Regression

For large sparse data with a huge number of instances and features, linear classification has become one of the most promising learning techniques. LIBLINEAR is an open source library for large-scale linear classification [34]. LIBLINEAR is very efficient on large sparse data sets. We use LIBLINEAR to develop classification models based on L1 and L2-regularized logistic regression (LR).

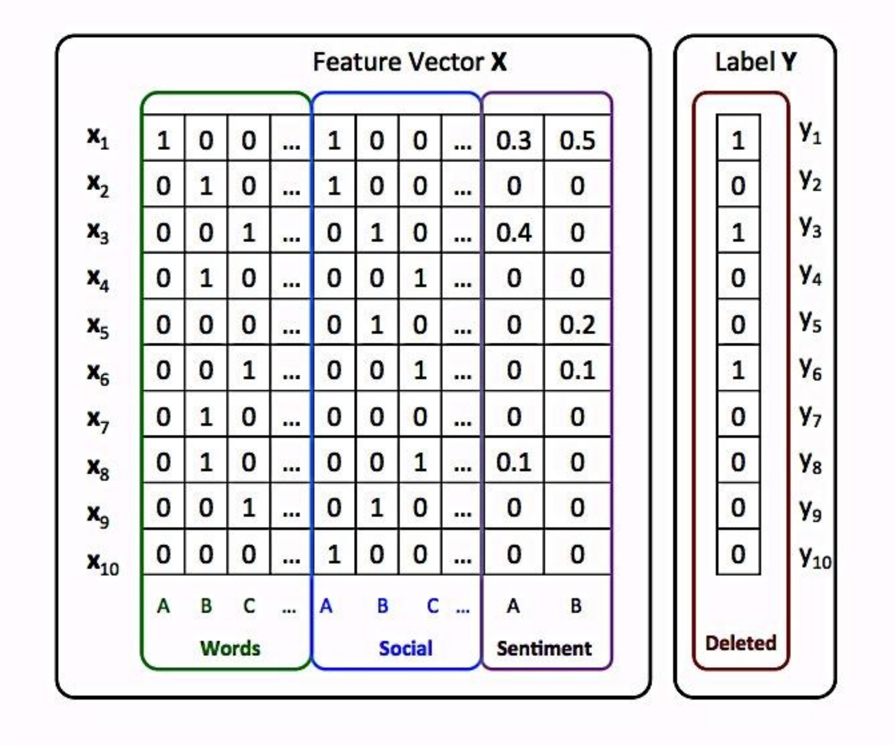
Given pairs of patterns and labels $(x_1, y_1) \dots (x_m, y_m)$ which constitute the set of training observations, both logistic regressions (LR) and linear SVM solve the following unconstrained optimization problem with different loss function $\xi\ (w; x_i, y_i)$:

$$\min_{w} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l} \xi(w; x_i, y_i)$$

where $C > 0$ is a penalty parameter. For LR, the loss function is $log\ (1 + e^{-y_i w^T x_i})$, which is derived from a probabilistic model.

[8] http://www.inside-r.org/packages/cran/e1071/docs/naiveBayes

We apply L1 and L2-regularized logistic regression (LR) and develop models utilizing different groups of features we extracted. Figure 3 is a visual illustration for the classification model setup with three groups of features we extracted. Features are denoted as x_t as the text of the microblog x and $x_u \in U$ as the user who posted x. A set of n labeled examples $\{<x_i, y_i> : i = 1...n\}$, where the label Y indicates whether the tweet has been censored or not.



Feature Vector **X**

Label **Y**

	Words				Social				Sentiment		Deleted	
	A	B	C	...	A	B	C	...	A	B		
x_1	1	0	0	...	1	0	0	...	0.3	0.5	1	y_1
x_2	0	1	0	...	1	0	0	...	0	0	0	y_2
x_3	0	0	1	...	0	1	0	...	0.4	0	1	y_3
x_4	0	1	0	...	0	0	1	...	0	0	0	y_4
x_5	0	0	0	...	0	1	0	...	0	0.2	0	y_5
x_6	0	0	1	...	0	0	1	...	0	0.1	1	y_6
x_7	0	1	0	...	0	0	0	...	0	0	0	y_7
x_8	0	1	0	...	0	0	1	...	0.1	0	0	y_8
x_9	0	0	1	...	0	1	0	...	0	0	0	y_9
x_{10}	0	0	0	...	1	0	0	...	0	0	0	y_{10}

Figure 3: Feature Representation.

Performance Measure

A confusion matrix as illustrated in Table 2 typically evaluates model performance. The columns are the predicted class and the rows are the actual class. In this confusion matrix, TN (True Negatives) is the number of negative examples correctly classified; FP (False Positive) is the number of negative examples incorrectly classified as positive. FN (False Negative) is the number of positive examples incorrectly classified as negative and TP (True Positive) is the number of positive examples correctly classified. Predictive accuracy is defined as $accuracy = \frac{TP+TN}{\mathrm{TP + FP + TN + FN}}$

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

Table 2: Confusion Matrix

Predictive accuracy might not be appropriate when the data is highly imbalanced. Take our tweet censorship problem as an example, uncensored data account for 91 % of total messages while censored data account only for 9 %. A simple guessing on the majority class would give a predictive accuracy of 91 %. However, in this problem, we are more interested in achieving a high rate of correct detection of minority class and allows for a small error rate in majority class. Thus, using simple predictive accuracy is not appropriate in our situation.

We choose to use Area Under the Receiver Operating Characteristic (ROC) Curve as a metric to measure classifier performance. ROC curve is a standard technique for summarizing classifier performance over a range of tradeoffs between true positive and false positive error rates where $\% FP = \frac{FP}{TN + FP}$ and $\% TP = \frac{TP}{TP + FN}$ .The ideal point on the ROC curve would be all positive examples are classified correctly and no negative examples are misclassified as positive. A single operating point of a classifier can be chosen from the trade-off between the %TP and %FP, thus one can choose the classifier giving the best %TP for an acceptable %FP.

CHAPTER 4 - RESEARCH RESULTS

Censored Microblog Example

Table 3 presents some of the examples of the microblogs that have been deleted due to censorship. These microblogs discuss about various topics, such as democracy, patriotism, state-owned enterprise, violence, pollution, etc. The table shows the information not only associated with message itself, such as the message content, retweet status, retweeted message content, user who post the message and user whose message was retweeted, timestamp that the message has been created at and been deleted at, etc., but also the self-reported information that user provided, such as the province, gender and verified status. Original microblogs are written in Chinese, we translated each of the messages into English for better understanding. For identifiable information such as user ID or hyperlink, we replaced them by ***. The information about province comes as numeric code, such as 44, we translate the code to the name of the province as shown in the bracket.

ID	Original Text	Translation	Created_At	Deleted_At	Province	Gender	Verified	Image	Source
1	免去的那点儿，比不上化肥、农药、种子的涨价幅度！	The amount being waived is nothing compared to the amount of increase of fertilizer, pesticide, and seeds!	2012-08-04 23:01:04	2012-08-05 21:11:24	53 (Yunnan)	Male	False	0	Sina Weibo
2	说得好。港人此次的行动有力表明了，谁是真正的爱国者，怎样才是真爱国。	Well done. People from Hong Kong strongly demonstrate who real patriots are and how to be real patriotic.	2012-08-17 12:32:05	2012-08-23 22:45:28	44 (Guangdong)	Male	True	0	Sina Weibo
3	参加砸车的和同情者，请看！不要再干”亲者痛，仇者恨“的傻事了！爱国不是这样爱的	Those who participate in car-smashing activities and sympathizers, watch out! Don’t do such stupid thing to make loved ones pain and enemies please! Being patriotic is not like that!	2012-08-20 11:23:21	2012-08-22 08:56:13	100 (Unknown)	Male	True	0	IPad Platform
4	转//@***: 污水一直排放长江，新建的排海工程被废止，只好继续排长	RT//@***: Waste water keeps being discharged into Yangtze River; due to the termination of Wastewater Release to Sea Project, it	2012-08-29 04:05:08	2012-08-30 10:04:13	100 (Unknown)	Male	False	1	Sina Weibo

	江。有问题么？	continues to be released to Yangtze river.							
5	蔑视舆论，是某些领导的基本原则。//@***：俺不相信党的高级干部这么说爱国群众[怒]	A distain for public opinion is some leaders' rule of thumb. //@***: I can't believe that those party's senior officials say such things about the patriots [angry].	2012-09-13 01:23:12	2012-09-13 10:04:46	11 (Beijing)	Male	True	0	Sina Weibo
6	中国人连上街游行的机会都是日本人给的，何其悲哀！！！	Even the chances for the Chinese to parade in streets are given by the Japanese, how sad!!!	2012-09-17 07:21:56	2012-09-21 10:40:27	31 (Shanghai)	Male	False	0	Sina Weibo
7	//@***：要真相，要法律！我们必须要声援前去了解真相的记者和律师！	//@***: We want the truth and law! We must support these correspondents and lawyers who seek the truth!	2012-09-25 02:11:23	2012-09-25 14:24:01	31 (Hangshai)	Male	True	1	Android Platform
8	网络是和谐社会建设的过程中监督的利器不是暴民的春秋，谨言慎行才是为官之道。//@***：有请摔杯子的李副省长主动公开全部财产，向"网络暴民"证明自己正大光明。	Online network is a useful monitoring tool in the process of building a harmonious society, not the world of mobs, be cautious of what you say is what a good official should do. //@***: Deputy governor Li, who throw a cup at the 'network mobs,' please reveal to us all you properties and prove your honesty and incorruption.	2012-09-25 20:01:03	2012-09-26 07:54:22	32 (Jiangsu)	Male	True	1	Sina Weibo Smartphone
9	这类所谓＂爱国者＂应该下地狱，给予重判。//@***：「爱国者」打残无辜同胞视频。悲哀！	This type of so-called 'patriot' should go to hell and be severely punished. //@***: the video about 'patriots' crippled innocent fellow citizens. Sorrow!	2012-09-26 17:24:25	2012-09-28 14:38:28	44 (Guangdong)	Female	True	1	IPad Platform
10	转//@***：中国已经进入了国企恐怖主义阶段。巨大的国企开始碾碎和吞噬一切它们认为应该碾碎和吞噬的。	RT//@***: China has entered an era of state-owned enterprise terrorism. Gigantic state-owned enterprises start to crush and swallow those which they believe need to be crushed and swollen.	2012-10-19 10:21:13	2012-10-22 23:33:51	11 (Beijing)	Male	True	0	Sina Weibo

