Efficient Location-based Event Detection in Social Text Streams

Abstract. Social networks provide a wealth of online sources about real-world events. Due to the large volume of data in social streams, the event detection suffers from high computational complexity. In this work, we present a location-based event detection approach using Locality-Sensitive Hashing to accelerate the similarity comparison. We use this approach to detect real-world events from Sina Weibo by clustering microblogs with high similarities. We propose a message-mentioned location extraction method based on the textual content based on Part-of-Speech tagging and a Support Vector Machine classifier and a novel similarity measurement considering content, location, and time between messages to improve the precision of event detection. We compare our approach with the state-of-the-art baselines on event detection, and demonstrate the effectiveness of our approach.

Keywords: Event detection, Location extraction, Social networks, Microblogs

1 Introduction

Currently, social networks, also known as the User Generated Content (UGC) platforms, provide a wealth of online sources about real-world events. Popular social media services, such as Twitter and Sina Weibo, allow people to report and share short messages (limited to 140 characters) about what is happening in their daily lives. Clearly, we can benefit from real-time event detection from social messages to support emergency managements and damage control.

The event, in this paper, is referring to an actual occurrence that happens at some specific time and place[1]. Under this definition, the social events range from widely known ones such as natural disasters or political affairs to local ones such as accidents or crimes. For instance, in Sina Weibo, there are a large number of messages discussing about the serious fire which burned 7 grain barns of SINOGRAIN on June 2, 2013, and also a certain amount of messages reporting the car accident near Nanchang University Commercial Street on the same day. Our research interest is to detect real-world events including both widely-known ones and local ones via monitoring the social text stream.

The majority of works in event detection from social messages rely on clustering algorithms[2,6,7,9,11,14], keyword co-occurrence graph[8,13] and topic models[3,15,16]. Generally, clustering algorithms have relatively high computational complexity due to the large volume of data, and thus the process of event detection is inevitably time consuming. To address the problem of time delay, researchers applied hashing algorithm to accelerate the similarity comparison[4,5]. The proposed algorithms based on Locality-Sensitive Hashing (LSH) only consider the cosine

similarity of text contents between message pairs but fail to take spatial and temporal similarity into account.

In this paper, we present a location-based event detection approach using LSH to avoid pair wise similarity computation. We use this approach to detect real-world events from Sina Weibo by clustering microblogs with high content and location similarities under the time constraint. The research challenges of our work are: (1) the amount of social message is huge, and we need to process the data in bounded space and time; (2) social messages are very noisy, because users can talk about whatever they choose, and it is often difficult to identify whether they are truly describing a real event; (3) the event location extraction is a challengeable problem, because the GPS tags and the registered locations in user profiles can only indicate where the message was sent out and where the user often hung around, and meanwhile it is difficult to extract message-mentioned locations from the text.

Our main contributions include: (1) a real-time event detection approach using LSH to accelerate the similarity comparison, (2) a message-mentioned location extraction method based on Part-of-Speech (POS) tagging and a Support Vector Machine (SVM) classifier, (3) a novel similarity measurement considering content, location, and time to improve the precision of event detection. We compare our approach with two state-of-the-art baselines on event detection, and demonstrate the effectiveness of our approach.

The remainder of this paper is organized as follows. Section 2 introduces some related work. Section 3 introduces the scheme of event detection in social streams and the proposed methods. We evaluate the performance of proposed methods in Section 4 and we finally conclude our work in Section 5.

2 Related Work

2.1 Social Event Detection

Event detection in social networks has received considerable attention in the fields of data mining and knowledge discovery[3]. The most common approach for event detection is text-based clustering, and a variety of clustering algorithms are applied, such as hierarchical clustering[2], single-pass incremental clustering algorithm with threshold[6,9,11,14], density-based clustering algorithm[7]. However, the pair-wise similarity comparison during the process of clustering is very expensive, and thus we need to limit the number of similarity comparison between messages by firstly finding candidate similar items.

