International Journal of Grid and Distributed Computing
Vol. 9, No. 8, (2016), pp.275-286
http://dx.doi.org/10.14257/ijgdc.2016.9.8.24

# Event Detection and Summarization on Microblogs

Jie Zhao[1], Shuhan Liu[1] and Tong Cui[2]

[1]*School of Business, Anhui University*
[2]*School of Computer Science and Technology, University of Science and Technology of China*
*zj_teacher@126.com*

***Abstract***

*Microblog has been a dominant social-network platform. Many social events spread fast on microblogging platforms, because of the high interactivity and large amounts of users of microblogging platforms. However, users can only search for interested events on microblogging platforms by keywords, yielding high redundancy of search results as well as the lack of complete description on events (owing to the limitation on the length of microblogs). Aiming to solve these problems, we propose a new approach to automatically generate event abstracts for microblog events. We summarize an event via two aspects, namely an event-description set and a user-opinion set. For extracting the event description for an event, we first partition microblogs into a set of short sentences. These sentences are further ranked by computing the relevance to the event based on a graph model and the most relevant sentences are selected as the set of event description. For extracting user opinions on an event, we propose a supervised learning model that is based on the set of long sentences extracted from microblogs. We conduct experiments on a data set consisting of 14 events crawled from the Sina Weibo. The experiment results suggest the effectiveness of the proposed approach.*

***Keywords****: microblog, spammer detection*

## 1. Introduction

Microblog has been a popular social utility recent years. Some famous platforms such as Twitter and Sina Microblog have attracted hundreds of millions of users. Everyday people post information about life, mood or opinions of some hot events. Microblog contains rich resources of news and hot topics. Besides, microblog is a more real-time media and users can easily access to the views of other people. Nowadays, people browse news transferring from traditional medias such as newspaper and website to microblog step by step. Most researches on microblog focus on its properties as a social network, while ignoring its contribution as a news media.

Generally, users could find posts related to an event through keyword search. However, because of the character length restriction, one post can hardly meet a user's requirement. In order to understand the outline of the event, users probably have to browse a large amount of posts along with redundant information inevitably. Traditional works based on microblog event analysis mostly concentrate on event detection and event tracking. Their objective was to extract events from a large microblog data set, and then attach every new event-related post to an existing event, which often ignore the description of an event after it is extracted from microblog. From the perspective of users, these tasks do not have much help to understand events conveniently. For example, we can obtain a data set related to an event "鹿儿岛 渔船 扣押 (Kagoshima fisher detain)" by event detection or keywords search, but the posts in the set are always disorder and verbose. It is probably that there are lots of posts sharing same texts about the topic "detain" and some other posts about "release" which means that people may take much time to filter useless information and read all the details of this event.

Based on above points, it is worth generating a summary for a given event data set. The summary should be concise. Meanwhile, it should cover important information as much as possible. To meet these conditions, in this paper, we present a method for generating a summary of a microblog event cluster. Besides, in order to let users understand the event better, we try to extract some "General sentences" in the data set [6]. General sentences are some informal sentences used for describing general opinions in microblog. By combining the sentences that directly describe the event with the General ones, people can receive a clear description of an event. The challenges in event summarization lie in sentences importance judging and General sentences extraction. The sentences importance judging problem refers to decide which sentence is more important to be selected into summary. Moreover, the selected sentences must be different from each other. The General sentences extraction problem refers to accurately extract a General sentence from a post. In this paper, we focus on those two problems and propose an effective framework to finally generate a meaningful summary from an event posts cluster. The major contributions of this paper can be summarized as follows:

(1) Different from traditional event detection and tracking on microblog, we pay attention to generate a summary for a given event. It can be seen as a follow-up procedure after event detection. We present a framework for event summarization on microblog posts. In this framework, we propose effective ways to solve the sentences importance judging and General sentences extraction problems.

(2) We propose a new approach for selecting the element of summary. We use a graph model to calculate the importance of the sentences. Different from previous work, we build the words graph by using dependency grammar. Then we use an algorithm to select the suitable sentences taking into consideration of both importance and novelty. We also use a supervising algorithm to extract some General sentences as a supplement of summary. In avoid of redundancy of using posts, or insufficiency of using words, we describe an event using short sentences splitting from posts.

