Thread Cleaning and Merging for Microblog Topic Detection

Jianfeng Zhang[1,2], Yunqing Xia[1], Bin Ma[1,2], Jianmin Yao[2], and Yu Hong[2]
[1]Dept. of Comp. Sci. & Tech., Tsinghua University, Beijing 100084, China
yqxia@tsinghua.edu.cn

[2]School of Comp. Sci. & Tech., Soochow University, Suzhou 215104, China
{jfzhang, bma, jyao, hongy}@suda.edu.cn

Abstract

As a classic natural language processing technology, topic detection recently attracts more research interests due largely to the rapid development of microblog. The most challenging issue in microblog topic detection is sparse data problem. In this paper, the temporal-author-topic (TAT) model is designed to accomplish microblog topic detection in two phases. In the first phase, the TAT model is applied to clean the thread, namely, to filter noisy microblog texts out of each thread. In the second phase, microblog texts within each thread are merged to form the thread text so that the TAT model is applied to find global topics. The new approach differs from the Hierarchical Agglomerative Clustering (HAC) algorithm by making use of microblog threads to overcome the sparse data problem. Experimental results justify our claims.

1 Introduction

Topic detection is the technique that discovers the latent topics from the given collection of text. Originated from the famous TDT evaluation workshop[1], topic detection research has attracted intensive and persistent interests from governments with security purpose. With the rapid advance of the Internet, the Web content becomes so plentiful that people start to explore how to make good use of the content with commercial purpose. Very recently, microblog becomes surprisingly popular. According to the recent Twitter statistics, 155 million tweets are created per day on average[2]. This leads to huge research passion on the microblog content.

Theoretically, topic detection from microblog text is similar to that from news articles. However, the microblog text is rather different from news articles. According to Ellen et al. (2011), microblog text is a typical microtext. Compared to regular long text such as news article, the microblog text exhibits the following characteristics.

- **Short**. Every microblog text finishes in less than 140. In fact, most texts are only a sentence or even a phrase.
- **Informal**. Spoken language is typically used, usually containing abbreviations and misspellings.
- **Semi-structured**. Each microblog text contains text as well as author and time.
- **Highly contextual**. Most microblog texts are created by replying or evaluating the existing texts. Meanwhile, they are replied or evaluated by others.
- **Conversational.** The microblog texts are usually naturally organized by thousands of conversation threads. In the thread, we name the top text by *head posting*, and the remaining texts by *followup postings*.

The following challenges in microblog text processing are worth noting. Firstly, microblog texts, especially the *followup postings*, contain very few characters. This inevitably leads to serious sparse data problem when machine learning algorithms are adopted to handle the microblog texts. Secondly, grammar is usually informal in microblog texts. Abbreviations and misspellings are constant. This makes standard language processing tools inapplicable on microblog texts. Thirdly, in the *followup postings*, anaphora and ellipsis are constantly used. This makes topic analysis rather difficult.

Some related work has been reported on microtext such as chat language and short messages (Dyke et al., 1999; Zhou et al., 2005; Shen et al., 2006; Peng et al., 2007). An earlier attempt on microblog text processing is Shen et al. (2009), which adopts TFIDF algorithm to analyze the Chinese microblog texts. Ramage et al. (2010) maps Twitter texts to four potential

[1] http://projects.ldc.upenn.edu/TDT/
[2] http://techcrunch.com/2011/04/06/twitter-q1-stats/

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 589–597,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

dimensions by labeled LDA, then sorts and recommends Twitter based on the result of LDA. However, the common drawback of the related work is sparse data problem because each posting is viewed as an individual text.

We argue that each microblog thread maintains a dominating topic, and that the thread structure plays an important role in microblog topic detection. In this work, the temporal-author-topic (TAT) model is designed to accomplish microblog topic detection in two phases. In the first phase, the TAT model is applied on each thread to organize the intra-thread postings into a few clusters, in which the cluster containing the head posting should dominate. The intention is to exclude the *followup postings* that are irrelevant to the head *posting* from the thread. In this way, the thread is made cleaner. In the second phase, postings within each cleaned thread are merged to form a bigger text, referred to as thread text. The TAT model is then applied on the thread texts to find global topics.

The new approach makes use of microblog threads to address the sparse data problem, which makes the approach different from any hierarchical text clustering approaches, e.g., Hierarchical Agglomerative Clustering (HAC). As it contains all relevant postings in the thread, the thread text is longer that any individual posting. In this manner, the sparse data problem can be relieved to great extent. Contributions of this work are summarized as follows.

(1) The thread structure in microblog is investigated in this work to address the sparse data problem in microblog topic detection. We argue cleaning and merging are crucial.

(2) A benchmark dataset is developed, which can be used by researchers to evaluate microblog topic detection approaches.

(3) The temporal-author-topic (TAT) model is proposed to model microblog text.