Research [4] presented a first story detection method based on LSH to overcome the limitations of traditional method which relied on pair-wise similarity comparison. The proposed method used the hashing scheme[10] in which the probability of two messages colliding was proportional to the cosine of the angle between them. Besides, a variance reduction strategy was introduced to reduce the false alarm rate returned by LSH. Further, research [5] incorporated paraphrase information in the LSH scheme proposed in [4] to improve the performance of streaming first story detection. Both

these two research only focused on the content text similarity between social messages without considering the location similarity and time similarity under the LSH scheme.

In this paper, we present our LSH scheme for streaming event detection based on the method proposed in [4], and furthermore extend the scheme to take both content and location similarities into consideration under the time constraint.

2.2 Event Location Extraction

An event location is a place where the event happened or is happening, and the message-mentioned location is the location mentioned in the text content of a message. According to research [2], compared to the GPS-tagged locations and the user profile locations, the message-mentioned locations are much more likely to be the actual event locations. Therefore, we use message-mentioned locations to identify event locations in this paper.

Researches on event location extraction are still relatively limited. [17] utilized a multinomial naive Bayes classifier to predict user-level location for each event-related tweet. [18] proposed a method to automatically identify location keywords and further estimate a Twitter user's city-level location based purely on the textural contents. [19] presented a method to predict the POI tag of a tweet based on its textual content and time of posting by using ranking algorithm and web pages retrieved by search engines as an additional source of evidence. All approaches above focused on assigning one or more locations from the existing geo-name datasets to a social message, while our work attempts to extract the message-mentioned locations from the text content of a message. According to current knowledge, our work is the first try to address this problem.

3 Event Detection in Social Streams

3.1 Text Stream Clustering based on LSH

In this paper, we detect real-world events from Sina Weibo by clustering similar microblogs. In order to accelerate the process of clustering and avoid pair-wise similarly comparison between messages, for each newly arrived social message, we firstly use LSH to hash the message into the same bucket with its candidate similar messages, and then compute the similarly between the new message and each message in its candidate set. If the similarity between the new message and its nearest neighbor is higher than the pre-specified threshold, the new message is assigned to the same cluster to which its nearest neighbor belonged.

Basically, our method extends the LSH scheme proposed in [4] by using two kinds of hash functions instead of one. The first put messages into the same bucket only if they have high cosine similarity in their textual content, which is the hash function utilized in [4]; the second put messages into the same bucket only if they have high Jaccard similarity[12] in their Message-Mentioned Location (MML) sets.

Under the LSH scheme for the cosine similarity in textual contents of messages ($LSH_{(content)}$, for short), the set of messages that collide with a newly arrived message $S_{content}(m)$ is defined as :

$$S_{content}(m) = \{m' : h_{ij}(m') = h_{ij}(m), \exists \in i[1 \ldots L], \forall j \in [1 \ldots k]\} \quad (1)$$

where x is the Vector Space Model (VSM) vector of a message with TFIDF term weights, L is the number of hash tables, k is the number of bits per key in the hashing scheme, and the hash functions h_{ij} are defined as :

$$h_{ij}(x) = \text{sgn}(u_{ij}^T x) \quad (2)$$

where the random vectors $u_{ij}$are drawn independently for each i and j by sampling a Gaussian function with mean 0 and variance 1.

Under the LSH scheme for Jaccard similarity in MML sets ($LSH_{(MML)}$, for short), the set of messages that collide with a new arrived message $S_{MML}(m)$ is defined as:

$$S_{MML}(m) = \{m' : h_{ij}(m') = h_{ij}(m), \exists \in i[1 \ldots B], \forall j \in [1 \ldots r]\} \quad (3)$$

where B is the number of bands, r is the number of rows per band in the hashing scheme, and the hash function h is the Minhashing function which was introduced in [12].

