# Event extraction from Twitter using Non-Parametric Bayesian Mixture Model with Word Embeddings

**Deyu Zhou**[†] **Xuan Zhang**[†] **Yulan He**[§]
[†] School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[§] School of Engineering and Applied Science, Aston University, UK
{d.zhou, xuanzhang}@seu.edu.cn, y.he@cantab.net

## Abstract

To extract structured representations of newsworthy events from Twitter, unsupervised models typically assume that tweets involving the same named entities and expressed using similar words are likely to belong to the same event. Hence, they group tweets into clusters based on the co-occurrence patterns of named entities and topical keywords. However, there are two main limitations. First, they require the number of events to be known beforehand, which is not realistic in practical applications. Second, they don't recognise that the same named entity might be referred to by multiple mentions and tweets using different mentions would be wrongly assigned to different events. To overcome these limitations, we propose a non-parametric Bayesian mixture model with word embeddings for event extraction, in which the number of events can be inferred automatically and the issue of lexical variations for the same named entity can be dealt with properly. Our model has been evaluated on three datasets with sizes ranging between 2,499 and over 60 million tweets. Experimental results show that our model outperforms the baseline approach on all datasets by 5-8% in F-measure.

## 1 Introduction

Event extraction from texts is to automatically extract key information of events such as what happened to whom, when and where. Previous research mainly focused on news articles, the best and abundant source of newsworthy events. With the increasing popularity of social media platforms, events are also reported and discussed in social media apart from news articles. It was reported in (Petrovic et al., 2013) that even $1\%$ of public Twitter stream covers $95\%$ of all events on newswire. Extracting events from social media makes it possible to quickly understand what is being discussed. It can be further integrated into downstream applications such as tracking the public's viewpoints towards a certain event. However, due to the difficulty in acquiring annotated data for training and the short and informal text commonly appeared in social media, traditional approaches (Grishman et al., 2005; Tanev et al., 2008; Piskorski et al., 2008) to event extraction from new articles are no longer applicable in social media data. Nevertheless, one important characteristic of social media data is that for most newsworthy events, there might be a high volume of redundant messages referring to the same event. An example of several tweets describing one event is given in Table 1.

Table 1: An example of several tweets describing the same event about "*Space shuttle Atlantis landed at Kennedy Space Center in Florida on 2011/07/08*".

| |
|---|
| Boom! #shuttle #Atlantis is back! |
| The shuttle is down, welcome back Atlantis, goodbye shuttle program. |
| Atlantis lands safely in Florida, marking the end of NASA's 30-yr space shuttle programme. |
| Space shuttle Atlantis lands at Kennedy space center, ending NASA's 30-year shuttle program. |

Approaches to event extraction from social media have largely explored the redundancy characteristic (Xia et al., 2015; Popescu et al., 2011; Abdelhaq et al., 2013). Most of the pervious methods aim to discover new or previously unidenti-



*Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 808–817, Valencia, Spain, April 3-7, 2017. 

fied events without extracting structured representations of events. Ritter et al. (2012) presented a system called TwiCal to extract and categorize events from Twitter. The strength of association between each named entity $y$ and date $d$ is measured based on the number of co-occurring tweets in order to form a binary tuple $\langle y, d \rangle$ to represent an event. However, TwiCal relies on a supervised sequence labeler trained on tweets annotated with event mentions for the identification of event-related phrases.

Assuming that 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 $k$, Zhou et al. (2014; 2015) proposed an unsupervised Bayesian model called latent event model (LEM) for event extraction from Twitter. However, LEM requires the number of events to be known beforehand, which is not realistic in practical applications. To address this limitation, in this paper, a non-parametric mixture model for event extraction is proposed, in which the number of events is inferred automatically from data. Moreover, the lexical variation of the same named entity, for example, "Charles" and "The Prince of Wales", if identified properly, could be exploited to help in detecting the same event described in tweets with different mentions. To this end, we further extend the non-parametric mixture model to incorporate word embeddings generated using neural language modelling.

The main contributions of the paper are summarized below:

- We propose a non-parametric approach called the Dirichlet Process Event Mixture Model (DPEMM) to extract structured events information. It avoids the problem of presetting the number of events, a common issue in latent Dirichlet allocation (LDA) based approaches.
- We extend DPEMM by incorporating word embeddings to deal with the issue of using multiple mentions to refer to the same named entity.
- The proposed approaches have been evaluated on three datasets and a significant improvement on F-measure compared to the baseline approach is observed.

## 2 Related Work

Research on event extraction of tweets can be divided into domain-specific and open domain approaches. Domain-specific approaches typically focus on one particular type of events. For example, Panem et al. (2014) proposed an algorithm to extract attribute-value pairs and map such pairs to manually generated schemas for natural disaster events. Evaluation was carried out on 58,000 tweets for 20 events and the system can fill such event schemas with an F-measure of 60%. TSum4act (Nguyen et al., 2015) was designed for disaster responses based on tweets and has been evaluated on a dataset containing 230,535 tweets. Anantharam et al. (Anantharam et al., 2014) 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.

