LEDS: Local Event Discovery and Summarization from Tweets

Zhi Liu, Yan Huang

Department of Computer Science & Engineering
University of North Texas, Denton, TX, 76201
zhiliu@my.unt.edu
yan.huang@unt.edu

Joshua R. Trampier

Department of Defense
Springfield, VA 22150, USA
Joshua.R.Trampier@nga.mil

ABSTRACT

Twitter is one of the most popular social media platforms where people can share their opinions, thoughts, interests, and whereabouts. In this work, we propose a Local Event Discovery and Summarization (LEDS) framework to detect local events from Twitter. Many existing algorithms for event detection focus on larger-scale events and are not sensitive to smaller-scale local events. Most of the local events detected by these methods are major events such as important sports, shows, or large natural disasters. In this paper, we propose the LEDS framework to detect both larger and smaller events. LEDS contains three key steps: 1) Detecting possible event related terms by monitoring abnormal distribution in different locations and times; 2) Clustering tweets based on their key terms, time, and location distribution; and 3) Extracting descriptions including time, location, and key sentences of local events from clusters. The framework is evaluated on a real world Twitter dataset with more than 60 million tweets. The results show that compared with previous work, LEDS can detect smaller-scale and greater variety of local events. More than 43 percent of detected local events do not have an official organizer, cannot be seen on news media, and only attract the attention from a small group of people.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data Mining*

Keywords

Local event detection; Tweets; Event summarization

1. INTRODUCTION

Social media has rapidly become one of the most important platforms where people can share their thoughts, opinions, interests, and whereabouts. A typical example is Twitter. There are 310 million average monthly active users on Twitter¹. Twitter's popularity carries it beyond a commu-

¹<https://investor.twitterinc.com/results.cfm>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

© 2016 ACM. ISBN 978-1-4503-4589-7/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2996913.2996928>

nication platform. With its users widely distributed, people are not only tweeting about their daily activities but can also reporting events happening anywhere and spreading through the network. The analytics on social media content can provide most recent topic trends, real-life events, and public sentiments. Compared with tradition information source, social media are informal but can provide the most up-to-date information of current events including earthquakes, sport matches, and variety kinds of local events.

This paper focuses on detecting local events from geotagged tweets. Monitoring local events is conducive to many real-world applications such as crime mapping, traffic monitoring, and emergency management. In fact, many local events in real world are “small” events. They are not reported by news media and only a relatively small number of local people discuss the events. However, these events are important and learning such events can help in location-based services, public security, and smart transportation.

For detecting local events, we have the following challenges: 1) People discuss events on social media in a very informal way and most of users will not list a clear time and location information of an event; 2) Collecting the tweets related to an event is challenging. Previous investigations show that 40.55% tweets are pointless babbles², and based on our analysis on the dataset, only less than 5% geotagged tweets are related to an event; 3) For many local events, there may be only a few users publishing a small number of related tweets. Existing methods based on outbreak detection of tweets or terms can only find large events such as sport games, shows, or disasters, which may be already on news and attract lots of attention. To address these challenges, we propose the LEDS framework to detect local events. The goal is to be sensitive to all kinds of local events no matter they are large or small. Many existing algorithms focus on larger events and detect events in limited categories, i.e. sports, shows, and disasters. Such methods use spikes of tweets or located users to detect events [6, 2, 8, 5]. However, they tend to ignore many small-scale events since most medium- or small-scale events cannot lead a obvious changing of number of tweets in an area. Terms are more sensitive to events than tweets. During an event, people are likely to use special terms which are not commonly used in daily lives. Thus we propose an anomaly-term-driven framework. The key idea is to compare term frequencies with overall distribution. The comparison needs to be done based on spatial and temporal scales instead of

²<http://pearanalytics.com/wp-content/uploads/2009/08/Twitter-Study-August-2009.pdf>

globally. Our method can detect the abnormal distribution effectively and is not affected by factors such as the flux of online users. In this way, our proposed system can detect much broader event categories which include many local events such as parties, and personal meetings.

