Open Domain Event Extraction Using Neural Latent Variable Models

Xiao Liu[1,2] and **Heyan Huang**[1,2] and **Yue Zhang**[3,4*]
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]Zhejiang Lab, China
{xiaoliu,hhy63}@bit.edu.cn
[3]School of Engineering, Westlake University
[4]Institute of Advanced Technology, Westlake Institute for Advanced Study
yue.zhang@wias.org.cn

Abstract

We consider open domain event extraction, the task of extracting unconstraint types of events from news clusters. A novel latent variable neural model is constructed, which is scalable to very large corpus. A dataset is collected and manually annotated, with task-specific evaluation metrics being designed. Results show that the proposed unsupervised model gives better performance compared to the state-of-the-art method for event schema induction.

1 Introduction

Extracting events from news text has received much research attention. The task typically consists of two subtasks, namely *schema induction*, which is to extract event templates that specify argument slots for given event types (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Sha et al., 2016; Huang et al., 2016; Ahn, 2017; Yuan et al., 2018), and *event extraction*, which is to identify events with filled slots from a piece of news (Nguyen et al., 2016b; Sha et al., 2018; Liu et al., 2018a; Chen et al., 2018, 2015; Feng et al., 2016; Nguyen and Grishman, 2016; Liu et al., 2018b). Previous work focuses on extracting events from single news documents according to a set of pre-specified event types, such as arson, attack or earthquakes.

While useful for tracking highly specific types of events from news, the above setting can be relatively less useful for decision making in security and financial markets, which can require comprehensive knowledge on broad-coverage, fine-grained and dynamically-evolving event categories. In addition, given the fact that different news agencies can report the same events, redundancy can be leveraged for better event extraction. In this paper, we investigate *open domain event extraction* (ODEE), which is to extract unconstraint types of events and induce universal event schemas from clusters of news reports.

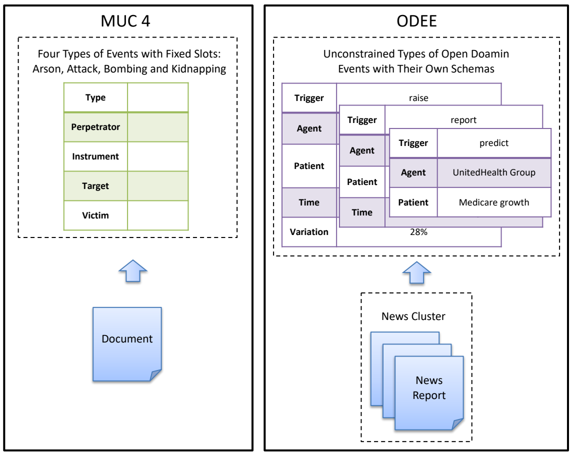


Figure 1: Comparison between MUC 4 and ODEE.

As shown in Figure 1, compared with traditional event extraction task exemplified by MUC 4 (Sundheim, 1992), the task of ODEE poses additional challenges to modeling, which have not been considered in traditional methods. First, more than one event can be extracted from a news cluster, where events can be flexible in having varying numbers of slots in the open domain, and slots can be flexible without identical distributions regardless of the event type, which has been assumed by previous work on schema induction. Second, mentions of the same entities from different reports in a news cluster should be taken into account for improved performance.

We build an unsupervised generative model to address these challenges. While previous work on generative schema induction (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015) relies on hand-crafted indicator features, we introduce latent variables produced by neural networks for better representation power. A novel graph model

[*]Corresponding author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2860–2871
Florence, Italy, July 28 - August 2, 2019.

is designed, with a latent event type vector for each news cluster from a global parameterized normal distribution, and textual redundancy features for entities. Our model takes advantage of contextualized pre-trained language model (ELMo, Peters et al. (2018)) and scalable neural variational inference (Srivastava and Sutton, 2017).

To evaluate model performance, we collect and annotate a large-scale dataset from Google Business News[1] with diverse event types and explainable event schemas. In addition to the standard metrics for schema matching, we adapt *slot coherence* based on NPMI (Lau et al., 2014) for quantitatively measuring the intrinsic qualities of slots and schemas, which are inherently clusters.

Results show that our neural latent variable model outperforms state-of-the-art event schema induction methods. In addition, redundancy is highly useful for improving open domain event extraction. Visualizations of learned parameters show that our model can give reasonable latent event types. To our knowledge, we are the first to use neural latent variable model for inducing event schemas and extracting events. We release our code and dataset at `https://github.com/lx865712528/ACL2019-ODEE`.

2 Related Work

The most popular schema induction and event extraction task setting is MUC 4, in which four event types - *Arson*, *Attack*, *Bombing* and *Kidnapping* - and four slots - *Perpetrator*, *Instrument*, *Target* and *Victim* - are defined. We compare the task settings of MUC 4 and ODEE in Figure 1. For MUC 4, the inputs are single news documents, and the output belongs to four types of events with schemas consisting of fixed slots. For ODEE, in contrast, the inputs are news clusters rather than the individual news, and the output is unconstrained types of open domain events and unique schemas with various slot combinations.

