Intelligent Data Analysis 21 (2017) 849–866
DOI 10.3233/IDA-160048
IOS Press

Unsupervised event exploration from social text streams

Deyu Zhou[a,*], Liangyu Chen[a], Xuan Zhang[a] and Yulan He[b]
[a] *School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China*
[b]*School of Engineering and Applied Science, Aston University, UK*

Abstract. Social media provides unprecedented opportunities for people to disseminate information and share their opinions and views online. Extracting events from social media platforms such as Twitter could help in understanding what is being discussed. However, event extraction from social text streams poses huge challenges due to the noisy nature of social media posts and dynamic evolution of language. We propose a generic unsupervised framework for exploring events on Twitter which consists of four major steps, filtering, pre-processing, extraction and categorization, and post-processing. Tweets published in a certain time period are aggregated and noisy tweets which do not contain newsworthy events are filtered by the filtering step. The remaining tweets are pre-processed by temporal resolution, part-of-speech tagging and named entity recognition in order to identify the key elements of events. An unsupervised Bayesian model is proposed to automatically extract the structured representations of events in the form of quadruples $<$ entity, keyword, date, location $>$ and further categorize the extracted events into event types. Finally, the categorized events are assigned with the event type labels without human intervention. The proposed framework has been evaluated on over 60 million tweets which were collected for one month in December 2010. A precision of 78.01% is achieved for event extraction using our proposed Bayesian model, outperforming a competitive baseline by nearly 13.6%. Moreover, events are also clustered into coherence groups with the automatically assigned event type labels with an accuracy of 42.57%.

Keywords: Social media, event extraction, bayesian model, unsupervised learning

1. Introduction

Newsworthy events describe what has happened around the world and might directly or indirectly affect everyone in the world. With the fast development of social media platforms, newsworthy events are widely scattered not only on traditional news media but also on social media. For example, Twitter, one of the most widely adopted social media platforms, appears to cover nearly all newswire events. As has been previously reported in [18], even 1% of the public stream of Twitter contains around 95% of all the events reported in the newswire. Furthermore, social networking sites allow people sharing their thoughts and opinions towards a wide range of events. Hence, it is possible to understand general public's reactions to events from social media stream data, which can facilitate downstream applications such as tracking public's viewpoints.

Therefore, it is crucial to extract events from social streams such as tweets. Events have been represented in different ways for different purposes. In Automated Content Extraction Program (ACE),

[*] Corresponding author: Deyu Zhou, School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing, Jiangsu 210096, China. E-mail: d.zhou@seu.edu.cn.

Table 1
Examples of event-related tweets without temporal information

Tweets
Cruise ship off Florida returning to port after rescuing six migrants on a raft.
ESPN and 9News report Broncos head coach Josh McDaniels – FIRED.
Brittany Mae Smith Missing http://t.co/wLdFLvC via @cbsnews I hope she is returned safe!
Doors' Jim Morrison pardoned for indecent exposure.
Elizabeth Edwards' Funeral: Controversial Westboro Baptist Church Plans Protest.
Matt Cardle is winner of The X Factor: Matt Cardle has been crowned winner of The X Factor, beating Rebecca Ferg ...

an event is represented as a '6W' tuple (*Who* did *What* to Whom, *Where* and *When*, through *What* methods and *Why*) with a varying number of components depending on the system task (http://www.itl.nist.gov/iad/mig/tests/ace/). We adopt such a representation and cast event extraction into a predication among arbitrary number of arguments and their relationships.

Previous research in event extraction focused largely on news articles. Event extraction techniques typically rely on the detection of event "triggers" with their arguments for slot filling in event frames. Classical approaches to event extraction can be roughly divided into three classes, pattern based [26], machine learning based [19] and a hybrid combining the previous two categories [11]. Compared to newswire text, the social stream data such as tweets have the following characteristics:

- *Noisy and informal writing styles.* Social media messages are often short, contain a large number of irregular and ill-formed words, and evolve rapidly over time. Comparing to news articles, it is more challenging to process such fragmented and noisy messages. Also, most social media messages are not event-related.
- *Unknown event types.* Social media data are produced continuously by a large and uncontrolled number of users. As such, it is not possible to know the event types a priori and hence makes it hard to apply the existing event extraction approaches which either rely on manually-defined linguistic patterns representing expert knowledge to extract events or make use of corpora annotated with event-specific information such as actors, date, place, etc., to learn event extraction patterns.
- *Redundancy.* For most newsworthy events, there may be a high volume of redundant messages referring to the same event.

The aforementioned characteristics of social stream data pose new challenges but also provide opportunities to employ unsupervised approaches for event extraction and categorization based on the redundancy property of event-related tweets. Recently there has been much interest in event extraction from Twitter. Ritter et al. [22] presented a system called TwiCal to extract and categorize events from Twitter. They relied on a sequence labeler trained from annotated data to extract event phrases from Twitter. In [1], a system called EvenTweet was constructed to extract localized events from a stream of tweets in real-time. The extracted events are described by start time, location and a number of related keywords. Instead of employing annotated data for event extraction, we have proposed an unsupervised Bayesian model called Latent Event & Category Model (LECM) for event extraction and categorization [28]. It is assumed that in the model, each tweet message $m \in \{1..M\}$ is assigned to one event instance e, while e is modeled as a joint distribution over the named entities y, the date/time d when the event occurred, the location l where the event occurred and the event-related keywords k.

However, a careful examination of the tweets data reveals that most tweets do not include temporal expression (the date/time when the event occurred). Taking the collected tweets in the month of December in 2010 as an example, out of a total of 706,815 tweets after filtering, only 133,031 (less than 20%) of tweets contain temporal expressions. Table 1 provides some examples of tweets without temporal expressions. Simply assigning the event date with the corresponding tweets' publishing date could

Table 2
Definition of notations

Notation	Definition
$\alpha, \tau, \rho, \delta, \beta, \eta, \lambda$	Hyperparameters of the LECM/LECM-d model
π	Parameter of event distribution
e	An event
$\boldsymbol{e}$	A set of events
w	A tweet
w_m	The mth tweet in the corpus
y, l, k, d	Named entities, locations, keywords and dates
θ, ψ, ω	Parameters of the distributions of the named entity, location and keyword
ν	Parameter of the distribution of event type
t	Event type
$\boldsymbol{t}$	A set of event types
ε, ζ	Parameters of the distributions of semantic classes and keywords for event type detection
y', k'	The semantic class index and keywords used for event type
n_i	The number of i, where i denotes a kind of data
x^{-i}	All x data excluding the ith data
Y, L, V	The total numbers of distinct named entities, locations, and keywords
Y_e, K_e	The set of y' and k' belonging to the event e
M, E, C, S	The number of tweets, events, event types and distinct non-location named entities' sematic classes

Table 3
Examples of tweets mentioning the same event but published in different dates

