STED: Semi-Supervised Targeted Event Detection

Ting Hua
Dept. of Computer Science
Virginia Tech
tingh88@vt.edu

Feng Chen
H.J. Heinz III College
Carnegie Mellon University
fchen1@cmu.edu

Liang Zhao
Dept. of Computer Science
Virginia Tech
liangz8@vt.edu

Chang-Tien Lu
Dept. of Computer Science
Virginia Tech
ctlu@vt.edu

Naren Ramakrishnan
Dept. of Computer Science
Virginia Tech
naren@vt.edu

ABSTRACT

Social microblogs such as Twitter and Weibo are experiencing an explosive growth with billions of global users sharing their daily observations and thoughts. Beyond public interests (e.g., sports, music), microblogs can provide highly detailed information for those interested in public health, homeland security, and financial analysis. However, the language used in Twitter is heavily informal, ungrammatical, and dynamic. Existing data mining algorithms require extensive manually labeling to build and maintain a supervised system. This paper presents STED, a semi-supervised system that helps users to automatically detect and interactively visualize events of a targeted type from twitter, such as crimes, civil unrests, and disease outbreaks. Our model first applies transfer learning and label propagation to automatically generate labeled data, then learns a customized text classifier based on mini-clustering, and finally applies fast spatial scan statistics to estimate the locations of events. We demonstrate STED's usage and benefits using twitter data collected from Latin America countries, and show how our system helps to detect and track example events such as civil unrests and crimes.

1. INTRODUCTION

Microblogs (e.g., Twitter and Weibo) have emerged as a disruptive platform for people to share their daily activities and sentiments on ongoing events. The rich up-to-date sensing information allows discovering and tracking important events even earlier than news, with important applications such as public health and emergency management. Although identifying events from newspaper reports has been well studied, analyzing messages in Twitter requires more sophisticated techniques. Twitter messages are irregular, contain misspelled or non-standard acronyms, and are written in informal style. Additionally, tweets are filled with trivial events discussing daily life. Twitter's noisy nature challenges traditional text-based event detection methods and therefore specifically designed event detection approaches are needed for Twitter text analysis.

Most previous work on Twitter event detection has focused on general and large-scale (breaking news) events, such as the Virginia Tech shooting and the Southern Califorinia wild fires. Unsupervised learning techniques, such as clustering, topic modeling, and burst detection, are mainly utilized. However, they have limited capabilities to detect small-scale events, such as city-level or even street-level protests or strikes. Recently, new attention has been paid to event detection of a targeted topic (e.g., civil unrests, disease outbreaks, or crimes). Supervised learning techniques are primarily applied, such as support vector machines and random forest classifiers. Although this work can detect small-scale events of the targeted topic, the requirements of expensive manual data labeling limits its efficiency and scalability. How to determine whether a tweet is interest-related or not is far more than simple keyword filtering. For example, if tweets related to shooting crimes are required, feedback from Twitter for the query word 'shooting' are motley: tweets like '2 shot to death, 1 wounded: A shooting erupted at Mexico City airport' are indeed related to shooting crime, but tweets like 'Shooting a music video' in fact have nothing to do with gunfire.

In this demo, we propose a novel approach, semi-supervised targeted event detection (STED), which takes users' specific interests as input, retrieves related tweets and summarizes events' spatial and temporal features into visualization results. The major contributions are as follows:

- **Automatical label creation and expansion:** To avoid burdensome human efforts required in previous work, we propose a method capable of generating labeled data automatically, which first transfers labels from newspapers to tweets, and further expands the initial label subspace using Twitter social ties.
- **Customized text classifier for Twitter:** The noisy nature of Twitter data is a new challenge for text classification. Using tweet mini-clusters obtained by graph partitioning, we build a specialized support vector machine classifier for tweet analysis.
- **Enhanced location estimation algorithms:** Utilizing tweet social ties and fast spatial scan statistics, we propagate geo-labels within location clusters for event separation.
- **Visualization and analysis:** Provision of event clusters, historical statistics, and related-tweets via a friendly interface promotes effective and efficient usage for human analysis.

2. FRAMEWORK AND METHODS

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
.