Open domain event extraction approaches are not limited to a specific event type or topic. Benson et al. (2011) proposed a structured graphical model which simultaneously analyzed individual messages, clustered, and induced a canonical value for each event. Popescu et al. (2011) 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. (2012) worked on social events extraction for social network construction using a factor graph by harvesting the redundancy in tweets. Experiments were conducted on manually annotated data set and results showed that it achieved a gain of 21% in F-measure. In (Abdelhaq et al., 2013), 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. Armengo et al. (2015) proposed a model named Tweet-SCAN based on 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. Xia et al. (2015) proposed a framework combining text, image and geo-location information to detect events with low spatial and temporal deviation.

Our proposed method belongs to the open do-

main category. Different from the previous methods, our model can automatically identify the number of events in the corpus and deal with lexical variations of named entities using word embeddings generating from neural language modelling.

## 3 Methodology

Our proposed model for event extraction is based on a typical non-parametric mixture model, Dirichlet Process Mixture Model (DPMM) (Green and Richardson, 2001; Ishwaran and Zarepour, 2002) in which the number of active clusters is automatically learned from the data. We first give a brief introduction to DPMM. In DPMM, observation $x_i$ is assumed to be derived from the following model:

$$\begin{aligned} \pi|\alpha &\sim \text{Dirichlet}(\alpha/K, ..., \alpha/K) \\ c_i|\pi &\sim \text{Multinomial}(\pi) \\ \phi_k|G_0 &\sim G_0 \\ x_i|c_i, \{\phi_k\}_{k=1}^K &\sim F(\phi_{c_i}) \end{aligned}$$

where $K$ denoting the number of components in the mixture model and can go to infinity, $\pi$ is the mixture weights of each component, $\phi_k$ is the parameter of the $k$th component, $c_i$ denotes the index of components, $F(\phi_{c_i})$ denotes the distribution of $x_i$ with parameter $\phi_{c_i}$. In this model, $\pi$ can be generated by stick-breaking model (Pitman, 2002) and Chinese restaurant process (Aldous, 1985).

Suppose that all the observations are generated by DPMM and the variable of observation $x_i$ is $\theta_i$, which has the following conditional distribution:

$$\theta_i|\theta_1, ..., \theta_{i-1} \sim \sum_{k=1}^{K} \frac{n_k}{i-1+\alpha}\delta_{\phi_k} + \frac{\alpha}{i-1+\alpha}G_0$$

where $\phi_1, ..., \phi_k$ are the distinct values of $\theta$, $n_k$ is the number of observations that belong to component $k$, $\delta_{\phi_k}$ is a probability measure concentrated on $\phi_k$, which returns 1 when $\theta_i = \phi_k$, $G_0$ is the base probability measure and generates new $\phi$ with probability $\frac{\alpha}{i-1+\alpha}$.

### 3.1 Dirichlet Process Event Mixture Model (DPEMM)

We propose a Dirichlet Process Event Mixture Model (DPEMM) in which each event is represented as a 4-tuple $\langle y, l, k, d\rangle$, where $y$ stands for non-location named entity, $l$ for location, $k$ for event-related keyword and $d$ for date. It is worth noting that $y, l, k$ is not atomic and could be a set by itself. One event can have multiple named entities, locations or keywords. Also, some elements of the 4-tuple might be absent if no associated information can be found in tweets. Assuming that the data contains an infinite number of events and each event is modeled as a joint distribution over $y$, $l$, $k$ and $d$, the model can be viewed as a Bayesian mixture model.

The generative process of the proposed model is given below.

- Draw event distribution $\pi \sim \text{Dirichlet}(\frac{\alpha}{K}, ..., \frac{\alpha}{K})$.
- For each event $e$, draw multinomial distribution $\theta_e \sim \text{Dirichlet}(\beta)$, $\psi_e \sim \text{Dirichlet}(\eta)$, $\omega_e \sim \text{Dirichlet}(\lambda)$, $\phi_e \sim \text{Dirichlet}(\gamma)$.
- For each tweet $\mathbf{t}$:
  - Draw an event from event distribution $e \sim \text{Multinomial}(\pi)$.
  - For each non-location named entity occurred in $\mathbf{t}$, choose a named entity $y \sim \text{Multinomial}(\theta_e)$.
  - For each location occurred in $\mathbf{t}$, choose a location $l \sim \text{Multinomial}(\psi_e)$.
  - For each keyword occurred in $\mathbf{t}$, choose a keyword $k \sim \text{Multinomial}(\omega_e)$.
  - For each date occurred in $\mathbf{t}$, choose a date $d \sim \text{Multinomial}(\phi_e)$.

