INTEGRATING CHINESE DATA FROM SINA WEIBO TO THE LITMUS LANDSLIDE DETECTION SYSTEM

A Thesis
Presented to
The Academic Faculty

by

Jiateng Xie

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science with the
Research Option in the
School of Computer Science

Georgia Institute of Technology
May 2016

INTEGRATING CHINESE DATA FROM SINA WEIBO TO THE LITMUS LANDSLIDE DETECTION SYSTEM

Approved by:

Dr. Calton Pu, Advisor
School of Computer Science
Georgia Institute of Technology

Dr. Ling Liu
School of Computer Science
Georgia Institute of Technology

Date Approved: 05/04/2016

[To the students of the Georgia Institute of Technology]

ACKNOWLEDGEMENTS

I would like to especially thank my mom, my dad, my girlfriend, and my best friends, without whose support and guidance I would not be here. I would also like to thank my graduate student advisor Aibek Musaev and my thesis advisor Dr. Calton Pu for their invaluable guidance in both my research and my career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

NER	Named Entity Recognition
USGS	United States Geological Survey
SVM	Support Vector Machine

SUMMARY

The detection of landslides has been a challenging problem for researchers since there are no dedicated physical sensors to detect landslides. LITMUS is a landslide detection system based on information from both social media platforms and physical sensors. It does have its own limitations, however, because it only supports English data. We propose to integrate the Chinese data from Sina Weibo to the LITMUS landslide detection system to extend its service. The Chinese LITMUS system pipeline starts off by collecting data from Sina Weibo using a web crawler. Then, it applies a few filtering techniques to tackle part of the noise that comes with the dataset. Subsequently, the system uses a combination of Named Entity Recognition (NER)-based and gazetteer-based approach to geo-tag the data items. The data-items that contain the same location entity are grouped to one cluster, which represents a candidate event. The system then classifies each data item to identify the remaining noise by using Word2Vec and Support Vector Machine (SVM). Lastly, the system makes a decision based on the majority label assigned to each cluster by the classifier as to whether or not a candidate event is an actual landslide event. Through our experiments, we show that the classification component of the system achieves about 0.96 in precision, recall and F-measure using the evaluation dataset, and that the system is able to detect a large number of landslides in China.

CHAPTER 1

INTRODUCTION

The detection and prediction of natural disasters, which can result in loss of lives and property damage, have always been a concern for modern society. The traditional methods of detecting natural disasters often rely on dedicated physical sensors, such as seismometers that are used to detect earthquakes. A landslide, however, is a kind of natural disaster that does not have dedicated physical sensors because of the fact that countless factors can cause a landslide. Apparently, the lack of physical sensors makes landslides a kind of disaster that is much harder to detect, which presents a challenge for researchers.

Figure 1. A Large Landslide in Shanxi Province

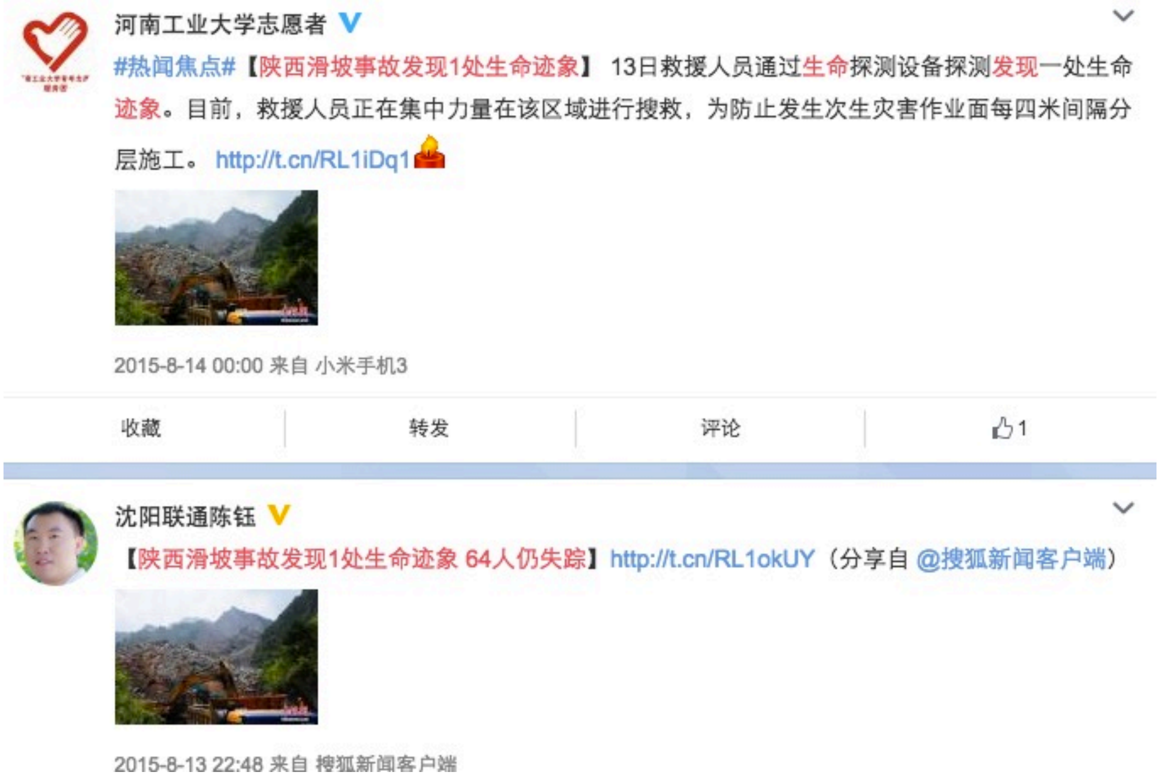
During recent years, social media platforms have experienced remarkable growth. For instance, Twitter has approximately 320 million monthly active users that post over 500 million tweets per day [3]. These platforms provide active communication channels during mass convergence and emergency events [6]. And because of the popularity of social media, not only emergency response agencies, but also regular users disseminate situation-sensitive information in safety-critical situations [11]. Reports of an event will usually appear in a social media platform before they appear in the columns of newspapers. With this phenomenon taken into account, it is believed that the data collected from social media platforms can contribute significantly to the detection of landslides.