2. RELATED WORK

Event detection on social media has attracted much attention in recent years. The approaches used can be broadly classified into two categories: geographic-anomaly-detection-driven methods and content-feature-driven methods. The basic idea of the methods in the first category is to detect events by monitoring abnormal occurrences of tweets or terms. Weng et al. [9] build signal models for individual words by applying wavelet analysis on the frequency-based raw signals of the words and capture the bursts in the words' appearance. The authors then detect events by grouping a set of words with similar patterns of burst using a community detection algorithm. Krumm et al. [5] use regression to estimate the number of tweets of an area using multi-level geographic division based method. The features used include time of the day, day of the week, and tweets counts from neighbors. If the actual number of tweets exceeds the threshold, a text summarization algorithm will be used on the tweets in the region to extract tweets to describe the event. In [6], the authors detect events by monitoring the moving crowd. Both increasing of tweets and Twitter users are considered in event detection. To monitor moving crowd, number of users in a region, incoming users, and outgoing users are recorded. These abnormal cases will be investigated and used in event detection. Cheng et al. [2] propose a space-time scan statistics based approach in event detection. The idea is to detect clusters on both space and time dimensions assuming people will tweet more than expected during the event. In [1], the authors propose a user-driven method to perform a geo-temporal analysis of information to detect events. The DBSCAN clustering algorithm is used to identify geo-temporal clusters of messages.

In a content-feature-driven method, text, time, location, and other information are used as features in learning algorithms. In [8], the authors detect earthquake location from related tweets. The authors build a classifier to detect tweets which are related to the target event. A probabilistic spatiotemporal model is then built to locate the center of the event. In [10], the authors propose a graphical model to capture the content, time, and location of social messages. Each message is represented as a probability distribution over a set of topics. The event is detected by conducting efficient similarity joins over social media streams. In [7], the authors extract four features to represent an event: name entity, event phrase, calendar date, and event type. The named entity is extracted by NLP tools and the event phrase and event types are obtained by a learning method. The calendar date is obtained from a rule based system.

3. ALGORITHM

Our event detection algorithm has three main steps: detecting index terms, tweets clustering, and event extraction. In this section, we will introduce these three steps in details.

3.1 Detecting Index Terms

Detecting outbreaks of tweets is one of the most important existing methods to detect events. However, only global events attract a huge amount of interests. Local and small

events may only involve a few users on Twitter and these events cannot raise an outbreak of number of tweets. So in this work, instead of looking at outbreak of number of tweets, we focus on detecting abnormal distribution of terms. Term level distribution is more sensitive to events since when people discuss an event, some terms are more likely to be rare in daily conversations.

Firstly, all the tweets will be added to the root node of a k -d tree, a space-partitioning data structure for organizing points in a k -dimensional space. Then the node will be divided into two parts iteratively by finding a line which can equally separate the tweets in the node. This process will be stopped until the number of average tweets in the node is less than 50,000. Then for each leaf node, we detect the index terms using correlation coefficient. Here we use t to represent a tweet and for each tweet, it contains a set of terms $\{e_1, e_2, \dots\}$. We use t^l to denote the location where the tweet is published and E is the set of all of the terms. For a term e , we count the frequency of term e in each day: $\vec{e} = \langle \text{fre}^1(e), \text{fre}^2(e) \dots \rangle$, and the overall distribution is denoted as $\vec{E} = \langle \text{fre}^1(E), \text{fre}^2(E) \dots \rangle$. We then use the Spearman correlation coefficient to measure the correlation between the two distributions as: $r = \frac{\text{cov}(\vec{e}, \vec{E})}{\sigma_{\vec{e}} \sigma_{\vec{E}}}$. Here $\text{cov}(\vec{e}, \vec{E})$ is the covariance of the two distributions, and $\sigma_{\vec{e}}$ and $\sigma_{\vec{E}}$ are the standard deviations of the distributions. Terms with correlation coefficient higher than 0.35 will be considered as index terms.