Here, $K$ is the number of events and can go to infinity. To estimate the parameters of the model, we employ Markov chain sampling methods (Neal, 2000). As $K$ goes to infinity, we cannot represent the infinite number of $\theta_e$, $\psi_e$, $\omega_e$ and $\phi_e$ explicitly. Therefore, we perform Gibbs sampling for only those parameters that are currently associated with some observations. Gibbs sampling for the event label $e_i$ of tweet $i$ is based on the following conditional probabilities:
If $e_i$ is assigned with a previously seen event $e$,

$$\begin{aligned} P(e_i = e|e_{-i}, s_i, t_{-i}) &= b\frac{n_e^{-i}}{n-1+\alpha} \\ &\prod_{y\in y_i}\int F_y(\theta_e)dH_y(\theta_e)\prod_{l\in l_i}\int F_l(\psi_e)dH_l(\psi_e) \\ &\prod_{k\in k_i}\int F_k(\omega_e)dH_k(\omega_e)\prod_{d\in d_i}\int F_d(\phi_e)dH_d(\phi_e) \end{aligned}$$

If $e_i$ is assigned with a new event,

$$P(e_i = e_{new}|e_{-i}, s_i\, d_i, t_{-i}) = b\frac{\alpha}{n-1+\alpha} \prod_{y\in y_i}\int F_y(\theta)dG_0(\theta)\prod_{l\in l_i}\int F_l(\psi)dG_0(\psi) \prod_{k\in k_i}\int F_k(\omega)dG_0(\omega)\prod_{d\in d_i}\int F_d(\phi)dG_0(\phi)$$

, where $b$ is the normalizing constant that makes the probabilities sum to 1, $e_{-i}$ is the event assignment of all the other tweets excluding the data from $i$th tweet, $s_i$ is the four-tuple $\langle y_i, l_i, k_i, d_i\rangle$, $n$ is the total number of tweets, $n_e^{-i}$ is the number of tweets assigned with event label $e$ excluding the current assignment, $F_y(\theta_e)$ is the multinomial distribution over non-location named entities with prior $\theta_e$, $F_l(\psi_e)$ over locations with $\psi_e$, $F_k(\omega_e)$ over keywords with $\omega_e$, and $F_d(\phi_e)$ over dates with $\phi_e$. $H_y(\theta_e)$ is the posterior distribution of parameters based on the prior $G_0(\theta_e) \sim \text{Dirichlet}(\beta)$ and all observations $y_j$ for which $j \neq i$ and $e_j = e$, and similarly for $H_l(\psi_e)$, $H_k(\omega_e)$ and $H_d(\phi_e)$.

We then derive the following formulae:
If $e_i$ is assigned with a previously seen event $e$,

$$P(e_i = e|e_{-i}, s_i, t_{-i}) = b\frac{n_e^{-i}}{n-1+\alpha} \prod_{y\in y_i}\frac{n_{e,y}^{-i}+\beta}{\Sigma_{t=1}^{Y}(n_{e,y,t}^{-i}+\beta)}\prod_{l\in l_i}\frac{n_{e,l}^{-i}+\eta}{\Sigma_{t=1}^{L}(n_{e,l,t}^{-i}+\eta)} \prod_{k\in k_i}\frac{n_{e,k}^{-i}+\lambda}{\Sigma_{t=1}^{K}(n_{e,k,t}^{-i}+\lambda)}\prod_{d\in d_i}\frac{n_{e,d}^{-i}+\gamma}{\Sigma_{t=1}^{D}(n_{e,d,t}^{-i}+\gamma)}$$

If $e_i$ is assigned with a new event $e'$,

$$P(e_i = e'|e_{-i}, s_i, t_{-i}) = b\frac{\alpha}{n-1+\alpha}\prod_{y\in y_i}\frac{1}{Y}\prod_{l\in l_i}\frac{1}{L}\prod_{k\in k_i}\frac{1}{K}\prod_{d\in d_i}\frac{1}{D}$$

, where the superscript $-i$ denotes a count excluding data from $i$th tweet, $n_{e,y}^{-i}$, $n_{e,l}^{-i}$, $n_{e,k}^{-i}$, and $n_{e,d}^{-i}$ denotes the occurrence count of non-location $y$, location $l$, keyword $k$ and date $d$ in event $e$, respectively. $t_{-i}$ denotes all other tweets. $\beta, \eta, \lambda, \gamma$ are the hyperparameters and are set to the same value 1 in the experiments in the paper.

### 3.2 DPEMM With Word Embeddings

In the proposed model described above, each distinct word is treated separately without considering their semantic relations. However, the knowledge of semantic relations of words might be useful for event extraction. For example, "`Putin`" and "`The President of Russia`" are two different mentions referring to the same person. Knowing such knowledge would help to cluster the following two tweets together, "*President of Russia attended the opening ceremony of the 119th session of the International Olympic Committee.*" and "*Putin took part in the presentation of Sochi, at the 119th of the IOC.*", and hence identify a single event. Moreover, there might exist partitive relations between two location names. For example, `Croydon` is a part of `London`. The information will help to identify the same event described as happened in `Croydon` and `London` and subsequently improve the accuracy of event extraction.

