# Semi-supervised New Event Type Induction and Event Detection

**Lifu Huang and Heng Ji**
Computer Science Department
University of Illinois at Urbana-Champaign
{lifuh2,hengji}@illinois.edu

## Abstract

Most previous event extraction studies assume a set of target event types and corresponding event annotations are given, which could be very expensive. In this paper, we work on a new task of semi-supervised event type induction, aiming to automatically discover a set of unseen types from a given corpus by leveraging annotations available for a few seen types. We design a Semi-Supervised Vector Quantized Variational Autoencoder framework to automatically learn a discrete latent type representation for each seen and unseen type and optimize them using seen type event annotations. A variational autoencoder is further introduced to enforce the reconstruction of each event mention conditioned on its latent type distribution. Experiments show that our approach can not only achieve state-of-the-art performance on supervised event detection but also discover high-quality new event types. [1]

## 1 Introduction

Event extraction is a task of automatically identifying and typing event trigger words (Event Detection), and extracting participants for each trigger (Argument Extraction) from natural language text. Traditional event extraction studies (Ji and Grishman, 2008; McClosky et al., 2011; Li et al., 2013; Chen et al., 2015; Yang and Mitchell, 2016; Liu et al., 2018; Nguyen and Nguyen, 2019; Lin et al., 2020; Li et al., 2020) usually assume there exists a set of predefined event types and argument roles, so that supervised machine learning models, e.g., deep neural networks, can be employed to extract events for each type based on human annotations. However, in practice, it is usually very expensive and time-consuming to manually craft an event schema, which defines the types and complex templates of the expected events. Moreover, the coverage of manually crafted schemas is often very low, making them fail to generalize to new scenarios.

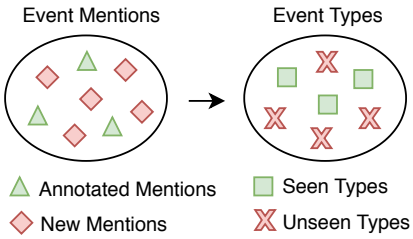


Figure 1: Semi-supervised new event type induction: discovering a set of new event types and their event mentions given the annotations for a few seen types.

Recent studies have shown that it's possible to automatically induce an event schema from raw text. Some researchers explore probabilistic generative methods (Chambers, 2013; Nguyen et al., 2015; Yuan et al., 2018; Liu et al., 2019) or ad-hoc clustering-based algorithms (Huang et al., 2016) to discover a set of event types and argument roles. Several studies (Huang et al., 2018; Lai and Nguyen, 2019) also explore zero-shot and few-shot learning approaches to leverage available resources and extend event extraction to new types. Generally, event schema induction can be divided into two steps: event type induction, aiming to discover a set of new event types for the given scenario, and argument role induction which discovers a set of argument roles for each type. In this work, we focus on tackling the first problem only.

We propose a task of semi-supervised event type induction, which is shown in Figure 1 and aims to leverage available event annotations for a few types, which are called as *seen* types, and automatically discover a set of new *unseen* types, as well as their corresponding event mentions. As a solution, we design a new Semi-supervised Vector Quantized Variational Autoeocoder framework (short as **SS-VQ-VAE**) which first assigns a discrete latent type

[1] The programs are publicly available for research purpose at https://github.com/wilburOne/SSVQVAE



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 718–724,
November 16–20, 2020. 

representation for each seen and unseen type, and optimizes them during the process of projecting each candidate trigger into a particular seen or unseen type. The candidate triggers are discovered with a heuristic approach.

Experiments under the setting of both supervised event detection and new event type induction demonstrate that our approach can not only detect event mentions for seen types with high precision, but also discover high-quality new unseen types.