The remainder of the paper is structured as follows. Section 2 introduces the related work. Section 3 is the description of the sentences' importance definition and calculation. Section 4 discusses the detailed algorithm for the opinion sentences extraction. Section 5 presents the experimental results, and conclusions are in Section 6.

## 2. Related Work

Event related analysis on microblog has always been a hot research topic since microblog appeared. However, most researches in this area focus on event detection or event extraction. There are few works concentrated on microblog event summarization. [1] is the first one to achieve this goal. They give a solution based on learning the underlying hidden state representation of the event via Hidden Markov Models to generate summary for certain events, i.e., American Football games. The limitation of their work is that this method is only suitable for certain event. [2] proposed an algorithm called “Phrase Refinement” to generate one sentence as a summary for a tweet event. But sometimes only one sentence is not enough for introducing an event. Rui Long et al. [3] proposed a unified workflow of event detection, tracking and summarization on microblog data. Their summarization step considered both the content coverage and evolution over time. Their summary consists of posts which may have a lot of redundant information.

A related task is automatic text summarization. Its goal is to generate a summary for documents such as news reports, articles and papers. Based on the number of target documents, this task can be divided into single document summarization [13,15] and multi documents summarization[5,9,10,14]. Daraksha Parveen et al. [4] proposed a graph-based method for extracting single document summarization which considers importance, non-redundancy and local coherence simultaneously. Piji Li et al. [5] proposed a sparse-coding-based method that calculated the salience of the text units by jointly considering news reports and reader comments for multi documents summarization. Our work can be seen as multi documents summarization. However, these approaches cannot be directly applied to microblog data. Different from general document data such as articles and news reports, a microblog post must express the topic in no more than 140 characters. Meanwhile, a post does not contain "title", "paragraph" or other structures in passages which are necessary for the documents summarization methods.

Another similar task is event description after event extraction. Some previous works use words or words tuple to describe an event. [8] uses some typical words to describe an event which requests readers to have a little background knowledge. [6] provides a 4-tuple (Time, Locations, Entities, and Keywords) structure of the detected events. Lizhou Zheng et al. extracted 5W1H-tuple to describe an event [7]. However, words only are uneasy for people to understand an event. Other works select some posts to represent an event [3]. Due to the characteristic of microblog, this method can involve much irrelevant information.

## 3. Framework of Event Detection and Summarization

### 3.1. System Architecture

In this section, we describe the details of event summarization on microblogs. Given a set of microblogs about an event, we first conduct some textual preprocessing on it. The preprocessing includes word segmentation, removing stop words and POS tagging. We define the event related sentences selection and General sentences selection tasks as ranking problem. Then we extract the event-related sentences and General sentences using different units separately.

For event-related sentences extraction, we split every microblog into several short sentences as units by recognizing punctuations. The reason for employing this step is that there is a lot of overlap among the posts, directly selecting some posts as summary will be redundant. Each part of a post (normally separated by punctuations like "，", "。" or space) may contain different aspects of an event. Taking short ones as summary will be concise and intuitive. After that, we get the dependency grammar between the words in the short units using the Stanford Parser. Then we construct a words dependency graph and using HITS algorithm to get the importance score of the words. The vertexes in the graph consist of the words and the edges are the dependency between words. Finally, we calculate the score of a unit by summing all the words importance score it contains. Top 50 units will be selected as candidates. We use the MMR (maximal marginal relevance) to rank the event related sentences and select top n units as event introduction.

For General sentences extraction, we split the post only by ".". Because different from event-related sentences, opinion sentences always need more complex information which cannot be contained by short sentences. What is more, other punctuations like '?' or '!' always represent opinion tendency. Then we extract some useful features and use Logistic Regression to rank the candidate sentences, select some of them as General sentences subset. In the following sections, we describe the details about the sub-routines respectively.