Table 3: Sample Data of Original Censored Microblogs

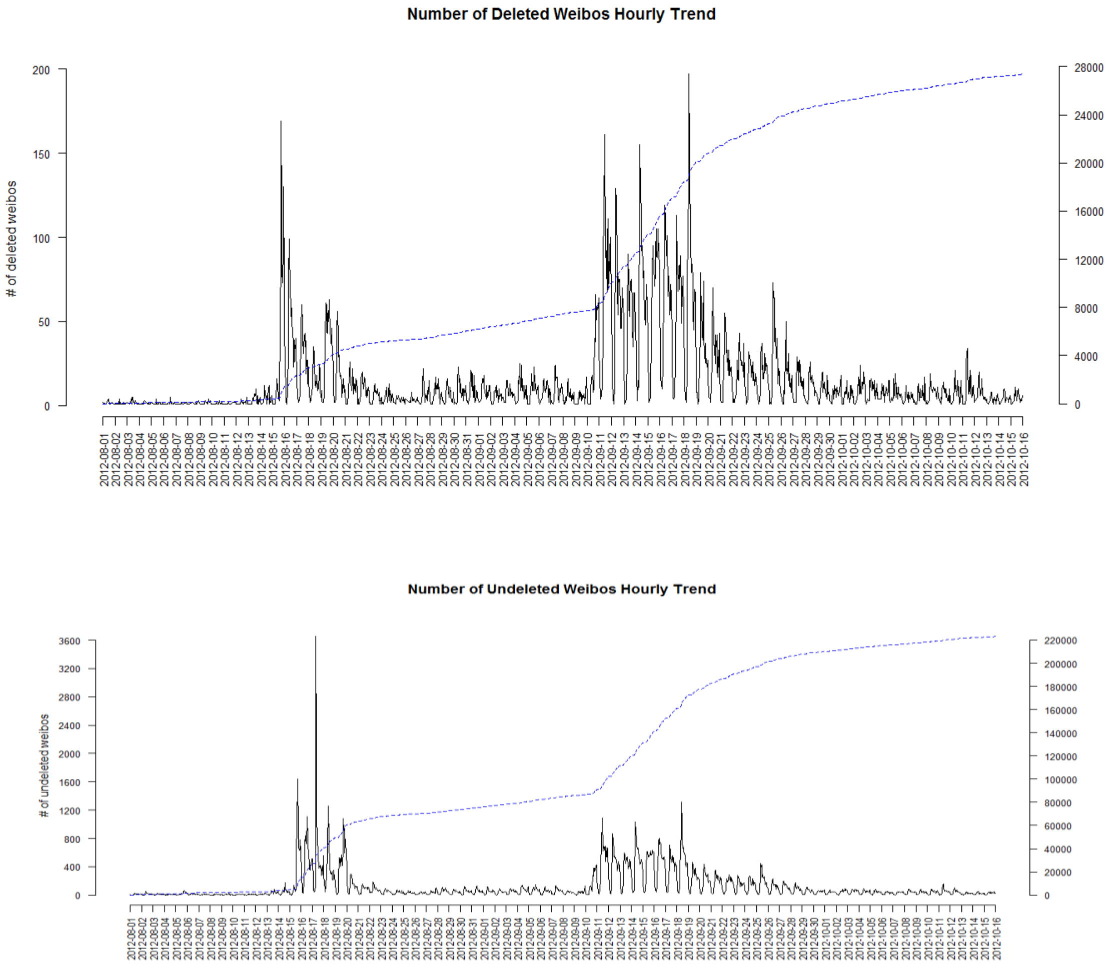
Empirical Analysis

In this research, we included all social and text features of representative microblogs and users collected from Sina Weibo microblogging platform by *Weiboscope* project a total of 219,835 microblogs with from August 1 to October 15, 2012.

Aggregate Statistic	Uncensored	Censored
Microblogs	**205,632 (91%)**	**14,203 (9%)**
Users	**149,277 (97%)**	**4,073 (3%)**
Retweeted Microblogs	104,872 (51%)	**11,984 (84%)**
Retweeted Users	70,160 (47%)	**10685 (75%)**
Message Sources	1061	186
Images	61689 (30%)	1,431 (10%)
Geographic Information	2072 (1%)	17 (0.1%)
Gender – Male	167,997 (81%)	12,301 (86%)
Verified Users	92,534 (45%)	**12,413 (87%)**

Table 4: Comparison of Uncensored and Censored Datasets

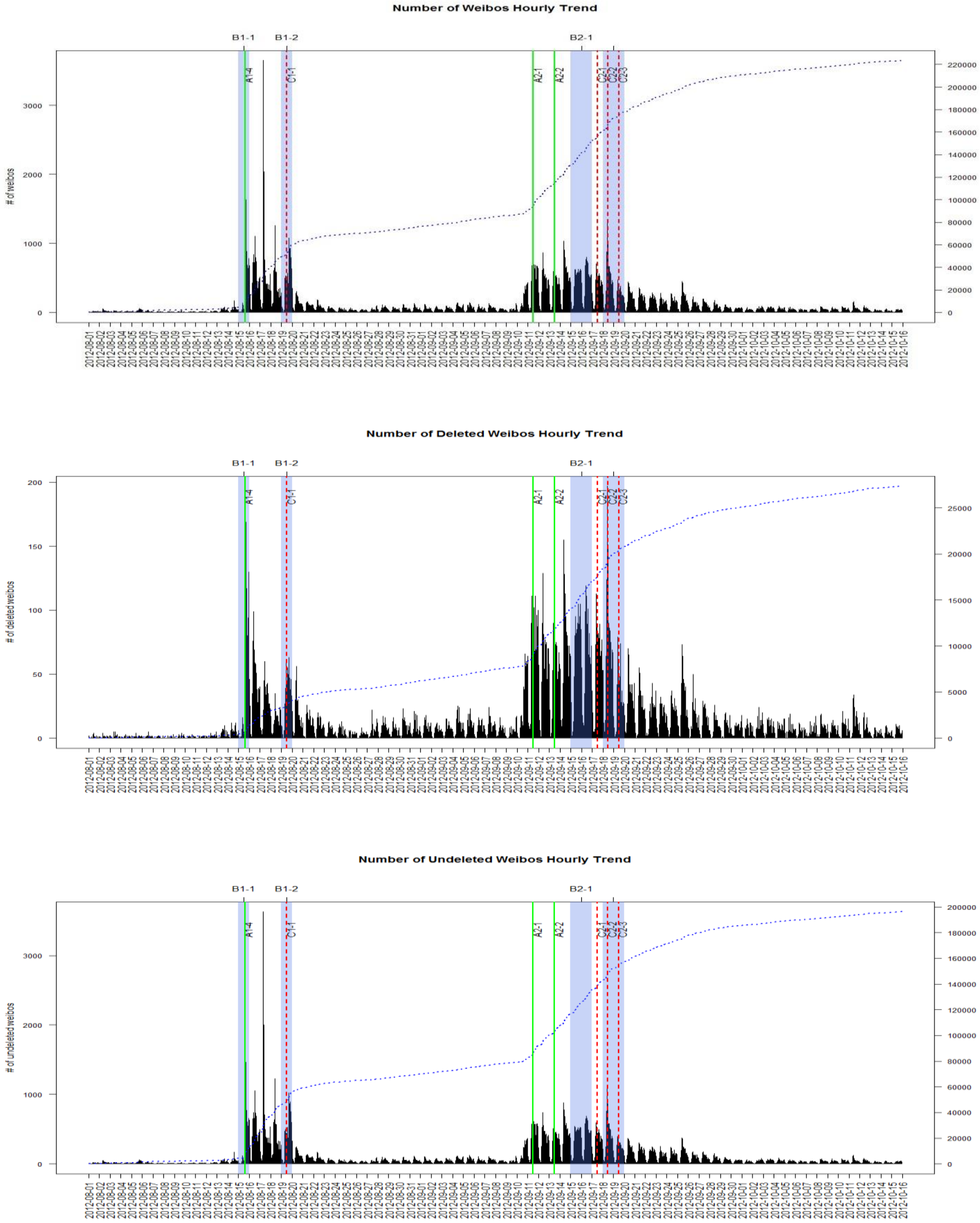
Table 4 presents descriptive statistics to compare the information of uncensored and censored data. While the ratio of uncensored to censored data size is close to 9:1, the proportion of retweeted microblogs (84%) is much higher than of uncensored microblogs (51%). The pattern also applies to retweeted users, which the proportion of unique users who retweeted the censored microblogs (75%) is much higher than of uncensored microblogs (47%). In the censored data pool, users whose identities have been verified in the real world account for a much higher proportion (86%) compared to the proportion of uncensored data (45%). It is also observed that digital information, image and geographical location, appears much less frequently in censored than in uncensored datasets. In addition, gender does not differentiate that much for censored (86%) and uncensored (81%) datasets.



Number of Deleted Weibos Hourly Trend

Number of Undeleted Weibos Hourly Trend

Figure 4: Censored/Uncensored Microblog Hourly Trend



Number of Weibos Hourly Trend

Number of Deleted Weibos Hourly Trend

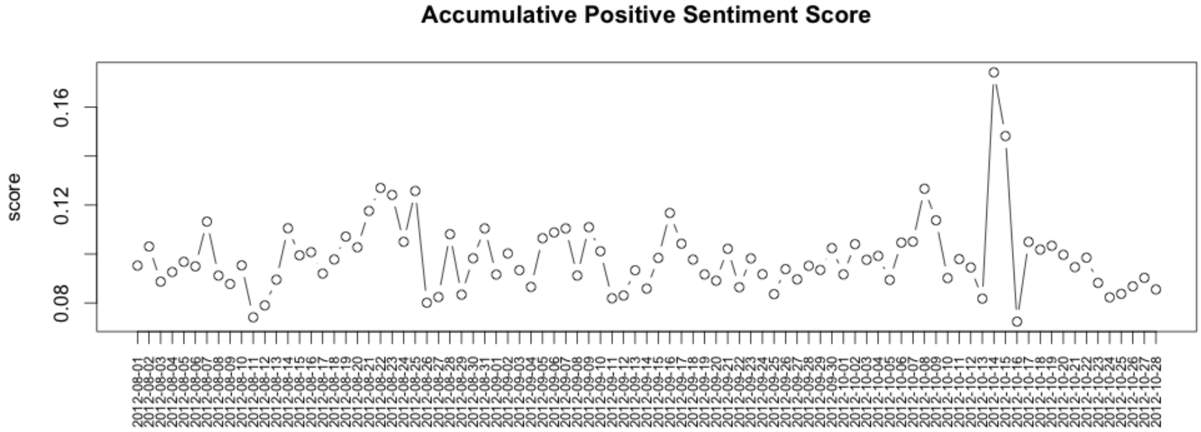
Number of Undeleted Weibos Hourly Trend

Figure 5: Volume Trend for Censored/Uncensored Microblogs

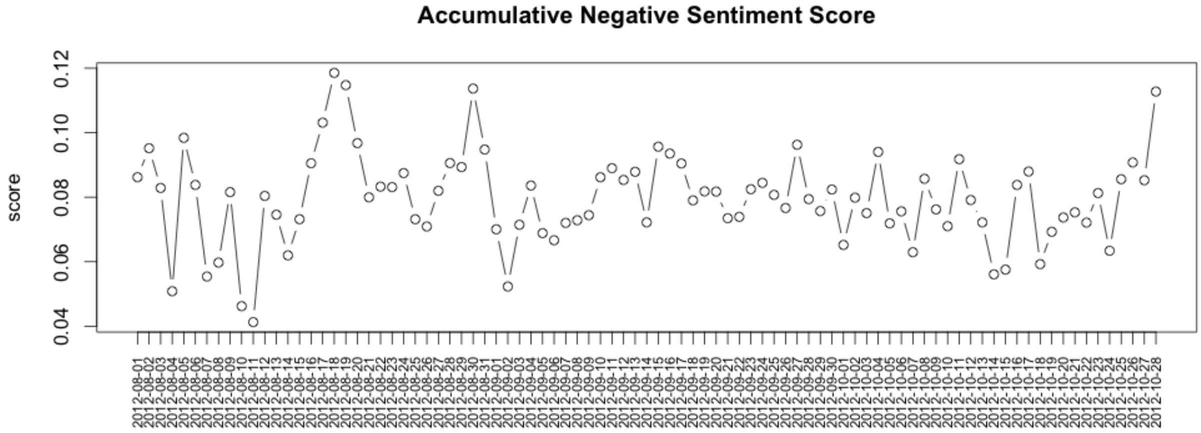
From Figure 4, we observe that there exist two peaks in the hourly trend of total number of censored vs. uncensored microblogs. When investigating on the causes to these sudden increases of microblog volume, we discovered that the peak times are in align with two major waves of a social protest, 2012 Anti-Japanese Protest, which was taking place in over 180 cities throughout the country. Table 5 presents a timeline of major events in the Chinese social protest movement in 2012.