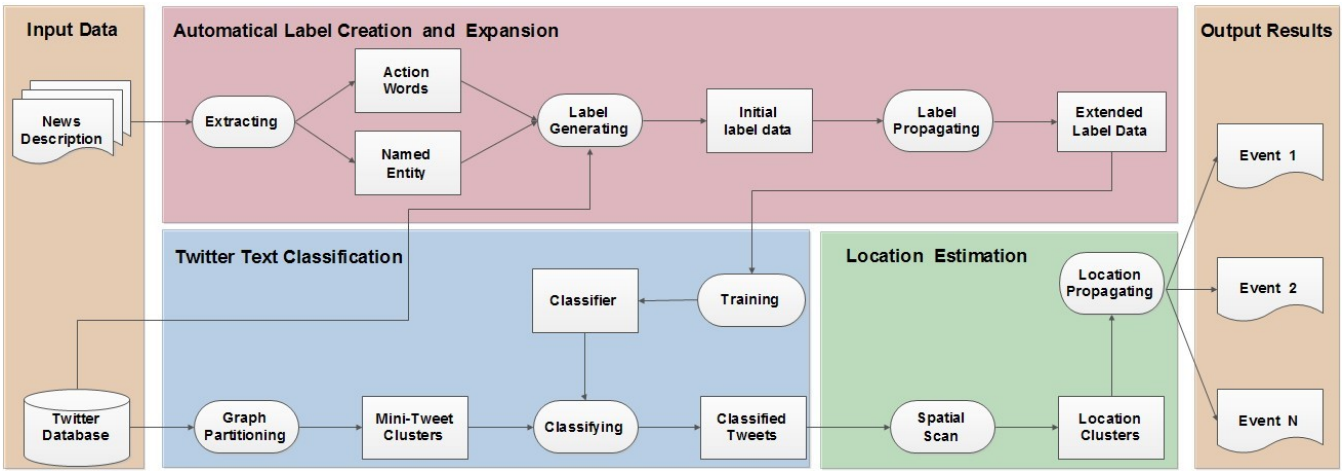


Figure 1: System Framework of STED

As shown in Figure 1, the architecture of STED can be divided into these parts. Using the Extracting and Label-Generating modules, we transfer labels from news to Twitter to generate initial label data. Module Label Propagating utilizes Twitter social features to obtain extended label data. Graph-Partition module clusters initial single tweets into mini-tweet-cluster, and then training module build a Support Vector Machine(SVM) text classifier to identify targeted topic related tweets. Finally, Spatial Scan and Location Propagating modules further group target topic related tweets into specific events according to location.

2.1 Automatical Label Creation and Expansion

In this step, we first automatically transfer labels from news descriptions to Tweets and further expand the initial label data by utilizing Twitter social ties like *Retweet(RT)*, *Hashtag(#)*, and *Mentions(@)*.

Term Extracting and Label Generating: We first collect domain specific news descriptions, such as news about crime, from public media. Though news reports are quite different from tweets in structure and expression style, elements that can specify an event remain the same: *Named Entity* and *Action Word*. By using NLTK [1], we extract *Named Entity(noun)* and *Action Words(verb)* from news description as candidate query word set for tweets. Given a query set as input, label generating module investigates Twitter data and selects tweets containing at least one Named Entity and one Action Word as positive label data.

Label Propagating: Social-ties terms appear between tweets in the form of *Mentions(@)*, *Retweets(RT)*, and *Hashtag(#)*. Tweets sharing common terms are more likely to discuss the same topic. We use social-ties terms to expand initial labeled dataset L obtained. First, we identify social-ties terms from labeled tweets, build Term-Tweet heterogeneous network S_1, and remove less popular terms. As shown in Figure 2(a), node degrees are approximately distributed in power law where most tweets are connected to few terms with high degree. These terms are expected to be more related to the event, while those low degree terms on the border are trivial. Then, we use the remaining popular terms as query to retrieve tweets, build Term-Tweet heterogeneous network S_2, and filter away terms with low ability to denote a specific event. Figure 2(b) illustrates an example of S_2, a Hashtag-Tweet heterogeneous network, where the core term of the central cluster is hashtag '#mexico', surrounded with more newly found tweets (orange nodes) than initial labeled tweets (blue nodes). These terms should be filtered away by our system, since they are popular but shared by too many topics to represent specific interests. Finally, we build Term-Tweet network with filtered term set connected to new found label tweets S_3, as shown in Figure 2(c). Iterate process above, until no new tweet satisfying the conditions can be found. Through Label Propagating module, we obtain an extended label dataset for further processing.