Event	Tweet	Publishing date
<Yao Ming Ankle, Breaking, Dec-16, ->	My reaction to Yao Ming breaking is like Riley's reaction to Gangstalicious being shot: "Again!?"	Dec-16
	Yao Ming out indefinitely with stress fracture in ankle	Dec-16
	Time to call it a career! RT @espn Yao Ming out indefinitely with stress fracture in ankle	Dec-16
	That Sucks! RT @espn Yao Ming out for season with stress fracture in ankle	Dec-17
	Yao Ming out for season with stress fracture in ankle	Dec-17
	damn... Poor Yao..@espn: Yao Ming out for season with stress fracture in ankle	Dec-17
<Protesters Prince Charles, Attack, Dec-09, Alphabet St>	'Protesters attack prince's car' Apparently whilst driving down Alphabet St yobs smashed up his little red corvette.	Dec-09
	Witness phone in on the news "Prince Charles was in a black cab"..that'll be the dark maroon, classic Rolls Royce not a hackney carriage!	Dec-09
	Protesters demonstrating against the increase in tuition fees have attacked a car containing Prince Charles and Camilla.	Dec-09
	Protesters attack Prince Charles' car in anti-fees demo	Dec-10
	Prince Charles and Camilla's car attacked by student protesters in London; both unhurt. princecharles	Dec-10
	Folks are in an uproar that student protesters managed to pelt Prince Charles & Camilla's car w/paint.	Dec-10

result in one event being assigned with multiple dates and hence cause ambiguities. Table 3 shows some examples of tweets mentioning the same event but with different publishing date.

To tackle this problem, we propose to modify our LECM model by dropping the date element. The event date information is inferred more accurately based on combining heuristic rules with the outputs generated by LECM-d. As will be discussed in the Experiments section, our results on a large dataset consisting of over 60 million tweets show that our modified model significantly improves upon TwiCal by nearly 13.6% in precision and outperforms the original LECM model by 9.75% in precision. Moreover, events are also clustered into coherence groups with the automatically assigned event type label with an accuracy of 42.57%.

2. Related work

Our work is related to two lines of research, event extraction and event detection. Here, an event refers to something that happens at certain time and place. We distinguish between event extraction and detection in which event extraction aims at extracting structured information from text while event detection focuses on discovering new or previously unidentified events. In the following, we present a brief survey of related work in event extraction and event detection.

2.1. Event extraction

Event extraction has been largely studied on news articles. Methods proposed include machine learning, pattern-based and a hybrid of both. In [16], event extraction is considered as a clustering problem and two novel distance metrics are employed on heterogeneous news sources. In [8], specific patterns are designed and employed for biomedical event extraction. In [11], a combination of pattern matching and statistical modeling techniques is used. Two types of patterns are constructed including the sequence of constituent heads separating anchor and its arguments and a predicate argument subgraph of the sentence connecting anchor to all the event arguments.

In recent years, event extraction from tweets has received an increased interest. Focusing on entertainment events, Benson et al. [6] proposed a structured graphical model which simultaneously analyzes individual messages, clusters them according to event, and induces a canonical value for each event property. The method yields up to a 63% recall against the city table and up to 85% precision evaluated manually. Popescu [20] focused on detecting events involving known entities from twitter. Experimental results showed that events centered on specific entities can be extracted with 70% precision and 64% recall. Liu et al. [15] work on social events extraction for social network construction using a factor graph by harvesting the redundancy in tweets. Experiments were conducted on a human annotated data set and results showed that the proposed method achieved an absolute gain of 21% in F-measure. Li et al. [14] paid attention to personal major life events such weddings and graduation. A pipeline based system was constructed to extract a fine-grained description of users' life events based on their published tweets.

Ritter et al. [22] presented a system called TwiCal to extract and categorize events from Twitter. The strength of association between each named entity and date based on the number of tweets they cooccur in is measured to determine whether the extracted event is significant. The approach achieved an increase in maximum F-measure over a supervised baseline. Anantharam et al. [3] focused on extracting and understanding city events. The problem is formulated as a sequence labeling problem. Evaluation was carried out on a real-world dataset consisting of event reports and tweets collected over four months from San Francisco Bay Area. Sandeep et al. [17] proposed algorithms to extract attribute-value pairs and map such pairs to manually generated schemas for natural disaster events. Evaluation was carried out on 58000 tweets for 20 events and the system can fill such event schemas with an F-measure of 60%.

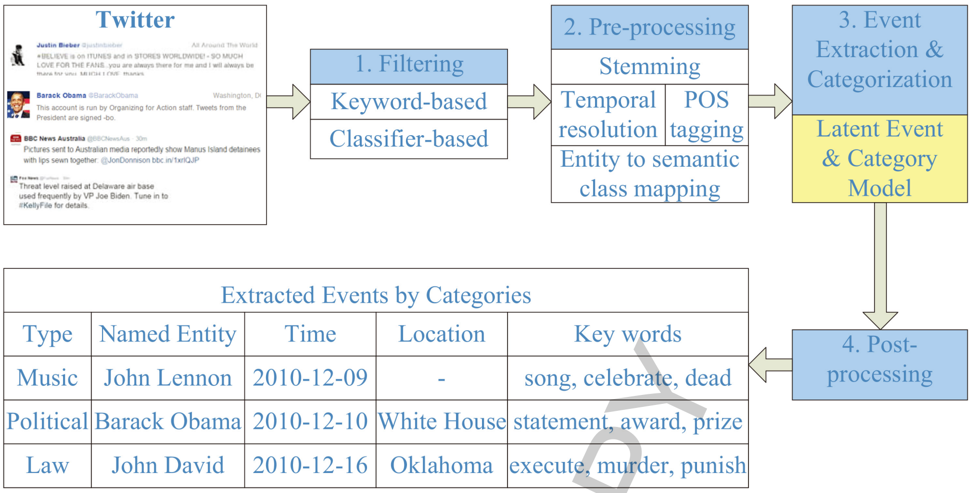
Our work is similar to TwiCal in the sense that we also focus on the extraction and categorization of structured representation of events from Twitter. However, TwiCal relies on a supervised sequence labeler trained on tweets annotated with event mentions for the identification of event-related phrases. We propose a simple Bayesian modelling approach which is able to directly extract event-related keywords from tweets without supervised learning. TwiCal uses G^2 test to choose an entity y with the strongest association with a date d to form a binary tuple $\langle y, d\rangle$ to represent an event. On the contrary, the structured representation of events can be directly extracted from the output of our Bayesian model. Moreover, the extracted event groups are categorized and assigned with event type labels automatically in our proposed framework. We have conducted experiments on a Twitter corpus and the results show that our proposed approach outperforms TwiCal, the state-of-the-art open event extraction system, by nearly 13.6% in precision.