Event Schema Induction seminal work studies patterns (Shinyama and Sekine, 2006; Filatova et al., 2006; Qiu et al., 2008) and event chains (Chambers and Jurafsky, 2011) for template induction. For MUC 4, the current dominant methods include probabilistic generative methods (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015) that jointly model predicate and argument assignment, and ad-hoc clustering algorithms for inducing slots (Sha et al., 2016; Huang et al., 2016; Ahn, 2017; Yuan et al., 2018). These methods all rely on hand-crafted discrete features without fully model the textual redundancy. There are also works on modeling event schemas and scripts using neural language models (Modi and Titov, 2014; Rudinger et al., 2015; Pichotta and Mooney, 2016), but they do not explore neural latent variables and redundancy.

Event Extraction work typically assumes that event schemas are given, recognizing event triggers and their corresponding arguments. This can be regarded as a subtask of ODEE. Existing work exploits sentence-level (McClosky et al., 2011; Li et al., 2013; Liu et al., 2016; Yang and Mitchell, 2016) and document-level statistics (Liao and Grishman, 2010b; Ji and Grishman, 2008; Hong et al., 2011; Reichart and Barzilay, 2012). There has also been work using RNNs (Nguyen et al., 2016b; Sha et al., 2018; Liu et al., 2018a; Chen et al., 2018), CNNs (Chen et al., 2015; Feng et al., 2016; Nguyen and Grishman, 2016) and GCNs (Liu et al., 2018b) to represent sentences of events. Event extraction has been treated as a supervised or semi-supervised (Liao and Grishman, 2010a; Huang and Riloff, 2012) task. In contrast, ODEE is a fully unsupervised setting.

Event Discovery in Tweet Streams extracts news-worthy clusters of words, segments and frames. Both supervised and unsupervised methods have been used. The former (Sakaki et al., 2010; Benson et al., 2011) are typically designed to monitor certain event types, while the latter cluster features according to their burstiness (Becker et al., 2011; Cui et al., 2012; Li et al., 2012; Ritter et al., 2012; Qin et al., 2013; Ifrim et al., 2014; McMinn and Jose, 2015; Qin et al., 2017). This line of work is similar to our work in using information redundancy, but different because we focus on formal news texts and induce structural event schemas.

First Story Detection (FSD) systems aim to identify news articles that discuss events not reported before. Most work on FSD detects first stories by finding the nearest neighbors of new documents (Kumaran and Allan, 2005; Moran et al., 2016; Panagiotou et al., 2016; Vuurens and de Vries, 2016). This line of work exploits textual redundancy in massive streams predicting whether or not a document contains a new event as a clas-

[1] `https://news.google.com/?hl=en-US&gl=US&ceid=US:en`, crawled from Oct. 2018 to Jan. 2019.

sification task. In contrast, we study the event schemas and extract detailed events.

3 Task and Data

Task Definition. In ODEE, the input consists of news clusters, each containing reports about the same event. The output is a bag of open-domain events, each consisting of an event trigger and a list of event arguments in its own schema. In most cases, one event is semantically sufficient to represent the output.

Formally, given an open-domain news corpus $\mathcal{N}$ containing a set of news clusters $\{c \in \mathcal{N}\}$, suppose that there are M_c news reports $\{d_i \in c | i = 1, \cdots, M_c\}$ in the news cluster c focusing on the same event $\mathcal{E}_c$. The output is a pair $(\mathcal{E}_c, \mathcal{T}_{\mathcal{E}})$, where $\mathcal{E}_c$ is the aforementioned set of open-domain events and $\mathcal{T}_{\mathcal{E}}$ is a set of schemas that define the semantic slots for this set of events.

Data Collection. We crawl news reports from Google Business News, which offers news clusters about the same events from different sources. In each news cluster, there are no more than five news reports. For each news report, we obtain the title, publish timestamp, download timestamp, source URL and full text. In total, we obtain 55,618 business news reports with 13,047 news clusters in 288 batches from Oct. 17, 2018, to Jan. 22, 2019. The crawler is executed about three times per day. The full text corpus is released as *GNBusiness-Full-Text*. For this paper, we trim the news reports in each news cluster by keeping the title and first paragraph, releasing as *GNBusiness-All*.

Inspired by the general slots in FrameNet (Baker et al., 1998), we design reference event schemas for open domain event types, which include eight possible slots: *Agent*, *Patient*, *Time*, *Place*, *Aim*, *Old Value*, *New Value* and *Variation*. *Agent* and *Patient* are the semantic agent and patient of the trigger, respectively; *Aim* is the target or reason for the event. If the event involves value changes, *Old Value* serves the old value, *New Value* serves the new value and *Variation* is the variation between *New Value* and *Old Value*. Note that the roles that we define are more thematic and less specific to detailed events as some of the existing event extraction datasets do (Sundheim, 1992; Nguyen et al., 2016a), because we want to make our dataset general and useful for a wide range of open domain conditions. We leave finer-grained role typing to future work.