A recent study first introduces the concept that each Twitter user can be treated as a social sensor, and then proposes a probabilistic spatiotemporal model based on data provided by these social sensors to detect real-time earthquakes [1]. It provides an excellent example of using information from social networks for event detection. Based upon a similar methodology, LITMUS is a landslide detection system that applies an integration of data from both social and physical sensors, such as Twitter, Instagram and United States Geological Survey (USGS) [4], [19]. Nonetheless, since LITMUS only supports English language data from social media platforms, it misses a vast amount of valuable information from non-English-speaking countries. The language barrier and the associated problems in data processing are the major difficulties LITMUS faces to extend its service. Having a wall that blocks people from using popular social networks like Twitter, and using a language, Chinese, that is largely different from English, China is the archetype of such a difficulty.

To date, Sina Weibo is the leading social media platform in China, generating hundreds of millions of posts per day. Integrating the Chinese data from Sina Weibo into LITMUS would not only help it detect more landslides in China, but also serve as a paradigm for the system to integrate other languages.

Research Challenges

The most straightforward method to collect data from social networks is to use their public APIs, and in this case, Sina Weibo does provide a polling API that returns the latest posts [8]. Nonetheless, upon further inspection, it is discovered that not only the polling API does not provide a search method so that the posts returned will all contain specific keywords, but Sina Weibo also restricts the number of requests allowed per hour and states that it is against their policy to use computer programs to repetitively collect data from their platforms. As a result, very few data regarding landslides can be collected using this method. A recent study suggests the use of a crawler to collect data from Sina Weibo [5]. Since Sina Weibo provides for its users a search interface, which can return up to 50 pages of the most recent posts that contain the search keywords, we decided to develop a web crawler that is able to parse the HTML documents returned by the search interface to collect data.



河南工业大学志愿者 V

#热闻焦点#【陕西滑坡事故发现1处生命迹象】 13日救援人员通过生命探测设备探测发现一处生命迹象。目前，救援人员正在集中力量在该区域进行搜救，为防止发生次生灾害作业面每四米间隔分层施工。 http://t.cn/RL1iDq1

2015-8-14 00:00 来自 小米手机3

沈阳联通陈钰 V

【陕西滑坡事故发现1处生命迹象 64人仍失踪】http://t.cn/RL1okUY（分享自 @搜狐新闻客户端）

2015-8-13 22:48 来自 搜狐新闻客户端

Figure 2. Relevant Posts from Sina Weibo

The keywords being used are “滑坡” (landslide, decline), “泥石流” (mudslide), “塌方” (landslide, landslip or collapse), “崩塌” (landslide or collapse) and “山崩” (landslide). See Figure 2 for examples of relevant posts returned from Sina Weibo’s search interface about a landslide in Shanxi Province in August 2015.

Since each landslide event has its own spatiotemporal point, a geo-location is needed to describe one such event. Currently, most social media platforms allow users to disclose their geo-information when they post. However, users rarely use this functionality. One study of Twitter shows that less than 0.42% of tweets contain geo-location [2]. Data from Sina Weibo showed similar results. Based on a dataset we collected using Sina Weibo’s polling API, only 20,673 out of 629,609 posts contain geo-location, which is less than 3.3%. Therefore, in order to geo-tag the events, we need to look for mentions of places within the text of the post. A few past studies have proposed methods to extract the location entity [7], [9]. For the English data, LITMUS uses a Named Entity Recognition (NER)-based approach to extract the location entity [4]. However, the NER-based approach for Chinese data is not as accurate, because it will miss some county-level locations and misidentify some words as locations. To increase accuracy without adding too much computing overhead, we use a combination of NER-based and gazetteer-based approach, with the gazetteer only containing Chinese geo-political entities. The location entity extracted by this approach will be subsequently passed to Google Geocoding API to obtain the geographic coordinates of the location [10]. Note that the geo-tagging process can also be considered as a filtering process, since only data items that contain location entities are useful for landslide detection.

Despite the large number of data collected, many of them are considered noisy information irrelevant to a landslide event. There are two categories the noise can fit into: old pieces of news and descriptions of places that are posted repetitively by spammers or zombie fans, or posts from regular users in which the keywords’ meanings are not relevant to landslide as a natural disaster. Below are a few examples of the irrelevant data items from the two categories:

