Distilling Discrimination and Generalization Knowledge for Event Detection via Δ-Representation Learning

Yaojie Lu[1,3], **Hongyu Lin**[1,3], **Xianpei Han**[1,2]*, **Le Sun**[1,2]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
{yaojie2017,hongyu2016,xianpei,sunle}@iscas.ac.cn

Abstract

Event detection systems rely on discrimination knowledge to distinguish ambiguous trigger words and generalization knowledge to detect unseen/sparse trigger words. Current neural event detection approaches focus on trigger-centric representations, which work well on distilling discrimination knowledge, but poorly on learning generalization knowledge. To address this problem, this paper proposes a **Δ**-learning approach to distill discrimination and generalization knowledge by effectively decoupling, incrementally learning and adaptively fusing event representation. Experiments show that our method significantly outperforms previous approaches on unseen/sparse trigger words, and achieves state-of-the-art performance on both ACE2005 and KBP2017 datasets.

1 Introduction

Event detection (ED) aims to identify triggers of specific event types. For instance, an ED system will identify *fired* as an Attack event trigger in the sentence "*An American tank fired on the Palestine Hotel*." Event detection plays an important role in Automatic Content Extraction (Ahn, 2006), Information Retrieval (Allan, 2012), and Text Understanding (Chambers and Jurafsky, 2008).

Due to the ambiguity and the diversity of natural language expressions (Li et al., 2013; Nguyen and Grishman, 2015), an effective approach should be able to distill both discrimination and generalization knowledge for event detection. Discrimination knowledge aims to distinguish ambiguous triggers in different contexts. As shown in Figure 1, to identify *fired* in S4 as an EndPosition trigger rather than an Attack trigger, an ED system needs to distill the discrimination knowledge from S1 and S2 that (*fired*, Attack) usually co-occurs with {tank, death, enemy, ...} and (*fired*, EndPosition) usually co-occurs with {work, fault, job, ...}. Unlike discrimination knowledge, generalization knowledge aims to detect unseen or sparsely labeled triggers, thus needs to be transferred between different trigger words. For example, to identify the unseen word *hacked* in S5 as an Attack trigger, an ED system needs to distill the generalized Attack pattern "*[Trigger] to death*" from S3.

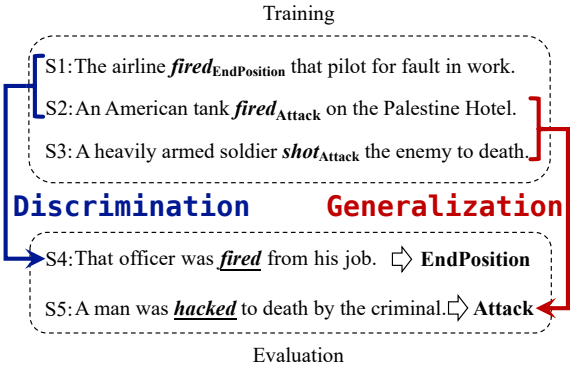


Figure 1: Examples of event instances. Identifing ambiguous word *fired* requires discrimination knowledge and identifing unseen word *hacked* requires generalization knowledge.

Currently, most neural network ED methods (Chen et al., 2015; Nguyen and Grishman, 2015, 2016; Duan et al., 2017; Yang and Mitchell, 2017) work well on distilling discrimination knowledge, but poorly on distilling generalization knowledge. Table 1 shows the performances of several models on both sparsely (OOV/OOL) and densely (Other) labeled trigger words. These models work well on densely labeled trigger words, i.e., they have a good discrimination ability. But they perform poorly on unseen/sparsely labeled trigger words, i.e., they have a poor generalization ability. This is because these approaches are mostly trigger-centric, thus hard to be generalized well to sparse/unseen words. Furthermore, the lack of

*Corresponding Author

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4366–4376
Florence, Italy, July 28 - August 2, 2019.

Models	OOV	OOL	Other
DMCNN	34.3	8.8	76.1
Bi-LSTM	35.3	9.3	75.5
ELMo	31.3	9.0	75.7

Table 1: F1 Scores of previous approaches on different types of triggers (ACE2005), where OOV words are the out-of-vocabulary words in the training corpus, OOL words are the out-of-label words, i.e., an instance whose (word, event type) never occurs in the training corpus but the word is not OOV. DMCNN (Chen et al., 2015) refers to dynamic multi-pooling based CNN; Bi-LSTM (Duan et al., 2017) refers to bidirectional LSTM based RNN. ELMo refers to the fixed task-independent word representations proposed by Peters et al. (2018).