## 2 Approach

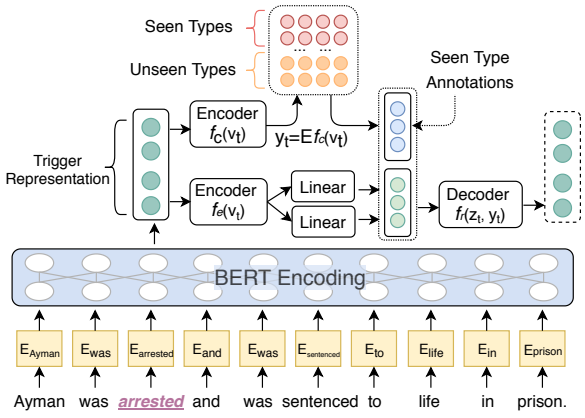


Figure 2: Architecture Overview.

As Figure 2 shows, given an input sentence, we first automatically discover all candidate triggers and encode each trigger with a contextual vector using a pre-trained BERT (Devlin et al., 2019) encoder. Then, we predict the type of each candidate trigger by looking up a dictionary of discrete latent representations of all seen and unseen types. Meanwhile, to avoid the type prediction to be over-fitted to seen types, we apply a variational autoencoder (VAE) as a regularizer to first project each trigger into a latent variational embedding and then reconstruct the trigger conditioned on its type distribution.

### 2.1 Event Trigger Identification

Similar to (Huang et al., 2016), we identify all candidate triggers based on word sense induction. Specifically, for each word, we disambiguate its senses and link each sense to OntoNotes (Hovy et al., 2006) using a word sense disambiguation system — IMS (Zhong and Ng, 2010) [2]. We consider all noun and verb concepts that can be mapped to OntoNotes senses as candidate triggers. In addition, the concepts that can be matched with verbs or nominal lexical units in FrameNet (Baker et al., 1998) are also considered as candidate triggers.

### 2.2 Trigger Representation Learning

Given a sentence $s = [w_1, ..., w_n]$, where we assume $w_i$ is identified as a candidate trigger, we use a pre-trained BERT encoder to encode the whole sentence and get a contextual representation for $w_i$. If $w_i$ can be split into multiple subwords or words, we use the average of all subword vectors as the final trigger representation.

### 2.3 Event Type Prediction with Vector Quantization

To predict a type for a candidate trigger, an intuitive approach is to learn a classifier using the event annotations of seen types. However, as we also aim to discover a set of unseen types, without any annotations, the classifier for the unseen types cannot be optimized.

To solve this problem, we employ a Vector Quantization (Gersho and Gray, 2012) strategy. We first define a discrete latent event type embedding space $\boldsymbol{E} \in \mathrm{R}^{k \times d}$, where $k$ is the number of candidate event types, and $d$ is the dimensionality of each type embedding $\boldsymbol{e}_i$. Each $\boldsymbol{e}_i$ can be viewed as the *centroid* of the triggers belonging to the corresponding event type. For each seen type, we initialize $\boldsymbol{e}$ with the contextual vector of a trigger which is randomly selected from the corresponding annotations. For each unseen type, we initialize $\boldsymbol{e}$ with the contextual vector of another trigger which is randomly picked from all unannotated event mentions. Assuming there are $m$ seen types, we arbitrarily assign $\boldsymbol{E}^{[1:m]}$ as their type representations.

Given a candidate trigger $t$ and its contextual vector $\boldsymbol{v}_t$, we first apply a linear encoder $f_c(\boldsymbol{v}_t) \in \mathrm{R}^d$ to extract type-specific features. Then, we compute a type distribution $\boldsymbol{y}$ based on $f_c(\boldsymbol{v}_t)$ by looking up all the discrete latent event type embeddings with inner-product operation

$$\boldsymbol{y}_t = \mathrm{Softmax}(\boldsymbol{E}^{[1:k]} \cdot f_c(\boldsymbol{v}_t)) \quad (1)$$

The feature encoder $f_c(.)$ is optimized using all event annotations for seen types (the cross-entropy term in Equation 2) and event mentions for unseen types (the second term in Equation 2 [3]). The intuition of the second term in Equation 2 is that, for each new event mention, we don't know the correct type but we know that the type must be from