[1] http://nltk.org/

2.2 Twitter Text Classification

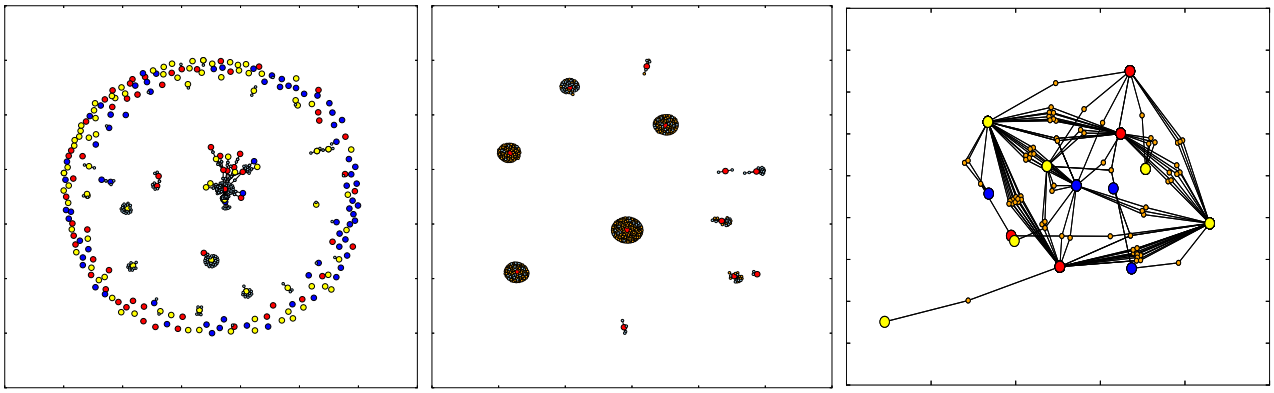
In this part, we first apply graph partitioning methods [6] to obtain event-related words groups and generate tweet mini-clusters, and then use support vector machine [2] for text classification.

Graph Partitioning: Given word w, we first build its wavelet signal represented by the following sequence.

$$f_w = [f_w(T_1), f_w(T_2), ..., f_w(T_n)] \quad (1)$$

where $f_w(T_i)$ is the TF-IDF score of word w during the period T_i . In this paper, to capture daily event emergence, we set duration of T_i to be one hour and number of segment n to be 24. Then, we compute the auto correlation A_w for each word w and filter away trivial words(appearing evenly day by day). From above, we get subset Ψ of rare and note worthy words. Next, we calculate cross-correlation X_{ij} of each word-pair in Ψ and construct a correlation matrix Γ containing all word pairs. This correlation matrix Γ can be viewed as a graph and related-word clustering becomes a graph partition problem: We apply graph partitioning [5] on correlation matrix Γ to obtain subgraphs that words within one subgraph are highly similar in form of high cross correlation, while words in different subgraphs have low cross correlation. Finally, tweet clusters are generated by obtained word groups: tweet containing at least two items of word group G_i can be considered as an item of tweet cluster C_i.

Classifier Training: The most important part for classifier training is feature selection. Words appear less than threshold ζ are first filtered out. Next, we calculate TF-IDF scores for words and filter out trivial words such as 'people', 'love', which appear more frequently in total Twitter space than labelled tweets space. Besides, to avoid overfitting problem, most words from the Named Entity set E should be removed, since they enjoy high TF-IDF scores and



(a) Term-Tweet Heterogeneous Network of Label Tweets S_1.

(b) Hashtag-Tweet Heterogeneous Network of Total Tweets Space S_2.

(c) Term-Tweet Heterogeneous Network of New Found Tweets and Filtered Terms S_3.

Figure 2: Tweets' Social Ties Networks. Big nodes represent terms: Red nodes are hashtags, blue nodes are mentions, and yellow nodes are Retweets. Small nodes denote tweets: blue ones are labeled tweets, orange nodes are newly found tweets from raw data. Edge (i, t) means tweet t contains term i.

will potentially be assigned heavy weights in the SVM classifier, but only represent one specific event.