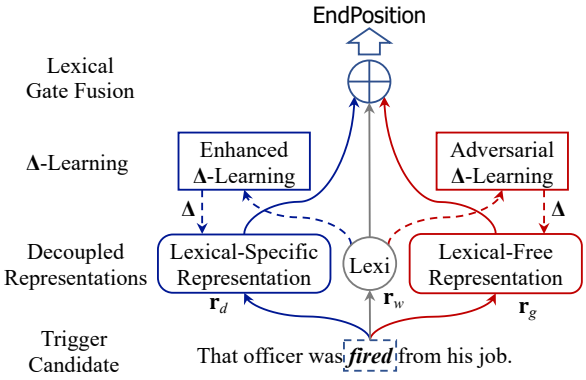


Figure 2: The framework of our Δ-learning approach. Dashed lines indicate the learning process; solid lines indicate the event detection process.

large-scale training data also limits the generalization ability of learned models. Table 1 also shows the performance of using general pre-trained word representation – ELMo (Peters et al., 2018). We can see that, this task-independent lexical-centric representation achieves nearly the same performance to task-specific representations.

In this paper, we propose a Δ-representation learning approach, which can incrementally distill both discrimination and generalization knowledge for event detection. Δ-representation learning aims to *decouple*, *learn*, and *fuse* alterable Δ-parts for event representation, instead of learning a single comprehensive representation. Specifically, we decouple an event representation $\mathbf{r}_{ed}$ into three parts $\mathbf{r}_{ed} = \mathbf{r}_w \bigoplus \mathbf{r}_d \bigoplus \mathbf{r}_g$ (Section 2), where $\mathbf{r}_w$ is the pre-trained word representation of trigger words, $\mathbf{r}_d$ is the lexical-specific event representation which captures discrimination knowledge for distinguishing ambiguous triggers, $\mathbf{r}_g$ is the lexical-free event representation which captures generalization knowledge for detecting unseen/sparse triggers, and $\bigoplus$ is the fusion function to fuse different parts. Here $\mathbf{r}_d$ and $\mathbf{r}_g$ are the Δ-parts of our representation, i.e., they are independently learned starting from $\mathbf{r}_w$ and are intended for capturing incremental knowledge for event detection. To incrementally learn the Δ-parts $\mathbf{r}_d$ and $\mathbf{r}_g$, we propose a Δ-learning framework (Section 3), i.e., a lexical enhanced Δ-learning algorithm is designed to learn the discrimination knowledge $\mathbf{r}_d$ which is both event-related and lexical-relevant part, and a lexical adversarial Δ-learning is designed to learn the generalization knowledge $\mathbf{r}_g$ which is event-related but lexical-irrelevant part. Finally, a lexical gate fusion mechanism $\bigoplus$ (Section 2.3) is proposed to adaptively fuse these learned representations. Figure 2 shows the architecture of our method.

We conduct experiments[1] on two standard event detection datasets: ACE2005[2] and TAC KBP 2017 Event Nugget Detection Evaluation[3] (KBP2017). Experimental results show that the proposed method significantly improves the performance on sparsely labeled triggers, and retains a high performance on densely labeled triggers.

The main contributions of this paper are:

1. We propose a new representation learning framework - Δ-learning, which can incrementally distill both discrimination and generalization knowledge during representation learning. Since the ambiguity and the diversity problem of natural language expressions are common in NLP, our framework can potentially benefit many other NLP tasks.

2. We design a new event detection approach. By effectively decoupling, independently learning, and adaptively fusing event representation, our approach works well on both sparsely and densely labeled triggers and achieves the state-of-the-art performance on both ACE2005 and KBP2017 datasets.

2 Decoupling Lexical-Specific and Lexical-Free Representations for Event Detection

To distill both discrimination and generalization knowledge, this section decouples event represen-

[1]Our source code is openly available at https://www.github.com/luyaojie/delta-learning-for-ed.

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

[3]https://tac.nist.gov/2017/KBP/data.html

tation into three parts: $\mathbf{r}_{ed} = \mathbf{r}_w \bigoplus \mathbf{r}_d \bigoplus \mathbf{r}_g$, where $\mathbf{r}_w$ is the word representation of trigger words, such as word embeddings/ELMo (noted that $\mathbf{r}_w$ is fixed during all our training process); $\mathbf{r}_d$ is a lexical-specific event representation which captures discrimination knowledge; $\mathbf{r}_g$ is a lexical-free representation which captures generalization knowledge. By decoupling event representations, $\mathbf{r}_d$ and $\mathbf{r}_g$ will be independently learned using our $\mathbf{\Delta}$-learning algorithm in Section 3. Finally, a gate mechanism is proposed to adaptively fuse the above representations for event detection.

Formally, an event detection instance is a pair of trigger candidate and its context, i.e., x = (t, c), where t is a trigger candidate, and $c = \{c_{-m}, ..., c_{-1}, c_1, ..., c_m\}$ is its context. For example, (*fired*, "That officer was _ from his job.") is an instance for candidate *fired*.

Following previous work (Nguyen and Grishman, 2015; Liu et al., 2018a), given an instance x, we embed each token t_i as $\mathbf{t}_i = [\mathbf{p}_w; \mathbf{p}_p; \mathbf{p}_e]$, where $\mathbf{p}_w$ is its word embedding, $\mathbf{p}_p$ is its position embedding, and $\mathbf{p}_e$ is its entity tag embedding. Therefore $\mathbf{t}_0$ is the representation of trigger candidate. In this paper, lexical-specific model $\mathbf{\Theta}_d$ and lexical-free model $\mathbf{\Theta}_g$ use independent embeddings.