(4) A two-phase approach is designed to accomplish microblog topic detection. In the first phase, the thread structure is used to clean every thread. In the second phase, each cleaned thread is merged to form a bigger text so that microblog topic detection is made more accurate.

The rest of this paper is organized as follows. In Section 2, related work is summarized. In Section 3, the principle of our approach is given. In section 4, the temporal-author-topic model is described. Section 5 presents the two-phase topic detection approach. Section 6 presents experimental results as well as discussions. We conclude this paper in Section 7.

2 Related Work

2.1 Topic Detection

Topic detection (TD) appeared as a subtask in TDT (topic detection and tracking) evaluation workshops. Since 1998, many research efforts have been made. The earlier work is carried out under TDT evaluation. Many famous universities and companies such as IBM Watson, BBN, CMU and CUHK, have participated in TDT workshop. TDT has been more and more important.

Two subtasks are included in TD evaluation, i.e., online topic detection and hierarchical topic detection. Online topic detection (OTD) is to detect new topic and collect the subsequent relevant news. The OTD systems usually focus on selection and combination of the clustering methods. Generally, k-means clustering algorithm is adopted by researchers. Yang et al. (1998) first adopted the hierarchy clustering to detect topics, but the results can be further enhanced. Therefore, Xu et al. (1999) and Wartena et al. (2008) used k-means to cluster the news streams to realize the topic detection, and the results are better than previous work. Papka et al. (1999) compared different clustering algorithms and attempted to combine the advantage of each algorithm. The results show the combination is efficient.

TDT 2004 defines a new TD task: hierarchical topic detection (HTD)[3]. HTD presents that the theme and topic of the news reports usually distribute in different levels. For example, *Financial Crisis in Wall Street* and *Rise of Gold Price* both belong to the topic *The 10 Financial Events in 2009*, but the emphasis of the theme makes two reports in different levels. Cutting et al. (1992) proposed a hybrid clustering algorithm to improve traditional HAC (Hierarchical Agglomerative Clustering). Trieschnigg et al. (2004) adopted incremental hierarchical clustering to implement HTD. Complexity of TD is decreased in this approach, on condition that remaining efficiency of clustering. All proposed TD approaches can achieve good performance in regular texts. However, it is not known whether the clustering algorithms are effective in microblog TD.

[3] http://ciir.cs.umass.edu/pubfiles/ir-389.pdf

Our approach is similar to HAC in nature. However, two differences are worth noting. First, each microblog thread is viewed as a priori cluster in our approach. The intra-thread topic detection helps to clean the thread. In contrast, the HAC approach does not use the thread structure. Second, the irrelevant postings are excluded in forming the thread text, which is used in higher level topic clustering. Differently, the HAC approach excludes no text.

2.2 Mircoblog Text Processing

Microblog is a user-relationship based platform to assist user sharing and gaining information. As microblog booms, microtext is made large scale. Microblog text processing has thus become an important topic. In this paper, we mainly summarize the related work on microblog topic detection. Microblog topic detection exhibits profound significance. Two functions are interesting. Firstly, it is able to remind users of the important events that has happened or is happening in a period. Sharifi et al. (2010) proposed a method to summarize the topic in microblog. Microblog texts were detected if they contain the same maximum common substring, and the substring is regarded as the title of microblog topic. Nevertheless, there exists much noise in the microblog postings. Thus, the maximum common substring might be a meaningless phrase or sentence. O'Connor et al. (2010) used document clustering and text summarization techniques to induce topics that are relevant to the query[4]. The main idea of the method is still to match the microblog texts that contain the key words or phrases, making the results less accurate.

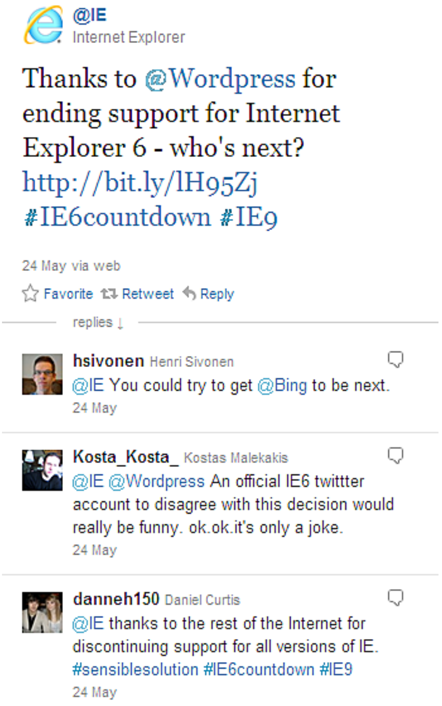
Secondly, irrelevant texts can be filtered out with topic detection approach. Wang et al. (2010) proposed a TwitterRank algorithm, which sorts the returned microblog texts by relevance score. In Liu et al. (2010), a feature selection method based on part-of-speech and HowNet is proposed, which can improve the performance of microblog classification. Similarly, Sriram et al. (2010) classified tweets into five categories, i.e. News, Events, Opinions, Deals and Private Messages by making use of author information within the tweets. With such a system, user can choose to view tweets based on their interest. Unfortunately, every posting is regarded as an individual text in previous methods, suffering the serious sparse data problem.