- “【泥石流冲进火车车厢掩埋乘客 武汉铁警徒手刨泥救人】台风“尤特”带来的强降雨导致隧道塌方，击中行驶中的k624次列车，2名成年男子瞬间被埋没，另有2名三岁小孩被飞溅的玻璃碴划伤。武汉铁路公安局乘警及时组织列车工作人员展开施救，徒手刨泥，及时将被埋旅客救出。” This post is an old piece of news regarding a mudslide in August 2013, and it was posted for 89 times by spammers and zombie fans in July 2015.
- “【西藏旅游小贴士】1.林芝东南部的波密、察隅和墨脱一带，5-9月大量降雨易引发山体滑坡及泥石流、塌方等，影响旅行。2.林芝地区很少ATM机，多带现金。3.巴松措每家饭店都有巴河鱼料理，但价格昂贵，建议自带食物。4.注意尊重当地各民族的信仰与风俗。” This post is a description of Tibet, and it was posted for 111 times by spammers and zombie fans in July 2015.
- “生买不起房，死买不起墓，半生不死住不起院，这才是真的悲哀。道德崩溃不是百姓商人的责任，而应多问几个为什么？为什么出现大面积道德滑坡，谁引领道德标杆？当政府假话连篇、媒体谎言不断、官员贪腐泛滥，就该知道问题核心所在。中国任何人可以谈道德，唯有政府不能谈，也没有资格谈” In this post, the keyword “滑坡” (landslide) means decline in the phrase “道德滑坡”, moral decline.
- “人生不是止水，总会出现许多出乎意料之事。泰山崩于前而色不变，风波骤起而泰然处之，就显得很重要。转危为安往往需要高超的心智，也需要好的心态。多思索少激动，多仁爱少仇恨，人生才变得更加美丽。” In this post, the keyword “山崩” (landslide) is used as part of the aphorism “泰山崩于前而色不变”, and in fact, the character “山” is in the word “泰山” and the character “崩” is a word

itself, but since Chinese sentences are not word-segmented, they form the word "山崩" in the sentence while the two characters are in different words.

- "没信任的感情早晚要崩塌". In this post, the keyword "崩塌" is used as a verb that means collapse in the phrase "感情崩塌", relationship collapse.

To filter out the noisy information posted by the spammers and zombie fans, we maintain an off-line database of such spam posts and update it periodically, because there is only a small set of distinct posts. As for the data items that are noisy due to the fact that the keywords are polysemous, a common approach is to use a list of stop words or phrases to filter out the data items. For example, "精神崩塌" (spirit collapse), "股市滑坡" (stock market collapse), and "公德滑坡" (social morality collapse) are some of the stop words that are being used. However, only a part of the irrelevant items can be identified using this technique, and many are left unfiltered. A recent study has shown that text classification can be applied effectively in labeling data items as either relevant or irrelevant to natural disasters [1]. Also, LITMUS adopts a similar technique based on Explicit Semantic Analysis to classify texts [15]. For the Chinese LITMUS system, a text classification approach based on Word2Vec is used to label the data items. It is important to note that the noisy data items identified by text classification are labeled as irrelevant items instead of being filtered out, because they will be useful later in determining if there is a landslide event in a specific location.

The rest of the paper is organized as follows. The next chapter presents an overview of the Chinese LITMUS system. Chapter 3 describes each component of the system in detail. In Chapter 4, we discuss our experiments and its outcomes. Chapter 5 and 6 discusses related work and our future work. The conclusion is given in Chapter 7.

CHAPTER 2

SYSTEM OVERVIEW

The Chinese LITMUS system first collects and subsequently processes its data before it outputs the detected landslide events – please refer to Figure 3 for an overview of the system.

The system starts off by collecting data from Sina Weibo using a web crawler that parses the HTML documents returned from Sina Weibo's search interface.

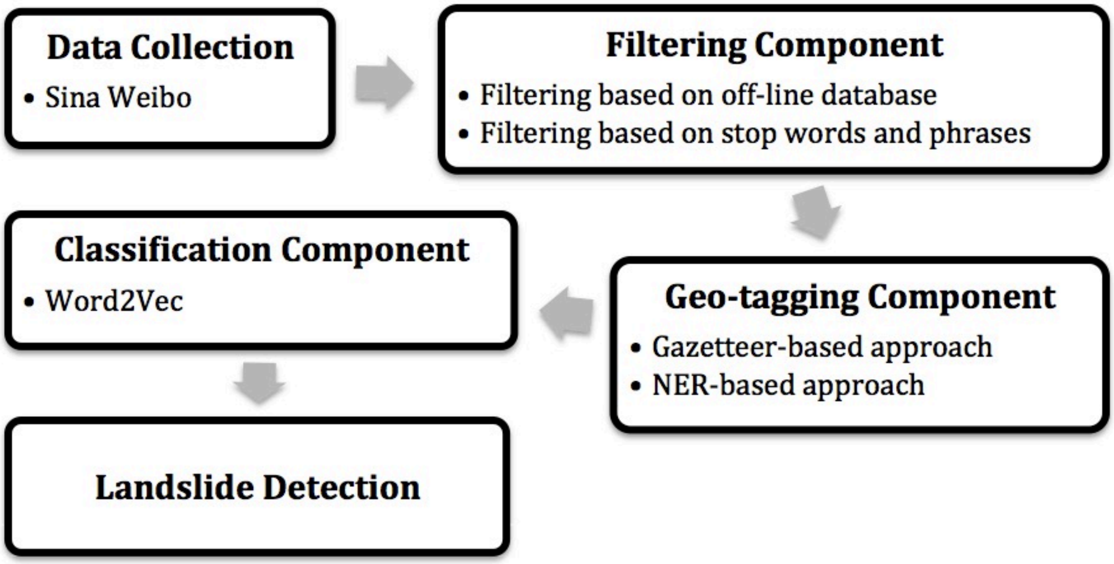


Figure 3. System Pipeline

Since the collected data are usually noisy, they are passed to the filtering component, which performs two filtering steps. Firstly, the component compares the data items with those that are stored in the off-line database. The items that match any entry from the off-line database get filtered out. Secondly, the component filters out the items that contain any stop words or phrases. The remaining data items are then passed to the geo-tagging component, which tries to extract a location entity from them using a combination of gazetteer-based and NER-based approach. The data items that do not contain any mentions of geographical names are filtered out. Subsequently, the data items that contain the same location entity are grouped into one cluster. After this step, many clusters are generated, and each cluster represents a candidate event. The classification component then uses a model built from the training dataset to label each data item as either relevant or irrelevant to landslide events. The detection component tries to determine if a cluster actually describes a landslide event by looking at the labels assigned to the data items that belong to the cluster. Specifically, every cluster whose majority label is relevant is treated as a landslide event, and every cluster whose majority label is irrelevant is not. Lastly, to confirm the results, we manually go through the data items of each cluster and try to decide if there is an actual landslide.