Furthermore, for the candidate similar messages which are sent into the same bucket by the two LSH schemes ($LSH_{(content)}$, and $LSH_{(MML)}$), we need to compute the pair-wise similarity between two messages using the similarity measurement which will be described in more detail in Section 3.3.

The pseudo code shown in Algorithm 1 summarizes our text stream clustering approach based on LSH.

```
Algorithm 1: LSH-based text stream clustering approach
input: threshold t
foreach message m in social stream do
  add m to LSH(content)
  Scontent(m) ←set of messages that collide with m in LSH(content)
  add m to LSH(MML)
  SMML(m) ←set of messages that collide with m in LSH(MML)
  Simmax(m) ←0
  NearestNeighbor(m) ←∅
  foreach message m' in Scontent(m)∪SMML(m)
    c=Sim(m,m')
    if c> Simmax(m) then
      Simmax(m) ←c
      NearestNeighbor(m) ←{m'}
    end
  end
  if Simmax(m)<=t then
    m is the first message of a new event cluster
  end
  assign m to EventCluster(NearestNeighbor(m))
```

```
  add m to InvertedIndex(content)
  add m to InvertedIndex(MML)
end
```

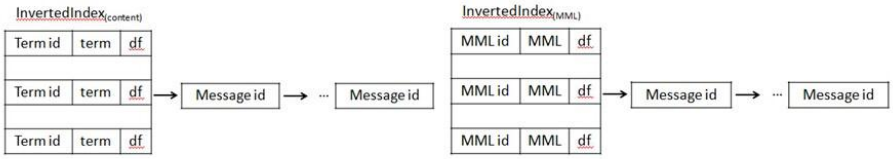

Fig. 1. The structures of $\text{InvertedIndex}_{(\text{content})}$ and $\text{InvertedIndex}_{(\text{MML})}$

We build indices to avoid unnecessary computation of the messages that have been processed. As shown in Figure 1, two indices are kept for all the saved messages: an inverted index of the textual content and an inverted index of the MML. The $\text{InvertedIndex}_{(\text{content})}$ has an entry for each term in the vocabulary, which is built though word segmentation and stop word elimination. The entry for term t is a linked list of the message ids of all the messages whose textual content contained t. The $\text{InvertedIndex}_{(\text{MML})}$ has an entry for each MML extracted from messages by using the method which will be described in more detail in Section 3.2. The entry for an MML is a linked list of the message ids of all the messages whose textual content mentioned it. The linked lists of message ids in both indices are sorted in descending order of messages' arrival time.

3.2 Message-mentioned Location Extraction

The investigation in [2] shows that the message-mentioned locations are much more likely to be the actual event locations, compared to the GPS-tagged locations and the user profile locations. Therefore, we use message-mentioned locations to identify event locations in this paper. However, the location extraction from text is one challenging problem.

Generally, by using POS tagging and Named Entity Recognition (NER) tools, such as ICTCLAS [1] and FudanNLP [2], all country-level, province-level and city-level locations can be identified after POS-tagging. We take the locations identified by POS tagging and NER tools from textual contents as one part of the MMLs.

However, these tools often fail to identify street-level locations, which are the other important part of the MMLs. For example, here is a message, such as "I see a car accident happens near Nanchang University Commercial Street.". ICTCLAS can identify the city-level location "Nanchang", but it fails to identify the street-level location "Nanchang University Commercial Street". To address this problem, we

[1] http://ictclas.org/
[2] http://jkx.fudan.edu.cn/nlp/

present a novel method based on POS tagging and a SVM classifier to extract Street-Level Message-Mentioned Locations (SLMMLs) from textual contents.

We use the classifier to predict one label out of four {B,M,E,N} for each term in the text of a message to indicate that whether this term is the beginning of a SLMML phrase (labeled as 'B'), or in the middle of a SLMML phrase (labeled as 'M'), or the end of a SLMML phrase (labeled as 'E'), or nothing to do with any SLMML phrase (labeled as 'N'). Figure 2 shows the SLMML phrase in the message "I see a car accident happens near Nanchang University Commercial Street" and the labels given by the SVM classifier.