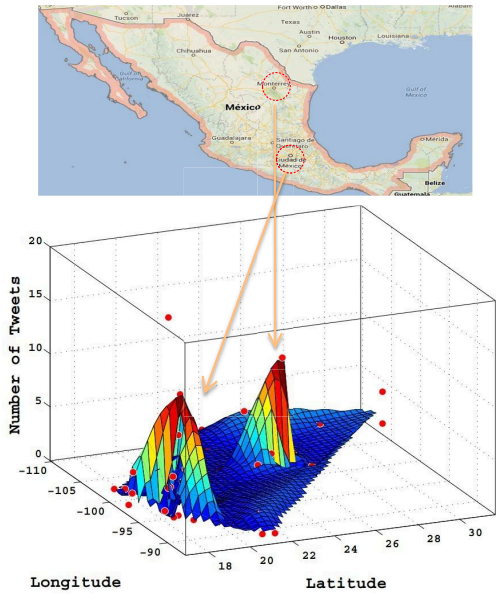


Figure 3: Example of Tweet Location Clusters. Red nodes denote the highest density of tweets of locations.

2.3 Location Estimation

To estimate event locations, we first identify spatial clusters using fast spatial scan methods [4] However, only 2% of tweets contain such geographic information. To make best use of the minority of tweets with geo-labels as well as the majority without labels, we further propagate geo-labels within each cluster to amplify spatial signal capabilities.

Spatial Scan: Geo-locations of tweets about a certain event are likely around the event's occurring location. We apply spatial scan statistics to detect significant spatial clusters, as shown in Figure 3. Specifically, we aggregate the count of event related tweets in city level and define the base of each city as the total count of the original tweets. Then we apply fast subset scan [4] to identify a set of H candidate clusters with the largest Kulldorff's statistics [3], which is defined as

$$K_r = (C_a - C_r)\lg(\frac{C_a - C_r}{B_a - B_r}) + C_r \lg(\frac{C_r}{B_r}) - C_a \lg(\frac{C_a}{B_a}) \quad (2)$$

where C_a and B_a refer to the total count and base in the country, respectively; and C_r and B_r refer to the count and base in the spatial region r, which is a set of neighbor cities. The empirical p-value of each candidate cluster is estimated by random permutation testing, and the clusters with empirical p-values smaller than a threshold η (e.g.$\eta = 0.05$), are returned as significant clusters. The parameter H is usually set greater than the maximum number of potential clusters that may exist, and the insignificant clusters can be filtered out later by randomization testing.

Location Label Propagating: Within each cluster, we further label tweets that lack geo information using social ties. Tweets contain common terms such as hashtag and mention are more likely to occur in the same location. We first compute a score $\omega_{ij} = \frac{m_{ij}}{M_i}$ by tweets with geo-labels to denote the relativity of term i and location j, where m_{ij} is number of tweets contain term i as well as location j, and M_i is the count of tweets contain term i.

$$l_t = \max_{j \in \varphi}\{1 - \prod_{i \in \phi_t}(1 - \omega_{ij})\} \quad (3)$$

Then, using Equation 3, we estimate location l_t of unlabelled tweet t, which contains a set ϕ_t of terms. We first compute the relativity between tweet t and each location j from location set φ, and then pick up the biggest value as this tweet's estimated location.

3. DEMONSTRATION

We showcase STED system using Twitter data from Latin America as the example application. The considered database is more than 400GB in size, from June, 2012 to Jan, 2013. With application to civil unrest event detection, STED achieved 72% in precision of and 74% in recall, with a lead time of 2.42 days ahead of traditional media (e.g., news sources). We implemented the STED interface in Python which provides users with the following capilities:

- Map visualization of targeted-interest event clusters in city-level.