[2] We use the OntoNotes based IMS word sense disambiguator (https://github.com/c-amr/camr)

[3] We only apply this term when we know the new event mentions do not belong to any seen types

a set of unseen types, so we maximize the margin between the probability of the most likely unseen type and the highest probability of the incorrect seen type.

$$\mathcal{L}_c = \sum_{(t,\tilde{y}_t)\in D_s} -\tilde{\boldsymbol{y}}_t \log(\boldsymbol{y}_t) + \sum_{t\in D_u} \max(\boldsymbol{y}_t^{[1:m]}) - \max(\boldsymbol{y}_t^{[m:k]}) \quad (2)$$

where $-\tilde{y}_t$ is the ground truth label. $D_s$ and $D_u$ denote the set of annotated event mentions for seen types and new event mentions for unseen types. $\boldsymbol{y}_t^{[1:m]}$ and $\boldsymbol{y}_t^{[m:k]}$ are the type prediction scores for seen and unseen types respectively.

To optimize the type embeddings $\boldsymbol{E}$, we follow the VQ objective (van den Oord et al., 2017) and use $l_2$ error to move the type vector $\boldsymbol{e}_i$ towards the type-specific feature $f_c(\boldsymbol{v}_t)$ (the first term in Equation 3) while $\boldsymbol{e}_i$ of $t$ is determined by $\boldsymbol{y}_t$ . To make sure $f_c(.)$ commits to an embedding, we add a commitment loss (the second term in Equation 3)

$$\mathcal{L}_{vq} = ||\text{sg}(f_c(\boldsymbol{v}_t)) - \boldsymbol{e}_i||^2 + ||f_c(\boldsymbol{v}_t) - \text{sg}(\boldsymbol{e}_i)||^2 \quad (3)$$

where sg stands for the stop gradient operator to make its operand to be a non-updated constant. The output of sg is the same as the input in the forward pass, and it is zero when computing gradients in the training process.

### 2.4 Variational Autoencoder as Regularizer

To avoid the type prediction to be over-fitted to the seen types, we employ a semi-supervised variational autoencoder as a regularizer. The intuition is that each event mention can be generated conditioned on a latent variational embedding $\boldsymbol{z}$ and its corresponding type distribution $\boldsymbol{y}$, which is predicted by the approach described in Section 2.3.

We first describe the semi-supervised variational inference process. It consists of an inference network $q(z|t)$ which is a posterior of the learning of a latent variable $z$ given the trigger $t$, and a generative network $p(t|z, y)$ to reconstruct the candidate trigger $t$ from the latent variable $z$ and type information $y$. For each candidate trigger $t$ with human annotated label $y$, the likelihood $p(t, y)$ can be approximated to a variational lower bound

$$\log p(t, y) \geq \log p(t|y, z) - KL(q(z|t)||p(z)) = -\mathcal{L}(t, y)$$

where $\log p(t|z, y)$ is the expectation of reconstruction of $t$ conditioned on $z$ and $y$, $p(z)$ is the prior Gaussian distribution. For each unlabeled candidate trigger $t$, the likelihood $p(t)$ approximates to another variational lower bound

$$\log p(t) \geq \sum_y q(y|t)(-\mathcal{L}(t, y)) - q(y|t) \log q(y|t) = -\mathcal{L}(t)$$

where $q(y|t)$ is obtained from Equation 1.

As for model implementation, given a candidate trigger $t$ and its contextual embedding $\boldsymbol{v}_t$, we first pass it through an encoder $f_e(\boldsymbol{v}_t)$ to extract features. As we assume the latent variational embedding $\boldsymbol{z}_t$ follows Gaussian distribution $\boldsymbol{z}_t \sim \boldsymbol{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$, we apply two linear functions to obtain the mean vector $\boldsymbol{\mu}_t = f_\mu(f_e(\boldsymbol{v}_t))$ and a variance vector $\boldsymbol{\sigma}_t = f_\sigma(f_e(\boldsymbol{v}_t))$. For decoding, we employ another linear function to reconstruct $\boldsymbol{v}_t$ from the concatenation of $\boldsymbol{z}_t$ and $\boldsymbol{y}_t$: $\boldsymbol{v}_t^{'} = f_r([\boldsymbol{z}_t : \boldsymbol{y}_t])$. We optimize the following objective for the semi-supervised VAE

$$\mathcal{L}_v = \sum_{t\in D_u} \mathcal{L}(t) + \sum_{(t,y)\in D_s} \mathcal{L}(t, y) \quad (4)$$