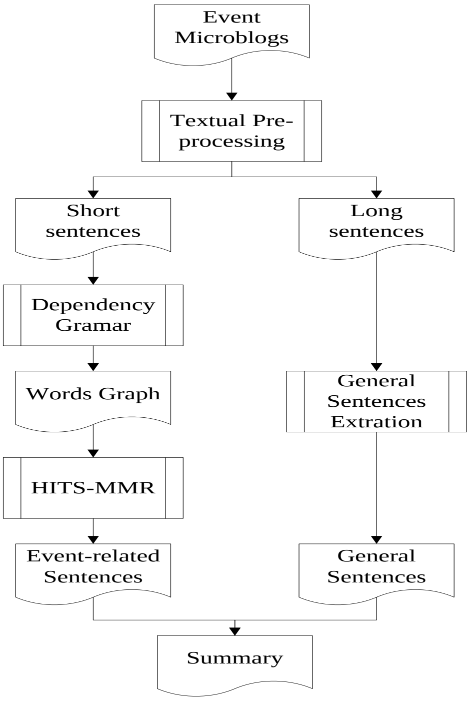


**Figure 1. The General Framework of Event Summarization On Microblog**

### 3.2. Textual Pre-processing

In this step, we pay attention to recognize punctuations for sentences partition. In particular, for event related sentence extraction, most of the short sentences less than 5 words are discarded. But a fraction of them which contain time or geographical position would be retained. These short sentences will be combined to the adjacent units. Stop words are also removed in this step.

### 3.3. Sentence Importance Judging

In order to select the most important sentences to generate our summary, we use a graph based model, HITS (Hypertext-Induced Topic Search), to rank our sentences. HITS has been adopted by many researchers for automatic summarization [4,13]. As a result, there have been many solutions to construct the graph. The vertex or node in the graph can be a word, a sentence or even a document. The edge between vertexes can be generated by words co-occurrence or text similarity. In this paper, considering that there is much semantic information existed between words, we adopt words as our vertexes and use directed edge to describe the relation between words. The edge between two vertexes is generated by dependency grammar technique. Dependency grammar is used to describe the dependency relations between words in a sentence. Each word is linked to another word with a special relation. Compared to words co-occurrence which may be meaningless, the word pairs produced by Dependency grammar contain more semantic information. For example, in the sentence “this car has a fantastic shape”, the word “fantastic” has a dependency relationship with “shape”. A lot of words pairs like <shape, fantastic> are generated by dependency grammar technique. "Shape" is defined as *governor* and “fantastic" is defined as *dependent*. A number of dependency relations are introduced in the dependency grammar. In the graph, we construct a directed edge from dependent to governor. There have been many tools that can be used to extract dependency relations of a sentence, such as Stanford Parser. Thus, in our study, we simply use the Stanford Parser for dependency grammar analysis. We select some typical ones to construct our graph such as "nsubj", "advmod". All of our dependency relations are shown in Table 1. These relations cover nouns, verbs and adjectives which play main role in summarizing an event.

**Table 1. Dependency Relations**

| Dependency Relation | Instruction |
|---|---|
| nsubj | nominal subject |
| advmod | adverb modifier |
| dobj | direct object |
| nsubjpass | passive nominal subject |

Part of a word graph is shown in Figure 2.

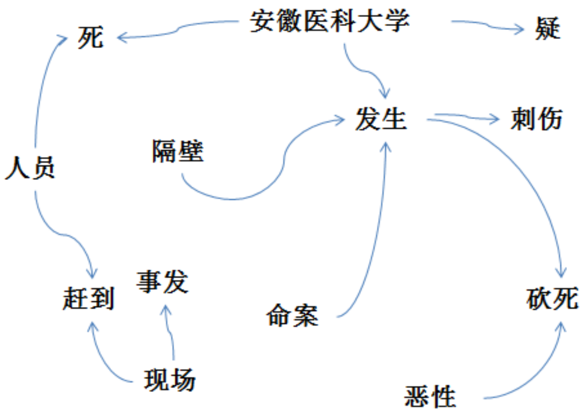


**Figure 2. Example of A Word Dependency Grammar Graph**

After the words graph is constructed, we apply HITS to our words graph. HITS is firstly used in optimizing the search performance on the Web. This graph based algorithm has two assumptions: (1) a node with a large number of outgoing links must be important. (2) a node with a large number of incoming links must be important. The former is called *hubs* and the latter is called *authorities*. The score of a vertex can be calculated as follows:

$$HITS_A(V_i)=\sum_{V_j\in In(V_i)} HITS_H(V_j) \quad (3.1)$$

$$HITS_H(V_i)=\sum_{V_j\in Out(V_i)} HITS_A(V_j) \quad (3.2)$$

After iterating the HITS until convergence, we can get two scores of a word. We sum the hub and authority as the final score of a word. Then we sort the units based on the sum of all the words score it contains. The score of a short sentence is calculated as follows:

$$Score(S_i)=\sum_{w_j\in(S_i)} Score(w_j) \quad (3.3)$$

```
Algorithm Words_HITS()
Input:   an adjacency matrix of a word graph G
Output: the hub score and authority score of all the vertexes in G.
for each word $w_i$ in G.vertices do
    $w_i$.auth = 1
    $w_i$.hub = 1
for step from 1 to k do   //run the algorithm for k steps
    norm = 0
    for each word $w_i$  in G do
        $w_i$.auth = 0
        for each word $w_j$ in $w_i$.incomingNeighbors do
            $w_i$.auth += $w_j$.hub;
        norm += square($w_i$.auth)
    norm = sqrt(norm)
    for each word $w_i$ in G do
        $w_i$.auth = $w_i$.auth / norm
    norm = 0
    for each word $w_i$ in G do
        $w_i$.hub = 0
        for each word $w_j$ in $w_i$.outgoingNeighbors do
            $w_i$.hub += $w_j$.auth
        norm += square($w_i$.hub)
    norm = sqrt(norm)
    for each word $w_i$ in G do
        $w_i$.hub = $w_i$.hub / norm
return hub score + authority score of the vertexes in G
End Words_HITS()
```

**Figure 3. Algorithm for HITS on Words Graph**

### 3.4. TF-IDF

We also adopt the classical TF-IDF to compute the importance of words as a comparison. Term Frequency Inverse Document Frequency (TF-IDF) has been applied to solve many problems. Here, we use a short sentence to refer to a document in the conventional sense. Note that the sentence is too short to contain enough terms, the terms appeared in the sentence almost appear only once. We compute the TF-IDF score of words in a sentence and add up all the scores as the score of a sentence. Once each sentence has been weighted, the sentences are ordered by their score from which the top k sentences with the most score are chosen as the result set.

## 4. Sentences Selection

### 4.1. Event Related Sentences Selection

We can select the top K sentences ranking by HITS score as summary. This method has been employed by some previous work [13]. However, after microblog splitting and duplicate removal, there is still a cast of potentially relevant short sentences, highly redundant with each other. Because the words scored higher probably appear in more than one sentence and different people tweet in different way for one thing. That is, the summary tends to describe one aspect of an event and ignored others. For example, if the words' score calculated by HITS are "渔船 /

0.5", "扣押 / 0.4", and "释放 / 0.2". The sentences contain "渔船" and "扣押" will be ranked in the front while the sentences contain "释放" will be in the back. If there're lots of sentences containing "扣押", the summary may have a lot redundant information about "扣押" and no information about "释放". It is obviously that the summary generated by this way cannot show the complete details of an event.

We attempt to solve this problem by using an algorithm called MMR (Maximal Marginal Relevance).The Maximal Marginal Relevance criterion strives to reduce redundancy while maintaining query relevance in re-ranking retrieved documents. It is also widely used in selecting appropriate passages for text summarization. MMR is originally defined as follows:

$$MMR = arg \max_{D_i \in R \setminus S} \left[ \lambda\, sim\left(D_i, Q\right) - \left(1-\lambda\right) \max_{D_j \in S} sim\left(D_i, D_j\right) \right] \tag{4.1}$$