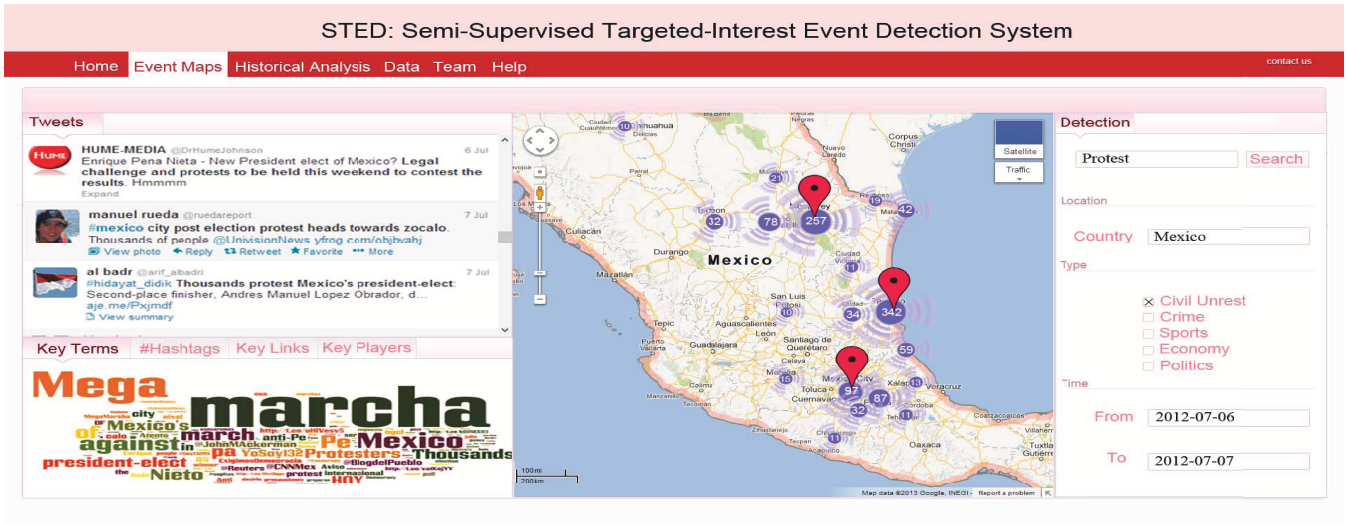


Figure 4: Interface of STED system

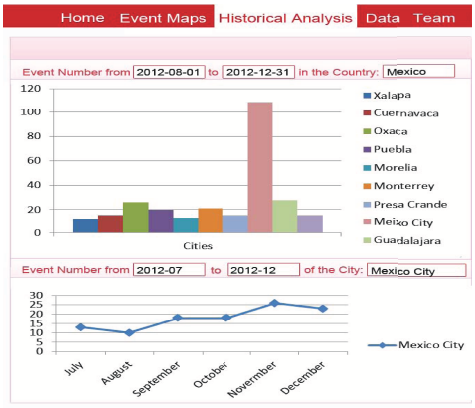


Figure 5: Historical Analysis Screenshot

- Detailed information about each event cluster, including related tweets and word cloud summary.
- Graphical analyses of historical statistics, including spatial comparison among regions and temporal trend within one region.

Figure 4 shows a screenshot of the STED interface. With STED, a user can search for events pertaining to their specific interests and analyze their spatial and temporal features. A targeted-interest includes time, location, topic and keywords. Users are allowed to choose date and topic as well as typing in keywords, in the right part of interface. As an ongoing project, we have applied our method to detect interests of crime and unrest. As shown in the screenshot, the users' interest is in detecting events about type *'Civil Unrest'* in country *'Mexico'*, with keyword *'protest'*, from date *'2012-07-06'* to *'2012-07-07'*. After users click on the 'Search' button, STED will return corresponding event clusters of targeted-interest, shown as balloons. By clicking on one of the balloons, users can find detailed information of corresponding event from left part of the interface: tweets ranked by their relativity to users' interests and word cloud event summary denoting terms' relative importance. System feedback of given interest shown in the screenshot reveals that there was a march (Spanish word 'marcha' in word cloud) held by YoSoy132 to protest president election results, which is also reported by public media [1].

It is also possible to study targeted interested events spatially and temporally, by using the historical statistics analysis interface. Given a city and historical period range, users can track interest-related event trend of this city. In the bottom of Figure 5, we show interest-related event summary of given city *'Mexico City'*, from historical date *'2012,July'* to *'2012,December'*. Users can also compare spatial features of interested event. In the upper part of Figure 5, we list top-10 cities in given country *'Mexico'* with restrict to historical event number.

Acknowledgement

This work is supported by the Intelligence Advanced Research Projects Activity(IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337.

4. REFERENCES

[1] L. Diaz, G. Stargardter, I. Grillo, and P. Cooney. Protesters march against mexico's president election. http://www.reuters.com/article/2012/07/07/us-mexico-election-protest-idUSBRE8660I820120707.

[2] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. pages 137–142. Springer, 1998.

[3] M. Kulldorff. Spatial scan statistics: models, calculations, and applications. pages 303–322. Springer, 1999.

[4] D. B. Neill. Fast subset scan for spatial pattern detection. Wiley Online Library, 2012.

[5] M. E. Newman. Fast algorithm for detecting community structure in networks. volume 69, page 066133. APS, 2004.

[6] J. Weng and B. Lee. Event detection in twitter. In *ICWSM'11*, volume 3, 2011.