An Improved Topic Detection Method for Chinese Microblog Based On Incremental Clustering

Gongshen Liu, Kui Meng, Jing Xie
School of Information Security, Shanghai Jiao Tong University, Shanghai, China
{lgshen, mengkui}@sjtu.edu.cn; xiejing1989@gmail.com

***Abstract*—A topic detection model based on hierarchical clustering for Chinese microblog is proposed in this paper. In order to minimize the impact of noise, we optimize the feature selection and weight computation method and use a new scoring method to filter out those topic-unrelated tweets. We also give an improved topic detection algorithm which uses a new vector distance calculation method and center vector updating method. It is shown by the experiment that this method can filter out majority of the topic-unrelated tweets and identify microblog topics accurately and efficiently. The study of microblog topic detection method can help users and service providers find out microblog hot topics dynamically.**

***Index Terms*—Incremental clustering; Microblog; topic detection**

I. Introduction

In recent years, microblogging services are more and more popular. And it is slowly moving into the mainstream. Unlike traditional blogging service, microblogging service is based on social network. People can share what they observe in their surroundings, information about events, their opinions about certain topics, and even their whereabouts updates with microblogging. Moreover, one can also follow other microbloggers to request their updates be delivered in real time. Microblogging also provides many other functions such as retweet or repost, commenting, etc. People can retweet microblog with the "//@username:" format. The "#hashtag#" format means the message is related to a particularly topic. In addition, microblogs can be written or received with a variety of computing devices, including cell phones. It has empowered people themselves to act as sensors or sources of data which could lead to important pieces of information. Moreover, various metadata can be extracted from the posts, such as location, time, and name. Aggregate analysis of these data includes different dimensions like space, time, theme, sentiment, network structure etc., and gives researchers an opportunity to understand social perceptions of people in the context of certain events of interest.

The target of topic detection is to classify the large amount of tweets according to their topic. Microblog topic detection differs from traditional topic detection in three aspects: firstly, microblogs or tweets are brief (typically 140 – 200 characters); secondly, tweet topics increase quickly; thirdly, there are too much topic noise involved in tweets.

Our research focus on hot tweet topic finding, related tweets clustering, and tweet topic keyword extraction. In this paper, we study data from Sina Weibo(one of the most visited microblogging website in China), and propose a topic detection method based on hierarchical clustering for Chinese microblog. Microblog topic detection can help users find out hot tweet topics more effectively, and help the providers improve their microblogging services.

II. Related Work

[1] proposes an algorithm for internet public opinion hotspot detection and analysis based on K-means and SVM. The authors use traditional vector space model in text expression, then perform K-means clustering and SVM classifiers on the documents to detect internet public opinion hotspot and classify following texts into corresponding classes. However, K-means is sensitive to noises, while there are many topic unrelated tweets in microblogs. This algorithm cannot reduce such noise influence. In fact, the algorithm is used for traditional websites, so it is not suitable for microblog. [2] studies characteristics of breaking news in Twitter and propose a method to collect, group, rank and track breaking news in Twitter. The authors index each tweet and grouped similar tweets together. They also propose a measurement to score each group and rank the groups according to the score. [3] proposes a detecting method for sudden topics on microblog based on the dynamic sliding window. The authors use windows to extract the information with potential sudden features, compute feature weight and build VSM with TF-IDF function which is combined with semantic. Then, they used improved Single-Pass clustering algorithm to generate the final clustering. This method is simple and accurate, but its miss rate is quite high. Furthermore, this method only focuses on finding sudden topics. [4] proposes a news topics mining approach from microblog. The author uses the word frequency and growing rate in the time window to generate a compound weight and extract news keywords, and then cluster keywords and detect news topic by incremental clustering method. But the experimental result shows that this method cannot get high precision

doi:10.4304/jsw.8.9.2313-2320

rate and high recall rate at the same time. The social network analysis layout algorithm propose in [11] is based on domain ontologies, which can help to find weibo topic. Besides the studies above, Sina Weibo also provides a hot topic list. But the topics are ranked simply by the number of tweets posted by users in specific micro-topic sites. It may involve in a lot noises, because those topic-unrelated tweets posted at these sites are included, while other topic-related tweets that are not posted at these sites are ignored.