2.1 Lexical-Specific Representation

Lexical-specific representation aims to capture discriminative information for distinguishing ambiguous trigger words. For example, we want our representation to capture {officer, job, ...} clues for distinguishing (*fired*, EndPosition) from (*fired*, Attack), and {tank, soldiers, ...} for distinguishing (*fired*, Attack) from (*fired*, EndPosition).

To capture discriminative clues for trigger candidates, we design a lexical-centered context selection attention. And we refer it as ATT-RNN and describe it as follows.

Lexical-Centered Context Selection. To select discriminative context words, the attentive context selection mechanism models the association between the trigger candidate and its context words. For instance, we want our attention mechanism to capture the association between "work" and *fired* in S1, and between "tank" and *fired* in S2.

Concretely, we first feed $[\mathbf{t}_{-m}, ..., \mathbf{t}_0, ..., \mathbf{t}_m]$ into a bidirectional GRU to get all tokens' context-aware token encoding $[\mathbf{h}_{-m}, ..., \mathbf{h}_0, ..., \mathbf{h}_m]$. Then our attention mechanism models (trigger, context word) pair's relevance with a Multi-Layer Perceptron (MLP), and uses a softmax function normalizing relevance scores to attention weights:

$$\alpha_i = \frac{\exp(\text{MLP}([\mathbf{h}_0; \mathbf{h}_i]))}{\sum_{j \in c} \exp(\text{MLP}([\mathbf{h}_0; \mathbf{h}_j]))} \quad (1)$$

Given the attention weights, the lexical-specific context representation is summarized as $\mathbf{c}_0 = \sum_{i \in C} \alpha_i \cdot \mathbf{h}_i$. And the final lexical-specific representation of instance x is the concatenation of its token representation $\mathbf{h}_0$ and the lexical-specific context representation $\mathbf{c}_0$, i.e., $\mathbf{r}_d = [\mathbf{h}_0; \mathbf{c}_0]$.

The lexical-specific representation can effectively disambiguate trigger words by capturing (trigger, context word) associations. However, this representation is lexical-specific, thus hard to generalize well to sparse/unseen words.

2.2 Lexical-Free Representation

In contrast to lexical-specific representation, lexical-free event representation $\mathbf{r}_g$ aims to capture generalization knowledge for ED, which can be transferred between different trigger words. For example, we want to capture the trigger word-irrelevant knowledge such as "*[Trigger] to death*" being a strong trigger pattern for Attack event, which can be used to detect many different trigger words, such as *fired*, *hacked*, *beat*. In this way, even an unseen trigger candidate t can be easily identified by leveraging such knowledge.

Obviously, the lexical-free event representation $\mathbf{r}_g$ should be lexical-irrelevant, but event-specific. To this end, we represent all tokens in x as $\mathbf{t}_i$, then employ a lexical-independent context selection module for $\mathbf{r}_g$. We simply use DMCNN (Chen et al., 2015) as our lexical-independent context selection module, but design a new adversarial $\mathbf{\Delta}$-learning algorithm in Section 3.2 which can eliminate lexical-relevant information from $\mathbf{r}_g$.

Lexical-Independent Context Selection. To select lexical-independent but event-relevant context words, we employ the same CNN architecture as Chen et al. (2015). For instance, we want to capture "to death" and "criminal" being relevant for Attack event in S5.

Given token sequence $[\mathbf{t}_{-m}, ..., \mathbf{t}_0, ..., \mathbf{t}_m]$, a h-width convolutional layer captures local context feature $\mathbf{l}_i$ from $\mathbf{t}_{i:i+h-1}$: $\mathbf{l}_i = tanh(\mathbf{w} \cdot \mathbf{t}_{i:i+h-1} + \mathbf{b})$, where $\mathbf{w}$ is the convolutional filter, and $\mathbf{b}$ is the bias term. To summarize important signals from different pieces of a sentence, a dynamic pooling layer (Chen et al., 2015) is used to produce the left

and right context features $\mathbf{l}^{left}$, $\mathbf{l}^{right}$:

$$\mathbf{l}^{left} = \max_{j<0} \mathbf{l}_j, \mathbf{l}^{right} = \max_{j\geq 0} \mathbf{l}_j \quad (2)$$

Finally, we concatenate the left context feature $\mathbf{l}^{left}$ and the right context feature $\mathbf{l}^{right}$ as our lexical-free representation $\mathbf{r}_g = [\mathbf{l}^{left}; \mathbf{l}^{right}]$.