Protest	Category	Code	Event Date	Event Description
First Wave	Trigger Incidents	A1-1	2012-04-16	Tokyo's prefectural governor Shintaro Ishihara publicly announced his decision to let Tokyo Municipality purchase the island from its private owner.
		A1-2	2012-07-04	Three Japanese coast guard boats made an official inspection of one Taiwanese ship near the disputed island
		A1-3	2012-07-07	Japanese prime minister Yoshihiko Noda expresses his consideration for the Japanese government to buy the disputed islands.
				Ministry spokesman Liu Weimin retorted "No one will ever be permitted to buy and sell China's sacred territory."
		A1-4	2012-08-15	The activists from Hong Kong and their ship were detained by Japanese authorities.
	Protests	B1-1	2012-08-15	In Beijing, citizens began protesting in front of the Japanese embassy.
		B1-2	2012-08-19	Protesters gathered in Shenzhen called for the boycott of Japanese goods and for the government to retake the islands lasted.
	Crackdown	C1-1	2012-08-19	Government sent in large numbers of armed police, who called for an end to the violent protests.
Second Wave	Trigger Incidents	A2-1	2012-09-11	China sent two patrol ships to the islands to demonstrate its claim of ownership.
				Japan formally nationalizes the three islands that were held in the ownership of Kunioki Kurihara.
		A2-2	2012-09-13	Chinese government submit nautical chart with baselines of the territorial sea on disputed islands to United Nations.
				Former president of Republic of China Lee Teng-hui remarked "The Senkaku islands were Japanese territory in the past and are still so at present."
	Protests	B2-1	2012-09-15 2012-09-16	Citizens in mainland China participated in protest marches and called for a boycott of Japanese products in 85 Chinese cities.
				Protests were held in 5 US cites as well as a petition to the US government and Congress to take a neutral stance over the dispute.
		B2-2	2012-09-18 2012-09-19	People in over 180 cities of China attend protests on 81st anniversary of Mukden Incident.
	Crackdown	C2-1	2012-09-17	Police in the city of Xi'an banned large protests and forbade the use of phone and online messages to organize illegal protests.
				Paramilitary troops in Shanghai provided round-the-clock protection to the Japanese consulate.
		C2-2	2012-09-18	Police in Qingdao city arrested six people in connection with the demonstrations.
				Guangzhou authorities arrested 18 people who committed anti-Japanese vandalism and warned citizens against being present in large crowds.
		C2-3	2012-09-19	National authorities deployed riot police to suppress existing protests and to prevent their re-occurrence.

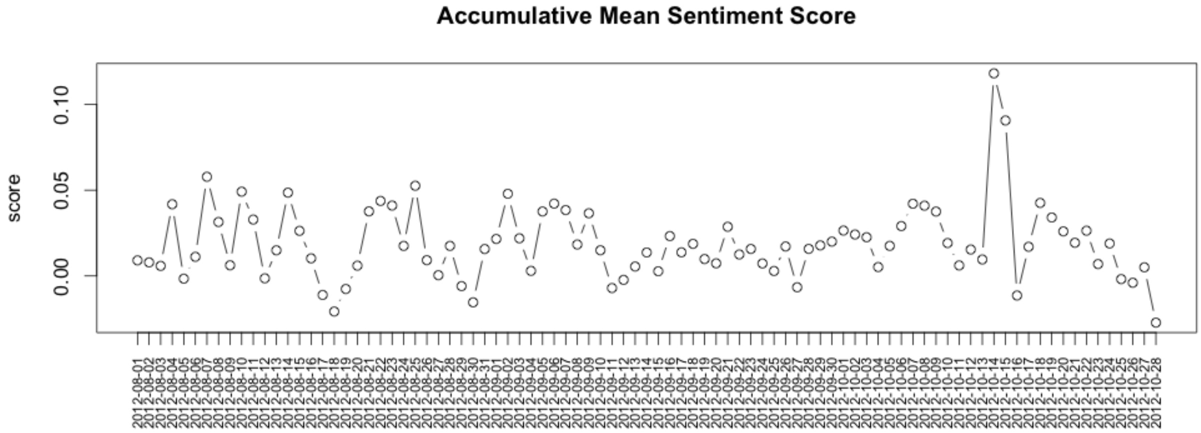
Table 5: Event Timeline of 2012 Anti-Japanese Protest in China



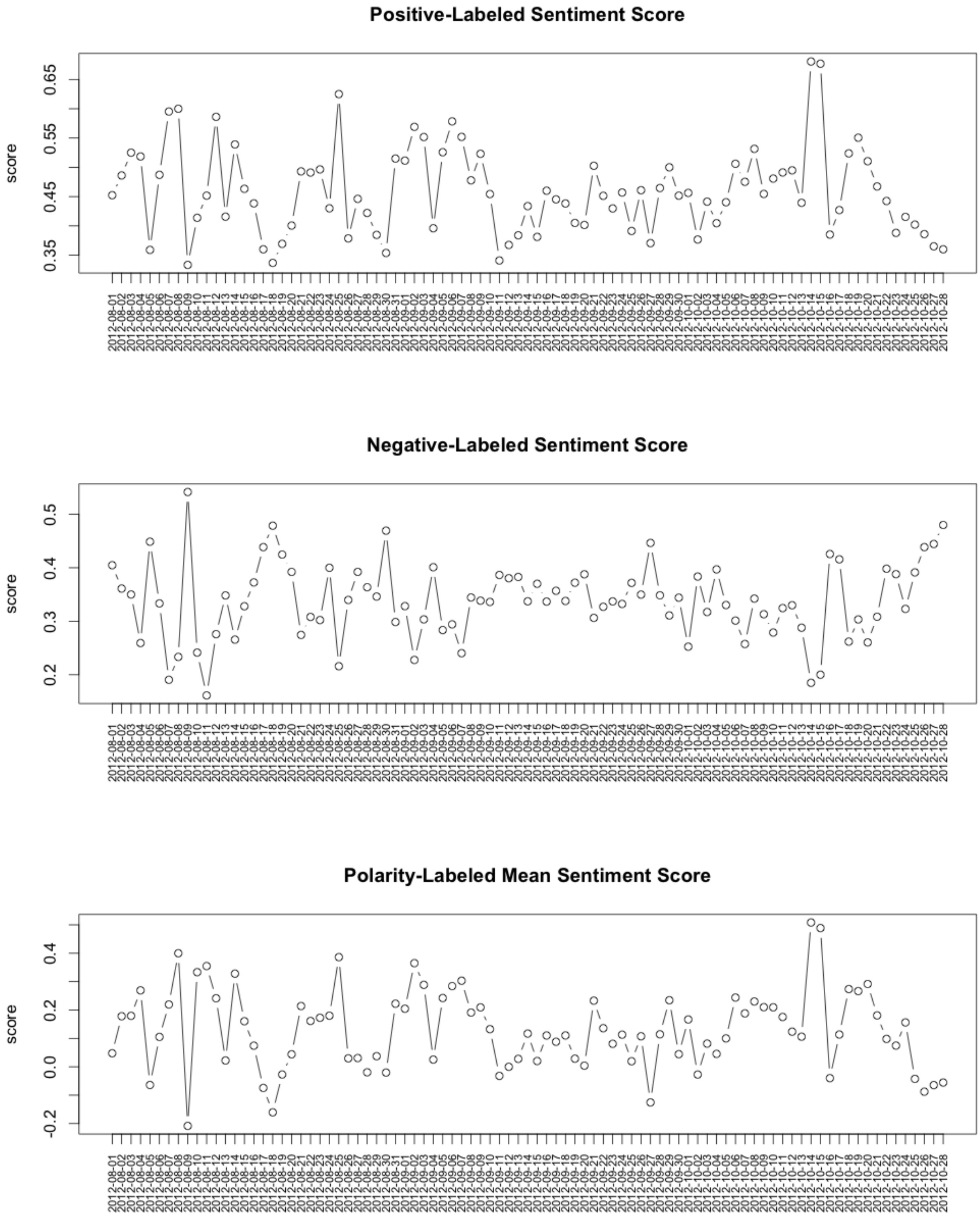
Accumulative Positive Sentiment Score



Accumulative Negative Sentiment Score



Accumulative Mean Sentiment Score



Positive-Labeled Sentiment Score

Negative-Labeled Sentiment Score

Polarity-Labeled Mean Sentiment Score

Figure 6: Positive and Negative Sentiment Trend

From Figure 5 and Table 5, it can be observed that there is a correlation between the major protest occurrence and the increase of total volume of censored microblogs. From above figure, we can tell that Internet censorship was strongly enforced especially in the second wave of the movement during the period of major protests taking place in most of the major cities in China following with a crack down period of the protests.

Figure 6 presents the results of our daily sentiment trend relating to the topic of “Senkaku Islands Disputes” via two daily sentiment score calculation approaches. The first two plots are generated by applying accumulative sentiment score calculation approach to combine the total polarity score on each day and divided by polarity word matches on that day. While positive sentiment score is remarkably high around October 15 after staying steady from August 1, negative sentiment score reaches to the highest point in as early as August 9. The third plot calculates the accumulative polarity mean by taking a difference between the two sets of sentiment scores. The plot clearly shows that the positive sentiment peaks around October 15, and in mid and late August, public sentiment stay relatively negative on the “Senkaku Islands Disputes” topic. The remaining three plots are generated by counting the polarity labels assigned to each of the microblogs each day and divided by the total of microblogs on that day. Sentiment scores may seem to be moving up and down more often than the results using our first approach; however, it appears that they have the similar trend where the positive sentiment peaks in mid-October and negative sentiment in mid-August. It is further confirmed from the mean sentiment score plot, which is the difference of positive and negative volume. Referencing from the event timeline of 2012 Anti-Japanese Protest in Table 5, we may conclude that in general during the first wave, public sentiment is more towards negative while in the second wave, public sentiment changes to be positive.

After exploring the descriptive information about the aggregate features on social features, we further investigate the microblog's textual content. We analyze microblog's text information by applying Natural Language Processing (NLP) techniques to get data pre-processed, including duplicates removal, word segmentation, stop-words removal, hyperlink and punctuation removal, etc. to get a clean corpus of meaningful terms our of the censored microblog content. Then, we construct document term matrix with weightings calculated as term-frequency - inverse document frequency (tf-idf) scores. The goal is to access the importance of the key terms and to detect sensitive terms containing of which would make the microblog more likely to be censored.

After pre-processing to get a clean corpus, we constructed document term matrix that reflects the number of times each word in the corpus is found in each of the documents. Our matrix is of 100% sparsity, which indicates there are too many terms in the matrix that do not occur very often and thus resulting a zero as tf-idf score. By visualizing term frequency distribution (Figure 7), we see that it is a long tail (power) distribution of which some distributions of numbers is the portion of the distribution having a large number of occurrences far from the "head" or central part of the distribution.

Because our document term matrix is very sparse, we reduce its size by choosing only less sparse terms to include. With a 2% reduction on matrix sparsity, we reduce the number of unique terms from 67,415 to 49. From the term frequency distribution plot in Figure 8, we now can get a clearer view on the important terms revealed from the corpus with frequency. To detect the most important key terms, we draw a list of top 20 terms with the highest frequency and present in Table 6 with English translation. Since we detected a correlation between 2012 Anti-Japanese protest against Japanese government purchase of Senkaku Islands and the volume of censored microblogs, it would be interesting to know what other meaningful terms that associated with this event are. We constructed a list of terms that have at least 0.1 correlations with the term “Senkaku Islands” in Table 7. Similarly, we then constructed a list of terms that have at least 0.09 correlations with “China,” which is the term that appears most frequently in the censored microblog content. Some terms may seem unexpected at the first sight, such as “Tofu” which literally refers to a food made by coagulating soy milk. However, under the context of social and political situation in 2012, it may refer to the poorly constructed buildings that are soft like tofu dreg.