3 The Principle

3.1 The Idea

To illustrate our idea, we take a *head posting* and its *followup postings* as a Twitter example in Figure 1.

With thousands of samples like Figure 1, we make observations and come up with the corresponding arguments as follows.



@IE Internet Explorer

Thanks to @Wordpress for ending support for Internet Explorer 6 - who's next? http://bit.ly/lH95Zj #IE6countdown #IE9

24 May via web

hsivonen Henri Sivonen
@IE You could try to get @Bing to be next.
24 May

Kosta_Kosta_ Kostas Malekakis
@IE @Wordpress An official IE6 twittter account to disagree with this decision would really be funny. ok.ok.it's only a joke.
24 May

danneh150 Daniel Curtis
@IE thanks to the rest of the Internet for discontinuing support for all versions of IE. #sensiblesolution #IE6countdown #IE9
24 May

Figure 1. A Twitter head posting and its *followup* postings

Observation 1: The *followup postings* are created to reply the head posting directly or indirectly.

Argument 1: Postings in one thread are usually topic-relevant.

For example, the three *followup postings* in Figure 1 are all topic-relevant.

Observation 2: There exist a few irrelevant postings, e.g., spam and meaningless postings.

Argument 2: The irrelevant postings can be distinguished from the relevant ones considering content similarity.

Observation 3: The individual postings are very short, i.e., up to 140 characters, while a thread usually contains more than 20 postings, which add up to more than 200 words.

Argument 3: The topic-relevant postings in a thread can be merged to form a bigger text so as to relieve sparse data problem.

With above arguments, we propose a two-phase microblog topic detection approach,

[4] http://tweetmotif.com/

in which irrelevant postings are filtered out of the threads in the first phase and relevant postings in each thread are merged to form a bigger thread text.

3.2 Definitions

For description convenience, we first give some definitions being related to microblog text.

***Definition* 1: Posting**. A *posting* is a piece of semi-structured microblog text that covers author, time and textual content, denoted with d.

***Definition* 2: Head Posting**. A *head posting* is a piece of microblog text that is spontaneously delivered, denoted with d^H.

***Definition* 3: Followup Posting**. A *followup posting* is a piece of microblog text that replies to another piece of microblog text, denoted with d^F.

***Definition* 4: Thread**. A *thread* is a set of microblog texts that contains the head posting and the *followup postings*, denoted with $T = (V, E)$. The thread complies with the tree structure.

***Definition* 5: Forest**. A *forest* is a set of microblog threads, denoted with $F = (V^{\cup}, E^{\cup})$.

An example microblog forest is given in Figure 2, in which three threads maintains three head postings and fourteen *followup postings*.

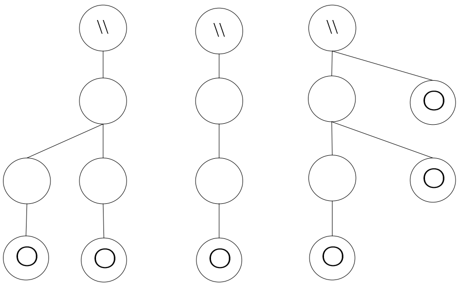


Figure 2. The example microblog forest contains three threads, where ⊘ represents the *head posting*, ○ the non-leaf *followup postings*, and ◎ the leaf *followup postings*.

The forest structure discloses some important information. As shown in Figure 2, every thread begins with a head posting and contains a few *followup postings*.

3.3 The Workflow

The workflow of our two-phase topic detection approach is given in Figure 3. In the first phase, intra-thread topic detection is run locally to find irrelevant *followup postings* within each thread. In the second phase, the relevant postings in each thread are merged to form a thread text so that the global topic detection is achieved with the thread texts. As thread texts are bigger in size, global topic detection can thus yield better results. In this way, sparse data problem in microblog topic detection can be alleviated to great extent.