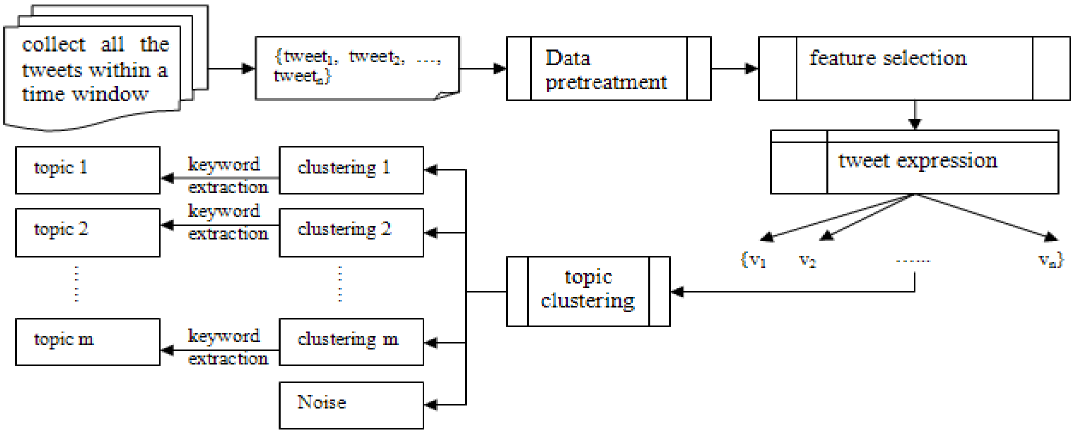

Figure 1. Procedure of the topic detection model

III. Topic Detection Model

The main task of topic detection is to recognize the beginning of any new topic from a large number of news, classify a news report by topic clusters, and establish new topic clusters when needed. Most topic detection algorithms are based on clustering algorithms. At first a vector space model is used to describe news report and topics, and then the similarities between different vectors are calculated to cluster those vectors based on some certain strategy.

For microblog, the goal of topic detection is to detect topics from large amount of tweets and classify those tweets into corresponding topic clusters while ignoring those topic-unrelated tweets (called *noise*) . Although traditional topic detection technology is quite mature, the topic detection method for microblog should pay more attention on following aspects: 1) the optimization of data pretreatment; 2) the optimization of feature selection; 3) the optimization of text representation model; 4) the optimization of topic clustering algorithm. In this paper, we propose a new topic detection model for Chinese microblog.

Figure 1 shows the basic procedure of the model. At first we collect all the tweets that are posted within a specified time window. These tweets are sent to the data pretreatment module. In this module, some useless information in the tweets is removed first. After word segmentation and POS (Part Of Speech) tagging, these tweets turn to feature selection module. Here, some topic representative words are selected as features, and we can calculate every tweet's feature weight and get its vector expression by vector space model. With this vector set, we use the topic clustering algorithm to get topic clusters.

A. Data Pretreatment

Data pretreatment is the first step for text processing. It transforms an original text string into term string or some specific symbol string. For each tweet in the collected dataset, there are two tasks in data pretreatment: useless information filtering, word segmentation and POS (Part Of Speech) tagging.

Filtering useless information means removing meaningless text or symbols in the tweet, such as some format related text content, url, special characteristics or emotion icons[5]. Sina Weibo provides some specific format to implement the function of retweeting, mentioning etc. For example, "@username" means mentioning a user in a tweet, "//@username:" is the format for retweeting. Such format related text should be removed at first, because usually they are not topic-related. Special characteristics, url and emotion icons are also topic-unrelated. They will lead to noise and influence word segmentation, so these texts should also be removed during data pretreatment. However, text in "#hashtag#" format should be reserved because they represent a topic directly. For example, after filtering, an original Chinese tweet "#请停止虐待儿童# 小孩太可怜了[愤怒] @小 Q http://t.cn/zluGttf" will be transformed to "请停止虐待儿童 小孩太可怜了".

TABLE I.

Part-of-Speech (POS) Tag Set

tag	POS	tag	POS
n	noun	nr	name
ns	location	nt	organization
nz	other proper nouns	t	time
s	place word	f	position word
v	verb	a	adjective
b	non-predicate adjective	z	status word
r	pronoun	m	numeral
q	quantifier	d	adverb
p	preposition	c	conjunction
u	particle	e	interjection
y	modal particle	o	onomatopoeia
h	prefix	k	postfix
x	string	w	punctuation

After filtering, it is turn to word segmentation and POS tagging. Here we use ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)[6] for this task. ICTCLAS is a Chinese grammar analysis system which is developed by Chinese academy of sciences. ICTCLAS is good at Chinese segmentation. It also supports POS tagging, named entity recognition and new word recognition. After word segmentation, we can get many pairs of "word / POS tagging". For example, after word segmentation and POS tagging the Chinese sentence "雷锋在我心中" will be transformed into the following pairs: "雷锋/nr 在/p 我/r 心中/s". Table 1 shows the POS tagging set we used.