Split	#C	#R	#S	#W
Test	574	2,433	5,830	96,745
Dev	106	414	991	16,839
Unlabelled	12,305	52,464	127,416	2,101,558
All	12,985	55,311	134,237	2,215,142
Full-Text	12,985	55,311	1,450,336	31,103,698

Table 1: Data split statistics. (C news clusters; R news reports; S sentences; W words.)

Dataset	#D	#L	#T	#S
MUC 4	1700	400	4	4
ACE 2005	599	599	33	36
ERE	562	562	38	27
ASTRE	1038	100	12	18
GNBusiness	12,985	680	–	8

Table 2: Comparison with existing datasets. (D documents or news clusters; L labeled documents or news clusters; T event types; S slots.)

We randomly select 18 batches of news clusters, with 680 clusters in total, dividing them into a development set and a test set by a ratio of $1 : 5$. The development set, test set and the rest unlabeled clusters are released as *GNBusiness-Dev*, *GNBusiness-Test* and *GNBusiness-Unlabeled*, respectively. One coauthor and an external annotator manually label the events in the news clusters as gold standards. For each news cluster, they assign each entity which participants in the event or its head word a beforehand slot. The inter-annotator agreement (IAA) for each slot realization in the development set has a Cohen's kappa (Cohen, 1960) $\kappa = 0.7$.

The statistics of each data split is shown in Table 1, and a comparison with existing event extraction and event schema induction datasets, including ASTRE (Nguyen et al., 2016a), MUC 4, ACE 2005[2] and ERE[3], is shown in Table 2. Compared with the other datasets, GNBusiness has a much larger number of documents (i.e., news clusters in GNBusiness), and a comparable number of labeled documents.

4 Method

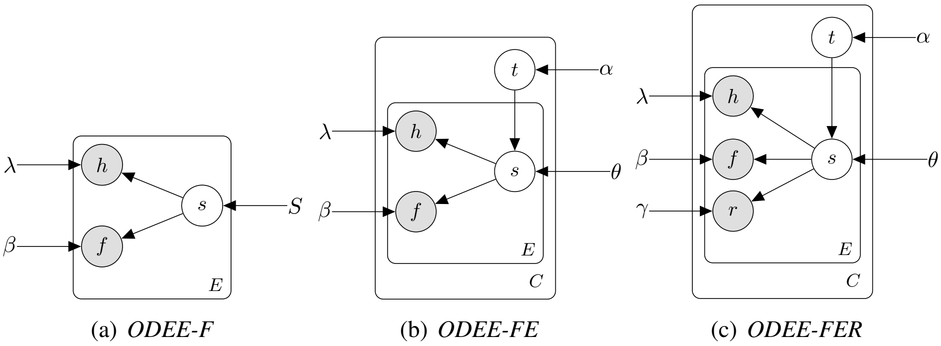
We investigate three incrementally more complex neural latent variable models for ODEE.

4.1 Model 1

Our first model is shown in Figure 2(a). It can be regarded as a neural extension of Nguyen et al.

[2] https://catalog.ldc.upenn.edu/LDC2006T06

[3] https://catalog.ldc.upenn.edu/LDC2013E64



(a) *ODEE-F* (b) *ODEE-FE* (c) *ODEE-FER*

Figure 2: Plate notations for models. (S – # of slots; E – # of entities; C – # of news clusters; V – head word vocabulary size; the grey circles are observed variables and the white circles are hidden variables.)

Algorithm 1 *ODEE-F*

```
for each entity $e \in E$ do
    Sample a slot $s \sim \text{Uniform}(1, S)$
    Sample a head $h \sim \text{Multinomial}(1, \lambda_s)$
    Sample a feature vector $f \sim \text{Normal}(\beta)$
end for
```

(2015). Given a corpus $\mathcal{N}$, we sample a slot s for each entity e from a uniform distribution of S slots, and then a head word h from a multinomial distribution, as well as a continuous feature vector $f \in \mathbb{R}^n$ produced by a contextual encoder. For simplicity, we assume that f follows a multivariable normal distribution whose covariance matrix is a diagonal matrix. We mark all the parameters (mean vectors and diagonal vectors of covariance matrixes) for the S different normal distributions for f as $\beta \in \mathbb{R}^{S \times 2n}$, where n represents the dimension of f, treating the probability matrix $\lambda \in \mathbb{R}^{S \times V}$ in the slot-head distribution as parameters under the row-wise simplex constraint, where V is the head word vocabulary size. We call this model *ODEE-F*.

Pre-trained contextualized embeddings such as ELMo (Peters et al., 2018), GPTs (Radford et al., 2018, 2019) and BERT (Devlin et al., 2018) give improvements on a range of natural language processing tasks by offering rich language model information. We choose ELMo[4] as our contextual feature encoder, which manipulates unknown words by using character representations.