2.3 Lexical Gate Mechanism for Representation Fusion

The above two representations are complementary to each other: $\mathbf{r}_d$ captures discrimination knowledge, and $\mathbf{r}_g$ captures generalization knowledge. However, simple concatenation is not effective for event detection: for frequently labeled trigger words in training data, lexical-specific representation is more useful; and for sparsely labeled or unseen trigger words, lexical-free representation is more helpful. Based on this observation, our system needs to rely more on $\mathbf{r}_d$ to detect frequent candidate *fired*, but more on $\mathbf{r}_g$ to detect the OOV candidate *hacked*. That is, we need to adaptively fuse different representations for different words, rather than simply concatenate them.

To adaptively fuse lexical-specific representation $\mathbf{r}_d$, lexical-free representation $\mathbf{r}_g$ and word representation $\mathbf{r}_w$, we design a lexical gate mechanism to fuse different representations: $\mathbf{r}_{ed} = \mathbf{r}_w \bigoplus \mathbf{r}_d \bigoplus \mathbf{r}_g$, where $\bigoplus$ is the fusion gate, and $\mathbf{r}_{ed}$ is the final event representation. Concretely, we first map these representations to a universal space:

$$\begin{aligned} \mathbf{r'}_d &= f_{Spec\to U}(\mathbf{r}_d) \\ \mathbf{r'}_g &= f_{Free\to U}(\mathbf{r}_g) \\ \mathbf{r'}_w &= f_{Lexi\to U}(\mathbf{r}_w) \end{aligned} \quad (3)$$

where $f_{Spec\to U}(\cdot)$, $f_{Free\to U}(\cdot)$ and $f_{Lexi\to U}(\cdot)$ are linear layers with a nonlinear function; then we fuse them via the gated mechanism:

$$\begin{aligned} \tilde{\mathbf{g}}_i &= f_{U\to G}(\mathbf{r'}_i), i \in \{d, g, w\} \\ \mathbf{g}_i &= \frac{\exp(\tilde{\mathbf{g}}_i)}{\sum_{j\in\{d,g,w\}} \exp(\tilde{\mathbf{g}}_j)} \end{aligned} \quad (4)$$

$\mathbf{g}_i$ ($i \in \{d, g, w\}$) correspondingly indicates the confidence of the evidences provided by $\mathbf{r'}_s$, $\mathbf{r'}_f$ and $\mathbf{r'}_l$; $\mathbf{g}_i$ and $\tilde{\mathbf{g}}_i$ have the same dimensions as $\mathbf{r'}_i$; $f_{U\to G}(\cdot)$ is a linear layer with a nonlinear function. Finally, we combine all representations:

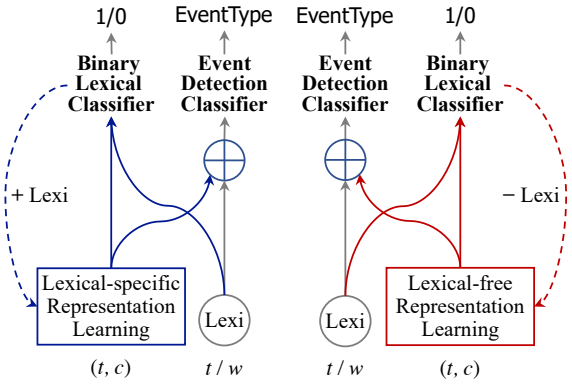
$$\mathbf{r}_{ed} = \mathbf{g}_d \odot \mathbf{r'}_d + \mathbf{g}_g \odot \mathbf{r'}_g + \mathbf{g}_w \odot \mathbf{r'}_w \quad (5)$$

where $\odot$ is element-wise multiplication.

After fusion, $\mathbf{r}_{ed}$ will be fed to the event detection classifier, which computes a classification probability for each event type y_t (including NIL for not a trigger):

$$P(y_t|x) = \frac{\exp(\mathbf{w}_t \cdot \mathbf{r}_{ed} + b_t)}{\sum_{t=1}^{T} \exp(\mathbf{w}_t \cdot \mathbf{r}_{ed} + b_t)} \quad (6)$$

where $\mathbf{w}_t$ is the weight vector, and b_t is the bias term. In this way, we identify trigger words of all pre-defined event types.



(a) Lexical-Enhanced (b) Lexical-Adversarial

Figure 3: The framework of our **Δ**-learning algorithms.

3 Distilling Discrimination and Generalization knowledge via Δ-Learning

This section describes our **Δ**-learning framework, which can learn lexical-specific representation $\mathbf{r}_d$ and lexical-free representation $\mathbf{r}_g$ independently. To distill discrimination knowledge to $\mathbf{r}_d$, we design a lexical-enhanced **Δ**-learning algorithm. To distill generalization knowledge to $\mathbf{r}_g$, we design a lexical adversarial **Δ**-learning algorithm. Finally, we fine-tune the full event detection model in Figure 2.

3.1 Distilling Discrimination Knowledge via Lexical-Enhanced Δ-Learning

This section describes our lexical-enhanced **Δ**-learning algorithm for lexical-specific representation $\mathbf{r}_d$. To ensure $\mathbf{r}_d$ be both event-relevant and lexical-specific, we use two types of supervision signals: first, we want the learned representation $\mathbf{r}_d$ can predict its event type y with the help of word representation $\mathbf{r}_w$; second, we want the learned $\mathbf{r}_d$ can also predict its trigger word t. For example, we want the learned $\mathbf{r}_d$ of the instance (*fired*, An solider _ to death) can predict both its event type Attack and its trigger word *fired*.