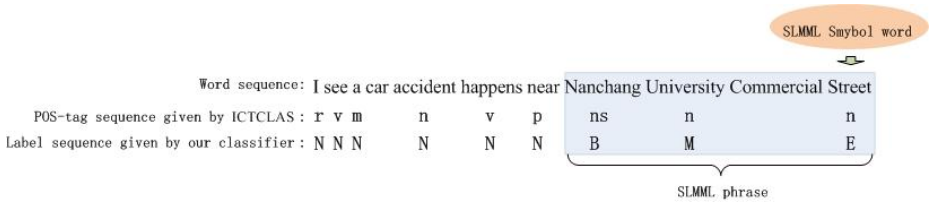

Fig. 2. The SLMML phrase and the labels given by our classifier

The classifier for labeling SLMML phrases is based on the POS tags given by POS tagging tools and a group of heuristic features. The features are described in Table 1.

Table 1. Heuristic features for labeling SLMML phrases

Feature	**Definition**
PosTag	The POS tag of term t
PosTagBefore	The POS tag of the term before t
PosTagAfter	The POS tag of the term after t
LabelAfter	The SLMML label of the term after t
IsSymWord	Where term t is a smybol word of SLMML phrases

We manually annotate 383 phrases of SLMML phrases from 370 event-related messages as the training set. Then, we extract 65 words which are commonly used at the end of SLMML phrases in the training set, such as "Road", "Bridge", "Avenue", "Square", "Street", and so on. We designate these words as the symbol words which can indicate SLMMLs. We construct the feature vectors according to the definitions in Table1, and train the SVM classifier using the training set. Finally, we use the extracted phrase as keywords to search the most similar geo-name in our Geo-Names dataset, and we take the most similar search result as the SLMML of the message. If there are not any similar geo-names in the dataset, we save the SLMML phrase as a new entry after the manual review.

3.3 Similarity Measurement for Event Detection

In this section, we propose a novel similarity measurement considering content, location, and time similarities between social messages for pair-wise similarity comparison between candidate similar messages in the union of $S_{content}(m)$ and $S_{MML}(m)$ returned by $LSH_{(content)}$ and $LSH_{(MML)}$). The proposed formula combines content, spatial and temporal dimensions. Supposed that we have a newly arrived message m, and its candidate near neighbor m', the similarity between m and m' can be denoted as follows:

$$Sim(m,m') = \cos(m,m') \times sp(m,m') \times tp(m,m') \tag{4}$$

$$\cos(m,m') = \sum_i \frac{m_i \times m'_i}{|m||m'|} \tag{5}$$

$$sp(m,m') = \frac{|MML(m) \cap MML(m')|}{|MML(m) \cup MML(m')|} \tag{6}$$

$$tp(m,m') = e^{\frac{\zeta|time(m)-time(m')|}{W}} \tag{7}$$

where the cosine similarity is used for content-based similarity measurement. To make the considerations of spatial and temporal similarities, the spatial penalty *sp(m,m')* is the Jaccard similarity between the MML sets of two messages, and the temporal penalty *tp(m,m')* is an exponential distribution which can reduce the similarity if the time distance between two messages is long. The parameter ζ can adjust the temporal decay rate, and W is the size of the sliding time window.

4 Experiments and Evaluation

4.1 Evaluation of Message-mentioned Location Extraction

We manually annotate 383 SLMMLs from 370 microblogs from Sina Weibo as the training set, and another 272 SLMMLs from another 312 microblogs as the test set.

As described in this paper, the task of SLMML extraction is to extract out the phrases describing SLMMLs from the text content of a message. To evaluate the performance of proposed method, we compare the phrase extracted by our method and the phrase annotated manually. If the difference is no more than one word, then we consider the phrase extracted by our method is precise, because the difference of one word can be corrected by using geo-names dataset in practical application or applying fuzzy matching strategy for phrases. Moreover, we use the recall to measure whether our proposed method can extract each SLMML from the text content. Table 2 shows the Precision, Recall and F-value using different features for SLMML extraction described in Table1.