The equation assures the sentence which has higher similarity with query and lower similarity with existing sentences could be ranked at the top. The weight of relevance and diversity is controlled by λ . A larger λ means a more important role of relevance. In this paper, we combine MMR with HITS to be suitable for summarization instead of IR. Our HITS-MMR is defined as follows:

$$MMR = arg \max_{D_i \in R \setminus S} \left[ \lambda Score(D_i) - \left(1-\lambda\right) \max_{D_j \in S} sim\left(D_i, D_j\right) \right] \tag{4.2}$$

The score ($D_i$) means the sentence's score calculated by HITS. The sim($D_i$,$D_j$) is calculated by text cosine similarity. We run MMR based on the sentences' HITS score. Firstly, we choose a sentence that scored highest as the initial summary set. Then we run our MMR to re-rank all the sentences. At last, we select the top K sentences to form the final summary. Another version of MMR called TFIDF-MMR could be obtained by replacing the HITS score in equation(4.2) with TFIDF score.

### 4.2. General Sentences Selection

Compared to the traditional formal news media, author of microblog tends to use some General sentences to describe an event. For example, "中国渔船被日本扣押了，敢不敢不给国人丢脸"（A Chinese fishing boat is detained by Japan. Please don't bring disgrace on Chinese）. "中国渔船被日本扣押了"（A Chinese fishing boat is detained by Japan) is a description on the event. "敢不敢不给国人丢脸" (Please don't bring disgrace on Chinese) is a General sentence. It is less rigorous but it expresses the attitude of author towards the event. We attempt to extract these General sentences so that readers can know more about an event. It can be regarded as an opinion sentences extraction problem.

```
Algorithm MMRonHITS()
Input:   the short sentences set $S$ with score
Output: result set $R$
Preliminary: $k$ is the size of the result set $R$.
select the highest score sentence in $S$ as $s_{max}$
initial $R$ with $s_{max}$
$S = S \setminus s_{max}$
Loop
    for each sentence $s_i$ in $S$ Do
        $sim(s_i, R) = max\ \{(s_i, s_j \text{ in } R)\}$
        $m_i = \lambda \cdot s_i.score - (1-\lambda) \cdot sim(s_i, R)$
    $R = R \cup \{ max(m_i)\}$
    $S = S \setminus max(m_i)$
Until $size(R)=k$
return $R$
End MMRonHITS()
```

**Figure 5. Algorithm of MMR on HITS**

There have been some mature studies in sentiment sentences extraction of reviews or opinion sentences extraction of microblog. Most of them can be divided into two types. One is rule-based, another is machine learning. Compared to sentiment sentences extraction of reviews which have a more formal sentence structure, the opinion sentences in microblog are always ill-formed. Thus it is hard to make effective rules for them. Based on this, we use the machine learning technique. We design 6 simple features to train our model. The features we used are shown in Table 2.

We use Logistic Regression (LR) to classify our sentences. In order to find a suitable subset of General sentences, we rank the sentences based on the probability of positive label from the output of LR, i.e., the sentence ranked in front is more likely to be a General one. We select top 20 as candidates. After that, we re-rank them based on their HITS score. As we have mentioned before, the sentence with higher HITS score has more information of event description. Finally we select top 10 as our General sentences set.

**Table 1. Details of the Features**

| features | |
|---|---|
| the number of sentiment words in sentiment words dictionary(HOWNET、NTUSD) | the number of interjection |
| the number of punctuations | the number of adverbs |
| the number of pronouns | the number of modal particles |

## 5. Performance Evaluation

### 5.1. Dataset

To evaluate our methods, we prepared a dataset crawling from Sina Weibo. The dataset contains posts 14 specific events such as Hangzhou fire/ 杭州 火灾, Henan kill/ 河南 砍伤, etc. We crawled each event by searching for posts containing several keywords about the specific event. Each event set contains about 200 ~ 500 microblogs. Table 3 shows the keywords about the dataset.