To achieve the above goal, we remove the lexical-free part in Figure 2 and show the lexical-enhanced $\mathbf{\Delta}$-learning framework in Figure 3 (a). The input of our lexical-enhanced learning framework is a triple (t, c, w), where t is the trigger, c is its context, and w is a sampled word. The output is two-fold: the event classifier will output the event type of (t, c), and the auxiliary lexical classifier will output 1 if $t = w$ and 0 otherwise. In this way, the event classifier can propagate the event type supervision signal to our lexical-specific representation learning component, and the auxiliary lexical binary classifier ensures that the learned representation $\mathbf{r}_d$ is lexical-specific.

Specifically, for each ED instance x = (t, c), we generate a positive lexical-enhanced training instance (t, c, t) with label $(y, 1)$, and n negative instances[4] (t, c, w) with label $(y, 0)$, where w is a word randomly sampled from context c.

For each ED instance $x = (t, c)$ in the train dataset $\mathcal{D}$, the event classifier loss is:

$$\mathcal{L}_{event} = -\sum_{(x_k, y_k)\in\mathcal{D}} \log P(y_k|x_k) \quad (7)$$

and the lexical binary classifier loss is:

$$\begin{aligned}\mathcal{L}_{lexical} = \sum_{x_k\in\mathcal{D}} &-\log P(1|x_k, t) \\ &-\sum_{j=1}^{n} \log P(0|x_k, w_{kj})\end{aligned} \quad (8)$$

Therefore, the loss function of lexical-enhanced $\mathbf{\Delta}$-learning is:

$$\mathcal{L}_{enhance} = \mathcal{L}_{event} + \mathcal{L}_{lexical} \quad (9)$$

By adding the auxiliary lexical classification task, this learning algorithm will ensure the learned representation be both event-related and lexical-relevant.

3.2 Distilling Generalization Knowledge via Lexical-Adversarial $\mathbf{\Delta}$-Learning

In contrast to lexical-specific representation $\mathbf{r}_d$, the lexical-free representation $\mathbf{r}_g$ needs to eliminate lexical-specific information, so that it can be transferred between different words. To achieve this goal, we adopt adversarial techniques and design a lexical-adversarial $\mathbf{\Delta}$-learning algorithm.

Specifically, we remove the lexical-specific part in Figure 2 and show the lexical-adversarial $\mathbf{\Delta}$-learning framework in Figure 3 (b). We can see that, the input and the output of our adversarial $\mathbf{\Delta}$-learning framework are still (t, c, w) and $(y, 1/0)$. The event classifier is used to propagate the event type supervision signal to our lexical-free representation learning component, so that $\mathbf{r}_g$ will capture event related information. The difference between Figure 3 (a) and 3 (b) is that they use different auxiliary tasks: Figure 3 (a) uses a lexical-enhanced auxiliary task, and Figure 3 (b) uses a lexical-adversarial auxiliary task.

To eliminate lexical-specific information, we design a two-player min-max game (Goodfellow et al., 2014) for the lexical-adversarial auxiliary task. Given (t, c, w), our binary lexical classifier $\mathbf{\Theta}_{DeLexi}$ attempts to predict whether $\mathbf{r}_g$ is specific to w, but the lexical-free model $\mathbf{\Theta}_g$ tries to produce $\mathbf{r}_g$ to confuse $\mathbf{\Theta}_{DeLexi}$. The min-max objective function for lexical-adversarial $\mathbf{\Delta}$-learning is:

$$\begin{aligned}\mathcal{L}_{minmax} = \sum_{x_k\in\mathcal{D}} &-\log P(1|x_k, t) \\ &-\sum_{j=1}^{n} \log P(0|x_k, w_{kj})\end{aligned} \quad (10)$$

$$\hat{\theta} = \min_{\mathbf{\Theta}_{DeLexi}} \max_{\mathbf{\Theta}_g} \mathcal{L}_{minmax}$$

In this way, we can remove the lexical-specific information from $\mathbf{r}_g$.

The above adversarial loss leads two different optimized directions for $\mathbf{\Theta}_g$ and $\mathbf{\Theta}_{DeLexi}$, which can be implemented by a gradient reversal layer (Ganin et al., 2016) during backpropagation. That is, $\mathcal{L}_{minmax}$ is jointly optimized with the main ED task objective $\mathcal{L}_{event}$, while gradients from adversarial loss are reversed with the factor λ_{adv} when they reach $\mathbf{r}_g$. By this means, we can unify the optimized directions of these components. Therefore, the loss function of our lexical-adversarial $\mathbf{\Delta}$-learning is:

$$\mathcal{L}_{adversary} = \mathcal{L}_{event} + \mathcal{L}_{minmax} \quad (11)$$

Following Liu et al. (2019), we divide the lexical-adversarial $\mathbf{\Delta}$-learning into two stages:

1. In the pretraining stage, we first update $\mathbf{\Theta}_g$ using the main ED task objective, then freeze $\mathbf{\Theta}_g$ and update $\mathbf{\Theta}_{DeLexi}$ using the Equation 10.