Table 2. Evaluation of SLMML Extraction

Feature	Precision	Recall	F-value
Pos	37.5%	55.8%	44.8%
PosBefore	42.1%	63.1%	50.5%
PosAfter	80.1%	83.2%	81.6%
LabelAfter	25.7%	33.5%	29.0%
IsSymWord	29.3%	43.3%	34.9%
All proposed features	**90.3%**	**87.6%**	**88.9%**

The experiment results show that the highest F-value is achieved on the set test when all proposed features are used for SLMML label prediction, and the proposed method is effective in SLMML extraction.

4.2 Evaluation of Event Detection based on LSH

In order to evaluate our approach, we collected microblogs between June 1, 2013 and June 3, 2013 from Sina Weibo to simulate a live social text stream. The search keywords that we used for data collection are "car accident", "fire", and "earthquake ", because messages containing these keywords may be related to actual important events. We collected 257872 messages containing the search keywords. Since it is impractical to manually label the overly large number of messages in the dataset, we labeled the messages as relevant and irrelevant in 5 event clusters, which are detected by our approach and both two baselines. The precision of one event cluster is defined as:

$$\text{precision(event cluster)}=\frac{\left|\text{relevent message set of an event}\right|}{\left|\text{detected message set of an event}\right|} \tag{8}$$

We also use the definition of the precision used in [2] to evaluate the ability of our approach to detect real-world events, which is defined as follows:

$$\text{precision(real-world event detection)}=\frac{\text{the number of real-world events}}{\text{the number of detected events}} \tag{9}$$

In the experiment, we compare our proposed approach with two state-of-the-art baselines. **Baseline 1** is the traditional 1-NN clustering which used cosine similarity and TFIDF weight document representation. **Baseline 2** is the LSH-based clustering approach proposed in [4], which also used cosine similarity between content text of messages and TFIDF weight scheme with k=13 and L=100 in equation (1). Both these two baselines and our approach utilized inverted indices to acculturate the similarity computation. The parameters B and r of our approach in equation (2) are set to B=5, r=20. The parameters ζ and W of our approach in equation (7) are set to ζ =-0.5, W=24 hours. Table 3 shows the descriptions of the five event clusters detected by our approach and both two baselines and the frequent terms for each event. Table 4 shows the precision of the five event clusters detected by our approach and two baselines, and our approach achieved the highest precision for five event clusters. Table 5 shows

the precision of real-world events detected by our approach and two baselines, and our approach achieves the highest precision for real-world event detection.

The experiment results demonstrate that the message-mentioned locations are likely to be the actual event locations and can improve the real-world event detection. Moreover, the similarity measure considering content, spatial and temporal dimensions is more effective than cosine similarity between textual contents.

Table 3. The five event clusters detected by our approach and two baselines

Event id	The frequent terms
1	Zhangzhou, traffic accident, family, child, truck, Taiwanese
2	Muxidi, car accident, AUDI, Beijing, driver, die,serious
3	Sinograin, fire, grain, barn, burn, Heilongjiang, last
4	Jilin, fire, explode, die, poutry, factory
5	Taiwan, earthquake, shake, feel, where

Table 4. The precision of five event clusters detected by our approach and two baselines

Event id	Number of messages in event cluster			Precision (event cluster)		
	Baseline 1	**Baseline 2**	**Our approach**	**Baseline 1**	**Baseline 2**	**Our approach**
1	526	483	567	78.3%	79.5%	**80.8%**
2	247	196	223	75.4%	76.3%	**79.6%**
3	1031	897	1192	72.6%	76.8%	**79.3%**
4	723	573	649	80.2%	81.3%	**83.2%**
5	1763	1488	2039	68.5%	69.3%	**73.8%**