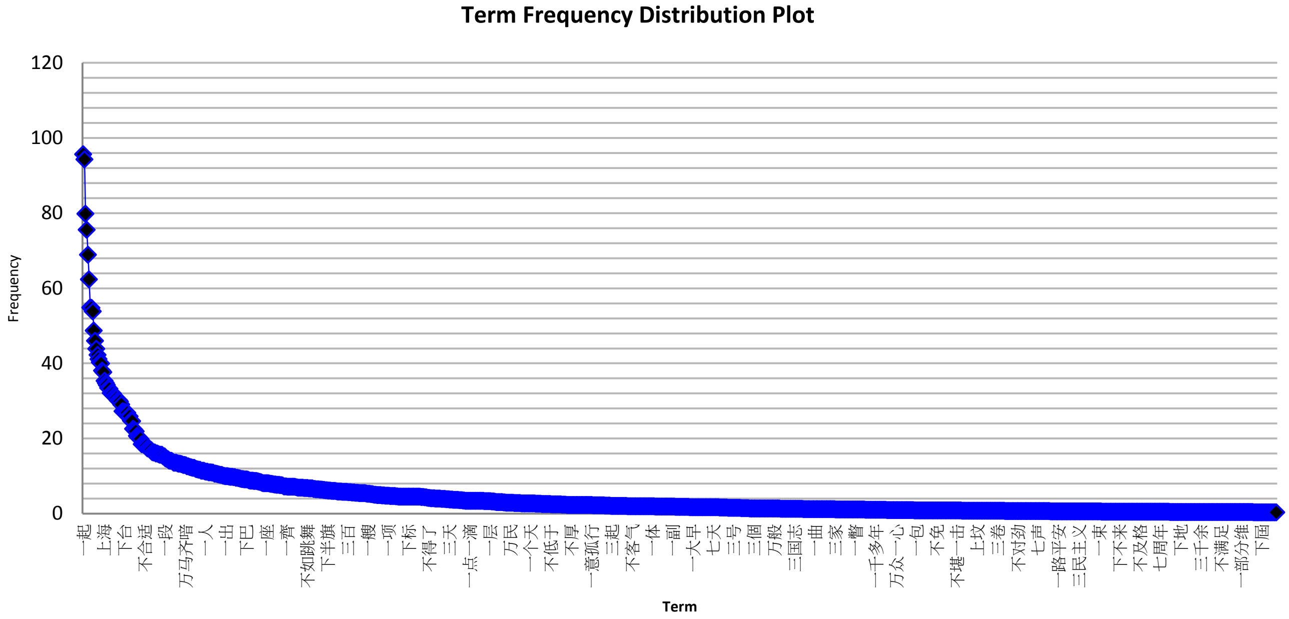


Figure 7: Term Frequency Distribution Plot with 100% Matrix Sparsity

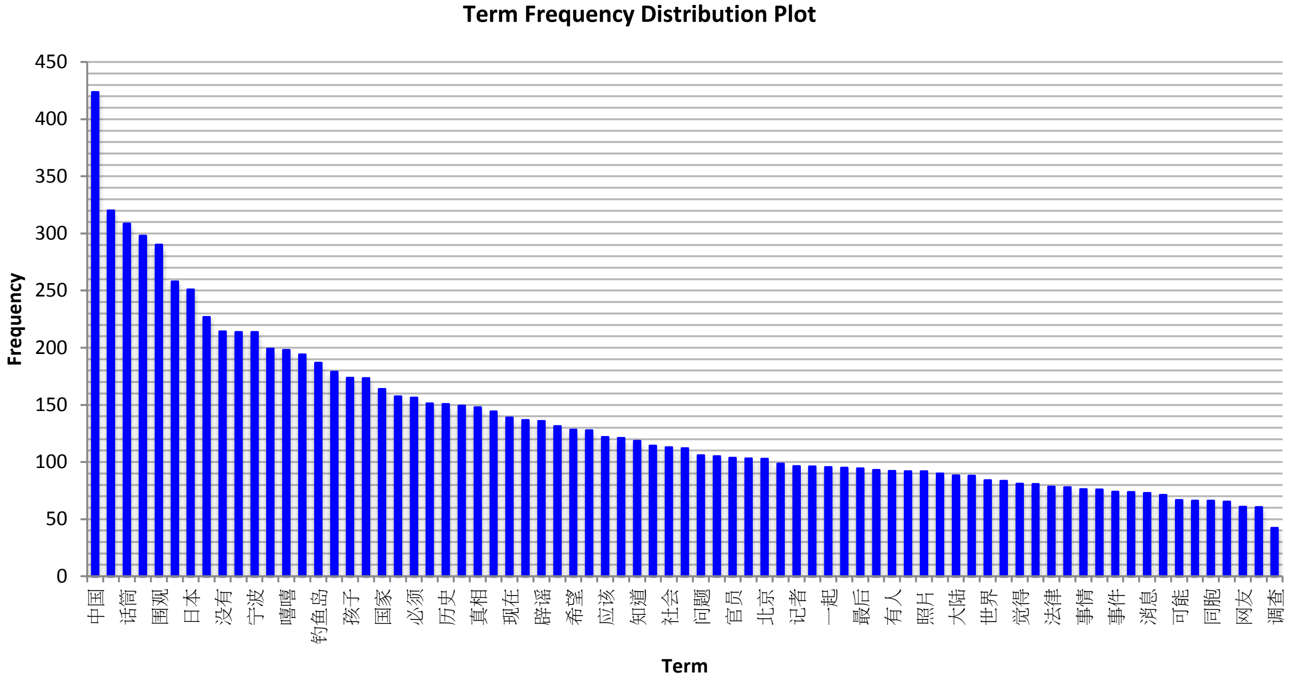


Figure 8: Term Frequency Distribution Plot 98% Matrix Sparsity

Term	English	Frequency
中国	China	423.79878
吃惊	Shock	320.25304
话筒	Anchor	308.78261
哈哈哈	Hahaha	298.01773
围观	Surround	290.28729
香港	HongKong	258.10466
日本	Japan	250.85721
人民	People	226.98542
没有	None	214.19842
表哥	Brother-In-Law	213.78053
宁波	Ning-Bo	213.75948
关注	Focus	199.2573
嘻嘻	Xixi	198.18083
支持	Support	194.03224
钓鱼岛	Senkaku	186.85814
领导	Leader	179.07458
孩子	Children	173.79866
政府	Government	173.44071
国家	Country	163.93524
看看	Look	157.41939

Table 6: List of Top 20 Terms with Highest Frequency in Microblogs

Original Terms	English Translation	钓鱼岛(Senkaku Island)
收复	Recover (Lost Land)	0.31
寸土不让	Not yield an inch of territory	0.20
占领	Occupy	0.19
买不起	Unable to Purchase	0.17
收回	Regain (Sovereignty)	0.15
不长	Not Long	0.14
社保	Social Security	0.14
解放	Liberation	0.14
放弃	Abandon	0.13
钓鱼台	Fishing terrace	0.13
养老	Provide for the Aged	0.11
三千	Three Thousand	0.10
城管	Urban Management Officer	0.10

Table 7: Terms associated with "Senkaku Islands" with correlation limit = 0.1

Original Terms	English Translation	中国 (China)
新闻界	Media Industry	0.11
豆腐	Tofu	0.11
一道	Go Along	0.1
女婿	Son-In-Law	0.1
新闻记者	News Reporter	0.1
此风	Social Conduct	0.1
肆意	Reckless	0.1
身影	Silhouette	0.1
不配	Mismatched	0.09
同类	Same Kind	0.09
官二代	Official's Second Generation	0.09
法国	France	0.09
百年	Hundreds of Years	0.09
苍蝇	Fly	0.09
获得	Obtain	0.09

Table 8: Terms associated with "China" with correlation limit = 0.09

Predictive Modeling

After data pre-processing on the text and social attributes of the datasets and using lexicon-based sentiment extraction technique and calculate sentiment scores for each of the microblogs, we now experiment the use of supervised learning techniques to automatically classify each of the incoming microblogs to be censored or not by adding censorship binary labels. 1 for censored and 0 for uncensored microblog.

Since our dataset is highly unbalanced, ratio of censored vs. uncensored microblog volume to be 9:1, we use a mix of up-sampling and down-sampling techniques to make the training datasets more balanced. We up-sample the minority class, which is censored class in our case, and triple the size by random subsample with replacement. We then down-sample the majority class, non-censored datasets, and randomly reduce the majority class data pool by 50%. Based on text, social and sentiment features, we then build separate models in four categories for each of the classification algorithms, Naïve Bayes, L1-regularized Logistic Regression and L2-regularized Logistic Regression and all features combined. Coefficients are computed and presented in Table 10 and variable significance is coded as 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1. Area-Under-Curve (ROC) is adopted to measure each model’s performance.

	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**	**Significance**
(Intercept)	-1.528e+01	3.800e-01	-40.212	< 2e-16	***
一天	-2.067e-01	1.588e-01	-1.302	0.192905	
一种	-2.841e-01	2.487e-01	-1.142	0.253440	
一起	-6.608e-01	1.885e-01	-3.506	0.000455	***
不错	-3.260e-01	1.406e-01	-2.318	0.020423	*
世界	-1.810e-01	2.030e-01	-0.891	0.372773	
中国	8.676e-01	8.871e-02	9.780	0.372773	***
京东	-5.193e+00	1.833e+00	-2.834	0.004599	**
人生	-1.452e+00	4.399e-01	-3.300	0.000966	***
今天	-3.797e-01	1.317e-01	-2.883	0.003944	**
价格	-2.832e+00	8.979e-01	-3.154	0.026692	**
关注	3.121e-01	1.409e-01	2.216	0.026692	*
发现	-9.549e-01	3.476e-01	-2.747	0.026692	**
哈哈哈	-1.361e-01	6.475e-02	-2.102	0.035577	*
喜欢	-1.110e+00	3.291e-01	-3.373	0.035577	***
嘻嘻	-2.055e-01	8.64e-02	-2.378	0.017421	*
围观	1.26e-01	7.55e-02	1.668	0.095341	.
地址	-5.19e+00	1.28e+00	-4.044	5.26e-05	***

女人	-1.292e-01	2.15e-01	-0.601	0.548011	
孩子	7.91e-02	1.11e-01	0.713	0.475909	
工作	-9.73e-01	2.95e-01	-3.293	0.000993	***
希望	2.65e-01	1.55e-01	1.71	0.087348	.
幸福	-3.44e-02	1.30e-01	-0.265	0.791187	
开心	-2.53e+00	4.95e-01	-5.123	3.01e-07	***
强烈	-4.57e-01	2.68e-01	-1.707	0.08777	.
很多	-1.28e-01	2.06e-01	-0.622	0.533936	
感觉	-8.44e-01	2.51e-01	-3.361	0.000778	***
手机	-3.59e+00	5.87e-01	-6.12	9.36e-10	***
推荐	-1.33e-01	1.71e-01	-0.775	0.438558	
支持	2.65e-01	9.93e-02	2.665	0.007689	**
日本	1.052e+00	1.171e-01	8.989	< 2e-16	***
时间	-7.375e-01	2.287e-01	-3.225	0.001259	**
明天	-1.026e-01	1.164e-01	-0.881	0.378459	
最后	9.623e-02	1.312e-01	0.734	0.463150	
朋友	-9.207e-01	2.311e-01	-3.983	6.79e-05	***
期待	-2.514e-01	1.740e-01	-1.444	0.148681	
没有	-3.831e-01	1.264e-01	-3.032	0.002430	**
活动	-4.833e+00	5.853e-01	-8.257	< 2e-16	***
现在	-2.865e-01	1.294e-01	-2.214	0.026836	*
生活	-3.066e+00	4.690e-01	-6.538	6.24e-11	***
男人	-8.782e-01	3.445e-01	-2.549	0.010798	*
看到	-6.321e-02	1.623e-01	-0.389	0.696955	
看看	1.904e-01	1.030e-01	1.849	0.064446	
知道	6.724e-02	1.281e-01	0.525	0.599570	
给力	1.035e-01	8.997e-02	1.150	0.250109	
觉得	-3.393e-01	1.568e-01	-2.165	0.030419	*
起来	-7.077e-02	1.231e-01	-0.575	0.565464	
问题	3.416e-01	1.666e-01	2.050	0.040329	*
需要	-1.966e-02	1.911e-01	-0.103	0.918052	
鼓掌	-4.531e-01	1.109e-01	4.086	4.38e-05	***
image	-2.920e-01	6.618e-02	-4.412	1.02e-05	***
source	4.312e-04	8.750e-05	4.927	8.33e-07	***
gender	1.255e+00	4.090e-02	30.677	< 2e-16	***
province	9.306e-04	1.471e-03	0.633	0.526958	
verified	1.952e-01	5.398e-02	3.617	0.000298	***
retweet_uid_indicator	-9.051e-01	5.398e-02	-13.424	< 2e-16	***
retweet_mid_indicator	1.994e+00	6.742e-02	23.725	< 2e-16	***
geo	-1.493e+00	8.406e-02	-5.663	1.49e-08	***
created_at	3.241e-02	8.406e-02	37.860	< 2e-16	***
uid	1.202e-05	2.637e-01	5.595	2.20e-08	***
delete	5.006e+00	8.560e-04	88.682	< 2e-16	***

score.pos	-1.856e+00	1.220e-01	-15.211	< 2e-16	***
score.neg	1.430e+00	1.165e-01	12.275	< 2e-16	***