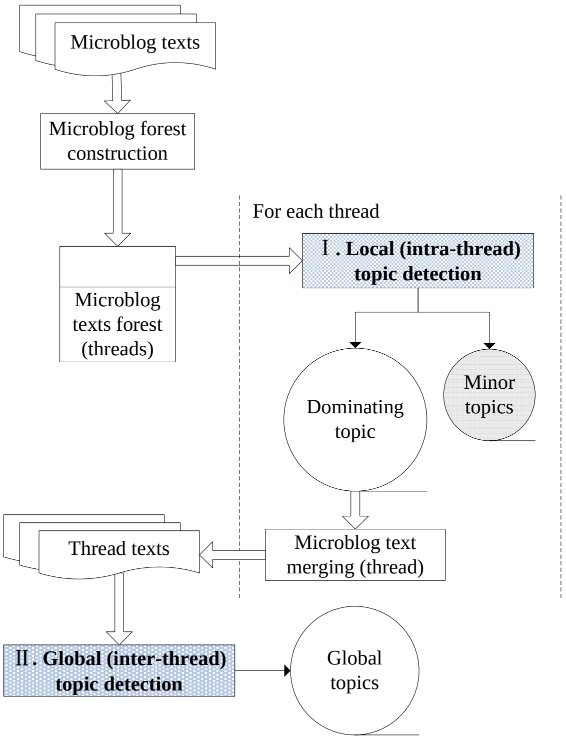

Microblog texts
Microblog forest construction
For each thread
Ⅰ. Local (intra-thread) topic detection
Microblog texts forest (threads)
Dominating topic
Minor topics
Thread texts
Microblog text merging (thread)
Ⅱ. Global (inter-thread) topic detection
Global topics

Figure 3. The workflow of our approach.

4 The Approach

Topic detection relies on a topic models. In this work, the temporal-author-topic model (TAT) is proposed to handle microblog texts. For description convenience, we first give the probabilistic topic model.

4.1 Probabilistic Topic Model

The common topic model used in topic detection is probabilistic topic model. Three distributions are given as follows:

(1) Word-topic distribution: $P(w, z) = \varphi(z)$;

(2) Word-topic dispatch: $P(w|z) = \delta(w)$

(3) Word-document distribution: $P(w, d) = \psi(d)$;

where z represents a topic, and w a word. The topic analysis actually judges the topic distribution $\theta(d)$ of document d.

4.2 Temporal-Author-Topic Model

As a kind of Internet microtext, the microblog text is intentional, conversational and