B. Vector Representation

The commonly used text expression model includes LM (Language Model) and VSM[7] (Vector Space Model). Here we choose VSM to express a tweet. The basic idea of vector space model is to express a text by a space vector, where each dimension of the vector represents a feature and the value of each dimension represents the weight of the corresponding feature. For example, for text D, the vector space model of D is $D = D(t_1, w_1; t_2, w_2; ......; t_n, w_n;)$, where t_i is the ith feature and w_i is the weight of t_i, $1 \le i \le n$.

1) Feature Selection

Method based on document frequency, mutual information, or information gain can be used as feature selection method to extract the most representative features [8,9,12]. In this paper, we use the feature selection method based on document frequency.

Given the dataset of tweets in a time window, we use all the words obtained after data pretreatment as the initial feature space. Words with different POS contribute differently to the expression of a topic. Usually, the key words of a topic-related tweet including nouns, names, location, time, numbers, verbs and adjectives, are much more meaningful in topic representation than the others like particle, pronouns, prepositions and modal particle. To improve the efficiency of the algorithm, we only take nine kinds of words into consideration. They are nouns, names, location, time, numbers, verbs, adjectives, organization names and other proper nouns. Therefore, we remove all other words except these nine kinds of words from the feature space at first.

We choose two suitable thresholds, a higher one, and a lower one. For each remaining feature in the feature space, say t_i, we count the number of the tweets that exist the feature. When the number or the frequency, say $df(t_i)$, is lower or higher than the specific threshold, the feature item t_i will be removed from the feature space. Because too low frequency of a feature reflects it is not representative, while too high frequency reflects it is not distinguishable. We also exclude the feature item whose length is smaller than 2, because one character word are not so representative either. The feature selection method based on frequency is quite simple, and it can exclude noise, reduce feature dimension quickly and efficiently. All of these are helpful to improve the accuracy and efficiency of the algorithm. The items in the final feature space will be used as features in the vector space model.

Also, we keep a vector FV that shows each feature's times of occurrences, which will be used for the computation of distance between two vectors later in section III.C.2.

$$FV = (fv_1, fv_2, ..., fv_n),$$
$$fv_i = df(t_i) \times boost(t_i)$$
(1)

Where $df(t_i)$ is t_i's number of occurrence in the dataset and $boost(t_i)$ is a constant value according to the POS tag of t_i, $1 \le boost(t_i) \le 2$. $boost(t_i)$ can adjust the importance of word terms with different POS tag for topic detection. For example, noun, location, time and name contribute more to the topic representation than adjective and verbs.

2) Feature Weight

There are two main methods to calculate feature weight: Boolean weight and TF-IDF (term frequency-inverse document frequency) weight. Usually, TF-IDF measure is used for general text model representation. Because tweet is very short and there's little length difference for most tweets, TF is meaningless. On the other hand, IDF makes the lower frequency features in the total dataset have higher weight since it stresses those distinguishable words. In fact, for topic detection, the words with higher frequency are more likely to be a topic keyword. That is to say, these words contribute more to topic representation. So TF-IDF measure is not appropriate for microblog topic detection.

Here, we use Boolean weight to compute the tweet feature weight. The formula is as follows:

$$w_{ij} = \begin{cases} 1, & tf_{ij} > 0 \\ 0, & otherwise \end{cases}$$
(2)

Where tf_{ij} represent the frequency of feature t_i in D_j.

C. Topic Detection

1) Noise Exclusion

There exist a large amount of tweets which are topic- unrelated. Such tweets not only lead to a lot of noise, but also influence the efficiency of the clustering method. So we should remove these topic-unrelated tweets as early as possible.