2. In the adversarial learning stage, we update parameters using Equation 11.

In practice, we find that the factor λ_{adv} is sensitive to the even of min-max game. A large λ_{adv} is easy to make the binary lexical classifier to be weak (the binary classfication accuracy tends to

[4] In this paper, we set n = 1.

50%). In this paper, λ_{adv} is set as 1^{-3}, and the accuracy of our binary lexical classifier $\mathbf{\Theta}_{DeLexi}$ always keep over 75% in the adversarial learning stage.

By adding the auxiliary lexical-adversarial task, this learning algorithm will ensure the learned representation be event-related but lexical-irrelevant.

3.3 Full Model Fine-Tuning

Given the pre-trained lexical-specific representation model $\mathbf{\Theta}_d$ and the pre-trained lexical-free representation model $\mathbf{\Theta}_g$, we finally fine-tune the full model $\mathbf{\Theta}$ in Figure 2 by optimizing the event classification loss function:

$$\mathcal{L}(\mathbf{\Theta}) = \mathcal{L}_{event} + \lambda_{reg} \cdot \|\mathbf{\Theta}\|^2 \qquad (12)$$

where λ_{reg} is the weight coefficient of regularization item and $\mathbf{\Theta}$ indicates all parameters. $\mathcal{L}(\mathbf{\Theta})$ can be optimized using mini-batch based stochastic gradient descent algorithms, such as Adadelta (Zeiler, 2012).

4 Experiments

4.1 Experimental Settings

Dataset. We conduct experiments on two standard English event detection datasets: ACE2005 and KBP2017.

ACE2005 (LDC2006T06) contains 599 documents annotated with 33 event types. Following previous studies (Liao and Grishman, 2010; Li et al., 2013; Chen et al., 2015; Liu et al., 2017, 2018a), we use the same 529/30/40 train/dev/test document splits in our experiments. We use ACE2005 as the primary dataset, as the same as previous studies (Nguyen and Grishman, 2018).

KBP2017 (LDC2017E55) contains 500 documents with RichERE annotations for TAC KBP 2017 evaluation. For model training, we use previously annotated RichERE datasets, including LDC2015E29, LDC2015E68, LDC2016E31 and TAC KBP 2015-2016 Evaluation datasets. Following previous work (Lin et al., 2018a), we randomly sample 20 documents from the 2016 evaluation dataset as the development set.

We evaluate different event detection systems using precision, recall, and F1-score. For ACE2005, we compute these criteria as the same as previous work (Li et al., 2013; Chen et al., 2015). For KBP2017, because TAC KBP2017 allows each team to submit 3 different runs, to make our results comparable with the evaluation results, we select 3 best runs of each system on the development set and report the best test performance among them using the official evaluation toolkit[5], which is referred as Best3 in previous work (Lin et al., 2018a).

Baselines. We compare our approach with three types of baselines:

Feature based Approaches rely on rich hand-designed features, including: *MaxEnt* (Li et al., 2013) which employs hand-designed features and uses Max-Entropy Classifier; *Combined PSL* (Liu et al., 2016b) – the best reported feature-based system which combines global and latent features using Probabilistic Soft Logic framework.

Representation Learning based Approaches employ neural networks to automatically extract features for event detection, including: *DMCNN* (Chen et al., 2015) which uses CNN as sentence feature extractor and concatenates sentence feature and lexical feature for event detection classifier; *NC-CNN* (Nguyen and Grishman, 2016) which extends traditional CNN by modeling skip-grams for exploiting non-consecutive k-grams; *Bi-RNN* (Nguyen et al., 2016) which embeds each token using additional dependency features for bi-directional RNN feature extractor, and jointly extracts triggers with its arguments.

External Resource based Approaches aim to enhance event detection with external resources, including: *SA-ANN-Arg* (Liu et al., 2017) which injects event arguments information via supervised attention mechanism; *GCN-ED* (Nguyen and Grishman, 2018) which exploits syntactic information to capture more accurate context using Graph Convolutional Networks (GCN); *GMLATT* (Liu et al., 2018a) which exploits the multi-lingual information for more accurate context modeling; *HBTNGMA* (Chen et al., 2018) which fuses both sentence-level and document-level information, and collectively detects different events in a sentence.

For our approach and all baselines, we adopt the pre-trained word embedding using Skip-gram [6] and the open released ELMo models[7]. We also report the performance of ELMo as a baseline for demonstrating the performance of universal pre-trained representations. All hyper-parameters are tuned on development set.