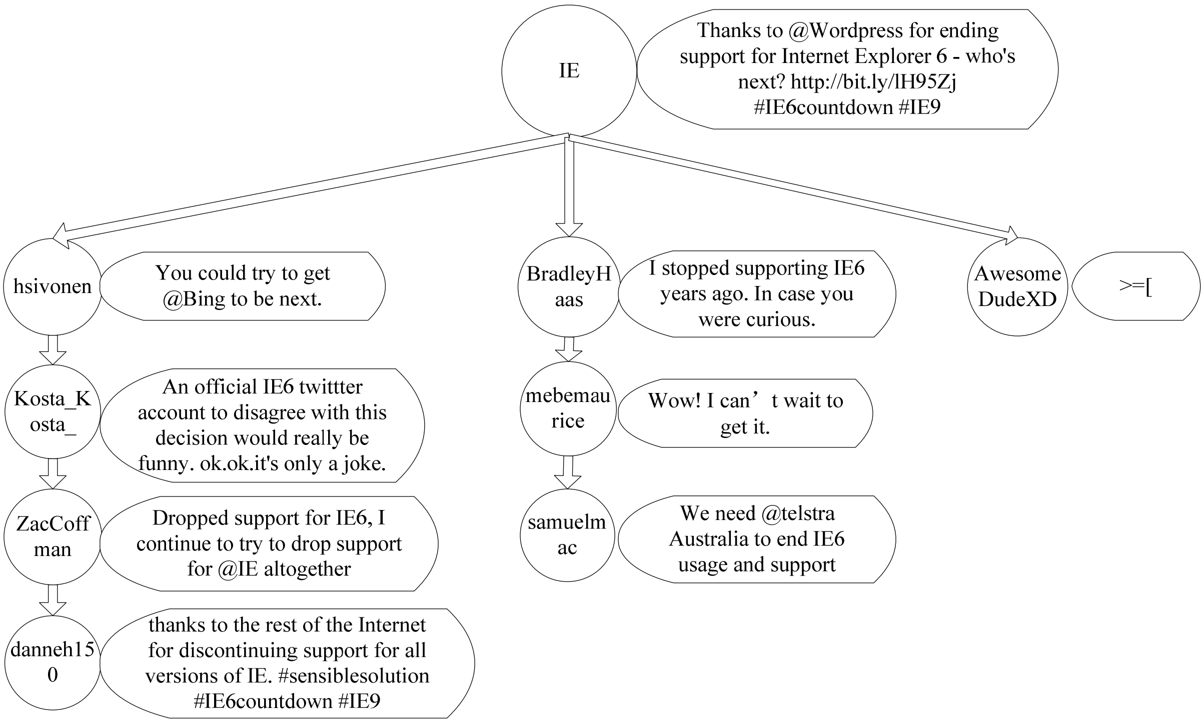

IE
Thanks to @Wordpress for ending support for Internet Explorer 6 - who's next? http://bit.ly/lH95Zj #IE6countdown #IE9
hsivonen
You could try to get @Bing to be next.
Kosta_Kosta_
An official IE6 twittter account to disagree with this decision would really be funny. ok.ok.it's only a joke.
ZacCoffman
Dropped support for IE6, I continue to try to drop support for @IE altogether
danneh150
thanks to the rest of the Internet for discontinuing support for all versions of IE. #sensiblesolution #IE6countdown #IE9
BradleyHaas
I stopped supporting IE6 years ago. In case you were curious.
mebemaurice
Wow! I can't wait to get it.
samuelmac
We need @telstra Australia to end IE6 usage and support
AwesomeDudeXD
>=[

Figure 4. Tree structure of the microblog thread T^O

personalized. The topic model for the microblog texts should reflect above characteristics. We consider the following information as the features in microblog topic detection.

- **Author**: In the microblog texts, author is a prominent feature. Observations show that one author usually participates in a limited number of topics. In this work, the author information is added in the common topic model, and the author-topic (AT) model is formed.
- **Timestamp**: We also find out that, if an author delivers several statements within a short period, these statements probably focus on a limited number of topics. Thus we define the interval as one hour, which means two posts are probably related to the same topic if they are delivered within one hour. Considering the temporal information, the AT model then evolves to the temporal-author-topic (TAT) model.
- **Thread**: Thread information is very important to microblog topic detection. Suppose a set of postings belong to the same conversation thread, they are assumed to talk about the same topic, which is initiated by the *head posting*. In this work, thread information is viewed as a key feature.

Finally, based on the common topic model, TAT adds in the following distributions:

(4) Temporal-Author-Topic distribution: $P(t,a,w,z)=\rho(z)$;

(5) Temporal-Author-Topic dispatch: $P(t,a,w|z)=\sigma(w)$;

where t is timestamp, a the author.

The distribution can be obtained from microblog text development dataset.

4.3 Local Topic Detection

Local topic detection, also called intra-thread topic detection, is the first phase in microblog topic detection. In this phase, the topic analysis is one-cluster based, which means that a dominating topic will be detected while the texts in the other topics are deemed irrelevant to the dominating topic.

Figure 4 gives a tree structure of the microblog thread T^O in Figure 1. The thread tree has three sub-trees, namely, there are three subtopics within thread T^O. However, seen from Figure 4, the right sub-tree is obviously not relevant to the dominating topic. Thus in this thread, the posting in the right sub-tree is deemed a spam posting. We adopt topic detection to filter such spam postings in every microblog thread.

The TAT model is used in thread topic detection in this work. We further consider more heuristics. In microblog threads, there exist many *followup postings*.

Given a pair of postings, in which one posting replies the other, the two postings P_1 and P_2 hold *A-reply-B* relation. They are assumed to talk about the same topic. The *A-reply-B* relation plays an important role in thread topic detection. We define that, if two postings hold the *A-reply-B* relation, in clustering, the similarity of two postings is increased by a parameter λ. The similarity formula is given as follows.

$$Sim(P_1, P_2) = \lambda + Sim(P_1, P_2), \quad \lambda = \frac{1}{|N|} \quad , \qquad (1)$$

where $Sim(P_1, P_2)$ represents the posting similarity calculated with VSM model, and $|N|$ is the number of postings in the thread.

Due to the quantity of spam postings is small, after calculating the similarity, the cluster which has the least postings may be filtered as the spam. As a result, a clean and topic-related thread T^R is obtained from thread T^O.

4.4 Global Topic Detection

Global topic detection, also referred to as inter-thread topic detection, is conducted on forest level. Microblog texts are usually very short, i.e., less than 140 characters. We propose to make use of thread structure to address the sparse data problem. We merge microblog texts in every clean thread to form a bigger thread text. Then the global topic detection is achieved with thread texts. Once a thread text is assigned a topic label, microblog texts in this thread are all assign the label. In this way, the ultimate goal of microblog topic detection is achieved.

Denote thread text being generated with thread T^R by X_i, which is combination of all the posts in thread T^R. Now we convert the microblog text into a set of thread texts $X_{i=1,\ldots,N}$. The global topic detection is executed on $X_{i=1,\ldots,N}$ to find microblog topics.

4.5 LDA-based Feature Weighting

TFIDF (term frequency and inverse document frequency) is widely used to calculate feature weights. In this paper, we also evaluate the Latent Dirichlet Allocation (LDA) in feature selection. LDA is proven better than TFIDF in regular texts (Madsen et al., 2005; Krestel et al., 2009). In this work, we evaluate how it works on microblog texts.

LDA is an unsupervised model proposed by Blei et al. (2003). It views every text as the combination of topics, and transfers the dimension of words into topics.