After feature selection and weights computation, for each tweet in the dataset, we can get its vector expression, and calculate its topic-related likelihood.

$$
\begin{aligned}
D_j &= (t_1, w_{1D_j}; t_2, w_{2D_j}; ......; t_n, w_{nD_j};) \\
Score(D_j) &= (w_{1D_j} \quad w_{2D_j} \quad \dots \quad w_{nD_j}) \cdot FV^T \\
&= (w_{1D_j} \quad w_{2D_j} \quad \dots \quad w_{nD_j}) \cdot \begin{pmatrix} fv_1 \\ fv_2 \\ \dots \\ fv_n \end{pmatrix} \\
&= \sum_{i=1}^{n} (w_{iD_j} \times fv_i)
\end{aligned}
\tag{3}
$$

fv_i corresponds to the contribution of t_i made in topic representing. A larger fv_i means t_i is more likely to be a topic keyword. According to the formula above, when a tweet contains the feature items with high frequency, it is very likely topic-related, and the score will be high. When a tweet does not contain or only contains the feature items with low frequency, its score will be low, and it might be topic-unrelated.

So, if we choose suitable score threshold, we can cut those topic-unrelated tweets with low score and reduce the noise in clustering algorithm.

2) Topic Clustering Algorithm

Our topic detection algorithm is based on incremental clustering. We use the remained topic-related tweets as the input data set for our algorithm.

The topic detection algorithm proposed in this paper is as follows:

```
Input: set of tweets D
Output: set of topic clusters C
steps：
(1)  for each tweet $D_j = (t_1, w_{1D_j}; t_2, w_{2D_j}; ......; t_n, w_{nD_j};)$ in D
(2)    if $D_j$ is already clustered in C
(3)      go to step (1) and turn to the next tweet
(4)    set $V_{center} = (w_{1D_j}, w_{2D_j}, ..., w_{nD_j})$
(5)    for each tweet $D'_j$ in D which is not already clustered in C
(6)      if $distance(V_{center}, D'_j)$ < 1
(7)        put $D'_j$ into the same cluster with $D_j$, and set $D'_j$ as already clustered
(8)        $update(V_{center})$
(9)    set $V_{center}$ as the representation of $D_j$ ’s cluster result
(10) Merge the cluster with the same $V_{center}$.
(11) for each resulted cluster
(12)   if the tweets number in the cluster is smaller or bigger than the set threshold
(13)     mark the cluster as noise
(14)   if $s(V_{center}) = c_1 + c_2 + ... + c_n <= 1$, assume $V_{center} = (c_1, c_2, ..., c_n)$
(15)     mark the cluster as noise
(16) output cluster result
```

In the proposed algorithm, the first tweet input will be an individual topic cluster. Then, for each new created topic clusters, it will check all un-classified tweets to see whether it can be classified into the new created topic cluster. If the answer is yes, we classify this tweet and update V_{center} of the topic cluster at the same time. The worst time complexity of this algorithm is $O(n^2)$. When all the tweet topics are discrete, it reaches the worst condition.

We can use the distance between a tweet vector and the topic cluster’s center vector to determine whether the tweet can be classified into the cluster. The distance between $V_{center} = (c_1, c_2, ..., c_n)$ and $D = (w_1, w_2, ..., w_n)$ is calculated as follows:

$$dis(V_{center}, D) = \sqrt{\frac{\sum_i ((c_i \oplus w_i) \times fv_i)^2}{\sum_i (c_i \times w_i \times fv_i)^2}} \tag{4}$$

In which c_i and w_i can only be 0 or 1 as described in section III.B.

$c_i \oplus w_i$ is calculated as follows:

$$c_i \oplus w_i = \begin{cases} 1, when\ c_i\ and\ w_i\ are\ different \\ 0, when\ c_i\ and\ w_i\ are\ the\ same \end{cases} \tag{5}$$

Here, $(c_i \oplus w_i)$ equals to 1 when and only when c_i and w_i have different values, while $(c_i \times w_i)$ equals to 1 when and only when c_i and w_i are both 1. Therefore, if two objects contain the same features, the distance will be short; when two objects contain different features, distance will be long. That is to say, the shorter distance between two tweets means they belong to the same cluster more likely. Moreover, fv_i can strengthen the impact of features with high frequency.

For example, given following data:

$$
\begin{aligned}
FV &= (23 \quad 5 \quad 40 \quad 12 \quad 3) \\
V_{center} &= (1 \quad 1 \quad 0 \quad 0 \quad 1) \\
D_1 &= (1 \quad 0 \quad 0 \quad 0 \quad 1) \\
D_2 &= (0 \quad 1 \quad 1 \quad 0 \quad 1)
\end{aligned}
$$

(6)

We can get:

$$dis(V_{center}, D_1) = \sqrt{\frac{5^2}{23^2 + 3^2 + 0.1}} \approx 0.2155$$

$$dis(V_{center}, D_2) = \sqrt{\frac{23^2 + 40^2}{5^2 + 3^2 + 0.1}} \approx 7.9$$

(7)

Obviously, the distance between D_2 and V_{center} is larger than that between D_1 and V_{center}.

