Jointly Event Extraction and Visualization on Twitter via Probabilistic Modelling

Deyu Zhou[†‡] **Tianmeng Gao**[†] **Yulan He**[§]
[†] School of Computer Science and Engineering, Key Laboratory of Computer Network
and Information Integration, Ministry of Education, Southeast University, China
[‡] State Key Laboratory for Novel Software Technology, Nanjing University, China
[§] School of Engineering and Applied Science, Aston University, UK
d.zhou@seu.edu.cn, gaotianmeng@seu.edu.cn, y.he@cantab.net

Abstract

Event extraction from texts aims to detect structured information such as what has happened, to whom, where and when. Event extraction and visualization are typically considered as two different tasks. In this paper, we propose a novel approach based on probabilistic modelling to jointly extract and visualize events from tweets where both tasks benefit from each other. We model each event as a joint distribution over named entities, a date, a location and event-related keywords. Moreover, both tweets and event instances are associated with coordinates in the visualization space. The manifold assumption that the intrinsic geometry of tweets is a low-rank, non-linear manifold within the high-dimensional space is incorporated into the learning framework using a regularization. Experimental results show that the proposed approach can effectively deal with both event extraction and visualization and performs remarkably better than both the state-of-the-art event extraction method and a pipeline approach for event extraction and visualization.

1 Introduction

Event extraction, one of the important and challenging tasks in information extraction, aims to detect structured information such as what has happened, to whom, where and when. The outputs of event extraction could be beneficial for downstream applications such as summarization and personalized news systems. Data visualization, an important exploratory data analysis task, provides a simple way to reveal the relationships among data (Nakaji and Yanai, 2012).

Although event extraction and visualization are two different tasks and typically studied separately in the literature, these two tasks are highly related. Documents which are close to each other in the low-dimensional visualization space are likely to describe the same event. Events in nearby locations in the visualization space are likely to share similar event elements. Therefore, jointly learning the two tasks could potentially bring benefits to each other. However, it is not straightforward to learn event extraction and visualization jointly since event extraction usually relies on semantic parsing results (McClosky et al., 2011) while visualization is accomplished by dimensionality reduction (Iwata et al., 2007; López-Rubio et al., 2002).

In this paper, we propose a novel probabilistic model, called Latent Event Extraction & Visualization (LEEV) model, for joint event extraction and visualization on Twitter. It is partly inspired by the Latent Event Model (LEM) (Zhou et al., 2015) where each tweet is assigned to one event instance and each event is modeled as a joint distribution over named entities, a date/time, a location and the event-related keywords. Going beyond LEM, we assume that each event is not only modeled as the joint distribution over event elements as in (Zhou et al., 2015), but also associate with coordinates in the visualization space. The Euclidean distance between a tweet and each events determines which event the tweet should be assigned to. Furthermore, the manifold assumption that the intrinsic geometry of tweets is a low-rank, non-linear manifold within the high-dimensional space, is incorporated in the learning framework using a regularization. Experimental results show that the proposed approach can effectively deal with both event extraction and visualization tasks and performs remarkably better than both the state-of-the-art event extraction method

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 269–278,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

and a pipeline approach for event extraction and visualization.

2 Related Work

Our proposed work is related to two lines of research, event extraction and joint topic modeling and visualization.

2.1 Event Extraction

Research on event extraction of tweets can be categorized into domain-specific and open domain approaches. Domain-specific approaches usually have target events in mind and aim to extract events from a particular location or for emergency response during natural disasters. Anantharam et al. (2015) focused on extracting city events by solving 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. TSum4act (Nguyen et al., 2015) was designed for emergency response during disasters and was evaluated on a dataset containing 230,535 tweets.

Most of open domain approaches focused on extracting a summary of events discussed in social media. For example Benson et al. (2011) proposed a structured graphical model which simultaneously analyzed individual messages, clustered, and induced a canonical value for each event. Capdevila et al. (2015) proposed a model named Tweet-SCAN based on the hierarchical Dirichlet process to detect events from geo-located tweets. To extract more information, a system called SEEFT (Wang et al., 2015) used links in tweets and combined tweets and linked articles to identify events. Zhou et al. (2014; 2015) proposed an unsupervised Bayesian model called latent event model (LEM) for event extraction from Twitter by assuming that each tweet message is assigned to one event instance and each event is modeled as a joint distribution over named entities, a date/time, a location and the event-related keywords. Our proposed method is partly inspired by (Zhou et al., 2015). However, different from previous methods, our approach not only extracts the structured representation of events, but also learns the coordinates of events and tweets simultaneously.

2.2 Joint Topic Modeling and Visualization

Since our proposed approach can be considered as a variant of topic model, we also review the related work of joint topic modeling and visualization here.

Traditionally, topic modeling and visualization are considered as two disjoint tasks and can be combined for pipeline processing. For example, probabilistic latent semantic analysis (Hofmann, 1999) can be first performed followed by parametric embedding (Iwata et al., 2007). Another pipeline approach (Millar et al., 2009) is based on latent Dirichlet allocation followed by self-organizing maps (López-Rubio et al., 2002).