To be specific, LDA models document as a mixture of *K* latent topics, each of which is a multinomial distribution over a word vocabulary *W*. For document j, we first draw a mixed proportion $\theta_{k|j}$ from a Dirichlet with parameter a. For the i^{th} word in the document, a topic z_{ij} is drawn with topic k chosen with probability $\theta_{k|j}$. Then word x_{ij} is drawn from the z_{ij}^{th} topic by taking on value *w* with probability $\phi_{w|k}$. Finally, a Dirichlet prior with parameter β is placed on the topics $\phi_{w|k}$. Thus, the generative process is given as follows.

$$\phi_{k|j} \sim D[\alpha] \quad \phi_{w|k} \sim D[\beta]$$
$$z_{ij} \sim \theta_{k|j} \quad x_{ij} \sim \phi_{w|z_{ij}} \qquad (2)$$

Given the observed words $X = \{x_{ij}\}$, Bayesian inference seeks to compute the posterior distribution over the latent topic indices $Z = \{z_{ij}\}$, the mixed proportion $\theta_{k|j}$, and the topics $\phi_{w|k}$. An efficient procedure is to use collapsed Gibbs sampling, where θ and ϕ are marginalized out, and the latent variables Z are sampled. Given the current state of all but one variable z_{ij}, the conditional probability of z_{ij} is given below.

$$p(z_{ij} = k | z^{\neg ij}), x, \alpha, \beta) \propto$$
$$(\alpha + n^{\neg ij}_{k|j})(\beta + n^{\neg ij}_{x_{ij}|k})(w\beta + n^{\neg ij}_{k}) \quad , \qquad (3)$$

where the superscript $\neg ij$ means that the corresponding data-item is excluded in the count values, and $n_{jkw} = \#\{i : x_{ij} = w, z_{ij} = k\}$. We use the convention that missing indices are summed out: $\sum_w n_{jkw}$ and $n_{w|k} = \sum_j n_{jkw}$.

4.6 VSM-based Document Similarity

Vector Space Model (VSM) is a widely used document representation model. Let *d* represent a document and $\{t_i\}_{i=1,\ldots,K}$ feature terms appearing in document *d*. Then document *d* can be represented by the following text vector V_d according to the TFIDF or LDA.

$$V_d = (t_1 : w_1^s; \ldots; t_K^s),$$

where w_i^s is weight of feature term t_i.

In VSM, document similarity is usually measured using the cosine function, which is given as follows.

$$Sim(d_1, d_2) = \frac{D_1^T, D_2}{\sqrt{D_1^T, D_1}\sqrt{D_2^T, D_2}}, \qquad (4)$$

where D_1 and D_2 denote the two document vectors.

4.7 Text Clustering

Any clustering algorithms can be used in our algorithm. In this work, but we choose K-means (Duda et al., 1973) and HAC (Voorhees, 1986).

HAC is similar to our approach due to the hierarchical nature. So we intend to compare our approach against HAC. We select K-means because it is a classical clustering algorithm.

5 Experiment

5.1 Data Preparation

There is no benchmark dataset that fits into our scenario. We have to compile the gold standard by ourselves.

We use SINA microblog API[5] to extract Chinese microblog texts. Then the gold standard is compiled by six human annotators. The annotation scheme complies with the TDT4 annotation guideline.

Finally, we constructed a microblog dataset containing 1,100 threads and 16,500 postings (i.e., 15 postings in each thread on average). The postings are managed in 100 topics.

5.2 Evaluation Metrics

We adopted the evaluation metrics proposed by Steinbach et al. (2000). The calculation starts from the maximum *F*-measure in each cluster. Let A_i represent the set of articles that are managed in a system-generated cluster c_i, A_j is the set of articles managed in a human-generated cluster c_i. *F* measure of the system-generated cluster c_i is calculated as follows.

$$\begin{aligned} p_{i,j} &= \frac{|A_i \cap A_j|}{|A_j|} & p_i &= \max_j\{p_{i,j}\} \\ r_{i,j} &= \frac{|A_i \cap A_j|}{|A_i|} & r_i &= \max_j\{r_{i,j}\}, \\ f_{i,j} &= \frac{2 \cdot p_{i,j} \cdot r_{i,j}}{p_{i,j} + r_{i,j}} & f_i &= \max_j\{f_{i,j}\} \end{aligned} \quad (5)$$

where $p_{i,j}$, $r_{i,j}$ and $f_{i,j}$ represent precision, recall and *F* measure of cluster c_i when compared with cluster c_j, respectively.

5.3 The Approaches

Three baseline approaches are developed in this work. The intention is to evaluate the influence of author, timestamp and thread information on topic analysis, respectively.

Baseline B1: Only posting text is used in topic detection. All postings are used equally in topic detection.