2.2. Event detection

Instead of extracting structured representations of events, event detection aims to discover new or previously unidentified events. Event detection has long been addressed in the Topic Detection and Tracking (TDT) program sponsored by the Defense Advanced Research Projects Agency. The concept of event in event detection [2] is defined as real-world occurrence e with an associated time period T_e and a time-ordered stream of news messages M_e, of substantial volume, discussing the occurrence and published during time T_e. There has been some recent work on detecting events or tracking topics on Twitter. Sankaranarayanan et al. [24] detected breaking news from tweets to build a news processing system, called TwitterStand. A Naïve Bayes classifier was employed to separate news from irrelevant information and an online clustering algorithm was used to group tweets into different clusters. Sakaki et al. [23] trained a classifier based on features derived from individual tweets (e.g., the keywords in a tweet and the number of words it contains) to detect a particular type of event such as earthquakes and typhoons. They formulated event detection as a classification problem and trained a Support Vector Machine (SVM) on a manually labeled Twitter dataset comprising positive events (earthquakes and typhoons) and negative events (other events or non-events). In [1], EvenTweet, a system to detect localized events from a stream of tweets, was constructed. Localized events were extracted from a stream of tweets in real-time by temporal and spatial keyword clustering and cluster scoring. The extracted events are described by a number of related keywords, start time and the location. Becker et al. [5] focused on online identification of real-world event content and its associated Twitter messages using an online clustering technique, which continuously clusters similar tweets and then classifies the clusters content into real-world events or non-events. Lee and Sumiya [13] proposed a geo-social event detection system based on modeling and monitoring crowd behavior via Twitter, to identify local festivals. A brief overview of event detection techniques applied to Twitter can be found in [4]. In [27] a unsupervised approach for detecting spatial events in targeted domains is presented. A dynamic query expansion algorithm is proposed to iteratively expand domain-related terms, and generate a tweet homogeneous graph. An anomaly identification method is utilized to detect spatial events over this graph by jointly maximizing local modularity and spatial scan statistics.

3. Methodologies

Our proposed framework consists of four main steps, filtering, pre-processing, event extraction and categorization, and post-processing, as illustrated in Fig. 1. Table 2 lists notations used in this paper. Given a raw stream of Twitter, irrelevant or noisy tweets are filtered out firstly. Only tweets which are more likely describing events are kept and processed by temporal resolution, part-of-speech (POS) tagging and named entity recognition in the pre-processing step. Afterwards, a Bayesian model is proposed and employed for event extraction and categorization. Here, an event is represented as a tuple $\langle y, d, l, k\rangle$ where y stands for non-location named entities, d for a date, l for a location, and k for event-related



Extracted Events by Categories				
Type	Named Entity	Time	Location	Key words
Music	John Lennon	2010-12-09	-	song, celebrate, dead
Political	Barack Obama	2010-12-10	White House	statement, award, prize
Law	John David	2010-12-16	Oklahoma	execute, murder, punish

Fig. 1. The proposed framework for exploring event from Twitter.

keywords. Each event mentioned in tweets can be closely depicted by this representation. It should be noted that for some events, one or more elements in their corresponding tuples might be absent since the information relating to certain event elements might not be available in tweets. The details of our proposed framework are described below.

3.1. Tweet filtering

Two approaches have been explored for filtering tweets. The first approach is through lexicon matching. By collecting news articles published around the same period as tweets, a lexicon is constructed by extracting keywords from these articles based on a measure such as TF-IDF (term frequency-inverse document frequency). Then, only the tweets containing words that can be found in the lexicon are kept.

Apart from the keyword-based approach, we have employed another feature based approach, which casts tweet filtering as a binary classification problem. Given a set of tweets $M = (m_1, ..., m_k)$, the classifier outputs a class label $C \in \{\text{event, non-event}\}$. The non-event tweets are removed at this stage. To build a good classifier, it is crucial to design a proper feature set. Considering that the number of event-related tweets is significantly less than non-event-related tweets, we propose to construct a feature set in the following way.

- *Binary word features*. We select words occurred more frequently in event-related tweets but rarely in non-event tweets as highly class-indicative features to build our feature set. The importance score of a word is defined as *TFP*/*TFN*, where *TFP* is the term frequency in the event-related tweets while *TFN* is the term frequency in non-event tweets. We sort the words by their importance scores and only select the top n words to construct binary features (presence of the word or not). We have tried $\{50, 100, 200\}$ for the top n words and 100 gives the best performance.
- *Other event-related features*. We notice that tweets containing information related to authoritative news agencies such as CNN or BBC and some phrases such as "breaking news" most likely describe real-world events. As such, we also include binary features indicating the presence of news agencies and some manually selected indicative phrases. Furthermore, we add other binary features [25] which consist of time-related phrases, opinionated words, currency and percentage signs, URLs, reply to other users such as "@username", etc.

– *Event elements*. As an event is described as "something that happens at a given place and time", the presence of named entity, location, and time information could be potentially useful to detect the occurrence of an event in text. Hence, they are also used as features to train a binary classifier.

3.2. Pre-processing

In the proposed framework, an event is represented as a tuple of named entities, date, location, and event-related keywords. Therefore, it is crucial to identify date, location and named entities in Tweets. As Twitter users might represent the same date in various forms, SUTime (http://nlp.stanford.edu/software/sutime.shtml) [7] is employed to resolve the ambiguity of time expressions. For example, temporal expressions such as "tomorrow" and "last Friday" are mapped to a specific date based on the tweet's publish date. Named entity recognition (NER) is a crucial step since the results would directly impact the final extracted 4-tuple $\langle y, d, l, k \rangle$. It is not easy to accurately identify named entities in the Twitter data since tweets contain a lot of misspellings and abbreviations. A named entity tagger trained specifically on the Twitter data (http://github.com/aritter/twitter-nlp) [21] is used to directly extract named entities from tweets. A POS tagger[1] trained on tweets [9] is used to perform POS tagging on the tweets and apart from the previously recognised named entities, only words tagged with nouns, verbs or adjectives are kept. These remaining words are subsequently stemmed and words occurred less than 3 times are filtered. We use the API provided by Freebase (http://www.freebase.com/) to map named entities to semantic classes. This is to provide a certain level of abstraction of named entities. For example, "Celine Dion" and "Justin Bieber" will be mapped to the "music" class. For named entities with more than one semantic class, we simply choose the one with the highest relevance score.

3.3. Event extraction and categorization

We have proposed an unsupervised latent variable model called LECM to extract and cluster event instances [28]. It is assumed that in the model, each tweet message $m \in \{1..M\}$ is assigned to one event instance e, while e is modeled as a joint distribution over the named entities y, the date/time d when the event occurred, the location l where the event occurred and the event-related keywords k. This assumption essentially encourages events that involve the same named entities, occur at the same time and in the same location and have the same keywords to be assigned with the same event. As the event distribution is shared across social media posts with the same named entities, dates, locations and keywords, it essentially preserves the ambiguity that for example, events comprising the same date and location may or may not belong to the same event. It is also assumed that each event e is assigned to one event type t, while t is modeled as a joint distribution over the semantic classes y' to which the named entities are mapped and the event-related keywords k. This assumption encourages events that involve the same semantic class and have similar keyword to be categorized into the same event type.