Jointly modeling topics and visualization is a new problem explored in very few works. The state-of-the-art is a joint approach proposed in (Iwata et al., 2008). In this model, both documents and topics are assumed to have latent coordinates in a visualization space. The topic proportions of a document are determined by the distances between the document and the topics in the visualization space, and each word is drawn from one of the topics according to the document's topic proportions. A visualization was obtained by fitting the model to a given set of documents using the EM algorithm. Following the same line, by considering the local consistency in terms of the intrinsic geometric structure of the document manifold, an unsupervised probabilistic model, called SEMAFORE, was proposed in (Le and Lauw, 2014a) by preserving the manifold in the lower dimensional space. In (Le and Lauw, 2014b), a semantic visualization model is learned by associating each document a coordinate in the visualization space, a multinomial distribution in the topic space, and a directional vector in a high-dimensional unit hypersphere in the word space.

Our work is partly inspired by (Le and Lauw, 2014a). However, our proposed approach differs from (Le and Lauw, 2014a) in that events, instead of topics, are modelled as the joint distribution over event elements. Both tweets and events are associate with coordinates in the visualization space.

3 Methodology

We follow the same pre-processing steps described in (Zhou et al., 2015) to filter out non-event-related tweets and extract dates, locations, and named entities by temporal resolution, part-of-speech (POS) tagging and named entity recognition. The pre-processed tweets are then fed into our proposed model for event extraction and visu-

Table 1: Definition of Notations.

Notation	Definition
e	event index, $e \in \{1..E\}$
$W = \{w_m\}$	tweets, $m \in \{1..M\}$
$Z = \{z_m\}$	event labels for tweets
N_{my}	number of named entities in w_m
N_{md}	number of dates in w_m
N_{ml}	number of locations in w_m
N_{mk}	number of keywords in w_m
θ_{ey}	probability of named entity y in event e
φ_{ed}	probability of date d in event e
ψ_{el}	probability of location l in event e
ω_{ek}	probability of keyword k in event e
$\beta, \gamma, \eta, \lambda$	Dirichlet hyperparameters
χ, δ	Normal hyperparameters
G	dimension of visualization space

alization. We describe our model in more details below.

3.1 Latent Event Extraction & Visualization (LEEV) Model

We propose an unsupervised latent variable model called the Latent Event Extraction & Visualization (LEEV) model which simultaneously extracts events from tweets and generates a visualization of the events. Table 1 lists notations used in this paper.

In LEEV, each tweet message w_m, $m \in \{1...M\}$ is associated with a latent coordinate x_m in the visualization space. Each event $e \in \{1...E\}$ is also associated with a coordinate ϕ_e. Assuming that each tweet message $w_m, m \in \{1...M\}$ is assigned to one event instance $z_m = e$ and e is modeled as a joint distribution over named entities y, the date d when e happened, the location l and the event-related keywords k, the generative process of the model is described as follows:

- For each event $e \in \{1..E\}$, draw multinomial distributions $\theta_e \sim$ Dirichlet(β), $\varphi_e \sim$ Dirichlet(γ), $\psi_e \sim$ Dirichlet(η), $\omega_e \sim$ Dirichlet(λ), draw event coordinate $\phi_e \sim$ Normal$(0, \chi^{-1}I)$;
- For each tweet $w_m, m \in \{1..M\}$
  - * Choose tweet coordinate: $x_m \sim$ Normal$(0, \delta^{-1}I)$;
  - * Choose an event $z_m = e \sim$ Multinomial$(\{P(e|x_m, \Phi)_{e=1}^{E}\})$;
  - * For each named entity in the tweet w_m, choose a named entity $y \sim$ Multinomial(θ_e);
  - * For each date in the tweet w_m, choose a date $d \sim$ Multinomial(φ_e);
  - * For each location in the tweet w_m, choose a location $l \sim$ Multinomial(ψ_e);
  - * For other words in the tweet w_m, choose a word $k \sim$ Multinomial(ω_e).

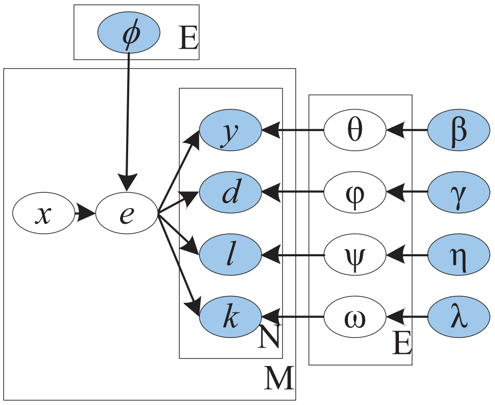


Figure 1: Latent Event Extraction & Visualization (LEEV) Model.

Here, $\beta, \gamma, \eta, \lambda, \chi, \delta$ are priors, I is an identity matrix, and $P(e|x_m, \Phi)$ is the probability of the tweet w_m with coordinate x_m belonging to the event e. It is defined as,

$$P(e|x_m, \Phi) = \frac{\exp(-\frac{1}{2} \parallel x_m - \phi_e \parallel^2)}{\sum_{e'=1}^{E} \exp(-\frac{1}{2} \parallel x_m - \phi_{e'} \parallel^2)}. \quad (1)$$

It is calculated as the normalized Euclidean distance between a tweet w_m and an event e. Using this equation, when the Euclidean distance between a tweet w_m and and an event e is small, the probability that tweet w_m belongs to event e becomes large. The graphical model of LEEV is shown in Figure 1.