To incorporate such information about semantic relations between words, we propose another model by employing word embeddings to describe the semantic relations among $y$ or $l$, which is called DPEMM-WE. Word embedding for each word is often represented in a vector form. In the embedded hyperspace, words that are more semantically or syntactically similar to each other are located closer. We use neural language modeling (Collobert et al., 2011) to learn word representations by discriminating the legitimate phrase from incorrect phrases. Given a sequence of words $p = (w_1, w_2, ..., w_d)$ with window size $d$, the goal of the model is to discriminate the sequence of words $p$ (the correct phrase) from a random sequence of words $p^r$. Thus, the objective function of the model is to minimize the ranking loss with respect to parameters $\theta$:

$$\sum_{p\in\mathfrak{p}}\sum_{r\in\mathfrak{R}}\max(0, 1 - f_\theta(p) + f_\theta(p^r)) \quad (1)$$

, where $\mathfrak{p}$ is the set of all possible text sequences with $d$ words coming from the corpus $U$, $\mathfrak{R}$ is the dictionary of words, $p^r$ denotes the window of words obtained by replacing the central word of $p$ by the word $r$ and $f_\theta(p)$ is the score of $p$. The dataset for learning the language model can be constructed by considering all the word sequences in the corpus. Positive examples are the word sequences from the corpus, while negative examples are the same word sequence with the central word replaced by a random one.

Different from DPEMM, in DPEMM-WE, non-location named entities $y$ and locations $l$ are as-

sumed to follow Gaussian distribution to incorporate word embeddings and their prior distributions are assumed to follow Normal-Inverse-Wishart (NIW) distribution, which is conjugated with Gaussian distribution. The probability density function is

$$NIW(\mu, \Sigma|\mu_0, \lambda, \Psi, \nu) = \mathcal{N}(\mu|\mu_0, \frac{1}{\lambda}\Sigma)\mathscr{W}^{-1}(\Sigma|\Psi, \nu)$$

$$\mathcal{N}(\mu|\mu_0, \frac{1}{\lambda}\Sigma) = \frac{e^{-\frac{1}{2}(\mu-\mu_0)^T(\Sigma/\lambda)^{-1}(\mu-\mu_0)}}{\sqrt{|2\pi\Sigma/\lambda|}}$$

$$\mathscr{W}^{-1}(\Sigma|\Psi, \nu) = \frac{|\Sigma|^{\frac{\nu}{2}}}{2^{\frac{\nu p}{2}}\Gamma_p(\frac{\nu}{2})}|\Sigma|^{-\frac{\nu+p+1}{2}}e^{-\frac{1}{2}tr(\Psi\Sigma^{-1})}$$

where $\Sigma$ and $\Psi$ are $p \times p$ positive definite matrices and $\Gamma_p(\cdot)$ is the multivariate gamma function. The graphical model of DPEMM-WE is shown in Figure 1.

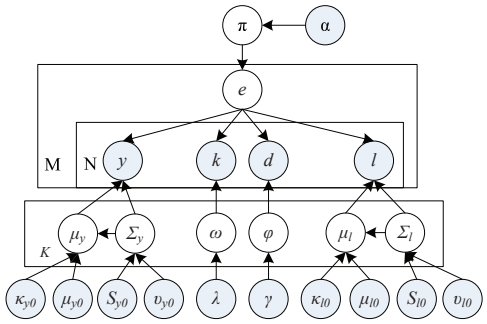


Figure 1: Plate notation of the graphical model DPEMM-WE.

The generative process of DPEMM-WE is given below.

- Draw event distribution $\pi \sim$ Dirichlet($\frac{\alpha}{K}, ..., \frac{\alpha}{K}$).
- For each event, draw Graussian distribution $\theta_e \sim$ NIW($\beta$), $\psi_e \sim$ NIW($\eta$); draw Multinomial distribution $\omega_e \sim$ Dirichlet($\lambda$), $\phi_e \sim$ Dirichlet($\gamma$).
- For each tweet $\mathbf{t}$:
  - Draw an event from event distribution $e \sim$ Multinomial($\pi$).
  - For each named entity occurred in $\mathbf{t}$, choose a named entity $y \sim$ Gaussian($\theta_e$).
  - For each location occurred in $\mathbf{t}$, choose a location $l \sim$ Gaussian($\psi_e$).
  - For each keyword occurred in $\mathbf{t}$, choose a keyword $k \sim$ Multinomial($\omega_e$).
  - For each date occurred in $\mathbf{t}$, choose a date $d \sim$ Multinomial($\phi_e$).

, where $\beta = (\kappa_{y0}, \mu_{y0}, \nu_{y0}, S_{y0})$, $\theta_e = (\mu_y, \Sigma_y)$, $\eta = (\kappa_{l0}, \mu_{l0}, \nu_{l0}, S_{l0})$, $\psi_e = (\mu_l, \Sigma_l)$.