However, after a close examination of the collected tweets data which will be discussed in more details in Section 4, we found that very few tweets contain temporal expression. Out of a total of 706,815 tweets after filtering, only 133,031 (less than 20%) of tweets contain temporal expressions. As such, for most tweets, their publish timestamps have been used to set the "date" element in the LECM model. However, tweets discussing a certain event could be published one or two days after the event actually happened. Also, LECM model allows event instances with similar keywords but different dates to be clustered into

[1] http://www.ark.cs.cmu.edu/TweetNLP.

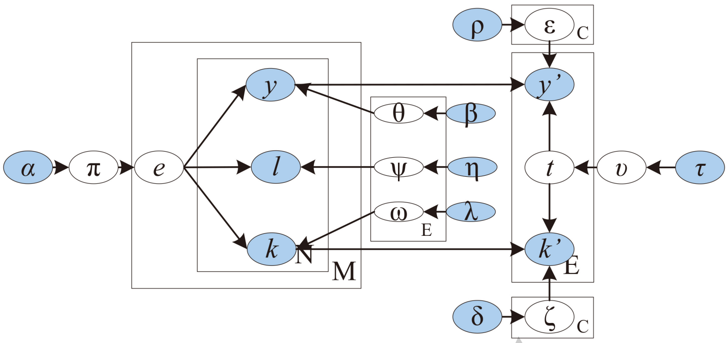


Fig. 2. LECM-d: A latent variable model for event extraction and categorization.

the same event. It results in the same event being associated with multiple dates. To tackle this problem, we propose to modify LECM by dropping the "date" element and called the modified model LECM-d. The graphical model of LECM-d is shown in Fig. 2. In addition, we propose to combine some heuristic rules with the outputs generated by LECM-d to infer the date information of the extracted events more accurately:

1. *Split tweets into bins where each bin corresponds to a specific date*. To infer event dates more reliably, we consider both tweets' publish timestamps and the temporal expressions found in tweets. For tweets without temporal expression, the events discussed in tweets are assumed to happen on the same day d as the tweets being published. These tweets are grouped in the bin corresponding to d. For a tweet with a temporal expression, if the resolved date d_t based on temporal expression differs from tweet's published date d_p and the tweet is published within 7 days after d_t, then the resolved date is considered as the potential event date and the tweet is moved to the bin corresponding to d_t instead of d_p.
2. *Extract events separately for each bin*. The LECM-d model is used to extract events and event types from tweets in each bin. Here, the events extracted do not have the date information.
3. *Infer the date information for each extracted event*. We assume that the earliest date when the event is mentioned on Twitter is the date when it happened. Therefore, for each extracted event, the date information is assigned based on the merging step proposed below. Firstly, we compared the events extracted from tweets in nearby bins. Events with overlapping entities and similar keywords are considered as the same event and merged. The merged events are then assigned with the date when the event was first mentioned.

The generative process of LECM-d is shown below.

- Draw the event distribution $\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha)$.
- Draw the event type distribution $\boldsymbol{\upsilon} \sim \text{Dirichlet}(\tau)$.
- For each event $e \in \{1..E\}$, draw distributions $\boldsymbol{\theta}_e \sim \text{Dirichlet}(\beta), \boldsymbol{\psi}_e \sim \text{Dirichlet}(\eta), \boldsymbol{\omega}_e \sim \text{Dirichlet}(\lambda)$.
- For each event type $t \in \{1..C\}$, draw distributions $\boldsymbol{\epsilon}_t \sim \text{Dirichlet}(\rho), \boldsymbol{\zeta}_t \sim \text{Dirichlet}(\delta)$.
- For each tweet w:
  - $*$ Choose an event $e \sim \text{Multinomial}(\boldsymbol{\pi})$.
  - $*$ For each named entity occur in tweet w, choose a named entity $y \sim \text{Multinomial}(\boldsymbol{\theta}_e)$.
  - $*$ For each location occur in tweet w, choose a location $l \sim \text{Multinomial}(\boldsymbol{\psi}_e)$.
  - $*$ For other word positions, choose a word $k \sim \text{Multinomial}(\boldsymbol{\omega}_e)$.

- For each event e:
  - Choose an event type $t \sim \text{Multinomial}(\boldsymbol{\upsilon})$.
  - For each named entity occur in event e, choose a semantic class $y' \sim \text{Multinomial}(\boldsymbol{\epsilon}_t)$.
  - For each keyword in event e, choose a keyword $k' \sim \text{Multinomial}(\boldsymbol{\zeta}_t)$.

Letting $\Lambda = \{\alpha, \beta, \eta, \lambda, \tau, \rho, \delta\}$, we obtain the marginal distribution of a tweet w by integrating over $\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\upsilon}, \boldsymbol{\omega}, \boldsymbol{\epsilon}$ and $\boldsymbol{\zeta}$ and summing over event $\boldsymbol{e}$ and type $\boldsymbol{t}$:

$$
\begin{aligned}
P(w|\Lambda) = & \int\int\int\int\int\int\int P(\boldsymbol{\pi};\alpha) \\
& \prod_{e=1}^{E} \left\{ P(\boldsymbol{\theta_e};\beta) P(\boldsymbol{\psi_e};\eta) P(\boldsymbol{\omega_e};\lambda) P(e|\boldsymbol{\pi}) \prod_{n=1}^{N} P(x_n|e, f_e) \right\} \\
& P(\boldsymbol{\upsilon};\tau) \prod_{t=1}^{C} \left\{ P(\boldsymbol{\epsilon_t};\rho) P(\boldsymbol{\zeta_t};\delta) P(t|\boldsymbol{\upsilon}) \prod_{e=1}^{E} P(z_e|t, g_t) \right\} \\
& d\boldsymbol{\pi}\, d\boldsymbol{\theta}\, d\boldsymbol{\psi}\, d\boldsymbol{\omega} d\boldsymbol{\upsilon}\, d\boldsymbol{\epsilon}\, d\boldsymbol{\zeta}
\end{aligned}
\tag{1}
$$

Here, depending on the word type at each word position n, w, x_n could be y_n, l_n, k_n and f_e is its corresponding Multinomial distribution, z_e could be y'_e, k'_e and g_t is its corresponding distribution.

Taking the product of marginal probabilities of tweets in a corpus gives us the probability of the corpus.

$$
P(\mathcal{D}|\Lambda) = \prod_{m=1}^{M} P(w_m|\Lambda) \tag{2}
$$

We use collapsed Gibbs sampling [10] to infer the parameters of the model and the latent class assignments for events and categories, given observed data $\mathcal{D}$ and the total likelihood. Gibbs sampling allows us repeatedly sample from a Markov chain whose stationary distribution is the posterior of e_m, t_e from the distribution over that variable given the current values of all other variables and the data. Different from Gibbs sampling, collapsed Gibbs sampling integrates out some variables which we are not interested in when sampling. For example, for a model consisting of 3 variables x, y, z in which we are not interested in z, Gibbs sampling needs to sample from $p(x|y,z), p(y|x,z), p(z|x,y)$ in turn, while collapsed Gibbs sampler samples from marginal distribution $p(x|y), p(y|x)$ by integrating z out. Such samples can be used to empirically estimate the target distribution.