[5] https://github.com/hunterhector/EvmEval
[6] https://code.google.com/archive/p/word2vec
[7] https://allennlp.org/elmo

	P	R	F1
Feature based Approaches			
MaxEnt	74.5	59.1	65.9
Combined-PSL	75.3	64.4	69.4
Representation Learning based Approaches			
DMCNN	75.6	63.6	69.1
Bi-RNN	66.0	**73.0**	69.3
NC-CNN	-	-	71.3
External Resource based Approaches			
SA-ANN-Arg (+Arguments)	78.0	66.3	71.7
GMLATT (+Multi-Lingual)	**78.9**	66.9	72.4
GCN-ED (+Syntactic)	77.9	68.8	73.1
HBTNGMA (+Document)	77.9	69.1	73.3
Our Approach			
ELMo	75.6	62.3	68.3
$\mathbf{\Delta}_{w2v}^{concat}$	71.8	70.8	71.3
$\mathbf{\Delta}_{ELMo}^{concat}$	73.7	71.9	72.8
$\mathbf{\Delta}_{w2v}$	74.0	70.5	72.2
$\mathbf{\Delta}_{ELMo}$	76.3	71.9	**74.0**

Table 2: Experiment results on ACE 2005. For a fair comparison, the results of baselines are adapted from their original papers.

4.2 Overall Performance

Table 2 shows the overall ACE2005 results of all baselines and our approach. For our approach, we show the results of four settings: our approach using word embedding as its word representation $\mathbf{r}_w$ – $\mathbf{\Delta}_{w2v}$; our approach using ELMo as $\mathbf{r}_w$ - $\mathbf{\Delta}_{ELMo}$; our approach simply concatenating $[\mathbf{r}_d, \mathbf{r}_g, \mathbf{r}_w]$ as instance representation - $\mathbf{\Delta}_*^{concat}$. From Table 2, we can see that:

1. **By distilling both discrimination and generalization knowledge, our method achieves state-of-the-art performance.** Compared with the best feature system, $\mathbf{\Delta}_{w2v}$ and $\mathbf{\Delta}_{ELMo}$ gain 2.8 and 4.6 F1-score improvements. Compared to the representation learning based baselines, both $\mathbf{\Delta}_{w2v}$ and $\mathbf{\Delta}_{ELMo}$ outperform all of them. Notably, $\mathbf{\Delta}_{ELMo}$ outperforms all the baselines using external resources.

2. **By incrementally distilling generalization knowledge, our method can achieve both high recall and high precision.** Our method obtains a high recall – 71.9, which outperforms most methods by a large margin, and retains a high precision – 76.3. We believe this is because the generalization knowledge is incrementally distilled using $\mathbf{\Delta}$-learning, so there is no need to make the precision-recall tradeoff during training.

3. **The lexical gate provides an effective mechanism for adaptively fusing discrimination and generalization knowledge.** Compared with the naive fusion baselines - $\mathbf{\Delta}_*^{concat}$, $\mathbf{\Delta}_{w2v}$ and $\mathbf{\Delta}_{ELMo}$ correspondingly gain 0.9 and 1.2 F1 improvements. This means that an adapative fusion mechanism can get benefits from both discrimination and generalization knowledge, rather than make tradeoff between them.

	P	R	F1
Top 3 in TAC 2017 ED Track			
3rd in TAC 2017	54.27	46.59	50.14
2nd in TAC 2017	52.16	48.71	50.37
1st in TAC 2017	56.83	**55.57**	56.19
Our Approach			
$\mathbf{\Delta}_{w2v}$	**62.84**	50.36	55.91
$\mathbf{\Delta}_{ELMo}$	62.30	53.77	**57.72**

Table 3: Experiment results on TAC KBP 2017 evaluation datasets.

4. **Although universal pre-trained representations can achieve a good performance, task-specific representations are still crucial.** Compared with the strong universal representation baseline ELMo, our task-specific event detection representations all achieve a significant performance improvements. This also verifies that $\mathbf{\Delta}$-learning is an effective way for incrementally learning task-specific representation.

Table 3 further compares our method with the Top 3 systems in TAC 2017 Event Detection Track (Mitamura et al., 2017). Because these teams had no access to gold entity information during evaluation, we exclude entity embedding in our KBP2017 experiments for a fair comparison. We can see that, the proposed method can significantly outperform the best ED systems in TAC 2017, despite these systems are ensemble models which have leveraged various external resources.

4.3 Detailed Analysis

To analyze the effect of our method in detail, Table 4 shows the performance of our method on different types of trigger words, including:

OOV (out-of-vocabulary) and **OOL** (out-of-label) are of the same as in Table 1.

Sparse instance means the event trigger rate of the given word $P(e|w) = \frac{\#(e,w)}{\#(w)}$ is less than 10% in training corpus, i.e, $< 10\%$ occurrences of word w are labeled with the event type e (NIL including).

Dense means all other instances except OOV,

OOL and Sparse.