Similar to DPEMM, parameters of the model can be estimated by Gibbs sampling. The sampling equation is given as below:
If $e_i$ is assigned with a previously seen event $e$,

$$P(e_i = e|e_{-i}, s_i, t_{-i}) = b\frac{n_e^{-i}}{n-1+\alpha} \prod_{y\in y_i}\int p(y|\theta)p(\theta|\nu_{e,y0}, \kappa_{e,y0}, \mu_{e,y0}, S_{e,y0}, t_{-i})d\theta \prod_{l\in l_i}\int p(l|\psi)p(\psi|\nu_{e,l0}, \kappa_{e,l0}, \mu_{e,l0}, S_{e,l0}, t_{-i})d\psi \prod_{k\in k_i}\int p(k|\omega)p(\omega|\lambda, t_{-i})d\omega \prod_{d\in d_i}\int p(d|\phi)p(\phi|\gamma, t_{-i})d\phi$$

If $e_i$ is assigned with a new event $e'$,

$$P(e_i = e'|e_{-i}, s_i, t_{-i}) = b\frac{\alpha}{n-1+\alpha} \prod_{y\in y_i}\int p(y|\theta)p(\theta|\nu_{e',y0}, \kappa_{e',y0}, \mu_{e',y0}, S_{e',y0})d\theta \prod_{l\in l_i}\int p(l|\psi)p(\psi|\nu_{e',l0}, \kappa_{e',l0}, \mu_{e',l0}, S_{e',l0})d\psi \prod_{k\in k_i}\int p(k|\omega)p(\omega|\lambda)d\omega \prod_{d\in d_i}\int p(d|\phi)p(\phi|\gamma)d\phi$$

, where $\theta$ and $\psi$ denote parameter $(\mu, \Sigma)$.

As

$$\int \mathcal{N}(x|\theta)NIW(\theta|\nu, \kappa, \mu, S)d\theta = \mathscr{T}(\nu - D + 1, \mu, \frac{S(\kappa+1)}{\kappa(\nu-D+1)})$$

the parameters of entities' $\mathscr{T}$ distribution are

given as:

$$
\begin{aligned}
\kappa_{e,y} &= \kappa_{y0} + N_e \\
\nu_{e,y} &= \nu_{y0} + N_e \\
\mu_{e,y} &= \frac{\kappa_{y0}\mu_{y0} + N_e\overline{v}_{e,y}}{\kappa_{e,y}} \\
S_{e,y} &= S_{e0} + C_{e,y} \\
&\quad + \frac{\kappa_{e0}N_e}{\kappa e,y}(\overline{v}_{e,y} - \mu_{e0})(\overline{v}_{e,y} - \mu_{e0})^T \\
\overline{v}_{e,y} &= \frac{\sum_{y\in e} v_y}{N_e} \\
C_{e,y} &= \sum_{y\in e}(v_y - \overline{v}_{e,y})(v_y - \overline{v}_{e,y})^T
\end{aligned}
$$

, where $v_y$ means the word embedding of entity $y$. The parameters of locations' $\mathscr{T}$ distribution can be calculated similarly.

### 3.3 Post-Processing

DPEMM or DPEMM-WE essentially outputs tweet clusters where each cluster represents one event. To further extract structured representation of an event, such as named entities, locations, dates and keywords, from each cluster, we simultaneously look into the probabilities of each event element returned by our models and their co-occurrence frequencies. We assumed that non-location named entities were the most important since an event is usually operated by somebody or something. If an event happened in someplace like "A bomb attack was happened in London", the location is the most important. Therefore, we first select the top 3 non-location named entities ranked by the probability $\theta_e$. For each non-location named entity $y$, its occurrence frequency needs to exceed $T_y$. If no such entities exist, the top 3 locations ranked by the probability $\psi_e$ are chosen; otherwise, the location $l$ is chosen based on its co-occurrences with the selected non-location named entities. After that, keywords $k$ are chosen among the top 10 $\omega_e$. Only those keywords with correlation coefficients with the chosen named entities and locations exceeding $T_c$ are selected. Then date $d$ is chosen in a similar way. Here, we define the correlation coefficient between $a$ and $b$ as $Corr(a,b) = \log\frac{\#(a,b)}{\#(b)}$, where $\#(a,b)$ denotes the co-occurrence count of $a$ and $b$ in the same tweet within a tweet cluster and $\#(b)$ denotes the occurrence count of $b$ in all tweets within a tweet cluster. In our experiments, we set the thresholds $T_y = 0.2, T_c = 0.4$.

If the entity or location is in the form of word embeddings, its occurrence frequency is calculated as the occurrence frequencies of all the neighboring words which have cosine similarity values greater than 0.85. The rationale behind our post-processing step is that although tweets have been filtered in the pre-processing step, tweet clusters generated by the proposed models still contain noisy event elements. As such, we select event elements from tweet clusters not only based on their probability distributions given by the proposed models but also taking into account their co-occurrences in each tweet cluster.