Letting the subscript $-m$ denotes a quantity that excludes data from the mth tweet, the conditional posterior for e_m is:

$$
\begin{aligned}
P(e_m = e|\boldsymbol{e}_{-m}, \boldsymbol{y}, \boldsymbol{l}, \boldsymbol{k}, \Lambda) \propto & \frac{n_e^{-m} + \alpha}{M + E\alpha} \times \prod_{y=1}^{Y} \frac{\prod_{b=1}^{n_{e,y}^{(m)}} (n_{e,y} - b + \beta)}{\prod_{b=1}^{n_e^{(m)}} (n_e - b + Y\beta)} \\
\times \prod_{l=1}^{L} \frac{\prod_{b=1}^{n_{e,l}^{(m)}} (n_{e,l} - b + \eta)}{\prod_{b=1}^{n_e^{(m)}} (n_e - b + L\eta)} & \times \prod_{k=1}^{V} \frac{\prod_{b=1}^{n_{e,k}^{(m)}} (n_{e,k} - b + \lambda)}{\prod_{b=1}^{n_e^{(m)}} (n_e - b + V\lambda)}
\end{aligned}
\tag{3}
$$

where n_e is the number of tweets that have been assigned to the event e; M is the total number of tweets, $n_{e,y}$ is the number of times named entity y has been associated with event e; $n_{e,l}$ is the number of times locations l has been assigned with event e; $n_{e,k}$ is the number of times keyword k has associated with event e, counts with (m) notation denote the counts relating to tweet m only. Y, L, V are the total numbers of distinct named entities, locations, and words appeared in the whole Twitter corpus respectively. E is the total number of events which needs to be set.

Letting the subscript $-e$ denote a quantity that excludes data from eth event, the conditional posterior for t_e is:

$$P(t_e = t|\boldsymbol{t}_{-e}, \boldsymbol{y'}, \boldsymbol{k'}, \Lambda) \propto \frac{\tau + n_t^{-e}}{E + C\tau} \times \prod_{\tilde{y}\in Y_e} \frac{\rho + n_{t,\tilde{y}}^{-e}}{\sum_{y'=1}^{S} n_{t,y'}^{-e} + S\rho} \times \prod_{\tilde{k}\in K_e} \frac{\delta + n_{t,\tilde{k}}^{-e}}{\sum_{k'=1}^{V} n_{t,k'}^{-e} + V\delta} \tag{4}$$

where C is the number of the event types, Y_e is the set of y' belonging to e, $n_{t,y'}$ is the times of non-location entity' semantic class y' being assigned with event type t, K_e is the set of k' belonging to e, $n_{t,k'}$ is the times of the keyword k' being assigned with event type t and S is the total number of distinct non-location named entities' semantic classes appeared in the whole Twitter corpus respectively.

Once the class assignments for all events are known, we can easily estimate the model parameters $\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\upsilon}, \boldsymbol{\omega}, \boldsymbol{\epsilon}$ and $\boldsymbol{\zeta}$. We set the hyperparameters $\alpha = \beta = \eta = \lambda = \tau = \rho = \delta = 0.5$ and run Gibbs sampler for 1000 iterations and stop the iteration once the log-likelihood of the training data converges under the learned model. Finally we select an entity, a date, a location, and the top 2 keywords of the highest probability of every event to form a 4-tuple as the representation of that event.

3.4. Post-processing

To improve the precision of event extraction and categorization, we remove the least confident event element from the 4-tuple in LECM and LECM-d using the following rules.

- If N (element) $< n_1$, the element will be removed from the extracted results. Here, N (element) is the number of occurrence of the element in the tweets with event e.
- If N (element) $> n_e/n_2$, the element will be kept. Here, n_e is the number of tweets with event e.

Here, n_e denotes the number of tweets describing event e. The element described here includes non-location named entities and locations, which are core elements of events. n_1, n_2, n_3 are the thresholds and we have tried $\{5, 10, 15\}$ for n_1, n_2, n_3 and $\{n_1 = 5, n_2 = 5, n_3 = 10\}$ gives the best performance.

Our model automatically groups events into different event clusters. For each event cluster, the most prominent semantic class obtained based on the event entities in the cluster is used as the event type label.

4. Experiments

In this section, we firstly describe the datasets used in our experiments and then introduce the baseline system for comparison. Then experimental results on filtering, extraction and categorization are subsequently presented. Finally, errors analyses are conducted to give the insights of the proposed framework.

4.1. Setup

Two datasets are constructed by collecting tweets in the month of December in 2010. Dataset I contains tweets which are manually annotated as event-related or not for the training of a binary classifier in the filtering step. Tweets are annotated as event-related if relevant news articles can be found in the one-week window before and after the tweets' publication dates. We argue that this is a reasonable choice since newsworthy events would be more interesting than others. In total, we have

Table 4
The performance of tweet filtering on Dataset I

Method	Precision (%)	Recall (%)	F-measure (%)
Keyword-based	73.03	25.73	38.05
SVM-based	81.65	11.22	19.73

2,891 event-related and 26,000 non-event-related tweets in Dataset I. Dataset II (http://cse.seu.edu.cn/people/zhoudeyu/AAAI2015-data.zip) contains 60 million unlabelled tweets which are used to evaluate the proposed framework.

The baseline we chose is TwiCal [22], the state-of-the-art open event extraction system on tweets. Each event extracted in the baseline are represented as a 3-tuple $\langle y, d, k\rangle$, where y stands for a non-location named entity, d for a date and k for an event phrase. We re-implemented the whole system and evaluate the performance of the baseline on the correctness of the exacted three elements only excluding the location element. Moreover, the parameters of TwiCal are optimized based on the suggestion mentioned in [22].

The evaluation is conducted in three aspects: filtering, extraction and categorization.

- *Tweet filtering.* As most tweets in Datasets I and II are not event-related, we only report the performance of classifying event-related tweets. Precision is defined as the proportion of the correctly identified event-related tweets out of the system returned event-related tweets. Recall is defined as the proportion of correctly identified true event-related tweets.
- *Event extraction.* Due to the large volume of tweets in Dataset II, it is almost impossible to know the exact number of events it contains. Therefore, we only report the precision of our event extraction results. For the 4-tuple $\langle y, d, l, k\rangle$, the precision value is calculated based on the two rules: (1) Do the entity y, location l and date d extracted refer to the same event? (2) Are the keywords k in accord with the event that other extracted elements y, l, d refer to and are they informative enough to tell us what happened?
- *Event categorization.* The performance is evaluated in two ways, only considering the correct extracted events and using all the extracted events.

4.2. Results of tweet filtering

As has been previously discussed in Section 3.1, we have explored both keyword-based and classifier-based approaches for tweet filtering. For classifier-based approach, we use Weka [12] to train an SVM with default parameters on Dataset I and perform 3-fold cross validation. For keyword-based approach, news articles were collected from GDELT Event Database (http://gdeltproject.org/). The GDELT Event Database contains over a quarter-billion event records including new articles in the world's news media. The results are shown in Table 4.