Baseline B2: Author information is added to the baseline system B1 and the author-topic (AT) model is formed. All postings are used equally in topic detection.

Baseline B3: Timestamp is added to baseline system B2 and the temporal-author-topic (TAT) model is formed. All postings are used equally in topic detection.

Our approach OUR: The TAT model is used in topic modeling. The thread information is considered, and topics within microblog texts are detected in two phases.

Note that TFIDF or LDA are adopted for feature selection and HAC or K-means for text clustering in all systems. To evaluate how topic number influences the approach, six predefined class numbers are defined in this experiment, ranging from 50 to 100.

5.4 Results and Discussions

Table 1 reports the experimental results of our approach and the baselines on gold-standard dataset of predefined cluster number of 100, which use TFIDF or LDA feature selection and HAC or K-means clustering algorithm.

	K-means + TFIDF (%)	K-means + LDA (%)	HAC + TFIDF (%)	HAC + LDA (%)
B1	21.1	23.2	17.2	21.7
B2	25.5	26.4	22.2	25
B3	25.2	27	21.8	25.4
OUR	26.6	31.2	24.7	27.5

Table 1. F measure values of approaches with predefined topic number 100.

Three observations are made on the experiment results. Firstly, according to Table 1, system B2 outperforms B1 by 4.6%, system B3 outperforms B2 by 1.1%, and our system outperforms B3 by 2.8% on average. It is thus proven that author, timestamp and thread information are important in microblog topic detection.

The significant outperformance can be explained by two types of errors that constantly happen in baseline systems but not in our system. Errors of the first type come from the conversation threads. In the baseline systems, each posting is considered as an individual text. The contextual information is ignored in the clustering process. For example, one posting reads: *It's really cute*! It is difficult for the baseline systems to figure out which topic it belongs to. In contrast, our approach can merge the posting to the head posting reads: *Beijing Kennel Club adopts six stray dogs.* It is no longer for our system to detect what is really cute.

[5] http://open.weibo.com/wiki/index.php/SDK

The second typical error comes from the sparse data problem. As aforementioned, a posting is considered as an individual text in baseline systems. When we use TFIDF or LDA in feature selection and adopt VSM to represent the posting, the data sparseness is serious. For instance, a text vector V_i looks like

$$V_i = (t_1 : 0; t_2 : 0; ...; t_{k-1} : 1, t_k : 0) ,$$

where only the feature t_{k-1} appears in the posting. We even find some extreme postings that contain no feature at all. It is difficult to calculate the similarity between two such postings.

In our approach, thread is viewed as a whole, and thread text is used to find global topics. The sparse data problem is alleviated to great extent. This is the major reason that leads to significant outperformance.

Secondly, we compare feature selection algorithms, i.e. TFIDF and LDA, in all experiments. LDA is demonstrated to be better than TFIDF in regular texts. In this work, we try to prove this conclusion with microblog texts. As shown in Table 1, the system using LDA outperforms that uses TFIDF. We thus conclude that the conclusion made by Madsen, et al. (2005) and Krestel et al. (2009) is also true on microblog texts.

Thirdly, we compare different clustering algorithms, i.e. HAC and K-means, in our experiments. Seen from Table 1, the system using K-means outperforms that uses HAC. We can conclude that K-means algorithm fits into our approach better than HAC.

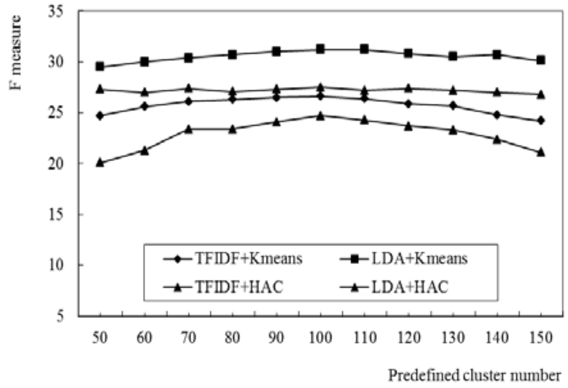


Figure 4. F measure curves of our approach with various topic numbers.

Finally, we evaluate how the predefined topic number influences the clustering algorithm. Seen from Figure 4, the approaches with LDA perform stably with different topic number. But for the approaches with TFIDF, different trend is disclosed. Performance of the approaches climbs gradually. Shown in Figure 4, our approach improves less when topic number is closer to 100. When the topic number is bigger than 100, F measure of our approach starts to drop. It can thus be concluded that TFIDF is sensitive to topic number than LDA. Note that the topic number in the gold standard dataset is 100. We thus conclude that the approach fits to the datasets well.