## 4 Experiments

We evaluate the proposed models on three datasets. Dataset I is the First Story Detection (FSD) dataset (Petrovic et al., 2013) containing 2,499 tweets manually annotated with 27 events. These tweets were published between 7th July and 12th September 2011, covering a range of categories such as accidents and science discoveries. Considering that events mentioned in a very few tweets are less likely to be significant, we remove events mentioned in less than 15 tweets and are left with 2,453 tweets annotated with 20 events. Dataset II and III were collected from tweets published in the month of December in 2010 using the Twitter streaming API. Dataset II consists of 6,297 tweets manually annotated with 73 events. All the annotated events in Dataset II are mentioned in at least 15 tweets. Dataset III contains 60 millions unlabelled tweets. We chose LEM (Zhou et al., 2014), the state-of-art approach based on Bayesian modelling for event extraction, as the baseline to compare with the proposed model. For all datasets, pre-processing is done as described in *baseline* (Zhou et al., 2014). A named entity tagger[1] specifically built for Twitter is used for extracting named entities including locations from tweets. A Twitter Part-of-Speech tagger (Gimpel et al., 2011) is used for POS tagging and only words tagged with nouns, verbs or adjectives are kept as candidate keywords. Word embeddings are trained on Dataset III (60 million tweets) using Word2Vec[2]. In this model, a word is used as an input to a log-linear classifier with continuous projection layer and the objective is to predict its neighboring words.

[1] http://github.com/aritter/twitter-nlp

[2] http://code.google.com/p/word2vec/

We train DPEMM, DPEMM-WE and LEM on an IBM 3850 X5 Linux server equipped with 1.86 Ghz processor and 8 GB DDR3 RAM. The number of Gibbs sampling iterations is set to 1,000 for LEM for all the datasets. For DPEMM, it converges in 16 iterations on Dataset I and 20 iterations on Dataset II and III. While for DPEMM-WE, it converges in 20 iterations on both Dataset II and III.

### 4.1 Experimental Results

To evaluate the performance of the proposed approaches, we calculate *precision*, *recall*, and *F-measure* on Dataset I and II and only *precision* on Dataset III since it is hard to know exactly how many events are mentioned in such a large dataset. The $precision$ is defined based on the following criteria: 1) Do the entity $y$, location $l$ and the date $d$ 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? If the extracted events does not have any keyword, such events are considered as incorrect.

The performance comparison of event extraction results is presented in Table 2. It can be observed that the proposed DPEMM achieves better performance on all the three datasets compared to the baseline approach, with the improvement in F-measure being $6.1\%$ and $7.7\%$ on Dataset I and II, respectively. After incorporating word embeddings into DPEMM, the proposed DPEMM-WE further improves upon DPEMM slightly by 1.45% in F-measure on Dataset II, but more significantly by 4.16% in precision on Dataset III. It verifies our hypothesis that the knowledge about the semantic relations of entities and locations could potentially improve the performance of event extraction. We also compared the proposed models with K-means on Dataset I to justify whether these proposed generative models are better than traditional clustering methods based on co-occurrence. The feature set was constructed by organizing the words in four categories such as $y, l, k, d$ and concatenating the four one-hot feature sets together.

It is worth noting that we did not apply DPEMM-WE on Dataset I because this dataset is very small, consisting of less than 2500 tweets. It is thus unreliable to learn word embeddings from such a small dataset. It is also hard to pre-train word embedding from extra dataset like Wikipedia corpus for Dataset I because some words in social media are informal and some words were only mentioned in some specific time slots such as "Dream Act". Also, word embeddings learned from Dataset III are not beneficial for event extraction in Dataset I since tweets collected in these two datasets were in different periods and a large number of words in Dataset I cannot be found in Dataset III. For example, more than 20% named entities in Dataset I can not be found in the word vocabulary constructed based on Dataset III.

Table 2: Comparison of the performance of event extraction on the three datasets.

| **Dataset I** | | | |
|---|---|---|---|
| Method | Precision(%) | Recall(%) | F-measure(%) |
| K-means | 91.23 | 55.40 | 68.93 |
| LEM | 79.17 | 85.00 | 81.98 |
| DPEMM | **86.21** | **90.00** | **88.06** |
| **Dataset II** | | | |
| Method | Precision(%) | Recall(%) | F-measure(%) |
| LEM | 62.35 | 68.49 | 65.28 |
| DPEMM | 70.80 | 75.34 | 73.00 |
| DPEMM-WE | **71.15** | **78.08** | **74.45** |
| **Dataset III** | | | |
| Method | Precision(%) | Number of correctly Events | |
| LEM | 68.25 | 215 | |
| DPEMM | 68.60 | 342 | |
| DPEMM-WE | **72.76** | **353** | |

Examples of events extracted by DPEMM and DPEMM-WE are shown in Table 3. It can be observed that the extracted results from DPEMM-WE contain more detailed and accurate information describing the events. For example, for the first event, DPEMM-WE is able to extraction the location information while DPEMM failed to do so. For the third event, DPEMM-WE gives more accurate location information compared to DPEMM. It might attribute to the advantage of incorporating word embeddings which are able to map semantically similar words into nearby locations in the embedding hyperspace. As such, although two tweets might contain different mentions of named entities and locations, they might still be clustered together if these named entities or locations have similar word embeddings.

