Open Domain Event Extraction Using Neural Latent Variable Models

Xiao Liu¹,² and Heyan Huang¹,² and Yue Zhang³,⁴∗
¹School of Computer Science and Technology, Beijing Institute of Technology
²Zhejiang Lab, China
{xiaoliu,hhy63}@bit.edu.cn
³School of Engineering, Westlake University
⁴Institute 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.

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...
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; Irим 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-

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 \( \{ d_i \in c \} | i = 1, \ldots, M_c \} \) in the news cluster \( c \) focusing on the same event \( \mathcal{E}_c \). The output is a pair \( (\mathcal{E}_c, \mathcal{T}_c) \), where \( \mathcal{E}_c \) is the aforementioned set of open-domain events and \( \mathcal{T}_c \) 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-All. For this paper, we trim the news reports in each news cluster by keeping the title and first paragraph, releasing as GNBusiness-Full-Text. 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.

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 participates 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 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.
(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 multivariate normal distribution whose covariance matrix is a diagonal matrix. We mark all the parameters (mean vectors and diagonal vectors of covariance matrices) 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

\[
p_{\alpha,\theta,\lambda}(e) = p(s) \times p_{\lambda}(h|s) \times p_{\beta}(f|s) \tag{1}\]

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.

![Diagram of ODEE-F models](image-url)

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**

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

**Algorithm 2 ODEE-FE**

1: for each news cluster $c \in \mathcal{C}$ do
2: Sample a latent event type vector $t \sim \text{Normal}(\alpha)$
3: for each entity $e \in \mathcal{E}_c$ do
4: Sample a slot $s \sim \text{Multinomial}(\text{MLP}(t; \theta))$
5: Sample a head $h \sim \text{Multinomial}(1, \lambda_s)$
6: Sample a feature vector $f \sim \text{Normal}(\beta_s)$
7: end for
8: 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 $\alpha$. We then use $t$ and a multi-layer perceptron (MLP) with parameters $\theta$ 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

\[
p_{\alpha,\theta,\lambda}(c) = p_{\alpha}(t) \times \prod_{e \in \mathcal{E}_c} p_{\theta}(s|t) \times p_{\lambda}(h|s) \times p_{\beta}(f|s) \tag{2}\]

### 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-
tion 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 \textit{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

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

(3)

4.4 Inference

We now consider two tasks for \textit{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 \textit{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:

\[
\log p_{\alpha,\beta',\gamma,\theta,\lambda}(c) = \log \int_t \prod_{e \in E_c} p_{\lambda}(h | t) p_{\beta',\theta}(f' | t) p_{\alpha}(t) \, dt
\geq \text{ELBO}_\alpha(\alpha, \beta', \theta, \lambda, \omega)
\]

\[
= \mathbb{E}_{q_\omega(t)} \log p_{\beta',\theta,\lambda}(c | t) - D_{\text{KL}}[q_\omega(t) || p_\alpha(t)]
\]

(4)

where \( D_{\text{KL}}[q_\omega || p_\alpha] \) is the KL divergence between the variational posterior \( q_\omega \) and the prior \( p_\alpha \). 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 \( \omega \), 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 \( \mu \) and the variance vector \( \sigma^2 \) of \( q_\omega(t) \).

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 maximize 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

\[
p_{\beta',\theta,\lambda}(s | e, t) \propto p_{\beta',\theta,\lambda}(s, h, f', t)
= p_\theta(s | t) \times p_\lambda(h | s) \times p_\beta(f' | s)
\]

(5)
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 \texttt{noun.ACT} or \texttt{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 \( E_c \).

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 \textit{precision}, \textit{recall} and \textit{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 rightmost 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 \textit{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 \textit{slot coherence}.

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

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

where \( \text{NPMI}(w_i, w_j) = \frac{\log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)}}{-\log(p(w_i, w_j) + \epsilon)} \)

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

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 running 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 effectiveness 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

---

**Table 4:** Overall performance of schema matching.

<table>
<thead>
<tr>
<th>Method</th>
<th>Schema Matching (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nguyen et al. (2015)</td>
<td>41.5</td>
</tr>
<tr>
<td>Clustering</td>
<td>41.2</td>
</tr>
<tr>
<td>ODEE-F</td>
<td>41.7</td>
</tr>
<tr>
<td>ODEE-FE</td>
<td>42.4</td>
</tr>
<tr>
<td>ODEE-FER</td>
<td>43.4</td>
</tr>
</tbody>
</table>

**Table 5:** Averaged slot coherence results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ave Slot Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nguyen et al. (2015)</td>
<td>0.10</td>
</tr>
<tr>
<td>ODEE-F</td>
<td>0.10</td>
</tr>
<tr>
<td>ODEE-FE</td>
<td>0.16</td>
</tr>
<tr>
<td>ODEE-FER</td>
<td>0.18</td>
</tr>
</tbody>
</table>

---

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

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® and China Scholarship Council No. 201806030142. The work is done when Xiao Liu is visiting Yue Zhang.

References


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.


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.