Table 9 Coefficients of Combined Logistic Regression Mode

Model	Feature	AUC
Naïve Bayes	Social	**0.9559007**
	Text	0.5881988
	Sentiment	0.6151021
	Combined	0.8725469
L1-regularized LR	Social	0.8603821
	Text	0.8859119
	Sentiment	0.001201103
	Combined	0.8565421
L2-regularized LR	Social	0.5685064
	Text	0.8947931
	Sentiment	0.0008888825
	Combined	0.5652492

Table 10: Performance Evaluation of Multiple Models

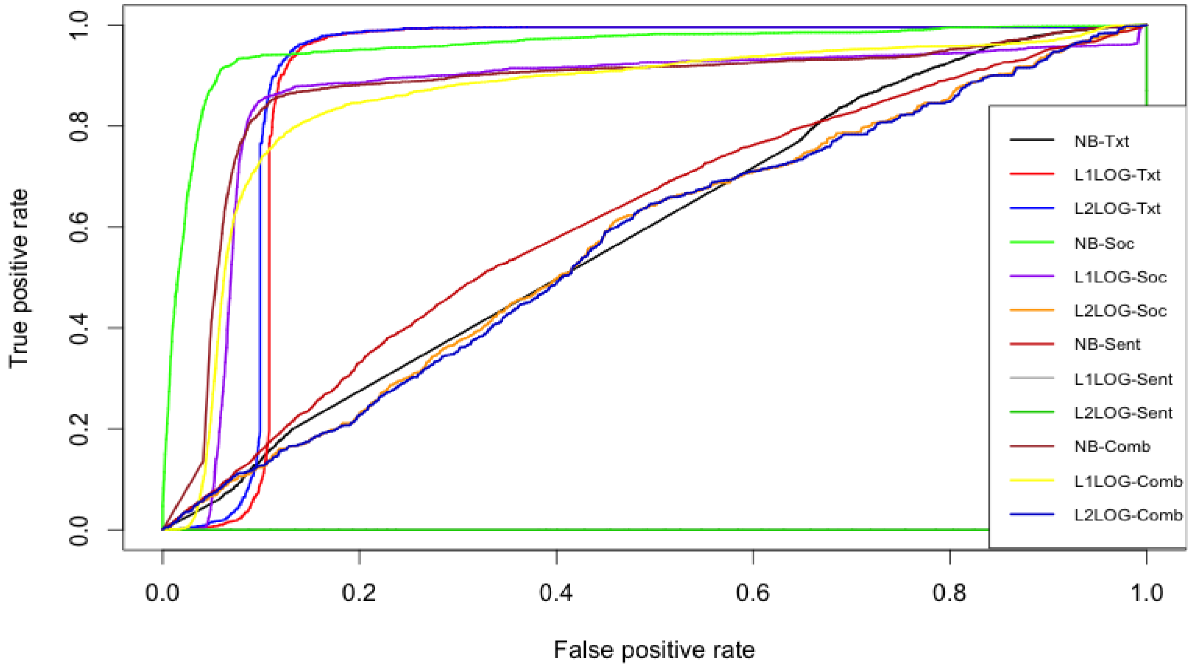


Figure 9 ROC Curve for All Models

Table 9 shows the results of our experiment. Comparing AUC scores in Figure 9, social and text features based models generate relatively similar good performance. The predictive ability of sentiment scores alone does not seem to be satisfying, especially when applying the regularized logistic regression models. However, when evaluating the significance of features, both sentiment features (positive and negative sentiment scores) have significantly small p-value (< 2e-16) and thus can be a good aggregate feature to use in modeling. As a result, naïve Bayes model trained by only social features provides the best score and has the best performance (AUC score = 95.6%).

Figure 9 visualizes the model performance by plotting Receiver Operating Characteristic curves, which is a plot of the true positive rate against the false positive rate for the different possible cut-offs in a diagnostic test. Model performance can be visualized and compared via ROC curves. From Figure 9, we can easily observe that NB-Soc (Naïve Bayes with social features) curves the most towards upper-left corner along with NB-Comb (Naïve Bayes with combined features), L1LOG-Soc (L1-regularized Logistic Regression with social features), L1LOG-Txt (L1-regularized Logistic Regression with text features), L1LOG-Comb (L1-regularized Logistic Regression with combined features), and L2LOG-Txt (L2-regularized Logistic Regression with text features). Thus, we demonstrated that Internet censorship can be quantitatively measured, and that microblogs censorship can be further predicted by constructing predictive models with selected social, text and sentiment features.

Applying our best performing model, Naïve Bayes with social features, on testing set to predict censorship outcomes, Table 10 shows 5 failure cases when our model fails to correctly detect and predict Internet censorship. In our code repository, we present a table of 188 failure cases[9].

ID	Original Text	Translation	Created	Deleted	Province	Gender	Verify	Image	Retweet	Source
1	是为了悼念这家化作灰烬的销售店以及警醒失去理智的伪爱国者阳光下的暴行来自同胞的痛诉！	This is to mourn those sales stores that turned into ashes and to alert those fake-patriots who lost their mind to assault their compatriots!	2012-09-17	0	44 (GuangDong)	Male	True	1	0	BitAuto
2	钓鱼岛是中国的周一表示抗议周二严正交涉周三深表遗憾周四密切关注周五强烈谴责周六周日休息	Senkaku Islands belongs to China. Monday: Protest; Tuesday: Negotiate; Wednesday: Deeply Regrets; Thursday: Pay Attention; Friday: Strongly Condemn; Saturday & Sunday: Take a Rest.	2012-09-11	0	440 (Oversea)	Female	True	1	0	Weibo Topic
3	公民第一课从学会如何上街抗议开始.	The first lesson on how to be a good citizen starts from how to do a protest arade.	2012-09-17	0	44 (GuangDong)	Male	True	0	0	Sina Weibo
4	北京事儿只有一辆军车遵纪守法其他多车都在逆行	In Beijing, there is only a military vehicle follows traffic rules, other cars are going against the traffic flow.	2012-09-17	0	11 (Beijing)	Male	False	1	0	Sina Weibo
5	我们没有国我们没有真正属于自己的土地房屋甚至墓地没有生育权没有出入境以及言论的自由没有保护自己不受不公平待遇的能力我们不是公民而是居民我们是一群暂住在中国这片土地上的租客在这一切没有得到改变之前你可以让我热爱地球但请不要跟我谈什么爱国我们根本没有国	We don't have our own country. We don't have our own land, home or even graveyard. We don't have freedom of speech nor the ability to protect ourselves. We are not the residence of this land but temporary renters. Before all of these changed, don't talk to me about loving this country, since we don't have one.	2012-08-17	0	44 (Guangdong)	Male	True	0	0	360 Safe Browser

Table 11 Cases when Model Fails to Detect Censorship

[9] https://github.com/just4jin/censorship_classification

CHAPTER 5 - CONCLUSION AND FUTURE WORK

Motivated by the increasing online censorship enforced on the Internet, we investigate the issue of social media censorship via statistical and machine learning approach. We analyze the patterns evolved from Internet censorship and build supervised learning models to automatically detect and predict Internet censorship. As a result of our research, we have shown how online social media have a correlation with real-world events, and how public opinions and sentiment can be detected through statistical analysis. Furthermore, similar to previous research on automatically detecting and predicting online activities such as cyber-recruitment by violent extremists[10], we have demonstrated that detecting and predicting Internet censorship on microblogging platform is also a feasible task.

In the future, our Internet censorship classifiers can be further improved by training with larger datasets and with more aggregate features. Our work has shown the effectiveness of incorporating sentiment score as an aggregate feature to boost modeling performance. Sentiment indicator can be further broke down into in-depth psychological mood expressed, such as happy, fear, angry, shocked, etc. to test their effectiveness in improving modeling performance. In our research, we constructed sentiment feature by using lexicon-based approach, in the future, supervised and unsupervised learning techniques can be used to perform the sentiment classification task. As a result, the accuracy of Internet censorship classifiers may be further improved with a more detailed and more precisely aggregated sentiment feature. Future research could also explore other classification algorithms. We implemented three supervised learning

[10] Scanlon, JacobR and Gerber, MatthewS. "Automatic detection of cyber-recruitment by violent extremists ." *Security Informatics* (Springer Berlin Heidelberg) 3 (2014): 1-10.

algorithms - Naïve Bayes, Logistic Regression with L1 and L2 Regularization. Other classification models such as Support Vector Machine (SVM) and Random Forest may be used to evaluate their performance in this task.

BIBLIOGRAPHY

[1] M. D. e. a. Conover, "Predicting the political alignment of twitter users.," pp. 192-199, 2011.

[2] H. M. X. Z. Johan Bollen, "Twitter mood predicts the stock market," *Journal of Computational Science,* vol. 2, no. 1, pp. 1-8, 2011.

[3] L. T. S. D. J. V. I. K. B. Zmazek, "Application of decision trees to the analysis of soil radon data for earthquake prediction," *Applied Radiation and Isotopes,* vol. 58, no. 6, pp. 697-706, 2003.

[4] E. S. M. a. M. M. Aramaki, "Twitter catches the flu: detecting influenza epidemics using Twitter," in *Proceedings of the conference on empirical methods in natural language processing*, 2011.

[5] J. a. R. A. a. G. I. a. C. E. a. F. A. a. F. D. a. F. D. a. I. D. a. P. M. P. a. R. G. a. o. Borge-Holthoefer, "Structural and dynamical patterns on online social networks: the Spanish May 15th movement as a case study," *PloS One,* vol. 6, no. 8, p. e23883, 2011.

[6] M. a. R. J. a. F. M. a. G. c. B. a. M. F. a. F. A. Conover, "Political polarization on twitter," *ICWSM,* pp. --, 2011.

[7] J. a. S. J. Fabrega, "The Emergence of Political Discourse on Digital Networks: The Case of the Occupy Movement," *arXiv preprint arXiv:1308.1176,* pp. --, 2013.

[8] Z. a. W. C. Tufekci, "Social media and the decision to participate in political protest: Observations from Tahrir Square," *Journal of Communication,* vol. 62, no. 2, pp. 363-379, 2012.

[9] M. a. A.-M. P. Pennacchiotti, "A Machine Learning Approach to Twitter User Classification.," *ICWSM 11,* pp. 281-288, 2011.

[10] J. a. G. M. Scanlon, "Automatic detection of cyber-recruitment by violent extremists," *Security Informatics,* vol. 3, pp. 1-10, 2014.

[11] L. T. a. W. P. a. C. W. a. P. W. a. Z. Y. Nguyen, "Predicting collective sentiment dynamics from time-series social media," *ACM,* p. 6, 2012.

[12] D. a. D. S. T. a. L. D. J. Ramage, "Characterizing Microblogs with Topic Models," *ICWSM,*

vol. 10, pp. --, 2010.

[13] T. A. a. B. W. Meyer, "Effective open-source, Bayesian based, email classification system," *CEAS,* 2004.

[14] D. a. O. B. a. S. N. Bamman, "Censorship and deletion practices in Chinese social media," *First Monday,* vol. 17, p. 3, 2012.

[15] T. a. P. D. a. P. A. a. C. J. R. a. W. D. S. Zhu, "The velocity of censorship: High-fidelity detection of microblog post deletions," *arXiv preprint arXiv:1303.0597,* pp. --, 2013.

[16] K.-w. a. C. C.-h. a. C. M. Fu, "Assessing censorship on microblogs in China: discriminatory keyword analysis and the real-name registration policy," *Internet Computing, IEEE,* vol. 17, no. 3, pp. 42-50, 2013.