The parameters to be learned are $\Theta = \{\theta_e, \varphi_e, \psi_e, \omega_e\}_{e=1}^{E}$, tweets' coordinates $\mathcal{X} = \{x_m\}_{m=1}^{M}$ and events' coordinates $\Phi = \{\phi_e\}_{e=1}^{E}$, which are collectively denoted as $\mathcal{B} = \langle \Theta, \mathcal{X}, \Phi \rangle$. Let

$$\begin{aligned} H(w_m, e) \quad = \quad & \prod_{n=1}^{N_{my}} P(y_n|\theta_e) \prod_{n=1}^{N_{md}} P(d_n|\varphi_e) \\ & \prod_{n=1}^{N_{ml}} P(l_n|\psi_e) \prod_{n=1}^{N_{mk}} P(k_n|\omega_e). \end{aligned}$$

The log likelihood of $\mathcal{B}$ given tweets W is,

$$\begin{aligned}\mathcal{L}(\mathcal{B}|W) &= \sum_{m=1}^{M} \log\Big\{\sum_{e=1}^{E} P(e|x_m,\Phi)\times H(w_m,e)\Big\}\\ &+ \sum_{m=1}^{M}\log(P(x_m)) + \sum_{e=1}^{E}\log(P(\phi_e)) \qquad (2)\\ &+ \sum_{e=1}^{E}\{\log(P(\theta_e)*P(\varphi_e)*P(\psi_e)*P(\omega_e))\}.\end{aligned}$$

For the events' coordinate ϕ_e and tweets' coordinate x_m, we use a Gaussian prior with a zero mean and a spherical covariance:

$$\begin{aligned} p(\phi_e) &= (\frac{\chi}{2\pi})^{\frac{G}{2}} \exp(-\frac{\chi}{2} \parallel \phi_e \parallel^2)\\ p(x_m) &= (\frac{\delta}{2\pi})^{\frac{G}{2}} \exp(-\frac{\delta}{2} \parallel x_m \parallel^2).\end{aligned}$$

3.2 LEEV with Manifold Regularization

Recent studies suggest that the intrinsic geometry of textual data is a low-rank, non-linear manifold lying in the high dimensional space (Cai et al., 2008; Zhang et al., 2005). We therefore assume that when two tweets w_i and w_j are close in the intrinsic geometry of the manifold Υ, their low-rank representations should be close as well. To capture this assumption, we consider Laplacian Eigenmaps (LE) (Belkin and Niyogi, 2003) which has been commonly used in manifold learning algorithms (Le and Lauw, 2014a). It constructs a k-nearest neighbors graph to represent data residing on a low-dimensional manifold embedded in a higher-dimensional space. In this paper, we use LE to incorporate neighborhood information of tweets. We construct a manifold graph with edges connecting two data points w_i and w_j. Set the edge weight $\upsilon_{ij} = 1$ if w_j is one of the k-nearest neighbors of w_i; Otherwise $\upsilon_{ij} = 0$. That makes LEEV an special case when $\xi = 0$. We represent each tweet as a word-count vector, i.e., each element of a vector is weighted by its corresponding term frequency, and use cosine similarity metric to measure the distance between tweets when constructing the manifold graph. We also tried vectors with the TFIDF weighting strategy to represent tweets and found word-count vectors give better results.

We apply a regularization framework to incorporate a manifold structure into a learning model. The new regularized log-likelihood function L is

$$L(\mathcal{B}|W,\Upsilon) = L(\mathcal{B}|W) - \frac{\xi}{2}\mathcal{R}(\mathcal{B}|\Upsilon), \qquad (3)$$

where ξ is the regularization parameter. The second component $\mathcal{R}$ is a regularization function, which consists of two parts:

$$\mathcal{R}(\mathcal{B}|\Upsilon) = \mathcal{R}_{+}(\mathcal{B}|\Upsilon) + \mathcal{R}_{-}(\mathcal{B}|\Upsilon), \qquad (4)$$

$$\mathcal{R}_{+}(\mathcal{B}|\Upsilon) = \sum_{i,j=1;i\neq j}^{M} \upsilon_{ij}\cdot\mathcal{F}(w_i,w_j), \qquad (5)$$

$$\mathcal{R}_{-}(\mathcal{B}|\Upsilon) = \sum_{i,j=1;i\neq j}^{M} \frac{1-\upsilon_{ij}}{\mathcal{F}(w_i,w_j)+1}, \qquad (6)$$

where $\mathcal{F}$ is a distance function that operates on the low rank space. We define $\mathcal{F}$ as the squared Euclidean distance of coordinates in the visualization space. $\mathcal{F}(w_i,w_j)$is computed as follows:

$$\mathcal{F}(w_i,w_j) = \parallel x_i - x_j \parallel^2 . \qquad (7)$$

Minimizing $\mathcal{R}_{+}$ leads to minimizing the distance between neighbors and minimizing $\mathcal{R}_{-}$ leads to maximizing the distance between non-neighbors. By enforcing manifold learning, we capture the spirit of keeping neighbors close and keeping none-neighbors apart.

