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
{lgshe, mengkui}@sjt.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
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. 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>
<thead>
<tr>
<th>PART-OF-SPEECH (POS) TAG SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>n</td>
</tr>
<tr>
<td>ns</td>
</tr>
<tr>
<td>nz</td>
</tr>
<tr>
<td>s</td>
</tr>
<tr>
<td>v</td>
</tr>
<tr>
<td>b</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>q</td>
</tr>
<tr>
<td>p</td>
</tr>
<tr>
<td>u</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>h</td>
</tr>
<tr>
<td>x</td>
</tr>
</tbody>
</table>

© 2013 ACADEMY PUBLISHER
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
/ p 我
/ t 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\(^{13}\) (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 \(D = (D_1, w_1; D_2, w_2; \ldots; D_n, w_n)\), where \(D_i\) is the \(i^{th}\) feature and \(w_i\) is the weight of \(D_i\), \(1 \leq i \leq n\).

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 \(D_i\), we count the number of the tweets that exist the feature. When the number or the frequency, say \(df(D_i)\), is lower or higher than the specific threshold, the feature item \(D_i\) 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.

\[
FV = (fv_1, fv_2, \ldots, fv_n),
\]

\[
fv_i = df(t_i) \times boost(t_i)
\]

(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 \(t_i\), \(1 \leq boost(t_i) \leq 2\). \(boost(t_i)\) 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:

\[
w_{ij} = \begin{cases} 
1, & tf_{ij} > 0 \\
0, & \text{otherwise}
\end{cases}
\]

(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.
\[ D_j = (t_1, w_{1D_j}, t_2, w_{2D_j}, \ldots, t_n, w_{nD_j};) \]

\[
\text{Score}(D_j) = (w_{1D_j}, w_{2D_j}, \ldots, w_{nD_j}) \cdot FV^T
\]

\[
= (w_{1D_j}, w_{2D_j}, \ldots, w_{nD_j}) \cdot \begin{pmatrix} f_{v_1} \\ f_{v_2} \\ \vdots \\ f_{v_n} \end{pmatrix}
\]

\[
= \sum_{i=1}^{w} (w_{iD_j} \times f_{v_i})
\]

(3)

\[ f_{v_i} \] corresponds to the contribution of \( t_i \) made in topic representing. A larger \( f_{v_i} \) 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 \( D_j = (t_1, w_{1D_j}, t_2, w_{2D_j}, \ldots, t_n, w_{nD_j};) \) in \( D \)
   2. if \( D_j \) is already clustered in \( C \)
   3. go to step (1) and turn to the next tweet
   4. set \( V_{\text{center}} = (w_{1D_j}, w_{2D_j}, \ldots, w_{nD_j}) \)
   5. for each tweet \( D'_j \) in \( D \) which is not already clustered in \( C \)
   6. if distance\( (V_{\text{center}}, D'_j) < 1 \)
   7. put \( D'_j \) into the same cluster with \( D_j \), and set \( D'_j \) as already clustered
   8. update\( (V_{\text{center}}) \)
   9. set \( V_{\text{center}} \) as the representation of \( D_j \)'s cluster result
   10. Merge the cluster with the same \( V_{\text{center}} \).
   11. if 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 \( \delta(V_{\text{center}}) = c_1 + c_2 + \ldots + c_n < = 1 \),
    assume \( V_{\text{center}} = (c_1, c_2, \ldots, c_n) \)

(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 \( V_{\text{center}} \) 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 \( V_{\text{center}} = (c_1, c_2, \ldots, c_n) \) and \( D = (w_1, w_2, \ldots, w_n) \) is calculated as follows:

\[
\text{dis}(V_{\text{center}}, D) = \sqrt{\sum_i ((c_i \oplus w_i) \times f_{v_i})^2}
\]

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

\[
c_i \oplus w_i = \begin{cases} 1, & \text{when } c_i \text{ and } w_i \text{ are different} \\ 0, & \text{when } c_i \text{ and } w_i \text{ are the same} \end{cases}
\]

(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, \( f_{v_i} \) can strengthen the impact of features with high frequency.

For example, given following data:

\( FV = (23, 5, 40, 12, 3) \)

\( V_{\text{center}} = (1, 1, 0, 0, 1) \)

\( D_1 = (1, 0, 0, 0, 1) \)

\( D_2 = (0, 1, 1, 0, 1) \)

(6)

We can get:

\[
\text{dis}(V_{\text{center}}, D_1) = \sqrt{\frac{5^2}{23^2 + 3^2 + 0.1}} \approx 0.2155
\]
\[
\text{dis}(V_{\text{center}}, D_2) = \sqrt{\frac{23^2 + 40^2}{5^2 + 3^2 + 0.1}} \approx 7.9
\]

(7)

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

Then, as mentioned in step 8, we need to update \( V_{\text{center}} = (c'_1, c'_2, \ldots, c'_n) \) 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 \( V_{\text{center}} \) 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 \( V_{\text{center}} \) are combined. For each topic cluster, we verify whether it is a noise set or not, according to its \( V_{\text{center}} \) and the number of tweets it contains. After Removing these noise sets, then we can get the final clusters.