**Table 3. Details of the Dataset**

| keywords | | | |
|---|---|---|---|
| 韩旅行团 失踪 | Korea tour group missing | 昆明 航班 | Kunming flight |
| 安徽医科大学 砍死 | An hui medical university kill | 李克强 恩施 | KeQiang Li Enshi |
| 贵溪 校车 | Guixi school bus | 河南 砍伤 | Henan kill |
| 杭州 火灾 | Hangzhou fire | 深圳 砍人 | Shenzhen wounding |
| 河北 烧车 | Hebei car burning | 福建 贩卖 婴儿 | Fujian selling baby |
| 河南 火灾 | Henan fire | 习近平 保定 | JinPing Xi Baoding |
| 吴凡 警察 | Wufan police | 彝良 陪酒 | Yiliang drink |

### 5.2. Event Summary Generation

The evaluation of summarization systems is a research issue in itself. In the traditional multi-document summarization task, ROUGE is a widely used standard which measures the similarity between a generated summary and some manual labeled summaries of experts[23]. In this paper, since we do not have manually labeled summary in hand, we reference the method mentioned in [3] which uses *precision@n* (Pre@n) to assess the relevance of an event summary. Note that n denotes the number of posts selected to represent an event.

We compare two algorithms. One is the HITS-MMR which uses HITS to calculate score and uses MMR to rerank the sentences. Another one named TFIDF-MMR takes the TFIDF as score and use MMR to rerank the short sentences. We run the two algorithms on 14 events sets.

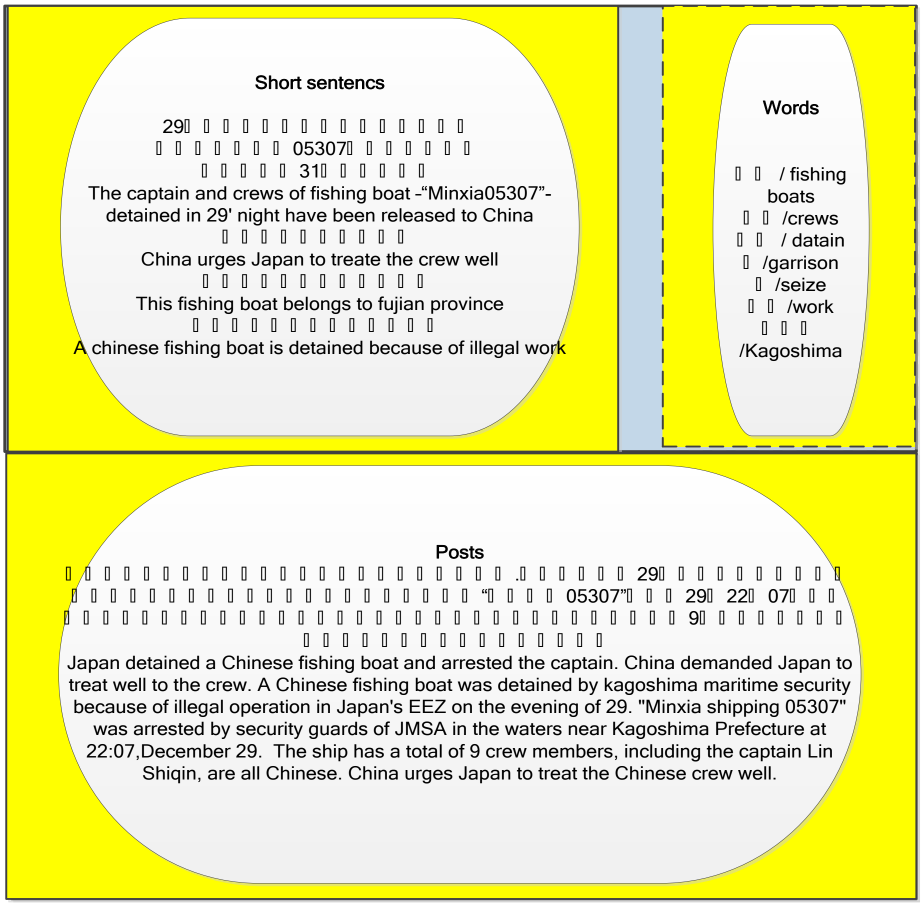
For evaluation, we firstly invite two volunteers to read some passage about an event. Then we mix up the result sets generated by the two algorithms. After that the volunteers manually label each sentence with "1" and "0". "1" means the sentence is relevant to the event and "0" means the sentence has nothing to do with the event. Finally we count the number of "1" in top K labeled by both of them, namely Pre@K. Due to the page limit, we demonstrate the first 5 events' precision in 10~50 in Table 4, name them e1~e5 together with average evaluation on 14 events to show the performance of our method.