CHAPTER 3

IMPLEMENTATION DETAILS

Data Collection

The Chinese LITMUS system uses five keywords, namely “滑坡” (landslide), “泥石流” (mudslide), “塌方” (landslide, landslip or collapse), “崩塌” (landslide or collapse) and “山崩” (landslide), to collect data from Sina Weibo. In order to do so, the web crawler parses the HTML documents returned from http://s.weibo.com, search interface provided by Sina Weibo, and extracts the data items. Note that the search interface is able to return up to 50 pages, with each page having around 20 posts, of the latest posts that contain the search phrase. Also, it provides the users the option to choose whether or not to use the filter provided by Sina Weibo. If the filter option is on, the posts returned will mostly come from verified users.

To form URLs used to gather the HTML documents, the keyword needs to be URL-encoded (percent-encoded) twice. The parameter “page” indicates the page number, and the parameter “nodup” indicates whether or not to user the filter provided by Sina Weibo, with 1 being true and 0 being false. For instance, the URL for the 5th page of search results of the keyword “滑坡” without filtering will be http://s.weibo.com/weibo/%25E6%25BB%2591%25E5%259D%25A1&page=5&nodup=0.

Within the returned HTML document, all the posts are located in a script block starting with “<script>STK && STK.pageletM && STK.pageletM.view”. There are a few such blocks in one document, but only one block is of our interest. After scanning, it is discovered that the text of a post is wrapped in a variable called “comment_txt”. Hence, we select the block that contains the string “comment_txt”.

Field	Description
UserID	The unique identification number of the user.
UserName	The screen name of the user.
IsVerified	Whether or not the user is officially verified by Sina Weibo.
CreatedAt	The time when the post is created.
MID	The unique identification number of the post.
Text	The content of the post.

Table 1. Fields of A Data Item

Before applying pattern matching on the selected block, however, it is important to note that the Chinese characters for each post are encoded in UTF-8 format in the document. Fortunately, the jsoup Java HTML parser library can handle the decoding of the UTF-8 hexes automatically, when the block is constructed as a JSON object and passed into the jsoup parser. With the Chinese characters now retrieved, we use a set of regular expressions to extract the data items from the block. See Table 1 for the fields of a data item.

Despite being able to collect data from Sina Weibo, the crawler is not fully automated due to two reasons. Firstly, Sina Weibo requires the user to log in to use the search interface. Otherwise, only the first page of the results can be accessed. Secondly, even if the user has logged in, Sina Weibo has an anti-crawling mechanism that will prompt the user to type in a captcha when the user has accessed around 40 pages of results. A fast approach to tackle the login problem is to manually log in first via a browser, obtain the cookie, and then pass the cookie to the network client. Although this approach can be automated, the software still needs to be run manually, since our system currently cannot automatically handle the captcha, which needs to be typed in about every 40 pages of data collected.

Filtering Component

As previously mentioned, one type of noise in the dataset is the old pieces of news and descriptions of places that are posted repetitively by spammers or zombie fans. These users are usually bots created by people for commercial purposes. For instance, zombie fans are accounts that are created just to increase the number of followers of some users and they can be purchased online. These accounts post trashy information repetitively in order to act like normal users so that they can avoid being identified by Sina Weibo as spammers or zombie fans. Based on our observation, each of such posts is usually duplicated by different bots many times, and the total number of different posts is small, so it is easy to collect a sample post for each. Therefore, in order to effectively filter out this type of noise, we maintain an off-line database that stores each different post, and update the database periodically. If a post is found to be on the database, it will be marked as irrelevant and filtered out subsequently. Currently, the off-line database contains 113 entries.

For the second type of noise, we initially use a common approach, stop words and phrases, to filter them out. It is important to point out that, based on our observation, two of the keywords used to collect data, namely "山崩" (landslide) and "崩塌" (landslide or collapse), are hardly ever used to describe a landslide event. "崩塌" is mostly used by people to express their emotion, such as "我内心崩塌了" (my heart collapsed), while "山崩" mostly happens to be part of the aphorism "泰山崩于前而色不变". Since they are rarely used to describe any landslide events, they are currently also used as stop words. However, because these two words do mean landslide in Chinese and, in the future, people may start using them to describe landslide events, they are still being used as keywords to collect data.

Despite the fact that using stop words and phrases is able to filter out much noise, the keywords that have meanings other than landslide can be combined with too many words to form phrases in Chinese so that it is not realistic to pick all such phrases. For example, the keyword “滑坡”, when it means decline, can be used with “股市” (stock market), “市值” (market value), “道德” (morality), and so on. Therefore, in order to identify the noise comprehensively, we apply a text classification approach in addition to stop words and phrases, which will be described in detail in a separate section in this chapter.

Geo-tagging Component

In order to extract geographical locations from posts, we need to look for mentions of geographical names within the texts. Since Chinese, unlike languages like English, is standardly written without spaces between words, each post needs to be word-segmented before it can be analyzed. We use Stanford Word Segmenter, which employs a conditional random field (CRF)-based model, to perform word-segmentation on the Chinese posts [13]. The benefit of segmentation is that we can avoid cases in which two neighboring characters from two different words in a sentence may be confused for a geographical name.