3.3 Parameter Estimation

As in Equation 2, the presence of the sum over e prevents the logarithm form directly acting on the joint distribution. Assuming that the corresponding latent event z_m of each tweet w_m is known, $\{W,Z\}$ is called the complete data. Maximizing the log likelihood of the complete data, $\log P(W,Z|\mathcal{B})$, can be easily done. However, in practice we don't observe the latent variables Z and only have the incomplete data W. Therefore, the expectation maximization (EM) algorithm is employed to handle the incomplete data. EM involves an efficient iterative procedure to compute the Maximum Likelihood estimation of probabilistic models with unobserved latent variables involved.

The class posterior probability of the m^{th} tweet under the current parameter values $\hat{\mathcal{B}}$, $P(z_m = e|m,\hat{\mathcal{B}})$, is given as follows:

$$\begin{aligned}&P(z_m = e|m,\hat{\mathcal{B}}) =\\ &\frac{P(z_m = e|\widehat{x_m},\hat{\Phi},\hat{\mathcal{B}})\times H(w_m,e)}{\sum_{e'=1}^{E} P(z_m = e'|\widehat{x_m},\hat{\Phi},\hat{\mathcal{B}})\times H(w_m,e')}, \qquad (8)\end{aligned}$$

which corresponds to the E-step in EM algorithm.

In M-step, model parameters $\mathcal{B}$ are updated by maximizing the regularized conditional expectation of the complete data log likelihood with priors defined as follows:

$$\begin{aligned}
&\mathcal{Q}(\mathcal{B}|\hat{\mathcal{B}}) = \sum_{m=1}^{M}\sum_{e=1}^{E}\{P(z_m = e|m, \hat{\mathcal{B}}) \\
&\times \log[P(e|x_m, \Phi) \times H(w_m, e)]\} \\
&+ \sum_{m=1}^{M}\log(P(x_m)) + \sum_{e=1}^{E}\log(P(\phi_e)) \\
&+ \sum_{e=1}^{E}\{\log(P(\theta_e) * P(\varphi_e) * P(\psi_e) * P(\omega_e))\} \\
&- \frac{\xi}{2}\mathcal{R}(\mathcal{B}|\Upsilon),
\end{aligned}$$

where $P(z_m = e|m, \hat{\mathcal{B}})$ is calculated in E-step.

By maximizing $\mathcal{Q}(\mathcal{B}|\hat{\mathcal{B}})$ w.r.t $\theta_{ey}, \varphi_{ed}, \psi_{el}, \omega_{ek}$, the next estimates are given as follows,

$$\begin{aligned}
\theta_{ey} &= \frac{\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{my}} I(y_{mn} = y)P(z_m = e|m, \hat{\mathcal{B}}) + \beta}{\sum\limits_{y=1}^{Y}\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{my}} I(y_{mn} = y)P(z_m = e|m, \hat{\mathcal{B}}) + Y\beta}, \\
\varphi_{ed} &= \frac{\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{md}} I(d_{mn} = d)P(z_m = e|m, \hat{\mathcal{B}}) + \gamma}{\sum\limits_{d=1}^{D}\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{md}} I(d_{mn} = d)P(z_m = e|m, \hat{\mathcal{B}}) + D\gamma}, \\
\psi_{el} &= \frac{\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{ml}} I(l_{mn} = l)P(z_m = e|m, \hat{\mathcal{B}}) + \eta}{\sum\limits_{l=1}^{L}\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{ml}} I(l_{mn} = l)P(z_m = e|m, \hat{\mathcal{B}}) + L\eta}, \\
\omega_{ek} &= \frac{\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{mk}} I(k_{mn} = k)P(z_m = e|m, \hat{\mathcal{B}}) + \lambda}{\sum\limits_{k=1}^{K}\sum\limits_{m=1}^{M}\sum\limits_{n=1}^{N_{mk}} I(k_{mn} = k)P(z_m = e|m, \hat{\mathcal{B}}) + K\eta},
\end{aligned}$$

where Y, D, L, K are the total numbers of distinct named entities, dates, locations, and words appeared in the whole Twitter corpus, respectively.

ϕ_e and x_m cannot be solved in a closed form, and are estimated by maximizing $\mathcal{Q}(\mathcal{B}|\hat{\mathcal{B}})$ using quasi-Newton method. The gradients of $\mathcal{Q}(\mathcal{B}|\hat{\mathcal{B}})$ w.r.t ϕ_e and x_m are as follows:

$$\begin{aligned}
\frac{\partial \mathcal{Q}}{\partial \phi_e} &= \sum_{m=1}^{M} p(e|m, \hat{\mathcal{B}})(p(e|x_m, \Phi) - 1)(\phi_e - x_m) \\
&\quad - \chi\phi_e, \\
\frac{\partial \mathcal{Q}}{\partial x_m} &= \sum_{e=1}^{E} p(e|m, \hat{\mathcal{B}})(p(e|x_m, \Phi) - 1)(x_m - \phi_e) \\
&\quad - \delta x_m - \frac{\xi}{2}\frac{\partial \mathcal{R}(\mathcal{B}|\Upsilon)}{\partial x_m},
\end{aligned}$$

where the gradient of $\mathcal{R}(\mathcal{B}|\Upsilon)$ w.r.t. x_m is computed as follows:

$$\begin{aligned}
\frac{\partial \mathcal{R}(\mathcal{B}|\Upsilon)}{\partial x_m} &= \sum_{j=1, j\neq m}^{M} 2v_{mj}(x_m - x_j) \\
&- \sum_{j=1, j\neq m}^{M} 2(x_m - x_j)\frac{1 - v_{mj}}{(\mathcal{F}(x_m, x_j) + 1)^2}.
\end{aligned}$$

We set the parameter $\chi = 0.00005$, $\delta = 0.05$, $\beta = \gamma = \eta = \lambda = 0.1$ and run EM algorithm for 50 iterations. Finally we select an entity y, a date d, a location l and two keywords k with the highest probabilities to form a tuple $\langle y, d, l, k\rangle$ to represent each potential event.