The generative story is shown in Algorithm 1. The joint probability of an entity e is

$$p_{\lambda,\beta}(e) = p(s) \times p_\lambda(h|s) \times p_\beta(f|s) \quad (1)$$

Algorithm 2 *ODEE-FE*

```
for each news cluster $c \in \mathcal{N}$ do
    Sample a latent event type vector $t \sim \text{Normal}(\alpha)$
    for each entity $e \in E_c$ do
        Sample a slot $s \sim \text{Multinomial}(\text{MLP}(t; \theta))$
        Sample a head $h \sim \text{Multinomial}(1, \lambda_s)$
        Sample a feature vector $f \sim \text{Normal}(\beta_s)$
    end for
end for
```

4.2 Model 2

A limitation of *ODEE-F* is that sampling slot assignment s from a global uniform distribution does not sufficiently model the fact that different events may have different slot distributions. Thus, in Figure 2(b), we further sample a latent event type vector $t \in \mathbb{R}^n$ for each news cluster from a global normal distribution parameterized by α. We then use t and a multi-layer perceptron (MLP) with parameters θ to encode the corresponding slot distribution logits, sampling a discrete slot assignment $s \sim \text{Multinomial}(\text{MLP}(t; \theta))$. The output of the MLP is passed through a softmax layer before being used. We name this model as *ODEE-FE*.

The generative story is shown in Algorithm 2. The joint probability of a news cluster c is

$$\begin{aligned} p_{\alpha,\beta,\theta,\lambda}(c) = & \, p_\alpha(t) \times \prod_{e \in E_c} p_\theta(s|t) \\ & \times p_\lambda(h|s) \times p_\beta(f|s) \end{aligned} \quad (2)$$

4.3 Model 3

Intuitively, the more frequently a coreferential entity shows up in a news cluster, the more likely it is with an important slot. Beyond that, different news agencies focus on different aspects of event arguments, which can offer complementary information through textual redundancy. One intu-

[4] In practice, we use the "small" ELMo model with 2 × 128-d output in https://allennlp.org/elmo as initial parameters and fine-tune it on *GNBusiness-Full-Text*.

Algorithm 3 *ODEE-FER*

```
for each news cluster $c \in \mathcal{N}$ do
    Sample a latent event type vector $t \sim \text{Normal}(\alpha)$
    for each entity $e \in E_c$ do
        Sample a slot $s \sim \text{Multinomial}(\text{MLP}(t; \theta))$
        Sample a head $h \sim \text{Multinomial}(1, \lambda_s)$
        Sample a feature vector $f \sim \text{Normal}(\beta_s)$
        Sample a redundancy ratio $r \sim \text{Normal}(\gamma_s)$
    end for
end for
```

ition is that occurrence frequency is a straightforward measure for word-level redundancy. Thus, in Figure 2(c), we additionally bring in the normalized occurrence frequency of a coreferential slot realization as an observed latent variable $r \sim \text{Normal}(\gamma_s)$. We call this model *ODEE-FER*.

Formally, a news cluster c receives a latent event type vector t where each entity $e \in E_c$ receives a slot type s. The generative story is shown in Algorithm 3. The joint distribution of a news cluster with head words, redundant contextual features and latent event type is

$$p_{\alpha,\beta,\gamma,\theta,\lambda}(c) = p_\alpha(t) \times \prod_{e \in E_c} p_\theta(s|t) \times p_\lambda(h|s) \times p_\beta(f|s) \times p_\gamma(r|s) \quad (3)$$

4.4 Inference

We now consider two tasks for *ODEE-FER*: (1) learning the parameters and (2) performing inference to obtain the posterior distribution of the latent variables s and t, given a news cluster c. We adapt the amortized variational inference method of Srivastava and Sutton (2017), using neural inference network to learn the variational parameters. For simplicity, we concatenate f with r as a new observed feature vector f' in *ODEE-FER* and merge their parameters as $\beta' \in \mathbb{R}^{S \times (2n+2)}$.

Following Srivastava and Sutton (2017), we collapse the discrete latent variable s to obtain an Evidence Lower BOund (ELBO) (Kingma and Welling, 2014) of the log marginal likelihood:

$$\begin{aligned}
&\log p_{\alpha,\beta',\theta,\lambda}(c) \\
&= \log \int_t [\prod_{e \in E_c} p_{\lambda,\theta}(h|t)\, p_{\beta',\theta}(f'|t)]\, p_\alpha(t)\, dt \\
&\geq \text{ELBO}_c(\alpha, \beta', \theta, \lambda, \omega) \\
&= \mathbb{E}_{q_\omega(t)} \log p_{\beta',\theta,\lambda}(c|t) - D_{\text{KL}}[q_\omega(t) \| p_\alpha(t)]
\end{aligned} \quad (4)$$

where $D_{\text{KL}}[q_\omega \| p_\alpha]$ is the KL divergence between the variational posterior q_ω and the prior p_α. Due to the difficulty in computing the KL divergence between different categories of distributions and the existence of simple and effective reparameterization tricks for normal distributions, we choose $q_\omega(t)$ to be a normal distribution parameterized by ω, which is learned by a neural inference network. As shown in Figure 3, our inference network takes the head word histograms h (the times of each head word appears in a news cluster) and contextual features f' as inputs, and computes the mean vector μ and the variance vector σ^2 of $q_\omega(t)$.