[17] D. Desilver, "China has more internet users than any other country.," 2 December 2013. [Online]. Available: http://www.pewresearch.org/fact-tank/2013/12/02/china-has-more-internet-users-than-any-other-country/. [Accessed 1 January 2014].

[18] China Internet Network Information Center, "Statistical Report on Internet Development in China (January 2013)," China Internet Network Information Center, 2013.

[19] D. Cohen, "Xi Jinping: China's First Social Media President?," 08 Jan 2013. [Online]. [Accessed 1 Jan 2014].

[20] G. a. P. J. a. R. M. E. King, "How censorship in China allows government criticism but silences collective expression," *American Political Science Review,* vol. 107, no. 02, pp. 326-343, 2013.

[21] N. a. R. G. a. X. X. a. M. Z. M. a. H. J. A. Spring, "Internet Censorship in China: Where Does the Filtering Occur?," vol. 6579, 2011.

[22] A.´. Fonseca, "Modeling political opinion dynamics through social media and multi-agent simulation," *First Doctoral Workshop for Complexity Sciences,* 2011.

[23] S. a. A. A. a. S. A. Valenzuela, "The social media basis of youth protest behavior: The case of Chile," *Journal of Communication,* vol. 62, no. 2, pp. 299-314, 2012.

[24] O. a. F. E. a. O. C. L. a. M. F. a. F. A. Varol, "Evolution of Online User Behavior During a Social Upheaval," in *Proceedings of the 2014 ACM Conference on Web Science,* New York, NY, USA, 2014.

[25] S. M. O. a. V. L. Petrovic, "I wish i didn't say that! analyzing and predicting deleted messages in twitter," *arXiv preprint arXiv:1305.3107*, 2013.

[26] M. D. a. F. E. a. M. F. a. F. A. Conover, "The digital evolution of occupy wall street," *PloS one*, vol. 8, no. 5, p. e64679, 2013.

[27] A. a. H. W. a. L. K. a. P. D. a. L. W.-K. Choudhary, "Social media evolution of the Egyptian revolution," *Communications of the ACM*, vol. 55, no. 5, pp. 74-80, 2012.

[28] D. Morrison, "Toward Automatic Censorship Detection in Microblogs," *Trends and Applications in Knowledge Discovery and Data Mining*, p. 8643, 2014.

[29] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

[30] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, no. 3, pp. 211-229, 1959.

[31] T. Mitchell, "Machine Learning," New York, McGraw Hill. ISBN 0-07-042807-7, 1997, p. 2.

[32] K.-w. a. C. M. Fu, "Reality check for the Chinese microblog space: a random sampling approach," *PloS One*, vol. 8, no. 3, p. e58356, 2013.

[33] M. Narasimha Murty and V. Susheela Devi, Pattern Recognition: An Algorithmic Approach, ISBN 0857294946, 2011.

[34] K.-W. C. C.-J. H. X.-R. W. a. C.-J. L. R.-E. Fan, "LIBLINEAR: A library for large linear classification Journal of Machine Learning Research," 9 2008. [Online].

[35] K. M. a. L. S. a. S. Y. DeLuca, "Occupy Wall Street on the public screens of social media: The many framings of the birth of a protest movement," *Communication, Culture and Critique*, vol. 5, no. 4, pp. 483-509, 2012.

[36] K. a. Z. B. a. Z. J. a. Y. H. Wu, "Sentiment Classification for Topical Chinese Microblog Based on Sentences' Relations," *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pp. 2221-2225, 2013.

[37] L. a. Z. C. a. W. C. Chen, "Tweeting under pressure: analyzing trending topics and evolving word choice on sina weibo," in *Proceedings of the first ACM conference on Online social networks*. 2013.

[38] C. F. S. a. L. V. Castellano, "Statis- tical physics of social dynamics," *Reviews of Modern Physics* , pp. 1-58, 1007.

[39] A. Samuel, "Some Studies in Machine Learning using the Game of Checkers," *IBM Journal of* .

APPENDIX

R Code is uploaded at https://github.com/just4jin/censorship_classification

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.1.0 (64-bit)

library("klaR")
library("FSelector")
library("maxent")
library("LiblineaR")
library("ROCR")
library("caret")
library("Rwordseg")
library("Rweibo")
library("tm")
library("RTextTools")
library("jiebaR")
library("e1071")
library("MASS")
library("AUC")
library("rpart")
library("ROCR")

# set up work directory
setwd("~./code")

# read in data
data <- read.csv("./data/data.csv",sep=',',header=T, quote = "\"",  encoding='UTF-8')

# variable names & size
names(data)
dim(data)

# exclude NA values
data <- na.omit(data)

# ratio of censored vs. uncensored
sum(data$censor_indicator==0)/sum(data$censor_indicator==1)




#********************************************************************
#
#                                    Text Features
#
#********************************************************************
```

```
# text data cleansing
data$text=gsub("[0-9 0 1 2 3 4 5 6 7 8 9]","",data$text)
data$text=gsub("[a-zA-Z]","",data$text)
data$text = gsub(pattern="http:[a-zA-Z\\/\\.0-9]+","", data$text)
data$text = gsub(pattern="@(\\w+)[,: ]","", data$text)
data$text = gsub('[[:punct:]]', '', data$text)
data$text = gsub('[[:cntrl:]]', '', data$text)
data$text = gsub('\\d+', '', data$text)
data$text=gsub(pattern="我在(\\w*)","",data$text)
data$text=gsub(pattern="我在这里(\\w*)","",data$text)
data$text=gsub(pattern="发表了博文(\\w*)","",data$text)
data$text=gsub(pattern="我上传了视频(\\w*)","",data$text)
data$text=gsub(pattern="视频(\\w*)","",data$text)
data$text=gsub(pattern="转发微博(\\w*)","",data$text)
data$text=gsub(pattern="转发微博","",data$text)
data$text=gsub(pattern="转发(\\w*)","",data$text)
data$text=gsub(pattern="回复(\\w*)","",data$text)
data$text=gsub(pattern="最新消息(\\w*)","",data$text)
data$text=gsub(pattern="(\\w*)美图秀秀","",data$text)
data$text=gsub(pattern="分享(\\w*)","",data$text)

# remove duplicates
data <- data[-which(duplicated(data$text)),]

# install word segmentation dictionary
installDict("./Dict/word_segmentation/宣传舆论学词库.scel","sougou_1")
installDict("./Dict/word_segmentation/网络流行新词官方推荐.scel","sougou_2")
installDict("./Dict/word_segmentation/网络流行语.scel","sougou_3")
installDict("./Dict/word_segmentation/真正的打字好秘书.scel","sougou_4")
installDict("./Dict/word_segmentation/政治学词库.scel","sougou_5")
installDict("./Dict/word_segmentation/五笔版网络流行新词官方推荐.scel","sougou_6")
installDict("./Dict/word_segmentation/2009年度百位华人公共知识分子.scel","sougou_7")
installDict("./Dict/word_segmentation/社会主义词汇.scel","sougou_8")
installDict("./Dict/word_segmentation/personalDict.txt","sougou_9")
installDict("./Dict/word_segmentation/2009年度百位华人公共知识分子.scel","sougou_10")

# word segmentation
doc_CN=list()
for(j in 1:length(data$text)){
  doc_CN[[j]]=c(segmentCN(data$text[j]))
}

# remove stopwords
stw <- readLines("./Dict/stopwords/stopwords.txt",encoding='UTF-8')
stw <- c(stw,"http","cn","www","里","称","不能","不要")
stopwords_CN<-as.vector(stw)
```

```
for(j in 1:length(data$text)){
  doc_CN[[j]] <-doc_CN[[j]][!(doc_CN[[j]] %in% stopwords_CN)]
}

# build corpus
corpus=Corpus(VectorSource(doc_CN))

# remove stopwords
stw <- readLines("./Dict/stopwords/stopwords.txt",encoding='UTF-8')
stw <- c(stw,"http","cn","www","里","称","不能","不要")
stopwords_CN<-as.vector(stw)

for(j in 1:length(data$text)){
  doc_CN[[j]] <-doc_CN[[j]][!(doc_CN[[j]] %in% stopwords_CN)]
}

# build corpus
corpus=Corpus(VectorSource(doc_CN))

# build document term matrix (dtm) with tf-idf weighting
control=list(removePunctuation=TRUE,minDocFreq=2, wordLengths = c(2, Inf),
             stopwords=TRUE, weighting = weightTfIdf)
dtm <-DocumentTermMatrix(corpus,control)


# reduce sparcity
dtms<-removeSparseTerms(dtm,0.99)

# frequent terms
findFreqTerms(dtms,900)

# write out dtm and dtms
write.csv(as.data.frame(inspect(dtm)) , file="dtm.csv")
write.csv(as.data.frame(inspect(dtms)) , file="dtms.csv")

# convert to data frame
dtms<-as.data.frame(inspect(dtms))



#**************************************************************
#
#                                   Social Features
#
#**************************************************************

# social attributes
image <- as.factor(data$image)
source <- as.factor(data$source)
province <- as.factor(data$province)
```

```
gender <- as.factor(data$gender)
verified <- as.factor(data$verified)
retweet_mid_indicator<-as.factor(data$retweet_mid_indicator)
retweet_uid_indicator<-as.factor(data$retweet_uid_indicator)
geo<-as.factor(data$geo)
created_at<-as.factor(data$created_at)
uid<-as.factor(data$uid)
delete<-as.factor(data$delete_indicator)

# social attributes
social <-
cbind(image,source,gender,province,verified,retweet_uid_indicator,retweet_mid_indicator,geo,created_at
,uid,delete)
social<-as.data.frame(social)




#***************************************************************
#
#       Sentiment Features
#
#***************************************************************

# install sentiment dictionary
installDict("./Dict/sentiment_words/HowNet_positive_review.txt","pos_1")
installDict("./Dict/sentiment_words/HowNet_positive_sentiment.txt","pos_2")
installDict("./Dict/sentiment_words/NTUSD_positive_simplified.txt","pos_3")
installDict("./Dict/sentiment_words/pos.txt","pos_4")
installDict("./Dict/sentiment_words/HowNet_negative_review.txt","neg_1")
installDict("./Dict/sentiment_words/HowNet_negative_sentiment.txt","neg_2")
installDict("./Dict/sentiment_words/NTUSD_negative_simplified.txt","neg_3")
installDict("./Dict/sentiment_words/neg.txt","neg_4")

# load dictionary of polarity words
pos_1 <- readLines("./Dict/sentiment_words/NTUSD_positive_simplified.txt", encoding='UTF-8')
pos_2 <- readLines("./Dict/sentiment_words/HowNet_positive_review.txt", encoding='UTF-8')
pos_3 <- readLines("./Dict/sentiment_words/HowNet_positive_sentiment.txt",encoding='UTF-8')
pos_4 <- readLines("./Dict/sentiment_words/pos.txt",encoding='UTF-8')
pos <- c(pos_1,pos_2,pos_3, pos_4)
pos <- pos[-which(duplicated(pos))]

neg_1 <- readLines("./Dict/sentiment_words/NTUSD_negative_simplified.txt", encoding='UTF-8')
neg_2 <- readLines("./Dict/sentiment_words/HowNet_negative_review.txt", encoding='UTF-8')
neg_3 <- readLines("./Dict/sentiment_words/HowNet_negative_sentiment.txt",encoding='UTF-8')
neg_4 <- readLines("./Dict/sentiment_words/neg.txt",encoding='UTF-8')
neg <- c(neg_1,neg_2,neg_3, neg_4)
neg <- neg[-which(duplicated(neg))]

# sentiment calculation
scores=rep(0,times=length(doc_CN))
```

```
score.pos=scores
score.neg=scores

for (j in 1:length(doc_CN)){

  # number of positive words in each row
  score.pos[j]<-sum(doc_CN[[j]] %in% pos)/length(doc_CN[[j]])
  # number of negative words in each row
  score.neg[j]<-sum(doc_CN[[j]] %in% neg)/length(doc_CN[[j]])
}

score.pos<-as.numeric(score.pos)
score.neg<-as.numeric(score.neg)
score.pos[is.nan(score.pos)] <- 0
score.neg[is.nan(score.neg)] <- 0
positive<- as.integer((score.pos-score.neg)>0)
negative<- as.integer((score.pos-score.neg)<0)