3.2 Tweets Clustering

After detecting index terms, we try to use these terms to help to find the tweets related to an event. For each index term, we first collect all the tweets which contain the term. The tweets published in the k -d tree nodes where the term is not considered as index term will be removed. Then these tweets will be clustered by their time and location. Considering that during an event, and there may be a quick increase of tweets related to the event and in most time, there may be only a few tweets discussing the event, we use a density based approach to cluster the tweets [3]. The basic idea of this approach is that the influence of each data point can be modeled by an influence function and this function can be seen as a description of the impact of data points within its neighborhood. We use this method because it works well in dealing with outliers and noises and flexible for clusters with different sizes and shapes.

Definition[Influence Function] The influence function between tweets are defined as: $f_{t_i}(t) = \exp(-\frac{d(t^l, t_i^l)^2}{2\sigma^2})$. Here $d(t, t_i)$ is the geographic distance between tweets t and t_i .

Definition[Density Function] With the influence function, the density function of tweet t is defined as the sum of total influence of other tweets: $\mathcal{I}(t) = \sum_i^N f_{t_i}(t)$. In the algorithm, we only count the tweets published within the distance of 4σ [3] and in the past or following ψ hours.

Definition[Gradient] The gradient of $\mathcal{I}(t)$ is defined as: $\nabla \mathcal{I}(t) = \sum_i^N (\vec{t}_i - \vec{t}) f_{t_i}(t)$. Here $\vec{t}_i$ and $\vec{t}$ mean a data point represented by the location (latitude, longitude) and time of the tweet in a three dimensions space.

Definition[Density-Attractor & Density-Attracted] A tweet t is a density-attractor iff $\mathcal{I}(t)$ is a local maximum of the density function. A tweet t_i is a density-attracted tweet for t when there exist a path to t that the gradient is continuously positive: $\vec{t}^0 = \vec{t}, \vec{t}^k = \vec{t}^{k-1} + \delta \cdot \frac{\nabla \mathcal{I}(\vec{t}^{k-1})}{\|\nabla \mathcal{I}(\vec{t}^{k-1})\|}$.

Different event has different types of locations. For example, for sport matches, the location can be considered as a point. For tornado, the location can be an area. Constrained by geographic boundaries, population distribution, and some other factors, the influence area of an event may have an arbitrary shape. So we use an arbitrary-shape clustering method here. We also choose the value of the parameters of σ , ψ , and ξ with the method provided in [3]. Based on the analysis of clustering results, we choose the value of ψ as two hours and use 7km as the value of σ .

Then we try to merge the clusters because an event may be related to several index terms. For each cluster C , we use normal distributions $C_{time} \sim \mathcal{N}(\mu_{Ct}, \sigma_{Ct}^2)$ to fit the distribution of published time of tweets of the cluster and the location distribution $C_{location} \sim \mathcal{N}(\mu_{Cl}, \sigma_{Cl}^2)$. If two clusters come from the same event, we can consider them as two different sampling of tweets from all the tweets related to the event. So our goal is to test the relationship between two clusters. Here if we assume that cluster C_1 and C_2 are sampled from the same event. With the location and time of all the tweets in C_1 and C_2 , we can obtain the mean location of tweets in these two clusters l_{C_1} and l_{C_2} , the variance of location distributions $\sigma_{C_{l1}}^2$ and $\sigma_{C_{l2}}^2$, mean time of tweets t_{C_1} and t_{C_2} , the variance of time distributions $\sigma_{C_{t1}}^2$ and $\sigma_{C_{t2}}^2$, and the number of tweets in these two clusters. Then we can estimate whether they are sampled from the same event by using student's t -test. If $|Z| < |Z_{\alpha/2}|$ on both location and time distributions, we will consider that C_1 and C_2 are related to the same event (in this paper, we set the value of α as 0.05).