According to its \( V_{\text{center}} \) and the feature space, for every topic cluster, if the dimensions in \( V_{\text{center}} \) 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 \( V_{\text{center}} \) of a topic cluster is \( V_{\text{center}} = \{1 \ 1 \ 1 \ 0 \ 1\} \), 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 “高考迟到母亲求无果” 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>
<thead>
<tr>
<th>Topic</th>
<th>Tweets number</th>
</tr>
</thead>
<tbody>
<tr>
<td>高考迟到母亲跪求无果</td>
<td>817</td>
</tr>
<tr>
<td>为高考隐瞒母亲死讯</td>
<td>727</td>
</tr>
<tr>
<td>英雄司机吴斌</td>
<td>1021</td>
</tr>
<tr>
<td>双胞胎孕妇救人</td>
<td>917</td>
</tr>
<tr>
<td>苹果全球开发者大会</td>
<td>741</td>
</tr>
<tr>
<td>李小璐被爆怀孕</td>
<td>675</td>
</tr>
</tbody>
</table>

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 \( i \) (\( i = 1, 2, \ldots, k \)) are calculated as follows:

\[
\text{miss}_i = \frac{\text{missed detected topic } i \text{ related tweets number}}{\text{total number of tweets related to topic } i}
\]

\[
\text{FA}_i = \frac{\text{false detected as topic } i \text{ related tweets number}}{\text{total number of tweets unrelated to topic } i}
\]

The average miss rate \( P_{\text{Miss}} \), and average false rate \( P_{\text{FA}} \) are:

\[
P_{\text{Miss}} = \frac{\sum_i \text{miss}_i}{k}
\]

\[
P_{\text{FA}} = \frac{\sum_i \text{FA}_i}{k}
\]

(8)

The smaller \( P_{\text{Miss}} \) and \( P_{\text{FA}} \) 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:

- 高考迟到母亲跪求无果 #高考迟到说#最近网上有则新闻很火，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。高考是最起码对他们来说是件重要的事，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？ #加分转给同学下 http://t.cn/zOgaMpv

- 英雄司机吴斌 #为高考隐瞒母亲死讯不能接受 #吴斌这名字真好听。看到他女儿高考完看到爸爸死亡的消息痛哭的样子……

- 李小璐被爆怀孕 #李小璐已怀孕四月#祝福吧，好想看他家孩子长什么样!! #李小璐已经怀孕四月了

- 苹果全球开发者大会 #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) 财风投：北京时间 6月11日早间 GordonXK4

- 双胞胎孕妇救人 #双胞胎孕妇跳深塘救落水儿童 正版西安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:

### TABLE III.

<table>
<thead>
<tr>
<th>POS tag</th>
<th>boost($t_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr, ns, t</td>
<td>1.8</td>
</tr>
<tr>
<td>n</td>
<td>1.2</td>
</tr>
<tr>
<td>m, v, a, nt, nz</td>
<td>1</td>
</tr>
</tbody>
</table>

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

$$ FV = (644.4, 988.2, 572, 702, 705.6, 706.8, 735.6, 622, 758.4, 763.2, 645, 778.8, 779.3, 1222.2, 683, 695, 837.6, 1269, 716, 771, 932.4, 797, 1461.6, 1479.6, 826, 1251.6, 1270.8, 1075, 1106, 1393.2, 1257, 1906.8, 1805) $$

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

$$
\begin{array}{c}
\text{time} \quad \text{中国} \quad \text{怀孕} \quad \text{全球} \quad \text{考生} \quad \text{孩子} \quad \text{开发者} \quad \text{不能} \quad \text{英雄} \quad \text{死讯} \quad \text{喜欢} \quad \text{时候} \\
\text{双胞胎} \quad \text{现在} \quad \text{流入} \quad \text{没有} \quad \text{资金} \quad \text{李小璐} \quad \text{救人} \quad \text{接受} \quad \text{大会} \quad \text{隐瞒} \quad \text{今天} \\
\text{吴斌} \quad \text{知道} \quad \text{苹果} \quad \text{母亲} \quad \text{迟到} \quad \text{可以} \quad \text{司机} \quad \text{分享} \quad \text{孕妇} \quad \text{高考}
\end{array}
$$

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.

<table>
<thead>
<tr>
<th>topic</th>
<th>number of remained</th>
<th>number of tweets filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td>高考迟到母亲跳楼无果</td>
<td>808</td>
<td>9</td>
</tr>
<tr>
<td>为高考隐瞒母亲死讯</td>
<td>722</td>
<td>5</td>
</tr>
<tr>
<td>英雄司机吴斌</td>
<td>708</td>
<td>313</td>
</tr>
<tr>
<td>双胞胎孕妇跳水救人</td>
<td>913</td>
<td>4</td>
</tr>
<tr>
<td>苹果全球开发者大会</td>
<td>715</td>
<td>26</td>
</tr>
<tr>
<td>李小璐被爆怀孕</td>
<td>493</td>
<td>182</td>
</tr>
<tr>
<td>other topics</td>
<td>973</td>
<td>18247</td>
</tr>
</tbody>
</table>
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.

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

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:

$$P_{\text{Miss}} = \frac{\sum_i \text{Miss}_i}{k} = 15.15\%$$

$$P_{\text{FA}} = \frac{\sum_i \text{FA}_i}{k} = 2.45\%$$

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. CONCLUSION

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.

ACKNOWLEDGMENTS

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


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.


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.