Since most tweets in Dataset I are not event-related, it makes sense to only report the results on the event-related class. It can be observed that the SVM-based approach achieves higher precision but with much lower recall rate. It might be attributed to the highly imbalanced training data in Dataset I where only about 10% tweets are event-related. We also tested both keyword-based and SVM-based approaches on Dataset II. Due to the large size of Dataset II, it is impossible to find out the actual performance of both approaches. We instead randomly selected 1,000 tweets identified as event-related by each approach and manually checked the accuracy. We found that the keyword-based approach gives higher precision compared to the SVM-based approach. As such, we chose to use the keyword-based approach for tweet filtering in all the subsequent experiments.

Table 5
Examples of the extracted events with or without filtering

Entity	Keywords
Extracted events without filtering	
Harry Potter	like, watch, movie
God	thank, wish, love
Lady Gaga	star, nightlife, blog
Justin Bieber	club, music, photo
Extracted events with filtering	
Windows Phone	os, release, mango
Philadelphia Eagles, Ubalo Jimenez	sign, championship, sense
Amy Winehouse	death, RIP, sad

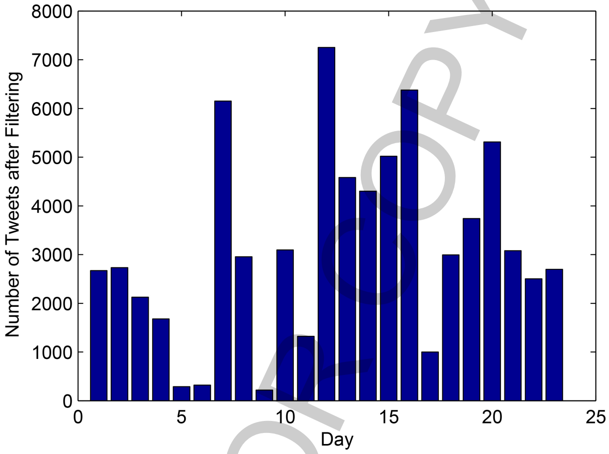


Fig. 3. The number of tweets in each day after filtering.

To further understand the effect of our filtering step, examples of the events extracted using our proposed framework with and without filtering are presented in Table 5. It can be observed that without filtering, some extracted events are not really newsworthy events although they also contain named entities and meaningful keywords. For example, there are many tweets talking about watching the movie “Harry Potter". However, it is not considered as a newsworthy event.

4.3. Results of event extraction

After filtering and pre-processing, less than 250,000 tweets in Dataset II are kept. Figure 3 shows the number of tweets after filtering in each day. It can be observed that the number of tweets varies significantly, of which the minimum is 220 and the maximum is 7,253. Comparing to 200,000 tweets per day in the original data, the filtering step has greatly filtered out non-event-related tweets for subsequent processing.

These tweets are fed into LECM and LECM-d for event extraction and categorization. For LECM-d, we need to group tweets by their potential event dates as mentioned in Section 3.3. Since for each potential event date d, we need to consider tweets published within 7 days after d, we only use tweets between 2010-12-03 and 2010-12-25 for LECM-d. Therefore, the input to LECM-d contains tweets in 23 days. In our experiments of LECM, the number of events is set to 400 which was chosen using the perplexity measure on the 10% held-out set from Dataset II. After post-processing, 315 events were

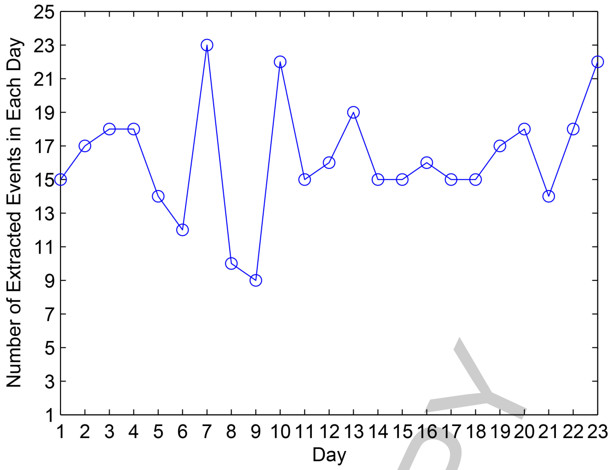


Fig. 4. The number of extracted events for each day using LECM-d.

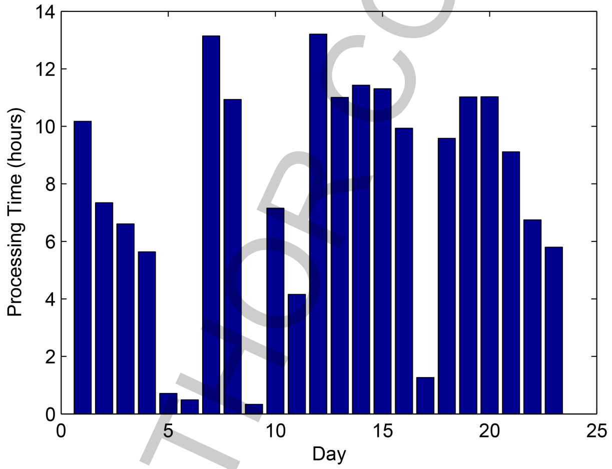


Fig. 5. The processing time on each day's data using LECM-d.

extracted finally. In LECM-d, the number of events is set to 30 for each day. Note that we have merged events in nearby days based on the named entities and keywords. As such, the exact number of events for each day may not be 30. Figure 4 shows the number of events extracted by LECM-d in each day. Altogether, 373 events were extracted after post-processing using LECM-d. The processing time using LECM-d on IBM 3850 X5 Linux server equipped with 1.86 Ghz processor and 8 GB DDR3 RAM is shown in Fig. 5.

The event extraction precisions using TwiCal, LECM, LECM-d are presented in Table 6. As TwiCal outputs a list of events ranked by confidence from high to low, the number of events to be extracted for TwiCal is set to 315 for fair comparison. It can be observed that the filtering step is really crucial to event extraction. By filtering out non-event-related tweets, the precision of our event extraction component increases dramatically from 28.33% to 68.25%. Our proposed framework using LECM-d has the best performance with the precision 78.01% and every event is assigned with a date.

When compared against the baseline approach, TwiCal, it can be observed from Table 6 that LECM

Table 6
Comparison of the performance of event extraction on Dataset II

Method	Precision	Number of extracted events
LECM without Filtering	28.33%	315
TwiCal	64.44%	315
LECM with Filtering	68.25%[2]	315
LECM-d with Filtering	78.01%	373

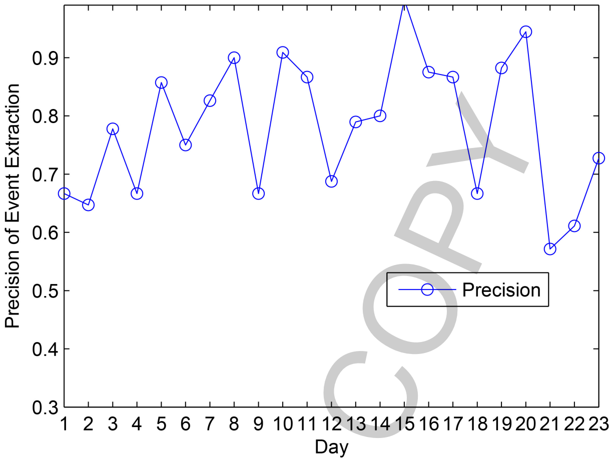


Fig. 6. The precision of event extraction in each day using LECM-d.

significantly outperforms the baseline with nearly 3.8% improvement on precision. Moreover, LECM-d further improves upon LECM by 9.75% and outperforms TwiCal by 13.6%. The accuracy of event extraction in each day by LECM-d is shown in Fig. 6. It can be observed that for some days, the precision of event extraction even reaches 100%.