# sentiment label as factor
senti_label <- as.data.frame(cbind(positive, negative))
# sentiment score as numeric
senti_score <- as.data.frame(cbind(score.pos, score.neg))

# feature coefficients
x<-as.matrix(x)
y<-as.factor(y)

glm.out = glm(y~x,family=binomial(logit))
summary(glm.out)


#####################################################################
#
#          Classification (Naive Bayes, Logistic w/ L1 & L2 Regulation)
#
#####################################################################

#******************************************************
#
#                         Model - Text
#
#*******************************************************

dim(dtms)

# combine dtms with censor indicator
text_data <- cbind(dtms,data$censor_indicator)

# randomize text data
text_data <- text_data[sample(1:nrow(text_data),nrow(text_data),replace=FALSE),]
```

```
# subsample for train and test datasets 8:2
set.seed(1)
trainIndicator = rbinom(length(data[,14]), size=1, prob=0.8)
train_text = text_data[trainIndicator == 1,]
test_text = text_data[trainIndicator == 0,]


#***************************************************************
#
#                       Naïve Bayes w/ Text – Non-Resampling
#
#***************************************************************

# construct training set for model
x <- train_text[,-50]
y <- as.factor(train_text[,50])

# NB without resampling
nb_text = naiveBayes(x,y)
table(nb_text_pred,truth=test_text[,50])
confusionMatrix(nb_text_pred, as.factor(test_text[,50]))

nb_text_pred_raw<-predict(nb_text,test_text[,-50],type='raw')
pred <- prediction(nb_text_pred_raw[,2],test_text[,50])
perf <- performance(pred, "tpr", "fpr")
perf.auc <- performance(pred,"auc")
auc <- perf.auc@y.values
auc


#**********************************************************************************
#
#           Upsampling on Minority Class & Downsampling on Majority Class
#
#**********************************************************************************

# train sets downsample majority
set.seed(1)

# up-sample minority
train_minority <- train_text[which(train_text[50]==1),]
train_minority1 <- train_text[sample(1:nrow(train_minority),
                                     length(train_minority[,50]),replace=T),]
train_minority2 <- train_text[sample(1:nrow(train_minority),
                                  length(train_minority[,50]),replace=T),]
train_us <- rbind(train_minority, train_minority1,train_minority2)

# down-sample majority
train_majority <- train_text[which(train_text[50]==0),]
majorityIndicator = rbinom(length(train_majority[,50]), size=1, prob=0.5)
train_ds <- train_majority[majorityIndicator==1,]
```

```
dim(train_ds)
dim(train_us)

# construct full datasets after sampling
train_text_ud <- rbind(train_us, train_ds)

# randomize
train_text_ud <- train_text_ud[sample(1:nrow(train_text_ud),nrow(train_text_ud),replace=FALSE),]

dim(train_text_ud)

#****************************************************************
#
#                    Naïve Bayes  w/ Text – Resampling
#
#****************************************************************

# naive bayes model with resampling techniques
nb_text_ud = NaiveBayes(train_text_ud[,-50],as.factor(train_text_ud[,50]))
nb_text_ud_pred <- predict(nb_text_ud,test_text)

table(nb_text_ud_pred$class,truth=test_text[,50])
confusionMatrix(nb_text_ud_pred$class, test_text[,50])

pred1 <- prediction(nb_text_ud_pred$posterior[,2],test_text[,50])
perf1 <- performance(pred1, "tpr", "fpr")
perf1.auc <- performance(pred1,"auc")
auc <- perf1.auc@y.values
auc

###################################################################
#
#          L1 & L2 Logistic Regression w/ Text - Resampling
#
#################################################################

# construct training data with resampling techniques
x = train_text_ud[,-50]
y = as.factor(train_text_ud[,50])

# 10-cross validation with accuracy metric
lib_text_l1_cv<-LiblineaR(x, y,type=6,cross=10) # accuracy 0.8899917
lib_text_l2_cv<-LiblineaR(x, y,type=0,cross=10) # accuracy 0.8899917

# L1 & L2 Logistic Model construction
lib_text_l1<-LiblineaR(x, y,type=6)
lib_text_l2<-LiblineaR(x, y,type=0)

lib_text_l1_pred<-predict(lib_text_l1,test_text,proba=TRUE)
lib_text_l2_pred<-predict(lib_text_l2,test_text,proba=TRUE)
```

```
table(lib_text_l1_pred$predictions,truth=test_text[,50])
table(lib_text_l2_pred$predictions,truth=test_text[,50])
confusionMatrix(lib_text_l2_pred$predictions, test_text[,50])
confusionMatrix(lib_text_l2_pred$predictions, test_text[,50])

# L1 - regularized
pred2 <- prediction(lib_text_l1_pred$probabilities[,2],test_text[,50])
perf2 <- performance(pred2, "tpr", "fpr")
perf2.auc <- performance(pred2,"auc")
auc <- perf2.auc@y.values
auc

# L2 - regularized
pred3 <- prediction(lib_text_l2_pred$probabilities[,2], test_text[,50])
perf3 <- performance(pred3, "tpr", "fpr")
perf3.auc <- performance(pred3,"auc")
auc <- perf3.auc@y.values
auc



######################################################
#
#                    Model - Social
#
######################################################

dim(social)

# combine dtms with censor indicator
social_data <- cbind(social,data$censor_indicator)

# randomize social data
social_data <- social_data[sample(1:nrow(social_data),nrow(social_data),replace=FALSE),]

# subsample for train and test datasets 8:2
set.seed(1)
trainIndicator = rbinom(length(social_data[,12]), size=1, prob=0.8)

train_social = social_data[trainIndicator == 1,]
test_social = social_data[trainIndicator == 0,]


#**************************************************************************
#
#                Upsampling on Minority Class & Downsampling on Majority Class
#
#**************************************************************************

# train sets downsample majority
```

```
set.seed(1)

# upsample minority
train_minority <- train_social[which(train_social[12]==1),]
train_minority1 <- train_social[sample(1:nrow(train_minority),
                                length(train_minority[,12]),replace=T),]
train_minority2 <- train_social[sample(1:nrow(train_minority),
                                length(train_minority[,12]),replace=T),]
train_us <- rbind(train_minority, train_minority1,train_minority2)

# downsample majority
train_majority <- train_social[which(train_social[12]==0),]
majorityIndicator = rbinom(length(train_majority[,12]), size=1, prob=0.5)
train_ds <- train_majority[majorityIndicator==1,]

dim(train_ds)
dim(train_us)

# construct full datasets after sampling
train_social_ud <- rbind(train_us, train_ds)

# randomize
train_social_ud <-
train_social_ud[sample(1:nrow(train_social_ud),nrow(train_social_ud),replace=FALSE),]

dim(train_social_ud)


####################################################
#
#           Naive Bayes w/ Social - Resampling
#
####################################################

# naive bayes model with resampling techniques
nb_social_ud = NaiveBayes(train_social_ud[,-12],as.factor(train_social_ud[,12]))
nb_social_ud_pred <- predict(nb_social_ud,test_social)

table(nb_social_ud_pred$class,truth=test_social[,12])
confusionMatrix(nb_social_ud_pred$class, test_social[,12])

pred4 <- prediction(nb_social_ud_pred$posterior[,2],test_social[,12])
perf4 <- performance(pred4, "tpr", "fpr")
perf4.auc <- performance(pred4,"auc")
auc <- perf4.auc@y.values
auc
```

```
####################################################
#
#           L1 & L2 Logistic Regression w/ Social - Resampling
#
###################################################

x <- train_social_ud[,-12]
y <- as.factor(train_social_ud[,12])

lib_social_l1_cv<-LiblineaR(x, y,type=6,cross=10) # accuracy 0.9413629
lib_social_l2_cv<-LiblineaR(x, y,type=0,cross=10) # accuracy 0.890448
lib_social_l1<-LiblineaR(x, y,type=6,bias=TRUE)
lib_social_l2<-LiblineaR(x, y,type=0,bias=TRUE)
lib_social_l1_pred<-predict(lib_social_l1,test_social,proba=TRUE)
lib_social_l2_pred<-predict(lib_social_l2,test_social,proba=TRUE)

table(lib_social_l1_pred$predictions,truth=test_social[,12])
table(lib_social_l2_pred$predictions,truth=test_social[,12])
confusionMatrix(lib_social_l1_pred$predictions, test_social[,12])
confusionMatrix(lib_social_l2_pred$predictions, test_social[,12])


# L1
pred5 <- prediction(lib_social_l1_pred$probabilities[,2],test_social[,12])
perf5 <- performance(pred5, "tpr", "fpr")
perf5.auc <- performance(pred5,"auc")
auc <- perf5.auc@y.values
auc

# L2
pred6 <- prediction(lib_social_l2_pred$probabilities[,2],test_social[,12])
perf6 <- performance(pred6, "tpr", "fpr")
perf6.auc <- performance(pred6,"auc")
auc <- perf6.auc@y.values
auc


###################################################
#
#                 Model - Sentiment
#
###################################################

dim(senti_score)

# combine dtms with censor indicator
sentiment_data <- cbind(senti_score,data$censor_indicator)

# randomize sentiment data
```

```
sentiment_data <-
sentiment_data[sample(1:nrow(sentiment_data),nrow(sentiment_data),replace=FALSE),]

# subsample for train and test datasets 8:2
set.seed(1)
trainIndicator = rbinom(length(sentiment_data[,3]), size=1, prob=0.8)

train_sentiment = sentiment_data[trainIndicator == 1,]
test_sentiment = sentiment_data[trainIndicator == 0,]


#****************************************************************************
#
#      Upsampling on Minority Class & Downsampling on Majority Class
#
#****************************************************************************

# train sets downsample majority
set.seed(1)

# upsample minority
train_minority <- train_sentiment[which(train_sentiment[3]==1),]
train_minority1 <- train_sentiment[sample(1:nrow(train_minority),
                                 length(train_minority[,3]),replace=T),]
train_minority2 <- train_sentiment[sample(1:nrow(train_minority),
                                 length(train_minority[,3]),replace=T),]
train_us <- rbind(train_minority, train_minority1,train_minority2)

# downsample majority
train_majority <- train_sentiment[which(train_sentiment[3]==0),]
majorityIndicator = rbinom(length(train_majority[,3]), size=1, prob=0.5)
train_ds <- train_majority[majorityIndicator==1,]

dim(train_ds)
dim(train_us)

# construct full datasets after sampling
train_sentiment_ud <- rbind(train_us, train_ds)

# randomize
train_sentiment_ud <-
train_sentiment_ud[sample(1:nrow(train_sentiment_ud),nrow(train_sentiment_ud),replace=FALSE),]

dim(train_sentiment_ud)


####################################################
#
#               Naive Bayes w/ Sentiment - Resampling
#
####################################################
```

```
# naive bayes model with resampling techniques
nb_sentiment_ud = NaiveBayes(train_sentiment_ud[,-3],as.factor(train_sentiment_ud[,3]))
nb_sentiment_ud_pred <- predict(nb_sentiment_ud,test_sentiment)

table(nb_sentiment_ud_pred$class,truth=test_sentiment[,3])
confusionMatrix(nb_sentiment_ud_pred$class, test_sentiment[,3])

pred7 <- prediction(nb_sentiment_ud_pred$posterior[,2],test_sentiment[,3])
perf7 <- performance(pred7, "tpr", "fpr")
perf7.auc <- performance(pred7,"auc")
auc <- perf7.auc@y.values
auc