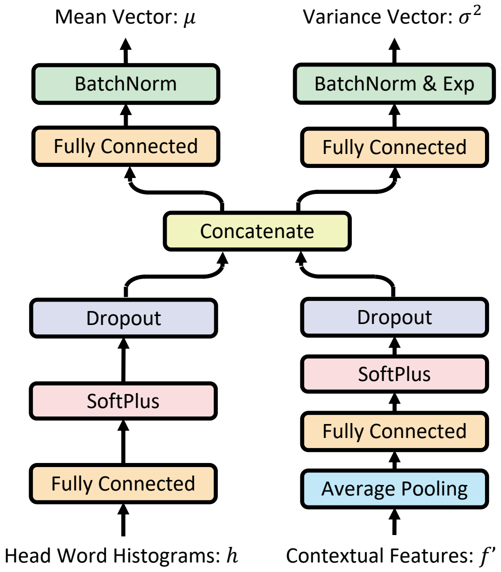


Figure 3: The framework of our inference network.

Equation 4 can be solved by obtaining a Monte Carlo sample and applying reparameterization tricks for the first term, and using the closed-form for the KL divergence term. We then use the ADAM optimizer (Kingma and Ba, 2014) to maximumize the ELBO. In addition, to alleviate the component collapsing problem (Dinh and Dumoulin, 2016), we follow Srivastava and Sutton (2017) and use high moment weight (> 0.8) and learning rate (in $[0.001, 0.1]$) in the ADAM optimizer, performing batch normalization (Ioffe and Szegedy, 2015) and dropout (Srivastava et al., 2014). After learning the model, we make slot assignment for each entity mention by MLE, choosing the slot s that maximizes the likelihood

$$\begin{aligned}
p_{\beta',\theta,\lambda}(s|e,t) &\propto p_{\beta',\theta,\lambda}(s,h,f',t) \\
&= p_\theta(s|t) \times p_\lambda(h|s) \times p_{\beta'}(f'|s)
\end{aligned} \quad (5)$$

Name	Value
Slots number S	30
Feature Dimension n	256
Fully connected layer size	100
MLP layer number	1
Activation function	softplus
Learning rate	0.002
Momentum	0.99
Dropout rate	0.2
Batch size	200

Table 3: Hyper-parameters setting.

4.5 Assembling Events for Output

To assemble the events in a news cluster c for final output, we need to find the predicate for each entity, which now has a slot value. We use POS-tags and parse trees produced by the Stanford dependency parser (Klein and Manning, 2003) to extract the predicate for the head word of each entity mention. The following rules are applied: (1) if the governor of a head word is *VB*, or (2) if the governor of a head word is *NN* and belongs to the *noun.ACT* or *noun.EVENT* category of WordNet, then it is regarded as a predicate.

We merge the predicates of entity mentions in the same coreference chain as a predicate set. For each predicate v in these sets, we find the entities whose predicate set contains v, treating the entities as arguments of the event triggered by v. Finally, by ranking the numbers of arguments, we obtain top-N open-domain events as the output $\mathcal{E}_c$.

5 Experiments

We verify the effectiveness of neural latent variable modeling and redundancy information for ODEE, and conduct case analysis. All our experiments are conducted on the GNBusiness dataset. Note that we do not compare our models and existing work on MUC 4 or ACE 2005 due to the fact that these datasets do not consist of news clusters.

Settings. The hyper-parameters in our models and inference network are shown in Table 3. Most of the hyper-parameters directly follow Srivastava and Sutton (2017), while the slot number S is chosen according to development experiments.

5.1 Evaluation Metrics

Schemas Matching. We follow previous work and use *precision*, *recall* and *F1-score* as the metrics for schema matching (Chambers and Jurafsky, 2011; Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Sha et al., 2016; Ahn, 2017). The matching between model answers and references is based on the head word. Following previous work, we regard as the head word the right-most word of an entity phrase or the right-most word before the first “of”, “that”, “which” and “by” if any.

In addition, we also perform slot mapping, between slots that our model learns and slots in the annotation. Following previous work on MUC 4 (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Sha et al., 2016; Ahn, 2017), we implement automatic greedy slot mapping. Each reference slot is mapped to a learned slot that ranks the best according to the *F1-score* metric on *GNBusiness-Dev*.

Slot Coherence. Several metrics of qualitative topic coherence evaluation have been proposed. Lau et al. (2014) showed that normalized pointwise mutual information (NPMI) between all the pairs of words in a set of topics the most closely matches human judgment among all the competing metrics. We thus adopt it as *slot coherence*[5].

Formally, the slot coherence $C_{\text{NPMI}}(s)$ of a slot s is calculated by using its top-N head words as

$$C_{\text{NPMI}}(s) = \frac{2}{N^2 - N} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \text{NPMI}(w_i, w_j) \tag{6}$$

$$\text{NPMI}(w_i, w_j) = \frac{\log \frac{p(w_i, w_j)+\epsilon}{p(w_i)\cdot p(w_j)}}{-\log(p(w_i, w_j) + \epsilon)} \tag{7}$$

where $p(w_j)$ and $p(w_i, w_j)$ are estimated based on word co-occurrence counts derived within a sliding window over external reference documents and ϵ is added to avoid zero logarithm.