The overall loss function for optimizing the whole **SS-VQ-VAE** framework is

$$\mathcal{L} = \alpha\mathcal{L}_c + \beta\mathcal{L}_{vq} + \gamma\mathcal{L}_v \quad (5)$$

where $\alpha$, $\beta$ and $\gamma$ are hyper-parameters to balance these three objectives.

## 3 Experiments and Results

### 3.1 Dataset

We perform experiments on Automatic Content Extraction (ACE) 2005 dataset and evaluate our approach under two settings: (1) supervised event extraction, where the target types include 33 ACE predefined types and *other*, thus $k$ is set as 34. Giving all candidate triggers, the goal is to correctly identify all ACE event mentions and classify them into corresponding types. We follow the same data split with prior work (Li et al., 2013; Nguyen et al., 2016) in which 529/30/40 newswire documents are used for training/dev/test set. (2) new event type induction, where we follow a previous study (Huang et al., 2018) and use top-10 most popular event types from ACE05 data as seen and the remaining 23 types as unseen. Given all ACE annotated event mentions, the goal of this task is to test whether the approach can automatically discover the remaining 23 unseen ACE types and categorize each candidate trigger into a particular seen or unseen type. In this experiment, $k$ is set as 500.

In terms of implementation details, we use the pre-trained bert-large-cased [4] model for fine-tuning,

[4] https://github.com/google-research/bert

| Methods | Encoder | Trigger Identification | | | Trigger Detection | | |
|---|---|---|---|---|---|---|---|
| | | P | R | F | P | R | F |
| DMCNN (Chen et al., 2015) | CNN | 80.4 | 67.7 | 73.5 | 75.6 | 63.6 | 69.1 |
| JRNN (Nguyen et al., 2016) | RNN | 68.5 | 75.7 | 71.9 | 66.0 | 73.0 | 69.3 |
| JMEE (Liu et al., 2018) | GCN | 80.2 | 72.1 | 75.9 | 76.3 | 71.3 | 73.7 |
| Joint3EE (Nguyen and Nguyen, 2019) | GRU | 70.5 | 74.5 | 72.5 | 68.0 | 71.8 | 69.8 |
| MOGANED (Yan et al., 2019) | GAN | - | - | - | 79.5 | 72.3 | 75.7 |
| BERT-CRF | BERT | 73.8 | 76.9 | 75.3 | 70.4 | 73.3 | 71.8 |
| DMBERT (Wang et al., 2019) | BERT | - | - | - | 77.6 | 71.8 | 74.6 |
| SS-VQ-VAE w/o VQ-VAE | BERT | 78.2 | 77.8 | 78.0 | 73.2 | 72.9 | 73.0 |
| SS-VQ-VAE w/o VAE | BERT | 80.8 | 80.2 | **80.5** | 76.2 | 75.7 | 75.9 |
| SS-VQ-VAE | BERT | 79.1 | 81.4 | 80.2 | 75.7 | 77.8 | **76.7** |

Table 1: Supervised Event Detection Performance on ACE 2005 (F-score %).

| Metrics | Normalized Mutual Info | Fowlkes Mallows | Completeness | Homogeneity | V-Measure |
|---|---|---|---|---|---|
| BERT C-Kmeans | 8.93 | 6.04 | 8.64 | 9.22 | 8.92 |
| SS-VQ-VAE w/o VAE | 33.45 | 25.54 | 42.76 | 26.17 | 32.47 |
| SS-VQ-VAE | 40.88 | 31.46 | 53.57 | 31.19 | 39.43 |