###############################################################
#
#	L1 & L2 Logistic Regression w/ Sentiment - Resampling
#
###############################################################

x <- train_sentiment_ud[,-3]
y <- as.factor(train_sentiment_ud[,3])

lib_sentiment_l1_cv<-LiblineaR(x, y,type=6,cross=10) # accuracy 0.8919885
lib_sentiment_l2_cv<-LiblineaR(x, y,type=0,cross=10) # accuracy 0.8919885
lib_sentiment_l1<-LiblineaR(x, y,type=6,bias=TRUE)
lib_sentiment_l2<-LiblineaR(x, y,type=0,bias=TRUE)
lib_sentiment_l1_pred<-predict(lib_sentiment_l1,test_sentiment,proba=TRUE)
lib_sentiment_l2_pred<-predict(lib_sentiment_l2,test_sentiment,proba=TRUE)

table(lib_sentiment_l1_pred$predictions,truth=test_sentiment[,3])
table(lib_sentiment_l2_pred$predictions,truth=test_sentiment[,3])
confusionMatrix(lib_sentiment_l1_pred$predictions, test_sentiment[,3])
confusionMatrix(lib_sentiment_l2_pred$predictions, test_sentiment[,3])

# L1
pred8 <- prediction(lib_sentiment_l1_pred$probabilities[,2],test_sentiment[,3])
perf8 <- performance(pred8, "tpr", "fpr")
perf8.auc <- performance(pred8,"auc")
auc <- perf8.auc@y.values
auc

# L2
pred9 <- prediction(lib_sentiment_l2_pred$probabilities[,2],test_sentiment[,3])
perf9 <- performance(pred9, "tpr", "fpr")
perf9.auc <- performance(pred9,"auc")
auc <- perf9.auc@y.values
auc
```

```
############################################################
#
#                        Model - Combined
#
############################################################

#*********************************************************
#
#                         Construct Full Datasets
#
#*********************************************************

# combine dtms, social and sentiment with censor indicator
data_full <- cbind(dtms, social, senti_score, data$censor_indicator)

# randomize text data
data_full <- data_full[sample(1:nrow(data_full),nrow(data_full),replace=FALSE),]

dim(data_full)

# # subsample - train:test = 8:2
set.seed(1)
trainIndicator = rbinom(length(data_full[,63]), size=1, prob=0.8)

train_full = data_full[trainIndicator == 1,]
test_full = data_full[trainIndicator == 0,]


#**************************************************************************
#
#       Upsampling on Minority Class & Downsampling on Majority Class
#
#**************************************************************************

# train sets downsample majority
set.seed(1)

# upsample minority
train_minority <- train_full[which(train_full[63]==1),]
train_minority1 <- train_full[sample(1:nrow(train_minority),
                                     length(train_minority[,63]),replace=T),]
train_minority2 <- train_full[sample(1:nrow(train_minority),
                                     length(train_minority[,63]),replace=T),]
train_us <- rbind(train_minority, train_minority1,train_minority2)

# downsample majority
train_majority <- train_full[which(train_full[63]==0),]
majorityIndicator = rbinom(length(train_majority[,63]), size=1, prob=0.5)
train_ds <- train_majority[majorityIndicator==1,]
```

```
dim(train_ds)
dim(train_us)

# construct full datasets after sampling
train_full_ud <- rbind(train_us, train_ds)

# randomize
train_full_ud <- train_full_ud[sample(1:nrow(train_full_ud),nrow(train_full_ud),replace=FALSE),]

dim(train_full_ud)


##################################################
#
#               Naïve Bayes w/ Combined - Resampling
#
##################################################

# naive bayes model with resampling techniques
nb_full_ud = NaiveBayes(train_full_ud[,-63],as.factor(train_full_ud[,63]))
nb_full_ud_pred <- predict(nb_full_ud,test_full)

table(nb_full_ud_pred$class,truth=test_full[,63])
confusionMatrix(nb_full_ud_pred$class, test_full[,63])

pred10 <- prediction(nb_full_ud_pred$posterior[,2],test_full[,63])
perf10 <- performance(pred10, "tpr", "fpr")
perf10.auc <- performance(pred10,"auc")
auc <- perf10.auc@y.values
auc

##########################################################
#
#           L1 & L2 Logistic Regression w/ Combined - Resampling
#
##########################################################

x <- train_full_ud[,-63]
y <- as.factor(train_full_ud[,63])

lib_full_l1_cv<-LiblineaR(x, y,type=6,cross=10) # accuracy 0.9498474
lib_full_l2_cv<-LiblineaR(x, y,type=0,cross=10) # accuracy 0.8899308
lib_full_l1<-LiblineaR(x, y,type=6,bias=TRUE)
lib_full_l2<-LiblineaR(x, y,type=0)
lib_full_l1_pred<-predict(lib_full_l1,test_full,proba=TRUE)
lib_full_l2_pred<-predict(lib_full_l2,test_full,proba=TRUE)

table(lib_full_l1_pred$predictions,truth=test_full[,63])
table(lib_full_l2_pred$predictions,truth=test_full[,63])
confusionMatrix(lib_full_l1_pred$predictions, test_full[,63])
confusionMatrix(lib_full_l2_pred$predictions, test_full[,63])


# L1
pred11 <- prediction(lib_full_l1_pred$probabilities[,2],test_full[,63])
perf11 <- performance(pred11, "tpr", "fpr")
perf11.auc <- performance(pred11,"auc")
auc <- perf11.auc@y.values
auc

# L2
pred12 <- prediction(lib_full_l2_pred$probabilities[,2],test_full[,63])
perf12 <- performance(pred12, "tpr", "fpr")
perf12.auc <- performance(pred12,"auc")
auc <- perf12.auc@y.values
auc


##########################################################
#
#                                   ROC Plots
#
##########################################################

# Plot ROC
plot(perf1,lwd = 2)
plot(perf2, add = TRUE, col="red",lwd = 2)
plot(perf3, add = TRUE, col="blue",lwd = 2)
plot(perf4, add = TRUE, col="green",lwd = 2)
plot(perf5, add = TRUE, col="purple",lwd = 2)
plot(perf6, add = TRUE, col="orange",lwd = 2)
plot(perf7, add = TRUE, col="red3",lwd = 2)
plot(perf8, add = TRUE, col="grey",lwd = 2)
plot(perf9, add = TRUE, col="green3",lwd = 2)
plot(perf10, add = TRUE, col="brown",lwd = 2)
plot(perf11, add = TRUE, col="yellow",lwd = 2)
plot(perf12, add = TRUE, col="blue3",lwd = 2)

# Legend
legend(.84, .84, legend = c("NB-Txt", "L1LOG-Txt", "L2LOG-Txt","NB-Soc", "L1LOG-Soc", "L2LOG-Soc","NB-Sent", "L1LOG-Sent", "L2LOG-Sent","NB-Comb", "L1LOG-Comb", "L2LOG-Comb"), lwd = 2, cex=0.7, col = c("black","red","blue", "green","purple","orange","red3", "grey","green3","brown","yellow","blue3"))





#****************************************************************
#
```

```
#                              Sentiment Trend
#
#****************************************************************

source("feature.R")

# read in data searched by Senkaku Islands keywords
twt <- read.csv("./data/keyword.csv",sep=',',header=T, quote = "\"",  encoding='UTF-8')

# variable names & size
names(twt)
dim(twt)

# exclude NA values
twt <- na.omit(twt)

# select all rows based on days
twt$created_at <- as.character(as.Date(twt$created_at))
text <- twt$text

length(date)
length(text)

twt$created_at<-as.Date(twt$created_at, format="%Y-%m-%d")

head(twt)
tail(twt)

# remove duplicates
twt <- subset(twt, !duplicated(twt[,1]) )

# exclude NA values
twt <- na.omit(twt)

dim(twt)

# install sentiment dictionary
installDict("./Dict/sentiment_words/HowNet_positive_review.txt","pos_1")
installDict("./Dict/sentiment_words/HowNet_positive_sentiment.txt","pos_2")
installDict("./Dict/sentiment_words/NTUSD_positive_simplified.txt","pos_3")
installDict("./Dict/sentiment_words/pos.txt","pos_4")
installDict("./Dict/sentiment_words/HowNet_negative_review.txt","neg_1")
installDict("./Dict/sentiment_words/HowNet_negative_sentiment.txt","neg_2")
installDict("./Dict/sentiment_words/NTUSD_negative_simplified.txt","neg_3")
installDict("./Dict/sentiment_words/neg.txt","neg_4")

# load dictionary of polarity words
pos_1 <- readLines("./Dict/sentiment_words/NTUSD_positive_simplified.txt", encoding='UTF-8')
pos_2 <- readLines("./Dict/sentiment_words/HowNet_positive_review.txt", encoding='UTF-8')
pos_3 <- readLines("./Dict/sentiment_words/HowNet_positive_sentiment.txt",encoding='UTF-8')
pos_4 <- readLines("./Dict/sentiment_words/pos.txt",encoding='UTF-8')
```

```
pos <- c(pos_1,pos_2,pos_3, pos_4)
pos <- pos[-which(duplicated(pos))]

neg_1 <- readLines("./Dict/sentiment_words/NTUSD_negative_simplified.txt", encoding='UTF-8')
neg_2 <- readLines("./Dict/sentiment_words/HowNet_negative_review.txt", encoding='UTF-8')
neg_3 <- readLines("./Dict/sentiment_words/HowNet_negative_sentiment.txt",encoding='UTF-8')
neg_4 <- readLines("./Dict/sentiment_words/neg.txt",encoding='UTF-8')
neg <- c(neg_1,neg_2,neg_3, neg_4)
neg <- neg[-which(duplicated(neg))]

# sentiment calculation
scores=rep(0,times=length(doc_CN))
score.pos=scores
score.neg=scores

for (j in 1:length(doc_CN)){

  # number of positive words in each row
  score.pos[j]<-sum(doc_CN[[j]] %in% pos)/length(doc_CN[[j]])
  # number of negative words in each row
  score.neg[j]<-sum(doc_CN[[j]] %in% neg)/length(doc_CN[[j]])
}

score.pos<-as.numeric(score.pos)
score.neg<-as.numeric(score.neg)
score.pos[is.nan(score.pos)] <- 0
score.neg[is.nan(score.neg)] <- 0

positive<- as.integer((score.pos-score.neg)>0)
negative<- as.integer((score.pos-score.neg)<0)

senti_label <- as.data.frame(cbind(positive, negative))
senti_score <- as.data.frame(cbind(score.pos, score.neg))

total_pos_score=total_neg_score=total_pos_label=total_neg_label=mean=mean1=rep(0,times=length(day
))

day=as.Date(seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"),"days"),format="%Y-%m-%d")

for(i in 1:length(day)){
  total_pos_label[i] <-
sum(positive[which(data$created_at==day[i])])/length(positive[which(data$created_at==day[i])])
  total_neg_label[i] <-
sum(negative[which(data$created_at==day[i])])/length(positive[which(data$created_at==day[i])])
}

for(i in 1:length(day)){
  total_pos_score[i] <-
sum(score.pos[which(data$created_at==day[i])])/length(score.pos[which(data$created_at==day[i])])
  total_neg_score[i] <-
sum(score.neg[which(data$created_at==day[i])])/length(score.neg[which(data$created_at==day[i])])
}

for(i in 1:length(day)){
  mean[i]<-(total_pos_score[i]-total_neg_score[i])
  mean1[i]<-total_pos_label[i]-total_neg_label[i]
}

# positive labeled ratio
plot(total_pos_label,type="b",xaxt='n',ylab="score",xlab=' ',main="Positive-Labeled Sentiment Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)

# negative labeled ratio
plot(total_neg_label,type="b",xaxt='n',ylab="score",xlab=' ',main="Negative-Labeled Sentiment Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)

# accumulative positive sentiment score
plot(total_pos_score,type="b",xaxt='n',ylab="score",xlab=' ',main="Accumulative Positive Sentiment
Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)

# accumulative negative sentiment score
plot(total_neg_score,type="b",xaxt='n',ylab="score",xlab=' ',main="Accumulative Negative Sentiment
Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)

# polarity mean
plot(mean,type="b",xaxt='n',ylab="sentiment score", xlab=' ',main="Polarity-Labeled Mean Sentiment
Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)

# sentiment score mean
plot(mean,type="b",xaxt='n',ylab="score",xlab=' ',main="Accumulative Mean Sentiment Score")
axis(1, at = seq(1,89,1),
     labels = seq(as.POSIXct("2012-08-01"),as.POSIXct("2012-10-28"), "days"),
     cex.lab = 1, las = 2,cex.axis=0.7)
```