The significant improvement over TwiCal can be attributed to two main reasons. One is that in a large scale Twitter dataset such as Dataset II, tweets with temporal keywords are rare and many event-related tweets have no date information. As such, TwiCal which relies on the association between named entities and dates for event extraction fails to handle tweets with no date information. The other reason is that TwiCal assumes that one event has only one named entity, which is not true in some cases. For example, in the tweet “Russian President Dmitry Medvedev on Thursday congratulated President Barack Obama on the Senate’s approval of a new nuclear arms control treaty between the countries”, both “Dmitry Medvedev” and “Barack Obama” are involved. Our proposed approach does not impose such a constraint.

To further understand the clustering effect of the proposed LECM-d, we analyze the number of tweets related to each extracted event. The statistics are shown in Fig. 7. It can be observed that most events are mentioned in less than 300 tweets. Only 9 events are mentioned in more than 600 tweets.

4.3.1. Impact of the number of event E in LECM-d

To see the impact of the event number chosen in LECM-d, we report the extracted results with different number of events E on one-day tweets only in Fig. 8. Similar results are observed for other days. It can be observed that when E is increased beyond 40, the number of correctly extracted events remains unchanged while the number of false positives keeps on increasing. As such, the optimal value of E is 30.

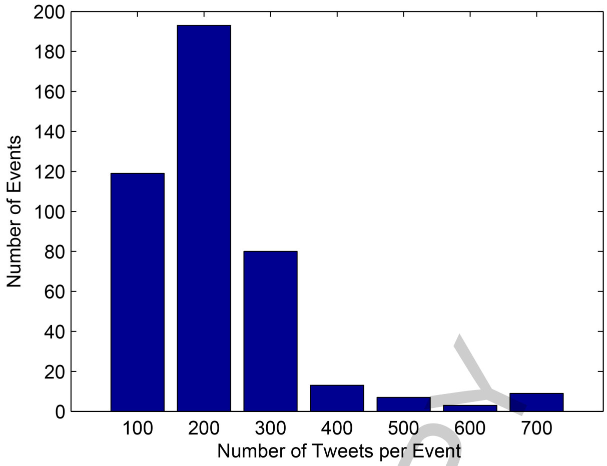


Fig. 7. The number of tweets versus the number of events.

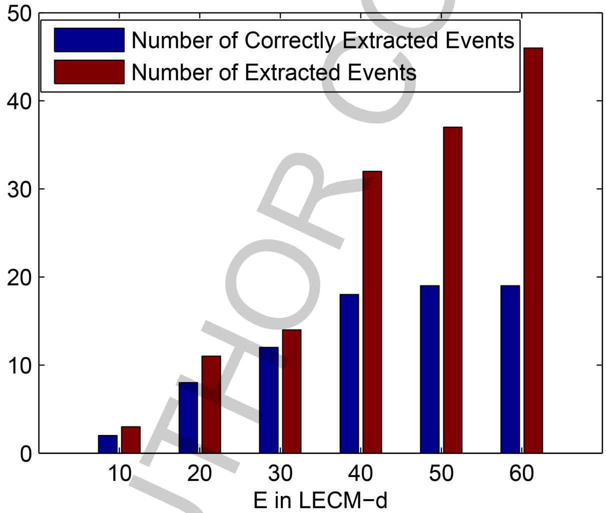


Fig. 8. The number of extracted events versus the number of correctly extracted events using different E in LECM-d.

4.3.2. LECM vs LECM-d

We have compared the events extracted by LECM and LECM-d and found that 160 events, about 74.4% of correctly extracted events by LECM, are also correctly extracted by LECM-d. However, 131 events correctly extracted by LECM-d are not discovered by LECM. This shows that our proposed method in inferring date information from tweets could potentially help in improving the recall rate of the system. Table 7 presents some examples of errors generated by simply using the publishing date as the event date but are corrected by our proposed method.

4.4. Results of event categorization

The event extraction and categorization component automatically clusters events into different event types. We empirically set the number of event types to 25 in both LECM and LECM-d. Some example

Table 7
Examples of extraction errors caused by using publishing date as event date. These errors are corrected by the proposed approach

Entity	Location	Keywords	Date by LECM-d	Date by using publishing date
Carmelo Anthony	–	Trade good sign told extense	12–12	12–13
Yao Ming	–	Season stress fracture ankle record-set	12–16	12–17
Google	–	People hire home work see	12–22	12–24
Elizabeth Edwards	–	Funeral say picket church issue	12–9	12–10
Prince Charles	–	Car attack protest contain tuition	12–09	12–11

Table 8
Examples of event categorization results. The event type labels are automatic assigned using the most frequent semantic class for entities in each event cluster

Event type	Event			
	Entity	Location	Date	Keywords
Goverment	–	South Korea	2010-12-20	drill, live-fire, artillery, heard, korea
	Senate	Russia	2010-12-20	warn, treaty, arm, start, senate
	Rupert Murdoch	–	2010-12-21	corporation, bskyb, say, bid, clear
	Dmitry Medvedev, President Obama	Russia	2010-12-23	treaty, army, laud, congratule, approval
Music	Larry King	–	2010-12-17	fan, show, year, meet, tv
	Lady Gaga	Germany	2010-12-03	song, steal, investigate, police, accuse
	McFly	–	2010-12-25	fan, world, troop, family, amaze
Sports	Cincinnati Bengals	–	2010-12-21	return, expect, re-sign, minist, die
	Matt Smith	North America	2010-12-25	special, treat, sign, front, cosy
	New York Knicks	–	2010-12-12	want, trade, team, sport, chance
	Boston Red Sox, Carl Crawford	–	2010-12-09	deal, sign, accord, report, multiple
Business	Google Chrome	–	2010-12-15	adblock, plus, say, fan, extense
	Carine Roitfeld	–	2010-12-17	leave, end, decide, threaten, record-set
	Google	–	2010-12-20	people, hire, work, home, gotta
	Google	–	2010-12-17	site, hack, highlight, pleas, may
Law	High Court	–	2010-12-10	verdict, gay, change
	Supreme Court	–	2010-12-07	law, case, hear
	Supreme Court	–	2010-12-06	hear, law, punish
	John David	Oklahoma	2010-12-16	execute, murder, punish
TV	Ryan Reynolds, Scarlett Johansson	LA	2010-12-24	divorce, take, break, credit, been
	Steve Landesberg	–	2010-12-21	die, cancer, star, onion, engag
	Candice Crawford	–	2010-12-17	engage, romo, tv, healthday, report

event categorization results generated by LECM-d are presented in Table 8. It can be observed from the results that our event categorization component does group similar events together. We evaluate the precision of event categorization on the correctly extracted events and also on the all extracted events. We found that when evaluated on the correctly extracted events, LECM and LECM-d give similar precision results of 43.87% and 42.57% respectively. However, when evaluated on the all extracted events, LECM-d achieves a precision of 38.3% on event categorization whereas LECM only gives a precision of 29.5%.