3.4 Post-processing

In order to filter out spurious events, we calculate the correlation coefficient of each event element. Remove the event element if its correlation coefficient is less than a threshold C_e and remove the event if the sum of the correlation coefficients of all its four event elements is less than C_t.

For an event element A, its correlation coefficient is calculated below:

$$C_A = \log\frac{\sum_{\substack{B\in\Omega \\ B\neq A}} \#(A, B)}{\#(A)}, \qquad (0)$$

where Ω is the set of the four event elements $\langle y, d, l, k\rangle$ and $\#(x)$ indicates the number of times x appeared in the whole corpus. We empirically set C_e to 0.4 and C_t to 4.

4 Experiments

In this section, we firstly describe the datasets used in our experiments and then present the experimental results.

4.1 Setup

We choose two datasets for model evaluation. The first one is the First Story Detection (FSD) dataset (Petrovic et al., 2013) (Dataset I) which contains 2,499 tweets published between 7th July and 12th September 2011. These tweets have been manually annotated with 27 events, covering a wide range of topics from accidents to science discoveries and from disasters to celebrity news. We filter out events mentioned in less than 15 tweets since events mentioned in very few tweets are less likely to be significant. The final dataset contains 2,453 tweets annotated with 20 events. This dataset has been previously used for evaluating event extraction models and the state-of-the-art results have been achieved using LEM (Zhou et al., 2015). We also create another dataset, called Dataset II, by manually annotating 1,000 tweets published in December 2010. A total of 20 events are annotated.

We compare our model with LEM (Zhou et al., 2015), which also extracts events as 4-tuples $\langle$

y,d,l,k ⟩. The main difference between LEM and our model is that LEM directly estimates the event distribution from the sampled latent event labels, while we derive the distribution from coordinates of tweets and events x_m, ϕ_e. We re-implemented the system described in (Zhou et al., 2015) and used the same evaluation metrics such as precision, recall and F-measure. Precision is defined as the proportion of the correctly identified events out of the system returned events. Recall is defined as the proportion of correctly identified true events. For calculating the precision of the 4-tuple $\langle y, d, l, k \rangle$, we use following criteria:

- Do the entity y, location l, date d and keyword k that we have extracted refer to the same event?

- If the extracted representation contains keywords, are they informative enough to tell us what happened?

As mentioned in Section 2, PE (Iwata et al., 2007) is a nonlinear visualization method which takes a set of class posterior vectors as input and embeds samples in a low-dimensional Euclidean space. By minimizing the sum of Kullback-Leibler divergences, PE tries to preserve the posterior structure in the embedding space. In order to evaluate the visualization results, we compare our proposed method with a pipeline approach, event extraction using LEM (Zhou et al., 2015) followed by event visualization using PE (Iwata et al., 2007), named as LEM+PE.

4.2 Event Extraction Results

Table 2 shows the event extraction results on the two datasets. LEEV+R is LEEV with manifold regularization incorporated, in which the model parameters are estimated by the EM algorithm described in Section 3.3. For LEEV and LEEV+R, the number of events, E, is set to 50 for both datasets. For LEEV+R, the number of neighborhood size k is set to 10 and the regularization parameter ξ is set to 1. For LEM, E is set to 25 for both datasets following the suggestion in (Zhou et al., 2015).

We ran our experiments on a server equipped with 3.40 GHz Intel Corel i7 CPU and 8 GB memory. The average running time of LEEV is 2328.1 seconds on Dataset I and 940.7 seconds on Dataset II for one iteration. The average running time of LEEV+R is 2612.7 seconds on Dataset I and 1296.4 seconds on Dataset II for one iteration.

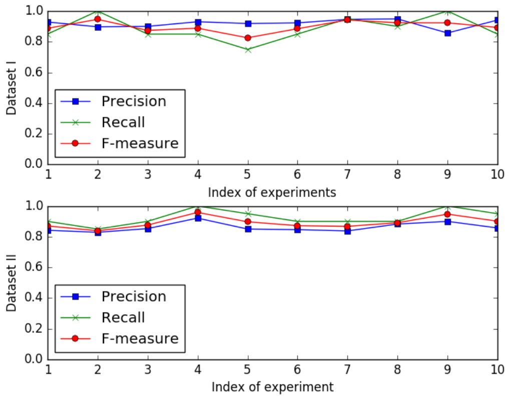


Figure 2: Experimental results of LEEV+R in 10 different runs.

Table 2: Comparison of the event extraction results on the two datasets.