Table 5. The precision of real-world events detected by our approach and two baselines

Method	Num of detected events	Num of detected real-world events	Precision (real-world event detection)
Baseline1	343	53	15.5%
Baseline2	421	52	12.4%
Our Approach	287	49	**17.1%**

5 Conclusions

In this paper, we presented a real-time event detection approach using LSH to accelerate the similarity comparison, a message-mentioned location extraction method based on POS tagging and a SVM classifier, and a novel similarity measurement considering content, location, and time to improve the precision of event detection. We compare our approach with two state-of-the-art baselines on event detection, and the experiment results demonstrate the effectiveness of our approach.

References

1. TDT 2004: Annotation manual, http://www.ldc.upenn.edu/Projects/TDT2004
2. S. Unankard, X. Li, M. A. Sharaf.: Location-based Emerging Event Detection in Social Networks. Technologies and Applications, pp. 280-291. Springer Berlin Heidelberg (2013)
3. X. Zhou, L. Chen.: Event Detection over Twitter Social Media Streams. VLDB J, 23(3): 381-400. (2014)
4. S. Petrović, M. Osborne, V. Lavrenko.: Streaming First Story Detection with Application to Twitter. In: NACL, pp. 181–189. ACL (2010)
5. S. Petrović, M. Osborne, V. Lavrenko.: Using Paraphrases for Improving First Story Detection in News and Twitter. In: NACL, pp. 338-346. ACL (2012)
6. H. Becker， M. Naaman， L. Gravano.: Learning Similarity Metrics for Event Identification in Social Media. In: WSDM, pp.291-300. ACM(2010)
7. C Lee.: Mining Spatio-temporal Information on Microblogging Streams Using A Density-based Online Clustering Method. Expert Systems with Applications, 39:9623–9641. ELSEVIER(2012)
8. M. Cataldi，L. D. Caro，C. Schifanella.: Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation. In: MDMKDD, Article No.4. ACM(2010)
9. O. Ozdikis，P. Senkul，H. Oguztuzun.: Semantic Expansion of Hashtags for Enhanced Event Detection in Twitter. In: VLDB. (2012)
10. M. S. Charikar.: Similarity Estimation Techniques from Rounding Algorithms. In: STOC, pp. 380-388. ACM(2002)
11. J. Sankaranarayanan， H. Samet， B. E. Teitler， M. D. Lieberman ， J. Sperling.: TwitterStand: News in Tweets. In: GIS, pp.42-51. ACM(2009)
12. A. Rajaraman, J. D. Ullman.: Mining of Massive Datasets. Cambridge University Press(2011)
13. H. Sayyadi, M. Hurst, A. Maykov.: Event Detection and Tracking in Social Streams. In: ICWSM, pp. 311-314. (2009)
14. H. Becker，M.r Naaman，L. Gravano.: Beyond Trending Topics: Real-World Event Identification on Twitter. In: AAAI, pp. 438-441. (2011)
15. Z. Tan, P. Zhang, J. Tan.: A Multi-layer Event Detection Algorithm for Detecting Global and Local Hot Events in Social Networks. Procedia Computer Science, 29: 2080-2089. (2014)
16. Y. Wang, E. Agichtein, M. Benzi.: TM-LDA: Efficient Online Modeling of Latent Topic Transitions in Social Media. In:KDD, pp. 123-131. ACM(2012)
17. T. Baldwin, P. Cook, B. Han, A. Harwood, S. Karunasekera, M. Moshtaghi.: A Support Platform for Event Detection Using Social Intelligence. In: EACL, pp. 69-72. ACL (2012)
18. Z. Cheng, J. Caverlee, K. Lee.: You Are Where You Tweet: a Content-based Approach to Geo-locating Twitter Users. In: CIKM, pp. 759-768. ACM (2010)
19. W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, M. Larson.: The Where in The Tweet. In: CIKM, pp. 2473-2476. ACM(2011)