From the table we can see that HITS-MMR is better than TFIDF-MMR in most cases. In fact, TFIDF is better than HITS only twice. We can come to the conclusion that HITS is a feasible solution.

There is no metrics for measuring the basic units of summary. Hence, we just analyze the advantage of using short sentences as summary units. Previous works use either words or full posts as units. For words, it is hard for readers to known the event without background knowledge. For posts, although this kind of summary can be easily understood by readers but lots of repeated information would be generated, especially in microblog. We use short sentences to generate summary which abandons a lot of useless information. Readers can see the content of event at the same time. From Figure 6. we can see that using words as summary is unintelligible. Meanwhile, using a post as summary has a lot of repeated information and it does not contain the content about "释放 / release" which is one-side.

**Table 4. Results of Summary Generation**

| Data set | HITS-MMR | | | | | TF-IDF-MMR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pre@ 10 | Pre@ 20 | Pre@ 30 | Pre@ 40 | Pre@ 50 | Pre@ 10 | Pre@ 20 | Pre@ 30 | Pre@ 40 | Pre@ 50 |
| e1 | 0.9 | 0.85 | 0.7 | 0.73 | 0.64 | 0.7 | 0.65 | 0.7 | 0.63 | 0.62 |
| e2 | 0.8 | 0.7 | 0.77 | 0.75 | 0.66 | 0.6 | 0.6 | 0.6 | 0.58 | 0.56 |
| e3 | 1.0 | 0.95 | 0.97 | 0.93 | 0.88 | 0.9 | 0.75 | 0.77 | 0.8 | 0.84 |
| e4 | 0.9 | 0.75 | 0.83 | 0.88 | 0.82 | 0.7 | 0.6 | 0.67 | 0.68 | 0.68 |
| e5 | 0.9 | 0.85 | 0.83 | 0.73 | 0.68 | 0.7 | 0.55 | 0.6 | 0.55 | 0.6 |
| avg14 | 0.88 | 0.83 | 0.82 | 0.78 | 0.73 | 0.7 | 0.64 | 0.65 | 0.63 | 0.63 |



Short sentences

The captain and crews of fishing boat -"Minxia05307"-detained in 29' night have been released to China

China urges Japan to treate the crew well

This fishing boat belongs to fujian province

A chinese fishing boat is detained because of illegal work

Words

/ fishing boats

/crews

/ datain

/garrison

/seize

/work

/Kagoshima

Posts

Japan detained a Chinese fishing boat and arrested the captain. China demanded Japan to treat well to the crew. A Chinese fishing boat was detained by kagoshima maritime security because of illegal operation in Japan's EEZ on the evening of 29. "Minxia shipping 05307" was arrested by security guards of JMSA in the waters near Kagoshima Prefecture at 22:07,December 29. The ship has a total of 9 crew members, including the captain Lin Shiqin, are all Chinese. China urges Japan to treat the Chinese crew well.

**Figure 6. Summary of Three Different Units**

### 5.3. General Sentence Extraction

Because we take a supervised learning method, we need a train set to learn our model. Our event data set is unlabeled and it is time consumed to find all the positive samples. Based on this, we use a labeled data set from NLPCC2012. It has a task of opinion sentences extraction from microblog which is similar to ours' and it provides a labeled train set.

We take a similar measure to evaluate the performance of opinion sentences extraction. After we get the result from opinion sentences extraction, we manually judge whether or not a sentence in the subset is an opinion one. Then we random select ten sentences as comparison. And part of the result is shown in Table 5.