Previous work on topic coherence uses Wikipedia and Gigaword as the reference corpus to calculate word frequencies (Newman et al., 2010; Lau et al., 2014). We use *GNBusiness-Full-Text*, in which there are 1.45M sentences and 31M words, which is sufficient for estimating the probabilities. To reduce sparsity, for each news report, we count word co-occurrences in the whole document instead of a sliding window. In addition, for each slot, we keep the top-5, top-10, top-20, and top-100 head words, averaging the $4 \times S$ coherence results over a test set.

[5] We use the implementation in https://github.com/jhlau/topic_interpretability.

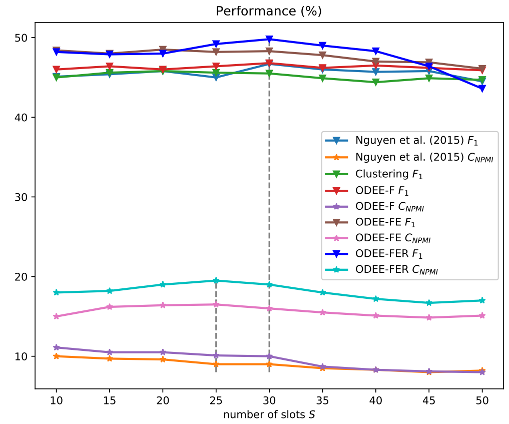


Figure 4: F_1 scores of schemas matching and averaged slot coherences C_{NPMI} of the five models with different numbers of slots S.

Method	Schema Matching (%)		
	P	R	F_1
Nguyen et al. (2015)	41.5	53.4	46.7
Clustering	41.2	50.6	45.4
ODEE-F	41.7	53.2	46.8
ODEE-FE	42.4	56.1	48.3
ODEE-FER	**43.4**	**58.3**	**49.8**

Table 4: Overall performance of schema matching.

5.2 Development Experiments

We learn the models on *GNBusiness-All* and use *GNBusiness-Dev* to determine the slot number S by grid search in $[10, 50]$ with the step equals to 5. Figure 4 shows the F_1 scores of schemas matching and averaged slot coherences of the five models we introduce in the next subsection with different numbers of slots S ranging from 10 to 50. We can see that for the best F_1 score of *ODEE-FER*, the optimal number of slots is 30, while for the best slot coherence, the optimal number of slots is 25. A value of S larger than 30 or smaller than 25 gives lower results on both F_1 score and slot coherence. Considering the balance between F_1 score and slot coherence, we chose $S = 30$ as our final S value for the remaining experiments.

5.3 Final Results

Table 4 and Table 5 show the final results. The p values based on the appropriate t-test are provided below in cases where the compared values are close. We compare our work with Nguyen et al. (2015), the state-of-the-art model on MUC 4 representing each entity as a triple containing a head word, a list of attribute relation features and a list of predicate relation features. Features in the model are discrete and extracted from dependency parse trees. The model structure is identical to our *ODEE-F* except for the features.

Method	Ave Slot Coherence
Nguyen et al. (2015)	0.10
ODEE-F	0.10
ODEE-FE	0.16
ODEE-FER	**0.18**

Table 5: Averaged slot coherence results.

To test the strengths of our external features in isolation, we build another baseline model by taking the continuous features of each entity in *ODEE-F* and runing spectral clustering (von Luxburg, 2007). We call it *Clustering*.

Schemas Matching. Table 4 shows the overall performance of schema matching on *GNBusiness-Test*. From the table, we can see that *ODEE-FER* achieves the best F_1 scores among all the methods. By comparing *Nguyen et al. (2015)* and *ODEE-F* ($p = 0.01$), we can see that using continuous contextual features gives better performance than discrete features. This demonstrates the advantages of continuous contextual features for alleviating the sparsity of discrete features in texts. We can also see from the result of *Clustering* that using only the contextual features is not sufficient for ODEE, while combining with our neural latent variable model in *ODEE-F* can achieve strong results ($p = 6 \times 10^{-6}$). This shows that the neural latent variable model can better explain the observed data.

These results demonstrate the effectivenesses of our method in incorporating with contextual features, latent event types and redundancy information. Among ODEE models, *ODEE-FE* gives a 2% gain in F_1 score against *ODEE-F*, which shows that the latent event type modeling is beneficial and the slot distribution relies on the latent event type. Additionally, there is a 1% gain in F_1 score by comparing *ODEE-FER* and *ODEE-FE* ($p = 2 \times 10^{-6}$), which confirms that leveraging redundancy is also beneficial in exploring which slot an entity should be assigned.