6 Conclusion and Future Work

In this paper, the Temporal-Author-Topic (TAT) model is proposed for microblog topic detection. Experimental results show that, the new model fits specially into microblog when information about timestamp and author is incorporated. We further make use of the thread information and propose a two-phase approach. Intra-thread topic detection is first executed to clean every thread, and then inter-thread topic detection is run to find global topics more precisely with bigger thread texts. The notable contribution lies in that the serious sparse data problem in microblog processing is alleviated to great extent.

However, the reported work is still preliminary. In the future, we will conduct full evaluation with microblog text in multiple languages. Meanwhile, we are aware that the maximum F measure (i.e., 31.2%) of the approach is rather low. We will incorporate various word similarity measures to achieve feature selection and document similarity in concept level.

Acknowledgments

This work is partially supported by NSFC (60703051, 60970057, 61003152), MOST (2009DFA12970) and Suzhou Municipal Foundation (SYG201030). We thank the reviewers for the valuable comments.

References

M. Blei, Y. Ng, I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3 (2003*): 993-1022.

D. Cutting, D. Karger, J. O. Pedersen, and J. W. Tukey. 1992. A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. NY: ACM, 1992, 318–329.

R. Duda and P. Hart. 1973. Pattern Classification and Scene Analysis. *John Wiley and Sons, Inc.*, New York, NY.

N. Dyke, H. Lieberman, P. Maes. 1999. Butterfly: A Conversation-Finding Agent for Internet Relay Chat. *Proc. of the 4th international conference on Intelligent User Interfaces, 1999*.

J. Ellen. 2011. All about microtext: A working definition and a survey of current microtext research within artificial intelligence and natural language processing. *Proc. of ICAART-11*.

R. Krestel, P. Fankhauser, W. Nejdl. 2009. Latent dirichlet allocation for tag recommendation. *Proceedings of the third ACM conference on Recommender system*, 2009.

Z. Liu, W. Yu, W. Chen, S. Wang, F. Wu. 2010. Short Text Feature Selection and Classification for Micro Blog Mining. *Proceedings of International Conference on Computational Intelligence and Software Engineering (CISE 2010)*, pp.1-4, 2010.

R. E. Madsen, D. Kauchak, C. Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. *ICML '05 Proceedings of the 22nd international conference on Machine learning, ACM, New York*, 2005.

B. O'Connor, M. Krieger and D. Ahn. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. *Proc. of ICWSM 2010*.

R. Papka. 1999. On-line New Event Detection, Clustering and Tracking. *Amherst: Department of Computer Science, UMASS*.

J. Peng, D. Yang, S. Tang, Y. Fu, H. Jiang. 2007. A Novel Text Clustering Algorithm Based on Inner Product Space Model of Semantic. Computer Journal, 2007, 8 (30): 1354-1363.

D. Ramage, S. Dumais and D. Liebling. 2010. Characterizing Microblogs with Topic Models. *In ICWSM'2010*.

B. Sharifi, M.-A. Hutton and J. Kalita. 2010. Summarizing Microblogs Automatically. *Proc. of NAACL-HLT'2010*: 685-688.

D. Shen, Q. Yang, J. Sun, Z. Chen. 2006. Thread Detection in Dynamic Text Message Streams. *Proc. of SIGIR'06*: 35-42.

Y. Shen, C. Tian, S. Li, S. Liu. 2009. The Grand Information Flows in Micro-blog. *Journal of Information & Computational Science 6*: 2 (2009): 683-690.

B. Sriram, D. Fuhry, E.Demir, H. Ferhatosmanoglu. 2010. Short Text Classification in Twitter to Improve Information Filtering. In Sigir'10, Performance Evaluation, 841-842.ACM.

M. Steinbach, G. Kapypis and V. Kumar. 2000. A Comparison of Document Clustering Techniques. *KDD Workshop on Text Mining*, 2000:109-111.

D. Trieschnigg and W Kraaij. 2004. TNO hierarchical topic detection report at TDT 2004. *The 7th Topic Detection and Tracking Conf.*

E. M. Voorhees. 1986. Implementing Agglomerative
Hierarchic Clustering Algorithms for Use in Document Retrieval. *Information Processing and Management*, 22(6): 465-76.

J. Wang, E.-P. Lim, J. Jiang, Qi He. 2010. TwitterRank: Finding Topic-sensitive Influential Twitterers. In *WSDM'10*.

C. Wartena and R. Brussee. 2008. Topic detection by clustering keywords. *In Proceedings of the 19th International Conference on Database and Expert Systems Application:* 54–58.

J. Xu and W. Croft. 1999. Cluster-based language models for distributed retrieval. *In Proceedings of the SIGIR 1999*: 254-261.

Y. Yang, T Pierce and J Carbonell. 1998. A study on Retrospective and On-Line Event detection. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*: 28-36.

L. Zhou and E. Hovy. 2005. Digesting Virtual "Geek" Culture: *The Summarization of Technical Internet Relay Chats. ACL 2005*: 298-305.