Method	Prec. (%)	Rec. (%)	F-measure (%)
Dataset I			
LEM	84.00	76.19	80.35
LEEV	**92.10**	80.00	85.62
LEEV+R	91.91	**88.50**	**89.88**
Dataset II			
LEM	80.00	90.00	84.70
LEEV	83.33	**95.00**	88.78
LEEV+R	**86.18**	92.50	**89.19**

It can be observed that both LEEV and LEEV+R outperforms the state-of-the-art results achieved by LEM on Dataset I. In particular, LEEV improves upon LEM by over 5% in F-measure and with regularization, LEEV-R further improves upon LEEV by over 4%. A similar trend is observed on Dataset II where both LEEV and LEEV+R outperforms LEM and the best performance is given by LEEV+R. This shows the effectiveness of using regularization in LEEV. We will further demonstrate its importance in visualization results. Overall, we see superior performance of LEEV+R over the other two models, with the F-measure of over 89% being achieved on both datasets.

As described in Section 3.1, the coordinates of tweets and events are randomly initialized. Therefore, we would like to see whether the performance of event extraction is influenced heavily by random initialization. We repeat the experiments

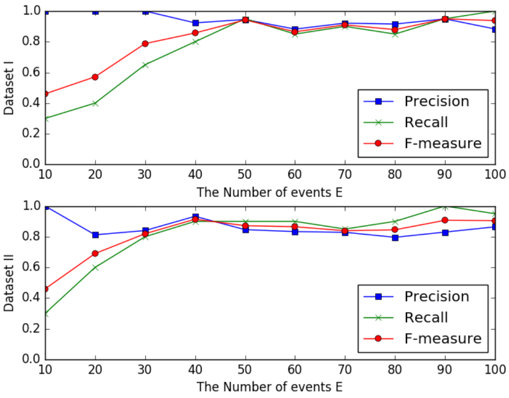


Figure 3: The performance of LEEV+R with different number of events E.

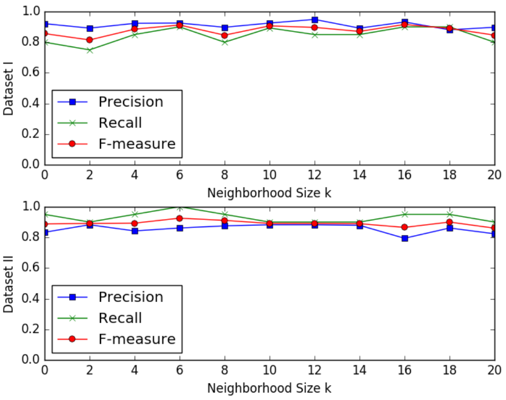


Figure 4: The performance of LEEV+R with different neighborhood size k.

on the two datasets for 10 times using LEEV+R. The experimental results are shown in Figure 2. It can be observed that the performance of LEEV+R is quite stable on both datasets. The standard deviation of F-measure on both Dataset I and II is 0.036, which shows that random initialization does not have significant impact on the final performance of the model.

4.3 Impact of Number of Events E

We need to pre-set the number of events E in the proposed approach. Figure 3 shows the performance of event extraction based on LEEV+R versus different values of E on the two datasets. It can be observed that the performance of the proposed approach improves with the increased value of E and when E goes beyond 50, we notice a more balanced precision/recall values and a relatively stable F-measure. This shows that the proposed approach is less sensitive to the number of events E so long as E is set to a relatively larger value.

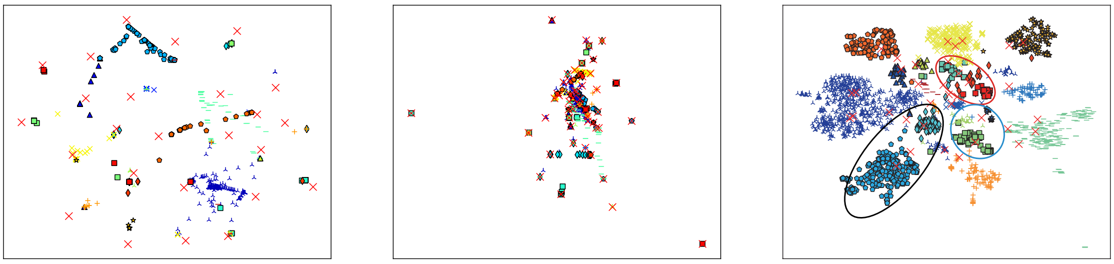
4.4 Impact of Neighborhood Size

As described in Section 3.2, the neighborhood information of tweets is incorporated into the learning framework. A manifold graph with edges connecting two tweets (or data points) w_i and w_j is constructed by setting the edge weight $v_{ij} = 1$ if w_j is among the k-nearest neighbors of w_i and $v_{ij} = 0$ otherwise. Therefore, it is crucial to see whether the performance of LEEV+R heavily depends on the setting of k. Figure 4 shows the performance of our proposed approach with different neighborhood size k. It can be observed that the performance of LEEV+R is quite stable and independent of the k value.