Then, as mentioned in step 8, we need to update $V_{center} = (c_1', c_2', ..., c_n')$ according to the tweets that are already clustered into the same cluster. The updating method is as follow: set $c_i' = 1$ when more than half of the tweets in the cluster have the weight of t_i equals to 1, otherwise set $c_i' = 0$. Thus V_{center} is more representative of those clustered tweets in the topic cluster.

After the first time topic clustering, all topic clusters with the same value of V_{center} are combined. For each topic cluster, we verify whether it is a noise set or not, according to its V_{center} and the number of tweets it contains. After Removing these noise sets, then we can get the final clusters.

According to its V_{center} and the feature space, for every topic cluster, if the dimensions in V_{center} have value greater than 0, we can find the corresponded feature terms in the feature space. These terms are keywords and can be used to represent that topic.

For example, assuming that the feature space is （母亲 救 美丽 高考 双胞胎）and the V_{center} of a topic cluster is $V_{center} = \begin{pmatrix}1 & 1 & 1 & 0 & 1\end{pmatrix}$, we can find the topic keywords of that cluster are “母亲，救，美丽，双胞胎”.

IV. Experiments

A. Data Set

We choose six hot topics on the date of June 11th, 2012. These topics range from social problems, education, science, technology to entertainment. Among these topics, the topic of “高考迟到母跪求无果” and “为高考隐瞒母亲死讯” have duplicate keywords such as“高考”、“母亲”, while the topic of “英雄司机吴斌” and “双胞胎孕妇救人” have duplicate keywords too, such as“英雄”、“最美”、“救”, etc. We use such topics to test our proposed algorithm. Whether it can tell the difference between these topics or not?

We collect tweets randomly through the website scratching method and open APIs provided by sina weibo, no matter the tweet is topic-related or not. Table 2 shows the topics and number of tweets associated with each topic.

TABLE II.

Topics and corresponding tweets number

Topic	Tweets number
高考迟到母亲跪求无果	817
为高考隐瞒母亲死讯	727
英雄司机吴斌	1021
双胞胎孕妇跳水救人	917
苹果全球开发者大会	741
李小璐被爆怀孕	675
Topic unrelated tweets (i.e. noise)	19220

B. Evaluation

We use miss rate and false rate to evaluate our algorithm according to the TDT evaluation method, where the miss rate and false rate of topic i (i = 1,2, …, k) are calculated as follows:

$$miss_i = \frac{missed\ detected\ topic\ i\ related\ tweets\ number}{total\ number\ of\ tweets\ related\ to\ topic\ i}$$

$$FA_i = \frac{false\ detected\ as\ topic\ i\ related\ tweets\ number}{total\ number\ of\ tweets\ unrelated\ to\ topic\ i}$$

The average miss rate P_{Miss}, and average false rate P_{FA} are:

$$P_{Miss} = \sum_i Miss_i / k$$
$$P_{FA} = \sum_i FA_i / k \qquad (8)$$

The smaller P_{Miss} and P_{FA} shows the better algorithm. Our goal is to keep both of them as small as possible.

C. Experiments

In our test data set, tweets topics are various, such as:

#高考迟到说#最近网上有则新闻很火啊，就是一位考生迟到 2 分钟，不让进考场。我想没有规矩不成方圆。高考最起码对他们来说是很重要的事，都能迟到。真是无语。有更重要的事耽误的话，就去做认为重要的事好了 你的观点呢？ http://t.cn/zOgaMpv

#为高考隐瞒母亲死讯不能接受#仅是一场考试，我们的道德底线到底在哪里..