Slot Coherence. Table 5 shows the comparison of averaged slot coherence results over all the slots in the schemas. Note that we do not report the slot coherence for the *Clustering* model because it does not output the top-N head words in each slot. The averaged slot coherence of *ODEE-FER* is the highest, which is consistent with the conclusion from Table 4. The averaged slot coherence

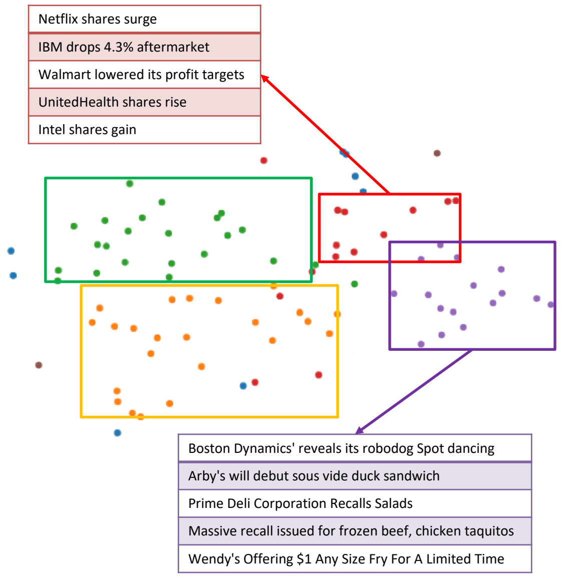


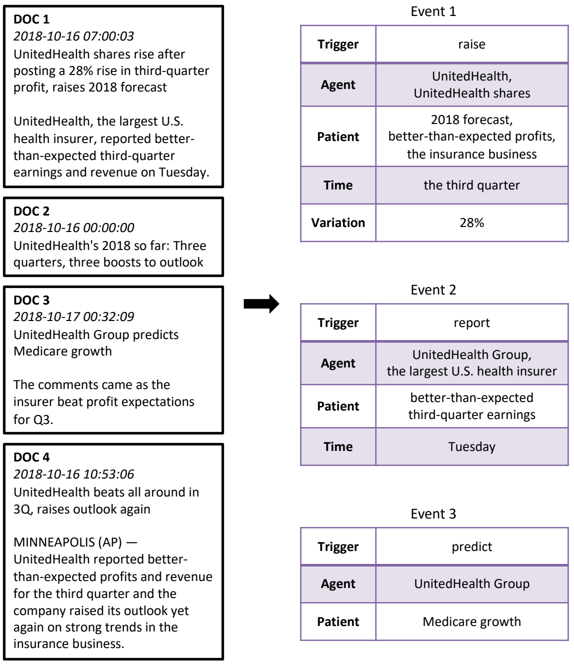
Figure 5: T-SNE visualization results of the latent event type vectors in the test set with colored labels produced by spectral clustering.

of *ODEE-F* is comparable to that of *Nguyen et al. (2015)* ($p = 0.3415$), which again demonstrates that the contextual features are a strong alternative to discrete features. The scores of *ODEE-FE* ($p = 0.06$) and *ODEE-FER* ($p = 10^{-5}$) are both higher than that of *ODEE-F*, which proves that the latent event type is critical in ODEE.

5.4 Latent Event Type Analysis

We are interested in learning how well the latent event type vectors can be modeled. To this end, for each news cluster in *GNBusiness-Dev*, we use our inference network in Figure 3 to calculate the mean μ for the latent event type vector t. T-SNE transformation (Maaten and Hinton, 2008) of the mean vectors are shown in Figure 5. Spectral clustering is further applied, and the number of clusters is chosen by the Calinski-Harabasz Score (Caliński and Harabasz, 1974) in grid search.

In Figure 5, there are four main clusters marked in different colors. Representative titles of news reports are shown as examples. We find that the vectors show salient themes for each main cluster. For example, the red cluster contains news reports about rise and drop of stocks such as *Netflix shares surge*, *IBM drops*, *Intel shares gain*, etc; the news reports in the purple cluster are mostly about product related activities, such as *Boston Dynamics' reveals its robodog Spot dancing*, *Arby's will debut sous vide duck sandwich*, *Wendy's Offering $1 Any Size Fry*, etc. The green cluster and the orange cluster are also interpretable. The former is about organization reporting changes, while the latter is about service related activities.



DOC 1
2018-10-16 07:00:03
UnitedHealth shares rise after posting a 28% rise in third-quarter profit, raises 2018 forecast

UnitedHealth, the largest U.S. health insurer, reported better-than-expected third-quarter earnings and revenue on Tuesday.

DOC 2
2018-10-16 00:00:00
UnitedHealth's 2018 so far: Three quarters, three boosts to outlook

DOC 3
2018-10-17 00:32:09
UnitedHealth Group predicts Medicare growth

The comments came as the insurer beat profit expectations for Q3.

DOC 4
2018-10-16 10:53:06
UnitedHealth beats all around in 3Q, raises outlook again

MINNEAPOLIS (AP) — UnitedHealth reported better-than-expected profits and revenue for the third quarter and the company raised its outlook yet again on strong trends in the insurance business.

Event 1

Trigger	raise
Agent	UnitedHealth, UnitedHealth shares
Patient	2018 forecast, better-than-expected profits, the insurance business
Time	the third quarter
Variation	28%

Event 2

Trigger	report
Agent	UnitedHealth Group, the largest U.S. health insurer
Patient	better-than-expected third-quarter earnings
Time	Tuesday

Event 3

Trigger	predict
Agent	UnitedHealth Group
Patient	Medicare growth

Figure 6: Extracted open domain events for *UnitedHealth shares rise*.