4.5 Visualization Results

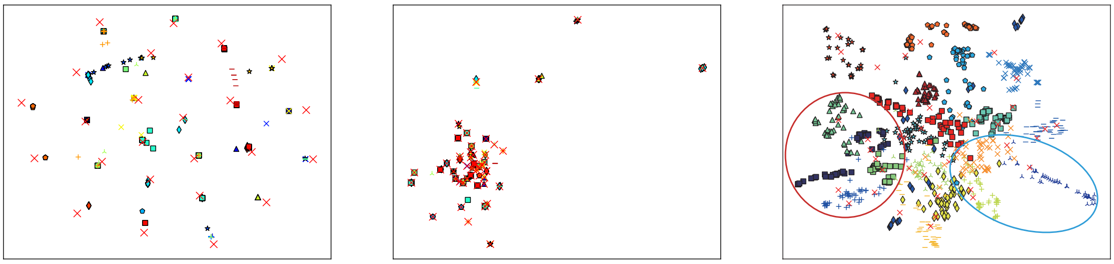
We show the visualization results produced by different approaches on the two datasets in Figure 5 and 6 respectively. We compare LEEV and LEEV+R with the pipeline approach LEM+PE. In the figures, each point represents a tweet and different shapes and colors represent the different events they are associated with. Each red cross represents an extracted event with coordinate ϕ_z.

For Dataset I, it can be observed from Figure 5(a) that the visualization result generated by LEM+PE is not informative. Tweets from different events are mixed together and events are evenly distributed across the whole visualization space. Thus, this visualization does not provide any sensible information about the relationships between tweets and events. The result generated by LEEV without manifold Regularization unit $\mathcal{R}$ seems better than that from LEM+PE, as shown in Figure 5(b). However, a large amount of tweets crowded together at the center, which makes it difficult to reveal the relations between tweets and events. The best visualization result is given by LEEV+R as shown in Figure 5(c) that different events are well separated and related events are located nearby. For example, the three events enclosed by a red circle represent "people died in terrorist attacks in Delhi, Oslo and Norway" respectively, while three events in the blue circle represent "riots in Ealing, Totteham and Croydon", respectively. And two events in the black circle represent "American credit rating" and "House



(a) LEM+PE (b) LEEV (c) LEEV+R

Figure 5: Visualization results on Dataset I.



(a) LEM+PE (b) LEEV (c) LEEV+R

Figure 6: Visualization results on Dataset II.

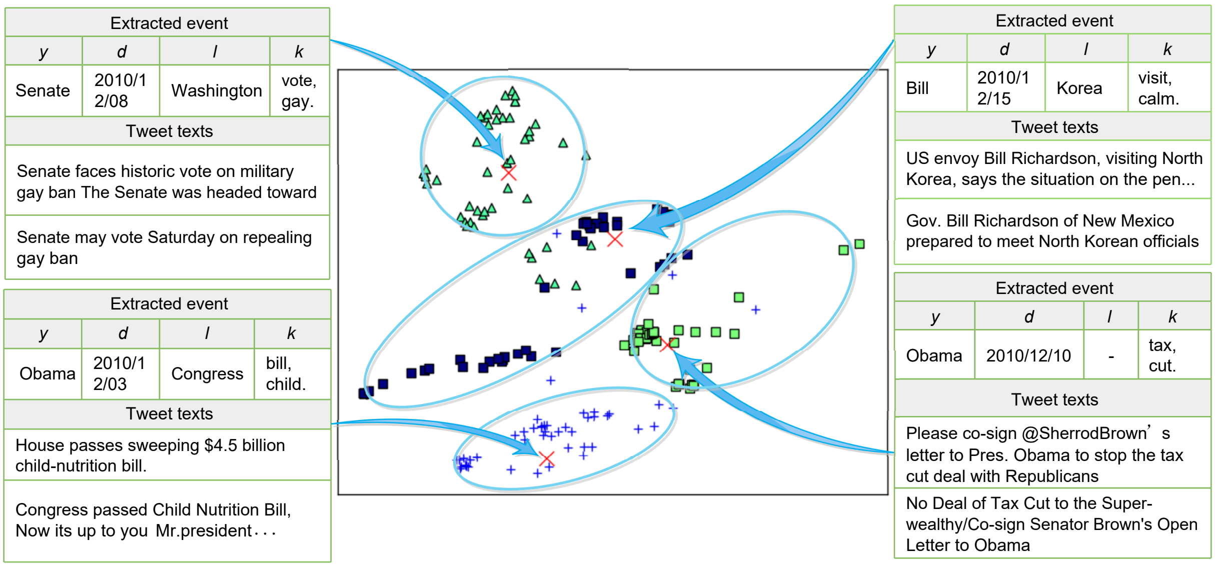
debt bill”, respectively. It shows that LEEV+R with manifold learning incorporated significantly improved upon LEEV without regularization and gives better visualization results. The relationships of events are directly reflected in the distances between their coordinates in the visualization space.

Similar visualization results have been obtained on Dataset II. Figure 6(a) and 6(b) failed to convey the semantic relations between different events. LEEV+R in Figure 6(c) is good at separating tweets from different events. The events in the red circle are the government activities of the United States. The events in the blue circle are categorized as traffic accidents. They are “ Transport chaos caused by heavy snow ”, “Train to Paris crushed ” and “Demonstrators attacked car carrying Prince Charles ”. Compared to LEM+PE and LEEV, LEEV+R gives much more informative visualization results.