#曝李小璐已怀孕四月#祝福吧，好想看他家孩子长什么样！！基因遗传好啊~~

期待：苹果 WWDC 明天凌晨开幕 iOS 6 成焦点 | 库克不仅将在大会上公布苹果未来一年的发展方向，还会展示一些新硬件、OS X“山狮”操作系统，以及下一版 iOS 系统。(银财风投配图) 银财风投：北京时间 6 月 11 日早间消 http://t.cn/zWvXmK4

这两天看微博，一直看到关于司机吴斌的英雄事迹，一直不敢打开看视频，害怕看到死亡的瞬间，现在看到电视播出这段视频，看到他女儿高考完看到爸爸死亡的消息痛哭的样子…… 吴斌，这名字真好听。 http://t.cn/zOs1rQv

【视频：“最美孕妇”怀双胞胎跳深塘救落水儿童 直播西安 120609】 http://t.cn/zOsoF7A （分享自 @优酷网）

睡不着。。。。。。。。。。

After data pretreatment, we can get the following results of the examples above:

#/x 高考/v 迟到/v 说/v #/x 最近/t 网上/s 有/v 则/q 新闻/n 很/d 火/a 啊/y ，/w 就/d 是/v 一/m 位/q 考生/n 迟到/v 2 分钟/t ，/w 不/d 让/v 进/v 考场/n 。/w 我/r 想/v 没有/v 规矩/n 不成方圆/v 。/w 高考/v 最/d 起码/d 对/p 他们/r 来说/u 是/v 很/d 重要/a 的/u 事/n ，/w 都/d 能/v 迟到/v 。/w 真/d 是/v 无/v 语/g 。/w 有/v 更/d 重要/a 的/u 事/n 耽误/v 的话/u ，/w 就/d 去/v 做/v 认为/v 重要/a 的/u 事/n 好/a 了/u 你/r 的/u 观点/n 呢/y ？ /w

#/x 为/v 高考/v 隐瞒/v 母亲/n 死讯/n 不能/v 接受/v #/x 仅/d 是/v 一/m 场/qv 考试/vn ，/w 我们/r 的/u 道德/n 底线/n 到底/d 在/p 哪里/r ./w ./w

#/x 曝/g 李小璐/nr 已/d 怀孕/v 四月/t #/x 祝福/v 吧/y ，/w 好/d 想/v 看/v 他/r 家/q 孩子/n 长/a 什么样/r ！ /w ！ /w 基因/n 遗传/vn 好/a 啊/y ~/x ~/x

期待/v ：/w 苹果/n WWDC/x 明天/t 凌晨/t 开幕/v iOS/x 6/g 成/v 焦点/n |/x 库克/nr 不仅/c 将/d 在/p 大会/n 上/f 公布/v 苹果/n 未来/t 一年/m 的/u 发展/vn 方向/n ，/w 还/d 会/v 展示/v 一些/mq 新/a 硬件/n 、/w OS/x X/x “/w 山/n 狮/g ”/w 操作系统/l ，/w 以及/cc 下/v 一/m 版/n iOS/x 系统/n 。/w (/w 银财风/nr 投/v 配/v 图/n)/w 银/b 财/n 风/n 投/v ：/w 北京/ns 时间/n 6 月/t 11 日/t 早间/t 消/v

这/r 两/m 天/qt 看/v 微/g 博/g ，/w 一直/d 看到/v 关于/p 司机/n 吴斌/nr 的/u 英雄/n 事迹/n ，/w 一直/d 不/d 敢/v 打开/v 看/v 视频/n ，/w 害怕/v 看到/v 死亡/v 的/u 瞬间/t ，/w 现在/t 看到/v 电视/n 播出/v 这/r 段/q 视频/n ，/w 看到/v 他/r 女儿/n 高考/v 完/v 看到/v 爸爸/n 死亡/v 的/u 消息/n 痛哭/v 的/u 样子/n …/w …/w 吴斌/nr ，/w 这/r 名字/n 真/d 好/a 听/v 。/w

【/w 视频/n ：/w “/w 最/d 美/b 孕妇/n ”/w 怀/v 双胞胎/n 跳/v 深/a 塘/g 救/v 落水/vn 儿童/n 直播/v 西安/ns 120609/m 】/w （/w 分享/v 自/p

睡/v 不/d 着/u 。/w 。/w 。/w 。/w 。/w 。/w 。/w 。/w 。/w 。/w

According to feature selection method we choose 33 words as feature items to generate the feature space. They are:

时间 中国 怀孕 全球 考生 孩子 开发者 不能 英雄 死讯 喜欢 时候 双胞胎 现在 流入 没有 资金 李小璐 救人 接受 大会 隐瞒 今天 吴斌 知道 苹果 母亲 迟到 可以 司机 分享 孕妇 高考

And we also record the times of occurrence $df(t_i)$ of the feature items t_i at the same time.