After segmentation, initially, we apply a Named Entity Recognition (NER) technique to identify the location entities mentioned in the texts. Specifically, we use the Stanford Named Entity Recognizer, which can run on the word-segmented Chinese texts using a model built from the Ontonotes Chinese named entity data, and label the words or sequence of words as one of the five classes, namely person, geo-political entity, miscellaneous, organization, and location [14]. We are interested in the geo-political entities and location entities that get extracted. After some experiments, however, it is discovered that the NER library will miss some county-level locations and misidentify some words as location entities. Below is an example that indicates a landslide event in Ziyan County in Ankang of Shanxi Province, but the NER library fails to identify the location entity.

- 【开展防汛排查】6 月 29 日，紫阳县红椿镇强降雨引发了山体滑坡和泥石流，为避免发生不安全事故，红椿派出所与交警中队民警冒雨开展重点路段巡查，疏导交通，排查险情。目前，共排查险情 3 处，救助因山体落石被砸的伤员 1 名。@安康警务

To tackle this problem, we employ a gazetteer-based approach to complement the NER approach. Particularly, the gazetteer contains about 2400 entries, including geographical names of every county-level and above place in China. Note that, contrary to the notion in the United States, a county is smaller than a city in China. The main reason for using only Chinese geographical names is that, based on our understanding of the dataset, the landslide events described are largely indigenous to China. Since the NER library is able to extract the location entities around the world, a larger gazetteer that contains worldwide geographical names is not used, due to the fact that the computing cost will be increased greatly.

To use the gazetteer, we search each segmented word of the sentence in the gazetteer to see if there is a match. Typically, each entry in the gazetteer is formed by the name of the place and its administrative level. For instance, “北京市” (Beijing), is formed by “北京” (Beijing), its geographical name, and “市” (city), its administrative level. Note that “北京市” actually means Beijing City in Chinese, although when translated to English, it is commonly referred to as just Beijing. As the administrative level of an entry in the gazetteer is usually omitted when it is used in Chinese, the last character of an entry, the administrative level, does not have to be matched. Either “北京” or “北京市” is considered a match to the “北京市” entry in the gazetteer, for example. However, this abbreviation of geographical names does introduce a problem. Some common words are used as geographical names, such as “资源县” (Ziyuan County), where the name “资源” means resources in Chinese. For common words like this, only a full match is considered a match, meaning that the administrative level has to be included in the search word.

After the location entities are extracted, the one that is closest to the keywords is selected for each post. If there are multiple keywords in one post, we use the average index of the keywords. If one location is mentioned multiple times in one post, we use the average of the distances from the index of each mention to the average index of all the keywords.

Subsequently, each post that has the same location entity is grouped into one cluster, and the cluster's location entity is passed as a parameter to Google Geocoding API to get the corresponding geographic coordinates and formatted geographic location for the cluster. Since each cluster now represents a candidate event, we need to estimate the location of landslide events based on the geographical information returned from Google. Our system, like LITMUS, applies a cell-based approach, which regards the surface of earth as a grid of cells [4]. Clusters are inserted to cells based on their geographical coordinates. The size of the cell is important, since when a cell is too large, multiple events may be inserted to the same cell, but when a cell is small, same event may be inserted to different cells so that they will end up being reported multiple times. Currently, a 2.5-minute grid is adopted, which means that each cell is 2.5 minute in latitude by 2.5 minute in longitude. To insert a cluster to its corresponding cell based on latitude (N) and longitude (E), the following formulas are used:

- $\text{row} = (90^\circ + N)/(2.5'/60') = (90^\circ + N) * 24$
- $\text{column} = (180^\circ + E)/(2.5'/60') = (180^\circ + E) * 24$

For instance, "雅安市" (Ya'an), whose geographical coordinates are N = 29.980537 and E = 103.013261, will be inserted to the cell (2879, 6792).

Classification

As previously mentioned, in addition to using stop words and phrases, a text classification technique is used to identify the noisy data items. In particular, we use Google's Word2Vec, a neural network model that takes a text corpus as input and outputs vectors for each unique word in the corpus [15]. The generated vector representations of words explicitly show many linguistic regularities and patterns [16]. For example, the words that are semantically close have been shown to have high cosine distance. To observe such strong regularities, a large text corpus is needed. We use a Chinese text corpus published by Sogou Lab that consists of a large number of all aspects of news as the input [17], [18]. Before the corpus can be used to train the model, it needs to be word-segmented. The resulting total word count of the model is 906,722,443 and its vocab size is 1,336,975.

For each word-segmented post, we extract from the built model the corresponding vector for each word, and compute a centroid vector for the post. Each post now has a corresponding centroid vector, and this vector is used as the feature for classification purpose. We select the state-of-the-art algorithm Support Vector Machine (SVM) to build a classifier using a manually labeled training dataset. The classifier is then used to classify each data item as either relevant or irrelevant to landslide events.

Landslide Detection

After the previous steps, each data item in the cluster has an associated label determined by the classifier as either relevant or irrelevant to landslide events. The system determines if a candidate event is an actual event using the following rule: each cluster has an original score of 0, and for each item in the cluster, if the assigned label is irrelevant, it decreases the score by 1, and if relevant, it increases the score by 1; if the final score of a cluster is positive, it will be treated as an actual event, if negative, it won't, and if it is zero, the cluster will be ignored.

CHAPTER 4

EXPERIMENTAL RESULTS

We use a dataset collected from Sina Weibo to demonstrate the effectiveness of our system. The process begins with running the dataset through the filtering and geo-tagging components. The outcome is then used to present an evaluation of the performance of the classification component. Based on the label assigned by the classifier, we show the landslide detection results and compare them with authoritative sources.