We observed that the precision achieved by DPEMM on Dataset I is signifcantly better than LEM on Dataset I and II while similar on Dataset III. We found that DPEMM tended to generate many but smaller clusters compared to LEM. As dataset III is huge, DPEMM might generate some small clusters which do not contain enough information to describe a correct event.

Table 3: Examples of extracted events based on DPEMM and DPEMM-WE.

| Event | Method | Entities | Locations | Keywords | Date |
|---|---|---|---|---|---|
| 1 | DPEMM | Biden | - | inevit marriage gay say | 2010-12-24 |
| | DPEMM-WE | Biden, Obama | WhiteHouse | marrige gay inevit say | 2010-12-24 |
| 2 | DPEMM | Charles | London | car protest attack contain | 2010-12-09 |
| | DPEMM-WE | Charles, Camilla | London, UK | protest car demonstrators attack | 2010-12-09 |
| 3 | DPEMM | - | London | snow close airport ice | 2010-12-18 |
| | DPEMM-WE | - | Europ, London, Gatwick | snow delay runaway airport | 2010-12-18 |
| 4 | DPEMM | DreamAct, Reid | - | pass bill vote will | 2010-12-09 |
| | DPEMM-WE | DreamAct, Harry, Reid | Senate | vote pass debate bill | 2010-12-09 |
| 5 | DPEMM | WorldCup | Russia | announce will host chose | 2010-12-03 |
| | DPEMM-WE | WorldCup, FIFA | Russia, Qatar | news host win will | 2010-12-03 |

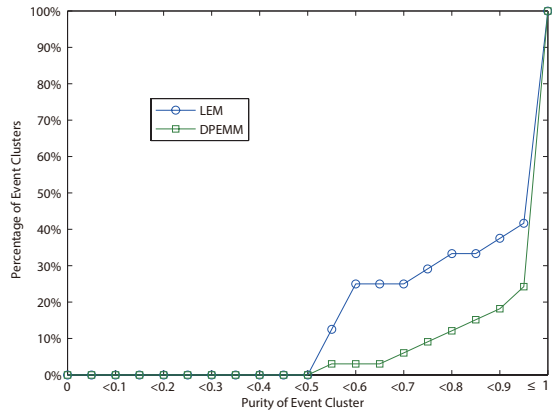


Figure 2: Purities of the clusters on Dataset I.

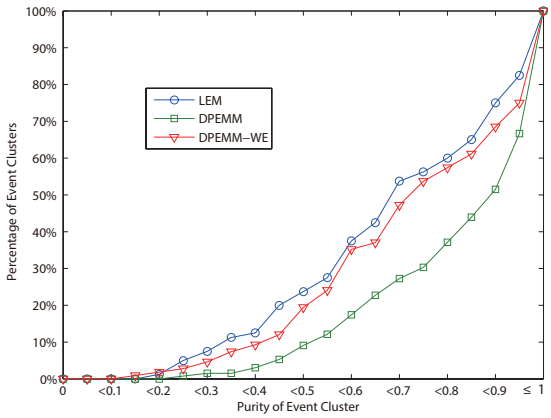


Figure 3: Purities of the clusters on Dataset II.

### 4.2 Quality of Clusters

As the proposed approaches essentially group tweets into different clusters with each cluster corresponding to an event, we conduct experiments to explore the quality of clusters by a measure of purity, which is defined as $P_e = \frac{n_e}{n}$, where $n_e$ denotes the number of tweets describing the event $e$ extracted from a cluster and $n$ denotes the total number of tweets in the cluster. Since it is difficult to calculate the purity on Dataset III, we only report the results on Dataset I and Dataset II as shown in Figure 2 and 3 respectively.

Each point $(x, y)$ in the figures denotes the percentage $y$ of the clusters whose purity is less than $x$. Obviously, if the curve is steeper, it means that the percentage of the clusters with low purity is smaller and the quality of the clusters is better. It can be observed that DPEMM achieves the best quality of cluster on both Dataset I and Dataset II, whose precision is lower than DPEMM-WE. Specifically, on Dataset I, more than $80\%$ of clusters generated by DPEMM has the purity value greater than $0.9$, compared to only $70\%$ in LEM. It might be attributed to the property of DPEMM that the cluster is generated dynamically without a preset number of clusters. On Dataset II, both DPEMM and DPEMM-WE achieve better clustering results compared to LEM. However, the purity of clusters generated by DPEMM is slightly higher than that generated by DPEMM-WE. This is somewhat contrary to our prior belief. By further analyzing the results, we found that as more tweets are clustered together using DPEMM-WE, more noisy information such as some named entities with similar word embeddings which are not related to the events is introduced. We present an example of the tweet clusters describing the same event generated by DPEMM and DPEMM-WE in Figure 4. For each method, we use a histogram to indicate the number of tweets which share the same event elements. Regions highlighted in dark or light red colors indicate that the corresponding tweets are event-related. Regions highlighted in blue denote the corresponding tweets are not event-related. It can be observed that the purity of the cluster generated by DPEMM is $91\%$ which is better than DPEMM-WE's $63\%$. However, the size of the cluster returned by DPEMM is smaller and it failed to extract the location information.