Table 2: Evaluation of New Event Type Induction on 23 Unseen Types of ACE 2005 (%).

and optimize our model with BertAdam. we optimize the parameters with grid search: training epoch 15, learning rate $l \in \{1e-5, 2e-5, 3e-5, 5e-5\}$, gradient accumulation steps $g \in \{1, 2, 3\}$, training batch size $b \in \{5g, 8g, 10g\}$, the hyper-parameters for the overall loss function $\alpha \in \{1.0, 5.0, 10.0\}$, $\beta \in \{0.1, 0.5, 1.0\}$, $\gamma \in \{0.1, 0.5, 1.0\}$. The dimensionality of type embedding as well as latent variational embedding, and the hidden states of $f_c(.)$ are all 500 while the hidden states of $f_e(.)$, $f_\mu(.)$, $f_\sigma(.)$ are all 1024.

### 3.2 Supervised Event Detection

Table 1 compares our approach with several baselines. We conduct ablation study to testify the impact of the VQ and VAE components: **SS-VQ-VAE w/o VQ-VAE** is only optimized with the classification loss (Equation 2) while **SS-VQ-VAE w/o VAE** is optimized with the classification loss (Equation 2) and the VQ objective (Equation 3).

As we can see, BERT based approaches generally outperform the methods using CNN, RNN or GRU. Our approach achieves the state-of-the-art among all methods. In particular, the recall of our approach is much higher than other methods, which demonstrate the effectiveness of the trigger identification step. It can narrow the learning space of the model. The ablation studies also prove the effectiveness of the VQ and VAE components.

### 3.3 New Event Type Induction

For new event type induction, we compare our approach with another intuitive baseline, **BERT-C-Kmeans**, which takes in the BERT based trigger representations and group all candidate triggers into clusters with a Constrained K-means (Wagstaff et al., 2001), a semi-supervised clustering algorithm which enforces all trigger candidates annotated with the same seen type to belong to the same cluster. Table 2 shows the performance with several clustering metrics (Chen and Ji, 2010), which measure the agreement between the ground truth class assignment and system based unseen type prediction.

**Normalized Mutual Info** is a normalization of the Mutual Information (MI) score and scales the MI score to be between 0 and 1.

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

where $Y$ denotes the ground truth class labels, $C$ denotes the cluster labels, $H(.)$ denotes the entropy function and $I(Y; C)$ is the mutual information between $Y$ and $C$.

**Fowlkes Mallows** (Fowlkes and Mallows, 1983) is to evaluate the similarity between the clusters obtained from our approach and ground-truth labels of the data.

$$FM(Y, C) = \frac{TP}{\sqrt{((TP + FP) \times (TP + FN))}}$$

where $TP$ means True Positive, which is calculated as the number of data point pairs that are in the same cluster in $Y$ and in $C$. $FP$ refers to False Positive and it is calculated as the number of data point pairs that are in the same cluster in $Y$ but not in $C$. $FN$ is False Negative and it is calculated as

the number of pair of data points that are not in the same cluster in $Y$ but are in the same cluster in $C$.

**Completeness** : A clustering result satisfies completeness if all members of a given class are assigned to the same cluster.

$$C(Y,C) = 1 - \frac{H(C|Y)}{H(C)}$$

where $H(C|Y)$ is the conditional entropy of the clustering output given the class labels.

**Homogeneity** : A clustering result satisfies completeness if all of its clusters contain only data points which are members of a single class.

$$C(Y,C) = 1 - \frac{H(Y|C)}{H(Y)}$$

**V-Measure** (Rosenberg and Hirschberg, 2007) is the weighted harmonic mean between homogeneity score and completeness score.

$$V(Y,C) = \frac{(1+\beta)\cdot h \cdot c)}{(\beta \cdot h) + c}$$

where $h$ denotes the homogeneity score and $c$ refers to the completeness score.