Dataset Description

The dataset contains data items collected from Sina Weibo during the period from July 2015 to November 2015 using the five keywords, “滑坡” (landslide, decline), “泥石流” (mudslide), “塌方” (landslide, landslip or collapse), “崩塌” (landslide or collapse) and “山崩” (landslide). To run experiments, we use the text and MID fields of the data item. Table 2 shows the total number of items and the number of items for each month during this period.

	July	**August**	**September**	**October**	**November**	**Total**
Number	71,404	110,744	52,517	47,761	40,972	323,398

Table 2. Dataset Overview

Filtering and Geo-tagging

Since the data items are noisy and lack geo-locations, we run them through both the filtering and geo-tagging components. Currently, the off-line database contains 121 sample entries of old news and descriptions of places posted by spammers and zombie fans. The results of this process are shown in Table 3. Each column shows the remaining number of posts after the corresponding step.

	Original	Filtering based on off-line database	Filtering based on stop words and phrases	Geo-tagging
July	71,404	60,967	26,930	9,067
August	110,744	100,126	50,073	27,801
September	52,517	45,019	20,582	7,032
October	47,761	39,749	18,946	5,774
November	40,972	38,573	23,327	14,230
Total	323,398	284,432	139,858	63,904

Table 3. Filtering and Geo-tagging Results

As can be seen from the table, most number of data items gets filtered out by the stop words and phrases, followed by geo-tagging and off-line database. The reason is that, as described earlier, two of the keywords, “崩塌” and “山崩”, are also used as stop words, and more data are collected on the keyword “崩塌” than any other keywords. Based on the results, 45.7% of the data items that remain after the filtering process contain location entities. The data items that contain the same location entity are grouped into clusters, which are then mapped into cells based on its geographic coordinates returned from Google Geocoding API. The number of clusters and cells for each of the month are shown in Table 4.

	July	August	September	October	November
Number of clusters	265	291	147	118	100
Number of mapped cells	265	291	147	118	100

Table 4. Number of Clusters and Mapped Cells

Evaluation of Classification

The data items that contain geo-locations are used as the dataset for evaluation of the classification component. Particularly, we use the data items from July and August as the training dataset, and the data items from September, October and November as the evaluation dataset.

Before conducting experiments, the data items need to be labeled manually. To label items as relevant, we look for trustworthy sources to confirm the landslide events described by the posts. For example, USGS publishes a list of confirmed landslides each month [19]. Also, if a data item contains an URL, we check whether or not the linked content describes a landslide and the source is trustworthy. Lastly, we can search for the described events online and see if there is any news or trustworthy source that confirms it. To label items as irrelevant, we see if keywords in the item actually means landslide as a natural disaster. See Table 5 for the number of relevant and irrelevant items for the training dataset and the evaluation dataset.

	Training Dataset			**Evaluation Dataset**			
	July	**August**	**Total**	**September**	**October**	**November**	**Total**
Relevant	6,964	25,703	32,667	3,537	3,519	12,628	19,684
Irrelevant	2,103	2,098	4,201	3,495	2,255	1,602	7,352
Overall	9,067	27,801	36,868	7,032	5,774	14,230	27,036

Table 5. Relevant and Irrelevant Items

After all the items are labeled, we perform word-segmentation on the post, convert each word within the post to a vector using the built Word2Vec model and compute a centroid vector for each item, which is the feature to be used for classification. Then, we use the Support Vector Machine (SVM) algorithm implemented by Weka, a Java open source collection of machine learning algorithms, to build a classifier from the training dataset [20]. The classifier is then used to classify each data item in the evaluation dataset as either relevant or irrelevant. We compare the label manually assigned to each item to the label assigned by the classifier, and measure the performance of the classification component.

The metrics used to measure the performance are precision, recall and F-measure (F1 score). The formulas of these three measures depend on the relationships between the manually assigned label and the label predicted by the classifier, which are true-positive, true negative, false positive and false negative. Refer to Table 6 for the definitions.

		Predicted Label	
		Relevant	**Irrelevant**
Actual Label	**Relevant**	True Positive	False Negative
	Irrelevant	False Positive	True Negative

Table 6. Definitions for Relationships between Labels

Based on the definitions, the formulas for the three metrics are:

$$(1)\ Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$(2)\ Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$(3)\ F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

We calculated these 3 metrics for the 3 months both separately and as a whole based on the results given by the classifier and plot them on Figure 4. As can be indicated by the graph, the classification component is able to achieve very good performance, with the overall precision, recall and F-measure all being around 0.96.