In our experiments, the relationship between the feature term t_i’s POS tag and $boost(t_i)$ is showed in Table III.

TABLE III.

VALUE OF BOOST(T_I) TO T_I’S POS TAG

POS tag	boost(t_i)
nr, ns, t	1.8
n	1.2
m, v, a, nt, nz	1

Based on $df(t_i)$ and $boost(t_i)$, we can get the vector FV from EQ(1):

FV = (644.4 988.2 572 702 705.6 706.8 735.6 622 758.4 763.2 645 778.8 793.2 1222.2 683 695 837.6 1269 716 771 932.4 797 1461.6 1479.6 826 1251.6 1270.8 1075 1106 1393.2 1257 1906.8 1805)

Using the vector space model, the above tweets can be expressed as:

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1

0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1

0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1

0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Then we calculate each tweet’s score to filter out those noise data. Here we set the lowest score as 1500.

After this noise exclusion, the tweets numbers of each topic are showed in Table IV. It shows that this noise exclusion algorithm can filters out most of the noise data, especially for those topic-unrelated tweets.

TABLE IV.

RESULT OF NOISE EXCLUSION

topic	number of remained	number of tweets filtered
高考迟到母亲跪求无果	808	9
为高考隐瞒母亲死讯	722	5
英雄司机吴斌	708	313
双胞胎孕妇跳水救人	913	4
苹果全球开发者大会	715	26
李小璐被爆怀孕	493	182
other topics	973	18247

TABLE V.

TEST RESULT OF CLASSES TO CLUSTERS AND CORRESPONDING KEYWORD

Assigned to cluster -- >	迟到|高考	不能|死讯|接受|隐瞒|母亲|高考	吴斌|司机|英雄	双胞胎|救人|孕妇	全球|开发者|大会|苹果	怀孕|李小璐	noise
高考迟到母亲跪求无果	757	14	0	0	0	0	46
为高考隐瞒母亲死讯	58	636	0	0	0	0	33
英雄司机吴斌	6	2	558	12	0	0	443
双胞胎孕妇跳水救人	0	0	0	829	0	0	88
苹果全球开发者大会	0	0	0	0	644	0	97
李小璐被爆怀孕	0	0	0	0	0	452	223
other topics	34	0	3	11	9	0	19163

At last, we use the remaining tweets as input for topic clustering and classifying. For each topic clusters that are not noise, we can get corresponding topic keywords based on its center vector and the feature space. Table V shows the final result.

D. *Experimental Results*

For evaluation, we calculate the corresponding miss rate and false rate of each topic as follows:

TABLE VI.

MISS RATE AND FALSE RATE OF EACH CLASS

topic	miss rate （%）	false rate （%）
高考迟到母亲跪求无果	7.344	0.42
为高考隐瞒母亲死讯	12.52	0.068
英雄司机吴斌	45.35	0.013
双胞胎孕妇跳水救人	9.597	0.099
苹果全球开发者大会	13.09	0.038
李小璐被爆怀孕	33.04	0
other topics	0.297	18.99

And the average miss rate and false rate of the method are:

$$P_{Miss} = \sum_i Miss_i / k = 15.15\% \quad ,$$

$$P_{FA} = \sum_i FA_i / k = 2.45\%$$

(9)

The result of our algorithm is quite satisfactory, because both miss rate and false rate are low enough.

The experiment result shows that our method can filter out most noise and resist against these noisy tweets, our clustering algorithm can identify the topics from large amount of tweets accurately and classify tweets to their corresponding topic clusters correctly.

V. CONCLUSTION

In this paper we analysis the characteristic and difficulty of microblog topic detection, and provide a topic detection model for Chinese microblogs. We describe the procedural of data pretreatment, feature selection, weight computation, text representation of the model, and noise tweet filtering. We also propose a new topic detection algorithm based on hierarchical clustering, using an improved method for the computing of distance between different tweets. This proposed topic detection method is easy to implement, and the following experiment shows that it is more efficient and more effective than traditional method. Moreover, this method has low miss rate and false rate, which means it is robust to noisy tweet influence.

ACKNOWLEDGMEN

This paper is supported by the National Key Basic Research Program of China (2013CB329603), the National Natural Science Foundation of China (61272441, 61171173) and the Opening Project of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security), whose number is C12609.