**Table 5. Results of General Sentence Extraction**

| Dataset | Our algorithm | Random |
|---|---|---|
| | Pre@10 | Pre@10 |
| e1 | 0.9 | 0.4 |
| e2 | 0.8 | 0.6 |
| e3 | 0.7 | 0.5 |
| e4 | 0.7 | 0.6 |
| e5 | 0.7 | 0.5 |

## 6. Conclusion

In this paper, we concentrated on the design of an event summarization framework for microblog event. We proposed a graph model based on dependency grammar to calculate the importance of sentences. In order to help users learn more, we also extracted some opinion sentences from the event set. In particular, our works were based on short sentences instead of posts or words. The experimental results on 14 microblog event datasets demonstrated the superiority of our methods when compared to the baseline methods. In future, we will try to analyze event trends based on the event summary and opinions.

## Acknowledgments

This paper is partially supported by the National Science Foundation of China (No. 71273010) and the Doctor Start-up Fund of Anhui University.

## References

[1] D. Chakrabarti, K. Punera. Event Summarization Using Tweets. ICWSM, **(2011)**, 66-73.
[2] B. Sharifi, M. A. Hutton, J. Kalita. Summarizing microblogs automatically. HLT-NAACL, **(2010)**, 685-688.
[3] R. Long, H. Wang, Y. Chen. Towards effective event detection, tracking and summarization on microblog data. WAIM, **(2011)**, 652-663.
[4] D. Parveen, M. Strube. Integrating Importance, Non-Redundancy and Coherence in Graph-Based Extractive Summarization. IJCAI, **(2015)**, 1298-1304.
[5] P. Li, L. Bing, W. Lam. Reader-Aware Multi-Document Summarization via Sparse Coding, IJCAI, **(2015)**, 1270-1276.
[6] Y. You, G. Huang, J. Cao. GEAM: A general and event-related aspects model for twitter event detection. WISE, **(2013)**, 319-332.
[7] L. Zheng, P. Jin, J. Zhao, L. Yue. A Fine-Grained Approach for Extracting Events on Microblogs. DEXA (**2014**), 275-283.
[8] Z. Liu, W. Huang, Y. Zheng. Automatic key phrase extraction via topic decomposition. EMNLP, (**2010**), 366-376.
[9] J. Yao, X. Wan, X. Jiao. Compressive Document Summarization via Sparse Optimization. IJCAI, **(2015)**, 1376-1382.
[10] S. Banerjee, P. Mitra, Sugiyama K. Multi-Document Abstractive Summarization Using ILP based Multi-Sentence Compression[.IJCAI, **(2015)**, 1208-1214.
[11] W. Yin, Y. Pei. Optimizing Sentence Modeling and Selection for Document Summarization. IJCAI, **(2015)**, 1383-1389.
[12] P. Wang, H. Wang, M. Liu. An algorithmic approach to event summarization. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, (**2010**), 183-194.
[13] M. Litvak, M. Last. Graph-based keyword extraction for single-document summarization. Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization. Association for Computational Linguistics, (**2008**), 17-24.
[14] M. A. Dhelaan. StarSum: A Simple Star Graph for Multi-document Summarization. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, (**2015**), 715-718.
[15] Z. Wei, W. Gao, Assistant, or Master? Using Tweets Linking to News for Extractive Single-Document Summarization. SIGIR, **(2015)**, 1003-1006.

[16] H. D. Kim, M. G. Castellanos, M. Hsu. Ranking explanatory sentences for opinion summarization. SIGIR, **(2013)**, 1069-1072.

[17] P. Jin, Y. Yu, J. Zhao, L. Yue. Extracting Appraisal Expressions from Short Texts. WAIM, (2015), 481-485.

[18] L. Zheng, P. Jin, J. Zhao, L. Yue. Multi-dimensional Sentiment Analysis for Large-Scale E-commerce Reviews. DEXA, (**2014**), 449-463.

[19] G. Erkan, D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, (**2004**), 22: 457-479.

[20] J. Carbonell, J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, (**1998**), 335-336.

[21] P. C. Chang, H. Tseng, D. Jurafsky. Discriminative reordering with Chinese grammatical relations features. Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation. Association for Computational Linguistics, (2009), 51-59.

[22] M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), (**1999**), 46(5): 604-632.

[23] C. Y. Lin. Rouge: A package for automatic evaluation of summaries. Text summarization branches out: Proceedings of the ACL-04 Workshop. (**2004**), 8.