4.5. Errors analysis

To further investigate the performance of the proposed framework, we conduct an analysis on the extraction errors, which can be categorized into three types:

- Filtering errors (30%): Some non-event-related tweets have not been filtered properly by the filtering step. This constitutes 30% of the errors.

- Temporal information errors (10%): Although we have reduced the temporal resolution errors with the pre-processing step and LECM-d, there are still some errors incurred by wrongly recognised event dates.
- NER errors (10%): Some extraction errors are caused by NER errors. For example, "Red" might denote a color or the name of a person. It might be wrongly extracted as a named entity from tweets.
- Keyword errors (20%) Event-related keywords might be wrongly identified for some events. For example, for the event "Amy Winehouse died", words such as "fans" are wrongly identified as event-related keywords.
- Other errors (30%): The model clusters the tweets with the same named entity, location, date and keywords as describing the same event. However, some different events might have the same date, location, and even share similar keywords. For example, two events "Car bomb explodes in Oslo, Norway" and "gunman opens fire in youth camp in Norway" happened on the same day and in the same country. They might even share the same keywords such as "fire" in some tweets. The LECM model could wrongly extract the same event from two tweets actually mentioning two different events.

5. Conclusions and future work

In this paper, we have proposed an unsupervised framework for event exploration on Twitter. A pipeline process consists of filtering, extraction and categorization is introduced. All the steps here are fully unsupervised, which makes our proposed framework specifically plausible for analyzing events in the large-scale social stream data. A new method of combining heuristic rules and outputs generated by the modified LECM model has been proposed to infer event dates more accurately. The proposed framework has been evaluated on a large Twitter dataset consisting of 60 million tweets and has achieved a precision of 78.01%, comfortably outperforming a baseline by nearly 13.6%. It also outperforms the previous proposed LECM model by 9.75%. Moreover, events are also clustered into coherence groups with the automatically assigned event type label with an accuracy of 42.57%. Our current model handles tweets in different dates separately. It is possible to explore a dynamic version of our proposed Bayesian model which can take into account the date dependencies to improve the event extraction performance.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (61528302), the Natural Science Foundation of Jiangsu Province of China (BK20161430), the Innovate UK under the grant number 101779 and the Collaborative Innovation Center of Wireless Communications Technology.

References

[1] H. Abdelhaq, C. Sengstock and M. Gertz, Eventweet: Online localized event detection from twitter, *Proceedings of the VLDB Endowment* **6**(12) (2013), 1326–1329.

[2] J. Allan, *Topic Detection and Tracking: Event-based Information Organization*, Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] P. Anantharam et al., Extracting city traffic events from social streams, *ACM Transactions on Intelligent Systems and Technology* **6**(4) (2015), 1–43.

[4] F. Atefeh and W. Khreich, A survey of techniques for event detection in twitter, *Computational Intelligence* **31**(1) (2015), 132–164.

[5] H. Becker, M. Naaman and L. Gravano, Beyond trending topics: Real-world event identification on twitter, in: *Proc of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[6] E. Benson, A. Haghighi and R. Barzilay, Event discovery in social media feeds, in: *Proc of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 389–398.

[7] A.X. Chang and C.D. Manning, Sutime: A library for recognizing and normalizing time expressions, in: *Proc of the 8th International Conference on Language Resources and Evaluation*, 2012.

[8] H.-W. Chun et al., Building patterns for biomedical event extraction, in: *Proc of the Fifteenth International Conference on Genome Informatics*, 2004.

[9] K. Gimpel et al., Part-of-speech tagging for twitter: Annotation, features, and experiments, in: *Proc of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[10] T.L. Griffiths and M. Steyvers, Finding scientific topics, in: *Proc of the National Academy of Sciences*, Vol. 101 (Suppl. 1), 2004, pp. 5228–5235.

[11] R. Grishman, D. Westbrook and A. Meyers, NYU's english ACE 2005 system description, in: *Proc of ACE Evaluation Workshop*, 2005.

[12] M. Hall et al., The WEKA data mining software: An update, *SIGKDD Explorations* **11**(1) (2009).

[13] R. Lee and K. Sumiya, Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection, in: *Proc of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, 2010, pp. 1–10.

[14] J. Li et al., Major life event extraction from twitter based on congratulations/condolences speech acts, in: *Proc of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1997–2007.

[15] X. Liu et al., Exacting social events for tweets using a factor graph, in: *Proc of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012, pp. 1692–1698.

[16] M. Naughton et al., Event extraction from heterogeneous news sources, in: *Proc of the 2006 AAAI Workshop on Event Extractionand Synthesis*, 2006, pp. 1–6.

[17] S. Panem, M. Gupta and V. Varma, Structured information extraction from natural disaster events on twitter, in: *Proc of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*, 2014, pp. 1–8.

[18] S. Petrovic et al., Can twitter replace newswire for breaking news? in: *Proc of the 7th International AAAI Conference on Weblogs and Social Media*, 2013.

[19] J. Piskorski et al., Cluster-centric approach to news event extraction, in: *Proc of the International Conference on New Trends in Multimedia and Network Information Systems*, 2008, pp. 276–290.

[20] A.-M. Popescu et al., Extracting events and event descriptions from twitter, in: *Proc of the 20th International Conference Companion on World Wide Web*, 2011, pp. 105–106.

[21] A. Ritter et al., Named entity recognition in tweets: an experimental study, in: *Proc of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1524–1534.

[22] A. Ritter et al., Open domain event extraction from twitter, in: *Proc of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1104–1112.

[23] T. Sakaki, M. Okazaki and Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: *Proc of the 19th International Conference on World Wide Web*, 2010, pp. 851–860.

[24] J. Sankaranarayanan et al., Twitterstand: News in tweets, in: *Proc of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 42–51.

[25] B. Sriram et al., Short text classification in twitter to improve information filtering, in: *Proc of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010, pp. 841–842.

[26] H. Tanev, J. Piskorski and M. Atkinson, Real-time news event extraction for global crisis monitoring, in: *Proc of the 13th International Conference on Applications of Natural Language to Information Systems*, 2008, pp. 207–218.

[27] L. Zhao et al., Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling, *PLoS ONE* **9**(10) (2014).

[28] D. Zhou, L. Chen and Y. He, An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization, in: *Proc of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 700–705.