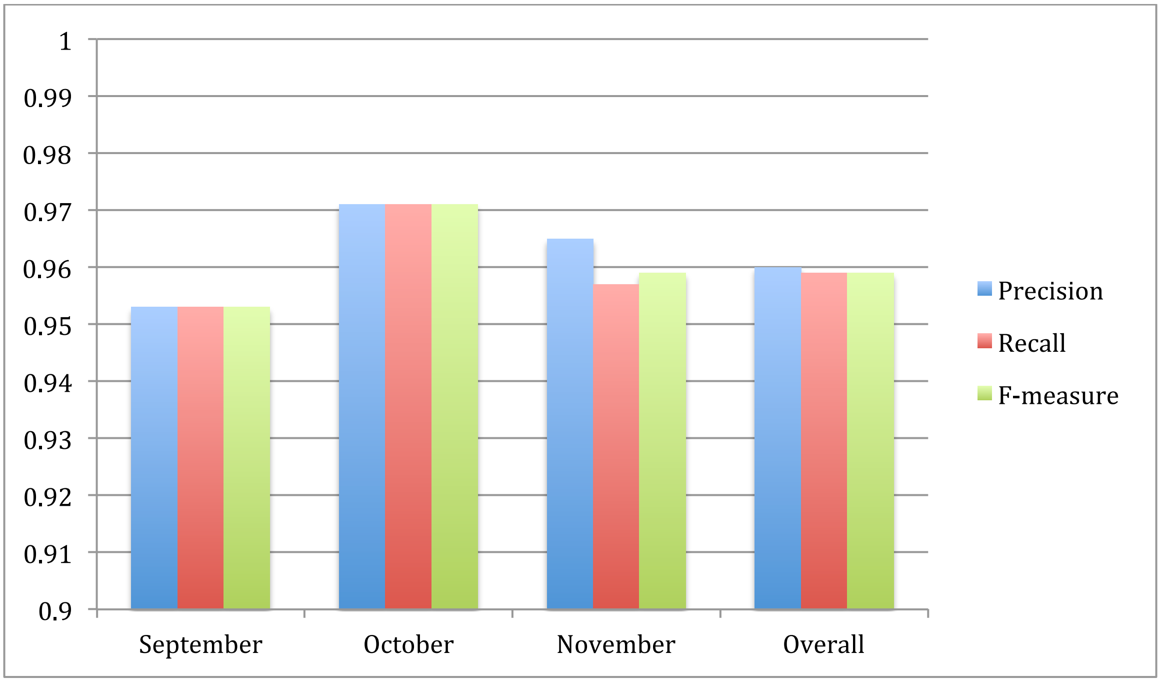


Figure 4. Results for the Classification Component

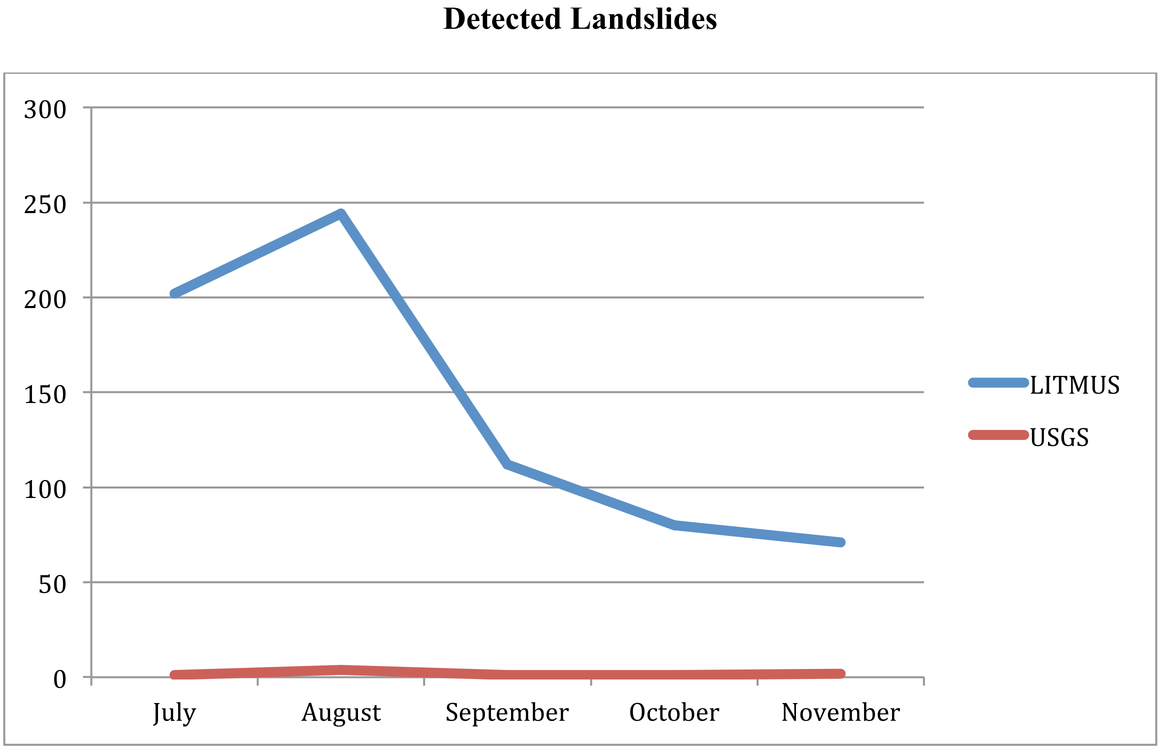


Figure 5. Landslide Detection Results

Based on the predicted labels assigned to all the data items, we compute a score as previously described for each cell based on the labels of the data items that belong to the cell. We mark the cells that have positive scores as real landslide events, and compare our results to those reported by the USGS for each month. Since the Chinese data from Sina Weibo mostly concern only landslides that happen in China, we only select the reported landslides in China. Figure 5 shows that the Chinese LITMUS system is not only able to detect all the landslides in China reported by USGS, but also to detect many landslides in China that are not reported by USGS. Over the five-month period, LITMUS discovers 700 more landslides in China than USGS, averaging 140 per month. These results indicate that integrating the Chinese data from Sina Weibo to the LITMUS landslide detection system can indeed help it detect more landslides in China.

CHAPTER 5

RELATED WORK

Social media platforms have been shown to be useful in detecting natural disaster events. Imran et al found out that social media platforms provided active communication channels during mass emergency and convergence events [6]. Moreover, Vieweg et al. discovered that not only emergency response agencies, but also regular users disseminate situation-sensitive information in safety-critical situations [11]. These works demonstrated to the researchers the potential usage of information from social media platforms in detecting natural disaster events. Taking advantage of this observation, Sakaki et al. regarded each Twitter user as a social sensor, and proposed a spatiotemporal model to detect earthquakes in real-time in Japan by using data collected from the social sensors [1]. In addition, Musaev et al. used a combination of data from both social information services and physical information sources to detect landslides throughout the world [4].