Some clusters will be filtered since they are related to non-local-event or meaningless. The first type of clusters is generated by events or news which have a global influence. For two clusters in different areas, they will be considered as discussing the same global event and removed when they share more than 80% common index terms with each other and their t -test of time distributions meet the requirement of $|Z_t| < |Z_{\alpha/2}|$. For the second case, we find that there may be lots of similar tweets with just one or two words difference published in a short time in an area. In most cases, they are just meaningless tweets. For each cluster, we will use a term level cosine similarity to measure the common terms between tweets. If there is a group of tweets in a cluster with: the cosine similarity between each two of them are higher than 0.5 and the number of tweets in the group is larger than half of the total number of the cluster, we will remove this cluster.

3.3 Event Extraction

After the clustering step, we extract the time, location, and content information from each cluster as the description of an event.

Time and location estimation: Different event has different type of influence area and time. For example, events such as sport games or concerts have a clear location where they are happening. However, many events do not have a specific point location and time, e.g. some natural disasters. In [8], the authors use a Poisson distribution to fit the distribution of number of tweets after an earthquake. It makes sense because users begin to discuss this only after it happened. The distribution of time of tweets can be fit into an exponential distribution. For other events, such as procession or a football match, people may discuss the event before it happens. In this work, we estimate the time and location

by the density attractors. For each cluster of tweets, we can obtain the density attractors of this cluster by the method in section 3.2. The average time and location of these density attractors will be used as the time and location of the event.

Tweet summarization: In [4], the authors propose a summarization method for a set of tweets. Each term in a tweet is assigned a weight based on its term frequency (tf) and inverse document frequency (idf). Then the weight of a tweet is defined as the average weight of terms it contains. For each cluster, we select the top 5 tweets with the highest weight to describe the event.

4. EXPERIMENT

In this section, we test our algorithm on Twitter dataset. Firstly we introduce the dataset used in this work and give the definition of local events. We use precision and case study to show the results of our event detection algorithm. Then we analyze the types of events we detected to show the differences between our algorithm and previous methods. Finally, we provide more details of the detected events to help to get a better understand of our algorithm. We collected tweets published from May 21 to July 6, 2015 using Twitter Streaming API³ with the spatial bounding box $[24^\circ 3'N, 127^\circ 7'W, 48^\circ 8'N, 65^\circ 5'W]$, which covers U.S. mainland. The dataset consists of over 60 million geotagged tweets from 2,200,402 Twitter users.

We propose a detailed description of the targeted local events. Currently, there is not a clear definition of what is a local event. An event is loosely defined as a happening which has a number of participants getting together at a specific location at a specific time. We also follow this idea in our work. However, compared with previous work, in our work we apply strict criteria when judging whether an event is a local event: 1) A local event must have a clearly defined set of participants, location, and time. The participants can be the people who attend an event e.g. sport games, or affected by an event e.g. earthquake. Only events which happen in a certain location or area will be considered as a local event; 2) A local event must belong to a certain location; 3) Events are different from news. Although some news have specific time, location, and participants, it is still news rather than a local event. For example, a soccer player transferring from a team to another is considered as news. For such news, it is difficult to provide a clear descriptions of time and location information. In this work, we only consider these happenings as news.

4.1 Experimental Results

We provide a detailed demonstration of the detection results. Both the precision analysis and case study will be used to show how our algorithm works and the advantages of our algorithm.

4.1.1 Overall Results

From the Twitter dataset we collected, our LEDS system first obtains a set of candidate clusters which may be related to local events. Then we filter the clusters. We divide the clusters into two classes based on whether they are related to a local event (LE) or not (NLE). The clusters which are not related to an local event (NLE) are further divided into two subclasses: a global but not local event (or news) (NLE_N) and the noise clusters (NLE_M). Please note that

³<https://dev.twitter.com/streaming/overview>

Table 1: Selected local events in Los Angeles.

Event id	Date	# of index terms	Descriptions
1	Jul 4	1	goonies + fireworks tonight! #hollywoodforevercemetery
2	May 30	1	mistakes we make can paint a better picture later on.
3	May 28	1	24 minutes till earthquake.
4	Jun 13	2	Today sat. june 13 4pm the smokin' cobras perform a benefit concert
5	Jun 16	3	#e32015 adventures! http://t.co/i2hqmq4jzw

based on our definition of local events, when a local event with a global influence happens, we only take it as a local event at the location where it happens. For other locations, we consider it as news rather than a local event. Figure 1 shows the results of our filtering method. The left pie chart is the filtering results. 31% of the clusters detected by our algorithm are considered as related to local events and 69% of the clusters are taken as non local event. Among the 69% clusters, 30% of them are considered as related to a news (NLE_N), and 39% of them are considered as noise clusters (NLE_M). After the filtering step, there are 31 percent of clusters left and considered as local events by our algorithm.