To further analyze the visualization results in more detail, the 4 representative events and their corresponding tweets in the red circle of Figure 6(c) are visualized in Figure 7. These four events are “Senate vote on repealing gay ban”, “US state governor plan to visit North Korea”, “Send letter to President Obama to stop tax cut deal” and “Congress passed the Child Nutrition Bill”. Their corresponding tweets are denoted as green ’△’, blue ’□’, green ’□’ and blue ’+’ individually in Figure 7. It can be observed that these four events are all about government activities of the United States, and they are located close to each other in the low-dimensional visualization space. Moreover, the tweets describing the same event are located close to each other and center around their corresponding events, while the tweets describing different events are far away from each other.

5 Conclusions

In this paper, we have proposed an unsupervised Bayesian model, called Latent Event Extraction & Visualization (LEEV) model, to extract the structured representations of events from social media and simultaneously visualize them in a two-dimensional Euclidean space. The proposed approach has been evaluated on two datasets. Experimental results show that the proposed approach outperforms the previously reported best result on



Extracted event

y	d	l	k
Senate	2010/12/08	Washington	vote, gay.

Tweet texts

Senate faces historic vote on military gay ban The Senate was headed toward

Senate may vote Saturday on repealing gay ban

Extracted event

y	d	l	k
Obama	2010/12/03	Congress	bill, child.

Tweet texts

House passes sweeping $4.5 billion child-nutrition bill.

Congress passed Child Nutrition Bill, Now its up to you Mr.president…

Extracted event

y	d	l	k
Bill	2010/12/15	Korea	visit, calm.

Tweet texts

US envoy Bill Richardson, visiting North Korea, says the situation on the pen...

Gov. Bill Richardson of New Mexico prepared to meet North Korean officials

Extracted event

y	d	l	k
Obama	2010/12/10	-	tax, cut.

Tweet texts

Please co-sign @SherrodBrown’s letter to Pres. Obama to stop the tax cut deal with Republicans

No Deal of Tax Cut to the Super-wealthy/Co-sign Senator Brown's Open Letter to Obama

Figure 7: Four representative events and their corresponding tweets in the red circle of Figure 6(c).

Dataset I by nearly 10% in F-measure. Visualization results show that the proposed approach with manifold regularization can significantly improve the quality of event visualization. These results show that by jointly learning event extraction and visualization, our proposed approach is able to give better results on both tasks. In future work, we will investigate scalable and parallel model learning to explore the performance of our model for large-scale real-time event extraction and visualization.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was funded by the National Natural Science Foundation of China (61528302), the Innovate UK under the grant number 101779 and the Collaborative Innovation Center of Wireless Communications Technology.

References

Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, and Amit Sheth. 2015. Extracting city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology*, 6(4):e110206.

Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June.

Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 389–398, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. 2008. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM ’08, pages 911–920, New York, NY, USA. ACM.

Joan Capdevila, Jess Cerquides, Jordi Nin, and Jordi Torres. 2015. Tweet-scan: An event discovery technique for geo-located tweets. In *Artificial Intelligence Research and Development: Proceedings of the 18th International Conference of the Catalan Association for Artificial Intelligence*, volume 277, pages 110–119.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’99, pages 50–57, New York, NY, USA. ACM.

Tomoharu Iwata, Kazumi Saito, Naonori Ueda, Sean Stromsten, Thomas L. Griffiths, and Joshua B. Tenenbaum. 2007. Parametric embedding for class visualization. *Neural Computation*, 19(9):2536–56.

Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. 2008. Probabilistic latent semantic visualization: Topic model for visualizing documents. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’08, pages 363–371, New York, NY, USA. ACM.

Tuan M. V. Le and Hady W. Lauw. 2014a. Manifold learning for jointly modeling topic and visualization. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1960–1967. AAAI Press.

Tuan M.V. Le and Hady W. Lauw. 2014b. Semantic visualization for spherical representation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1007–1016, New York, NY, USA. ACM.

Ezequiel López-Rubio, José Muñoz-Pérez, and José Antonio Gómez-Ruiz. 2002. Self-organizing dynamic graphs. *Neural Processing Letters*, 16(2):93–109(17).

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1626–1635, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeremy Millar, Gilbert Peterson, and Michael Mendenhall. 2009. Document clustering and visualization with latent dirichlet allocation and self-organizing maps. In *Proceedings of Florida Artificial Intelligence Research Society Conference*.

Yusuke Nakaji and Keiji Yanai. 2012. Visualization of real-world events with geotagged tweet photos. In *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops*, ICMEW '12, pages 272–277, Washington, DC, USA. IEEE Computer Society.

Minh-Tien Nguyen, Asanobu Kitamoto, and Tri-Thanh Nguyen. 2015. Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *Advances in Knowledge Discovery and Data Mining*, pages 64–75. Springer.

Saša Petrovic, Miles Osborne, Richard McCreadie, Craig Macdonald, Iadh Ounis, and Luke Shrimpton. 2013. Can twitter replace newswire for breaking news? In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*.

Yu Wang, David Fink, and Eugene Agichtein. 2015. Seeft: Planned social event discovery and attribute extraction by fusing twitter and web content. In *Proceedings of the Ninth International AAAI Conference on Web and Social Media*, pages 483–492.

Dell Zhang, Xi Chen, and Wee Sun Lee. 2005. Text classification with kernels on the multinomial manifold. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 266–273, New York, NY, USA. ACM.

Deyu Zhou, Liangyu Chen, and Yulan He. 2014. A simple bayesian modelling approach to event extraction from twitter. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 700–705. ACL.

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2468–2474. AAAI Press.