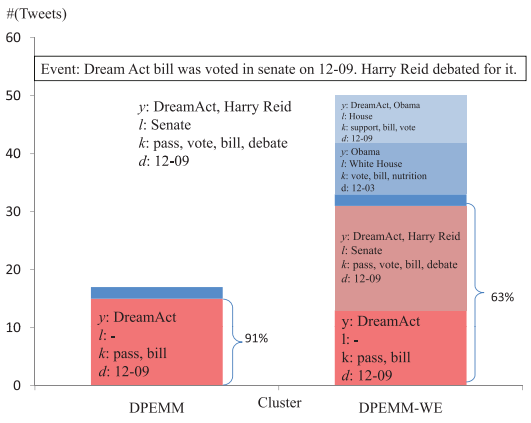


Figure 4: Example tweet clustering results generated by DPEMM and DPEMM-WE.

On the contrary, DPEMM-WE generated a larger cluster and for some tweets, it successfully extracted the location "Senate". However, more spurious tweets are included because "Harry Reid" is close to both "DreamAct" and "Obama", and "White House" is close to "Senate" in the word embedding space. Therefore, although DPEMM-WE gives better extraction results overall compared to DPEMM as shown in Table 2, it returns lower purity results because of some noisy information introduced through word embeddings.

## 5 Conclusions and Future Work

In this paper, we have proposed a model based on the Dirichlet Process mixture model to extract structured event information from social media data. Different from previous approaches for event extraction which require setting the number of events beforehand, it can infer the number of events automatically from data. It is specifically appealing for processing large-scale social media data. Moreover, considering different mentions of names could refer to the same person (and similarly for other named entities such as location), we have proposed to incorporate word embeddings into DPEMM so as to more effectively capture semantically similar words. Experiments have been conducted on three datasets and the proposed approaches achieve better performance on all the datasets in comparison with the baseline approach. In the future, we plan to investigate more effective way in reducing the noise introduced by word embeddings and incorporate emotion information into the proposed models to simultaneously extract public opinions of the extracted event.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was funded by the National Natural Foundation of China (61528302), the Natural Science Foundation of Jiangsu Province of China (BK20161430), the National Key Research and Development Program of China (2016YFC1306704) and the Collaborative Innovation Center of Wireless Communications Technology.

## References

Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. 2013. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, pages 1326–1329.

David J. Aldous. 1985. *Exchangeability and related topics*. Springer.

Pramod Anantharam, Payam Barnaghi, T. K. Prasad, and Amit P. Sheth. 2014. Extracting city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology*, 9(10):e110206.

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.

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.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter J. Green and Sylvia Richardson. 2001. Modelling heterogeneity with and without the dirichlet process. *Scandinavian journal of statistics*, 28(2):355–375.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description. In *ACE 05 Evaluation Workshop*.

Hemant Ishwaran and Mahmoud Zarepour. 2002. Exact and approximate sum representations for the dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283.

Xiaohua Liu, Xiangyang Zhou, Zhongyang Fu, Furu Wei, and Ming Zhou. 2012. Exacting social events for tweets using a factor graph. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1692–1698.

Radford M. Neal. 2000. Markov chain sampling methods for dirichlet process mixture models. *Computational and Graphical Statistics*.

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.

Sandeep Panem, Manish Gupta, and Vasudeva Varma. 2014. Structured information extraction from natural disaster events on twitter. In *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*, Web-KR '14, pages 1–8, New York, NY, USA. ACM.

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*.

Jakub Piskorski, Hristo Tanev, Martin Atkinson, and Erik Van Der Goot. 2008. Cluster-centric approach to news event extraction. In *International Conference on New Trends in Multimedia and Network Information Systems*, pages 276–290.

Jim Pitman. 2002. Poisson–dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition. *Combinatorics, Probability & Computing*, 11(05):501–514.

Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. 2011. Extracting events and event descriptions from twitter. In *Proceedings of the 20th international conference companion on World Wide Web (WWW)*, pages 105–106.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1104–1112, New York, NY, USA. ACM.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In *13th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 207–218.

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

Chaolun Xia, Jun Hu, Yan Zhu, and Mor Naaman. 2015. What is new in our city? a framework for event extraction using social media posts. In *Advances in Knowledge Discovery and Data Mining*, pages 16–32. Springer.

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

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorisation. In *Proceedings of the 29th AAAI Conference (AAAI)*, pages 2468–2474.