Figure 1: Proportion of different clusters.

Then we manually label these clusters. For each cluster, we extract the time, location, keywords and top 5 tweets in the clusters to describe the event. For each detected event, we have three people to label whether the cluster is a local event (LE), news (NLE_N) or noise cluster (NLE_M) based on these information. We randomly select 2,000 clusters from the results and manually labeled the clusters by three people and here we use the majority rule. Each result will be considered as a local event only when it has two or more positive labels. The right pie chart of Figure 1 demonstrates the results of the human labeling. 47% of the clusters after filtering are labeled as a local event. So the precision of our algorithm is 47%. There are also 20 percent of the clusters are labeled as noise clusters and 33 percent of clusters are labeled as news.

4.1.2 Case Study

In Table 1, we randomly selected 5 detected events happened in Los Angeles from May 25 to July 8, 2015. In this table we show the date of the event, the number of index terms related to the event, and the top one tweets which are selected to describe the events. There are one large event which have global influence, E3 2015, but more of them are “smaller” events. They don’t have a large influence and some of them even will not be mentioned in any news media. Compared with previous methods, this is an important advantage of our algorithm. For example, in July 4th, we detected some events such as firework for celebrating the Independence Day. Local events such as the firework, protesting, smaller earthquake, local festival, or business events, which may be mentioned by some news media, may still not have enough related tweets to be detected by the previous methods. We also detected many events which only attract the attention from a certain group of people. For these events,

only the users who attend or directly related to the event will discuss it. For example, parties, meetings, or events held by some communities like photography lovers. These events may have only tens of related tweets and are very hard to detect by the tweet spike detection based methods. Based on the analysis of experimental results, only 26% of events detected belong to “large” events, and 43% of them are small events which even cannot be seen on any news medias.

5. CONCLUSION

In this paper, we propose a framework for local event detection. The proposed LEDS framework can detect smaller-scale and more types of local events. The framework uses terms instead of number of tweets to detected events which are more sensitive. The density-based clustering and time-location-distribution-based method can help to cluster tweets which may be related to the same event effectively. Finally, we extract the time, location, and content information from each cluster. The results show that greater variety of local events can be detected by our method including shows, disasters, parties, business activities, and personal meetings.

6. REFERENCES

- [1] P. Arcaini, G. Bordogna, D. Ienco, and S. Sterlacchini. User-driven geo-temporal density-based exploration of periodic and not periodic events reported in social networks. *Information Sciences*, 2016.
- [2] T. Cheng and T. Wicks. Event detection using twitter: a spatio-temporal approach. *PloS one*, 9(6):e97807, 2014.
- [3] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *KDD*, volume 98, pages 58–65, 1998.
- [4] D. Inouye and J. K. Kalita. Comparing twitter summarization algorithms for multiple post summaries. In *PASSAT and SocialCom*, pages 298–306. IEEE, 2011.
- [5] J. Krumm and E. Horvitz. Eyewitness: Identifying local events via space-time signals in twitter feeds. *ACM SIGSPATIAL*, 2015.
- [6] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *LBSN*, pages 1–10. ACM, 2010.
- [7] A. Ritter, O. Etzioni, S. Clark, et al. Open domain event extraction from twitter. In *ACM SIGKDD*, pages 1104–1112. ACM, 2012.
- [8] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, pages 851–860. ACM, 2010.
- [9] J. Weng and B.-S. Lee. Event detection in twitter. *ICWSM*, 11:401–408, 2011.
- [10] X. Zhou and L. Chen. Event detection over twitter social media streams. *The VLDB journal*, 23(3):381–400, 2014.