5.5 Case Study

We further use the news cluster *UnitedHealth shares rise* in Figure 5 for case study. Figure 6 shows the top-3 open-domain events extracted from the news cluster, where four input news reports are shown on the left and three system-generated events are shown on the right with mapped slots.

By comparing the plain news reports and the extracted events, we can see that the output events give a reasonable summary for the news cluster with three events triggered by “raise”, “report” and “predict”, respectively. Most of the slots are meaningful and closely related to the trigger, while covering most key aspects. However, this example also contains several incorrect slots. In the event 1, the slot “Variation” and its realization “28%” are only related to the entity “better-than-expected profits”, but there are three slot realizations in the event, which causes confusion. In addition, the slot “Aim” does not appear in the first event, whose realization should be “third-quarter profit” in document 1. The reason may be that we assemble an event only using entities with the same predicate, which introduces noise. Besides, due to the preprocessing errors in resolving coreference chains, some entity mentions are missing from the output.

There are also cases where one slot realization is semantically related to one trigger but eventually appears in a different event. One example is the entity "better-than-expected profits", which is related to the predicate word "report" but finally appears in the "raise" event. The cause can be errors propagated from parsing dependency trees, which confuse the syntactic predicate of the head word of an entity.

6 Conclusion

We presented the task of open domain event extraction, extracting unconstraint types of events from news clusters. A novel latent variable neural model was investigated, which explores latent event type vectors and entity mention redundancy. In addition, GNBusiness dataset, a large-scale dataset annotated with diverse event types and explainable event schemas, is released along with this paper. To our knowledge, we are the first to use neural latent variable model for inducing event schemas and extracting events.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. We thank Kiem-Hieu Nguyen from Hanoi University of Science and Technology for providing source code and solving confusions for their work. We thank Katherine Keith from University of Massachusetts at Amherst for sharing valuable experiences on probabilistic models. This work is supported by National Natural Science Foundation of China No. 61751201, National Key Research and Development Plan No. 2016QY03D0602, Research Foundation of Beijing Municipal Science and Technology Commission No. Z181100008918002, the funding from Rxhui Inc[6] and China Scholarship Council No. 201806030142. The work is done when Xiao Liu is visiting Yue Zhang.

References

Natalie Ahn. 2017. Inducing event types and roles in reverse: Using function to discover theme. In *Proceedings of the Events and Stories in the News Workshop@ACL 2017*, pages 66–76.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90.

Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, pages 438–441.

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*, pages 389–398.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 976–986.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 167–176.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. 2012. Discover breaking events with popular hashtags in twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1794–1798.

[6] https://rxhui.com

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Laurent Dinh and Vincent Dumoulin. 2016. Training neural bayesian nets. Technical report.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 66–71.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen R. McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and the 21st International Conference on Computational Linguistics*, pages 207–214.

Yu Hong, Jianfeng Zhang, Bin Ma, Jian-Min Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *roceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1127–1136.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 258–268.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295.

Georgiana Ifrim, Bichen Shi, and Igor Brigadir. 2014. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference*, pages 33–40.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 254–262.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the 2014 International Conference on Learning Representations*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Giridhar Kumaran and James Allan. 2005. Using names and topics for new event detection. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing*, pages 121–128.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 155–164.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 73–82.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 680–688.

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 4865–4872.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2134–2143.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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*, pages 1626–1635.

Andrew James McMinn and Joemon M. Jose. 2015. Real-time entity-based event detection for twitter. In *Proceedings of the Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association*, pages 65–77.

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 49–57.

Sean Moran, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2016. Enhancing first story detection using word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 821–824.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 100–108.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 188–197.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2016a. A dataset for open event extraction in english. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 1939–1943.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016b. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 886–891.

Nikolaos Panagiotou, Cem Akkaya, Kostas Tsioutsiouliklis, Vana Kalogeraki, and Dimitrios Gunopulos. 2016. First story detection using entities and relations. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3237–3244.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

Karl Pichotta and Raymond J. Mooney. 2016. Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Yanxia Qin, Yue Zhang, Min Zhang, and Dequan Zheng. 2013. Feature-rich segment-based news event detection on twitter. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 302–310.

Yanxia Qin, Yue Zhang, Min Zhang, and Dequan Zheng. 2017. Semantic-frame representation for event detection on twitter. In *Proceedings of the 2017 International Conference on Asian Language Processing*, pages 264–267.

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2008. Modeling context in scenario template creation. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 157–164.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Roi Reichart and Regina Barzilay. 2012. Multi-event extraction guided by global constraints. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 70–79.

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*, pages 1104–1112.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860.

Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Joint learning templates and slots for event schema induction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 428–434.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 5916–5923.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 304–311.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *Proceedings of the 2017 International Conference on Learning Representations*.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Beth Sundheim. 1992. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the 4th Conference on Message Understanding*, pages 3–21.

Jeroen B. P. Vuurens and Arjen P. de Vries. 2016. First story detection using multiple nearest neighbors. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 845–848.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 587–596.