REFERENCES

[1] Hong Liu. Internet public opinion hotspot detection and analysis based on Kmeans and SVM algorithm[C]. 2010 International Conference of Information Science and Management Engineering. pp. 257-261 (2010)

[2] Phuvipadawat, S., Murata, T. Breaking News Detection and Tracking in Twitter[C]. 2010 IEEE/WIC/ACM International conference on Web Intelligence and Intelligent Agent Technology. pp.120-123 (2010)

[3] Qiu Yun-fei, Cheng Liang. Research on Sudden Topic Detection Method for Microblog[J]. Computer Engineering, Vol. 38(9), pp. 288-290 (2012)

[4] Zheng Fei-ran, Miao Duo-qian, etc. News Topic Detection Approach on Chinese Microblog[J]. Computer Science, Vol. 39(1), pp. 138-141 (2012)

[5] Zhiyuan Liu, Xinxiong Chen, etc. Mining the interests of Chinese microbloggers via keyword extraction[J]. Frontier of Computer Science in China, Vol. 6, pp. 76-87. (2012)

[6] Zhang H P, Yu H K, Xiong D Y, et al. HHMM-based Chinese lexical analyzer ICTCLAS[A]. Proceedings of the second SIGHAN workshop on Chinese language processing[C]. Sapporo, Japan: Associations for Computational Linguistics, pp. 184-187. (2003)

[7] Chengqing Zong. Statistical natural language processing. Edited by Tsinghua University Publisher, pp.342-343. (2008)

[8] D. Wu, Performance evaluation: an integreated method using data envelopment analysis and fuzzy preference relations[J]. European Journal of Operational Research. Vol.194 (1) pp. 227-235. (2009)

[9] Quanlong Guan, Saizhi Ye, Guoxiang Yao, Huanming Zhang, Linfeng Wei, Gazi Song, Kejing He. Research and Design of Internet Public Opinion Analysis System[C]. 2009 IITA International Conference on Services Science, Management and Engineering. pp. 173-177. (2009)

[10] DINOFF, R., HO, T., HULL, R., KUMAR, B., LIEUWEN, D., SANTOS, P., REN, H.. Intuitive Network Applications: Learning for Personalized Converged Services Involving Social Networks. Journal of Computers, North America, 2, aug. 2007.

[11] WU, P., LI, S.. Social Network Analysis Layout Algorithm under Ontology Model. Journal of Software, North America, 6, jul. 2011.

[12] XU, Y.. A Data-drive Feature Selection Method in Text Categorization. Journal of Software, North America, 6, apr. 2011.

Gongshen Liu. Shandong, China. Feb. 12th, 1974. He got his Ph.D. on computer science from Shanghai Jiao Tong University (SJTU), 2003; M.A. on computer science from Shandong University, 2000 and B.A. on computer science from Shandong University of Technology 1997.

He is an Associate Professor of School of Information Security Engineering of SJTU. He has many research experiences in the field of Natural Language Processing, Social Network and Content-based Security, some of which are published in International conferences and journals, such as China Communication, Journal of Systems Engineering and Electronics and so on.

Dr. Liu is the member of ACM, China Computer Federation and Chinese Information Processing Society of China.

Kui Meng. Jiangsu, China. Nov. 1st, 1973. She got doctor of science, in computer application technology, from Fudan University, Shanghai, China, 2006.

She is a lecturer of Shanghai Jiao Tong University, Shanghai, China. Publications: Computer Security (Beijing: Publishing House of Electronics Industry, 2003), Information Security Practice (Beijing: Tsinghua University Press, 2010), Malicious Code Prevention (Beijing: Higher Education Press, 2010). Current research interests include network trust management, Access control management and mobile security.

Dr. MENG, the second prize of Shanghai scientific and technical progress reward in 2008.

Jing Xie. Shanghai, 1989.9.3. Bachelor's Degree on information security, Shanghai Jiao Tong University, Shanghai, China, 2010; Master's Degree, information security, Shanghai Jiao Tong University, Shanghai, China, 2013.

She focuses her research in the areas of content security. She has participated in the National Natural Science Foundation of China(61171173, 61272441) and the National High Technology R&D Program of China (2010AA012505). Her research articles includes: The Prediction of User's Retweet Behavior in Social Network, accepted by Journal of Shanghai Jiao Tong University; A Topic Detection Method for Chinese Microblog, Proceedings of the 2012 Fourth International Symposium on Information Science and Engineering, 2012, pp: 100-103. Her current research interests are mainly about content security for social network.