As qualitative analysis, we further pick 6 unseen ACE types and randomly select at most 100 event mentions for each type. We visualize their type distribution $\boldsymbol{y}$ using TSNE [5]. As Figure 3 shows, most of the event mentions that are annotated with the same ACE type tends to be predicted to the same new unseen type.

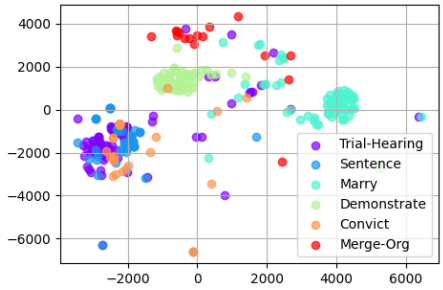


Figure 3: Type Distribution of 6 Unseen Types of Event Mentions.

## 4 Related Work

Traditional event extraction studies (Ji and Grishman, 2008; McClosky et al., 2011; Li et al., 2013; Chen et al., 2015; Yang and Mitchell, 2016; Feng et al., 2016; Liu et al., 2018; Nguyen and Nguyen, 2019; Lin et al., 2020; Li et al., 2020) assume all the target event types and annotations are given. They can extract high-quality event mentions for the given types, but cannot extract mentions for any new types. Recent studies (Huang et al., 2018; Chan et al., 2019; Ferguson et al., 2018) leverage annotations for a few seen event types or several keywords provided for the new types to extract mentions for new types. However, all these studies assume all the target types are given, which is very costly when moving to a new scenario.

Recent studies have explored probabilistic generative methods (Chambers, 2013; Nguyen et al., 2015; Yuan et al., 2018; Liu et al., 2019) or ad-hoc clustering based algorithms (Huang et al., 2016) to automatically discover a set of event types as well as argument roles. Most of these studies are completely unsupervised and mainly rely on statistical patterns or semantic matching, while our work tries to leverage the knowledge learned from available annotations to discover new event types.

## 5 Conclusion and Future Work

We have designed a semi-supervised vector quantized variational autoencoder approach which automatically learns a discrete representations for each seen and unseen type and predict a type for each candidate trigger. Experiments show that our approach achieves the state-of-the-art on supervised event extraction and discovers a set of high-quality unseen types. In the future, we will extend this approach to argument role induction to discover complete event schemas.

## Acknowledgement

This research is based upon work supported in part by U.S. DARPA KAIROS Program Nos. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014 and Air Force No. FA8650-17-C-7715. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

## References


Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

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.

Yee Seng Chan, Joshua Fasching, Haoling Qiu, and Bonan Min. 2019. Rapid customization for event extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36.

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 (Volume 1: Long Papers)*, pages 167–176.

Zheng Chen and Heng Ji. 2010. Graph-based clustering for computational linguistics: A survey. In *Proc. ACL 2010 TextGraphs5*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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 (Volume 2: Short Papers)*, pages 66–71.

James Ferguson, Colin Lockard, Daniel S Weld, and Hannaneh Hajishirzi. 2018. Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 359–364.

Edward B Fowlkes and Colin L Mallows. 1983. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569.

Allen Gersho and Robert M Gray. 2012. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media.

Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare 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 (Volume 1: Long Papers)*, pages 258–268.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: Hlt*, pages 254–262.

Viet Dac Lai and Thien Nguyen. 2019. Extending event detection to new types with learning from keywords. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 243–248.

Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020. Cross-media structured common space for multimedia event extraction. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

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 (Volume 1: Long Papers)*, pages 73–82.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint end-to-end neural model for information extraction with global features. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

Xiao Liu, He-Yan Huang, and Yue Zhang. 2019. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871.

Xiao Liu, Zhunchen Luo, and He-Yan Huang. 2018. 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.

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

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 (Volume 1: Long Papers)*, pages 188–197.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. 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.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6851–6858.

Aaron van den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. 2001. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5770–5774.

Bishan Yang and Tom 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.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.