A few studies in the past examined the user behavior in Sina Weibo. Lin et al used a web crawler to collect data from Sina Weibo and analyzed the spammer's behavior [5]. F. Yang extracted a large set of features from posts collected from Sina Weibo and trained a classifier based on the features to automatically detect rumors [23].

Previous research has proposed methods to geo-tag data items from social media platforms. Cheng et al. found that less than 0.42% of the tweets contained geo-locations, and hence proposed a probabilistic framework to estimate a Twitter user’s city-level location [2]. Watanabe et al. proposed to assign geo-location information to non-geo-tagged tweets creating a geo-name database [7]. R. Lee and K. Sumiya learnt that cell-based approach was efficient in clustering tweets to detect geo-social events [9].

Text classification has been shown to be efficient in classifying data items from social media. Musaev et al. proposed reduced Explicit Semantic Analysis to label data items as either relevant or irrelevant to landslide events [12]. Similarly, Sakaki et al. extracted features from tweets to be used by SVM to classify tweets as either relevant or irrelevant to earthquakes.

CHAPTER 6

FUTURE WORK

Since the Sina Weibo crawler we develop is not fully automated, we intend to firstly automate the software to mimic browser login, and then train a model that can automatically read the captcha using a large set of captchas from Sina Weibo.

Also, in order to describe a landslide event, we need not only the geographic location, but also the temporal information. For our future work, we wish to build a time-tagging component that is able to extract temporal entities from the data items and then assign a temporal tag to each cluster.

Lastly, our system currently cannot distinguish rumors or predictions from actual events. If a post falsely reports an event or predicts an event, it will be treated by the classifier as a relevant item. Hence, we plan to add more features or train the classifier differently to identify rumors and predictions.

CHAPTER 7

CONCLUSION

Throughout the thesis, we present both an overview and a detailed description of how to integrate the Chinese data from Sina Weibo to the LITMUS landslide detection systems. The system firstly tackles the challenge associated with data collection by introducing a web crawler that parses the HTML documents returned from the Sina Weibo search interface. Since the data items collected contain two types of noise, irrelevant information posted by zombie fans and spammers and the posts from regular users in which the keywords have irrelevant meanings, the paper proposes a few approaches to solve the problem. Firstly, we implement an off-line database that stores samples of the irrelevant information by zombie fans and spammers. Then, we filter out the noise by using stop words and phrases. To further identify the noise, we adopt a classification method based on Word2Vec and SVM. Through our experiments, this classification component shows excellent outcomes in terms of identifying the noise. Besides filtering, a geo-tagging approach that is based on a combination of gazetteer and NER is developed. Applying the classification algorithm on the data items that can be geo-tagged, we successfully detect a handful of landslides in China during the period from July 2015 to November 2015. The system is able to identify many more landslides than can be reported by USGS during the same period of time. In general, we demonstrate that the Chinese LITMUS system can convey good results in regards to processing Chinese data and detecting landslides in China.

REFERENCES

[1] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real-time event detection by social sensors,” in WWW, 2010.

[2] Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: a content-based approach to geo-locating twitter users,” in CIKM, 2010.

[3] Twitter Inc., https://about.twitter.com/company, accessed on 04/10/2016.

[4] A. Musaev, D. Wang, and C. Pu, “LITMUS: a Multi-Service Composition System for Landslide Detection,” IEEE Transactions on Services Computing, vol. 8, no. 5, 2015.

[5] C. Lin, J. He, Y. Zhou, X. Yang, K. Chen, and L. Song, “Analysis and Identification of Spamming Behaviors in Sina Weibo Microblog,” in Proceedings of the 7th Workshop on Social Network Mining and Analysis, ser. SNAKDD, Chicago, Illinois, 2013.

[6] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, “Processing social media messages in mass emergency: A survey,” ACM Comput. Surv., 47(4):67:1–67:38, June 2015.

[7] K. Watanabe, M. Ochi, M. Okabe, and R. Onai, “Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs,” in 20th ACM international conference on Information and knowledge management, 2011.

[8] Sina Corp, “Sina Weibo API”, http://open.weibo.com/wiki/API, accessed on 04/08/2016

[9] R. Lee and K. Sumiya, “Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection,” in 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, 2010.

[10] Google Inc., “The Google Geocoding API,” https://developers.google.com/maps/documentation/geocoding/, accessed on 04/05/2016.

[11] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. “Microblogging during two natural hazards events: What twitter may contribute to situational awareness,” CHI ’10, pages 1079–1088, New York, NY, USA, 2010. ACM.

[12] A. Musaev, D. Wang and C. Pu, “Toward a Real-time Service for Landslide Detection: Augmented Explicit Semantic Analysis and Clustering Composition Approaches,” in ICWS, 2015.

[13] H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning, “A Conditional Random Field Word Segmenter,” in 4th SIGHAN Workshop on Chinese Language Processing, 2005.

[14] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling,” in Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370, 2015.

[15] Google Inc., “Word2Vec”, https://code.google.com/archive/p/word2vec/, accessed on 04/10/2016.

[16] T. Mikolov, I. Stuskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality”, in NIPS, pp. 3111-3119, 2013.

[17] Sohu Inc., “SogouCS”, http://www.sogou.com/labs/dl/cs.html, accessed on 04/12/2016.

[18] C. Wang, M. Zhang, S. Ma, and L. Ru, “Automatic Online News Issue Construction in Web Environment”, in WWW, 2008.

[19] USGS, “United States Geological Survey agency: Earthquake activity feed from the United States Geological Survey agency,” http://earthquake.usgs.gov/earthquakes/, accessed on 3/18/2016

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The WEKA data mining software,” ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, 2009

[21] F. Yang, Y. Liu, X. Yu and M. Yang, “Automatic detection of rumor on Sina Weibo,” in Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, 2012.