Let $\bigoplus$ be our lexical gate, Table 4 shows the results of following settings: $\mathbf{r}_d \bigoplus \mathbf{r}_w$, $\mathbf{r}_g \bigoplus \mathbf{r}_w$ and $\mathbf{r}_d \bigoplus \mathbf{r}_g \bigoplus \mathbf{r}_w$ (i.e., our full model). To demonstrate the effects of $\mathbf{\Delta}$-learning, Table 4 also shows the results of the non-$\mathbf{\Delta}$-learning version of $\mathbf{r}_d \bigoplus \mathbf{r}_w$, $\mathbf{r}_g \bigoplus \mathbf{r}_w$ and $\mathbf{r}_d \bigoplus \mathbf{r}_g \bigoplus \mathbf{r}_w$. In this setting, $\mathbf{r}_d$ and $\mathbf{r}_g$ are trained without using auxiliary tasks. From Table 4, we can see that:

1. **Previous approaches work well on distilling discrimination knowledge, but poorly on distilling generalization knowledge.** Previous approaches achieve high F1-scores on Dense instances, but their performance on sparsely labeled instances is poor. The task specific representation (ATT-RNN and DMCNN) merely achieves a similar performance with the general word representation $\mathbf{r}_w$ (ELMo).

2. **$\mathbf{\Delta}$-learning is effective for incrementally distilling knowledge.** Compared with its non-$\mathbf{\Delta}$-learning version, $\mathbf{r}_g \bigoplus \mathbf{r}_w$ can distill generalization knowledge, i.e., gains 8.5, 12.3 and 9.8 F1 improvements on OOV, OOL and Sparse instances. And $\mathbf{r}_d \bigoplus \mathbf{r}_w$ can distill more discrimination knowledge than its non-$\mathbf{\Delta}$-learning version.

3. **The decomposition strategy, i.e, learning and fusing independent knowledge is effective for representation learning.** Through decomposition, $\mathbf{r}_g \bigoplus \mathbf{r}_w$ can capture generalization knowledge, $\mathbf{r}_d \bigoplus \mathbf{r}_w$ can capture discrimination knowledge, which are complementary to each other. Starting from $\mathbf{r}_w$, our method can incrementally distill knowledge in both $\mathbf{r}_d$ and $\mathbf{r}_g$ via $\mathbf{\Delta}$-learning. By fusing the independent knowledge in $\mathbf{r}_d$ and $\mathbf{r}_g$ via an effective lexical gate, $\mathbf{r}_d \bigoplus \mathbf{r}_g \bigoplus \mathbf{r}_w$ achieves the best performance on OOV, OOL and Dense instances.

Representations	OOV	OOL	Sparse	Dense
ATT-RNN	38.7	6.2	36.7	77.7
DMCNN*	32.4	9.0	43.1	77.6
ELMo	31.3	9.0	47.1	78.0
$\mathbf{r}_d \bigoplus \mathbf{r}_w$ (w/o $\mathbf{\Delta}$)	40.0	8.8	50.0	78.7
$\mathbf{r}_g \bigoplus \mathbf{r}_w$ (w/o $\mathbf{\Delta}$)	47.1	11.1	54.6	78.8
$\mathbf{r}_d \bigoplus \mathbf{r}_g \bigoplus \mathbf{r}_w$ (w/o $\mathbf{\Delta}$)	40.0	11.4	52.8	78.8
$\mathbf{r}_d \bigoplus \mathbf{r}_w$	32.3	12.3	43.1	79.1
$\mathbf{r}_g \bigoplus \mathbf{r}_w$	55.6	23.4	**64.4**	78.2
$\mathbf{r}_d \bigoplus \mathbf{r}_g \bigoplus \mathbf{r}_w$	**57.4**	**26.7**	55.6	**80.0**

Table 4: The results (F1-scores) of different representations (ELMo as word representation $\mathbf{r}_w$) on different types of trigger words. For a fair comparison, different from standard DMCNN (Chen et al., 2015) in Table 1 and Table 2, DMCNN* excludes lexical feature but includes entity feature.

5 Related Work

Event Detection. In recent years, neural approaches have achieved significant progress in event detection. Most neural approaches focus on learning effective instance representations (Chen et al., 2015; Nguyen and Grishman, 2015, 2016; Nguyen et al., 2016; Feng et al., 2016; Ghaeini et al., 2016; Lin et al., 2018b). The main drawback of these methods is that they mostly only learn a single and lexical-specific representation, which works well on distilling discrimination knowledge but poorly on generalization knowledge.

Some approaches enhance representation learning using external resources. One strategy is to employ extra knowledge for better representation learning, such as document (Duan et al., 2017; Chen et al., 2018; Liu et al., 2018b), syntactic information (Nguyen and Grishman, 2018; Sha et al., 2018; Orr et al., 2018; Liu et al., 2018c), event arguments (Liu et al., 2017), knowledge bases (Yang and Mitchell, 2017; Lu and Nguyen, 2018) and multi-lingual information (Liu et al., 2018a). The other strategy is generating additional training instances from extra knowledge bases (Liu et al., 2016a; Chen et al., 2017) or news paragraph clusters (Ferguson et al., 2018). Our method does not use any external resources, which could be a good complementary to these methods.

Representation Learning via Auxiliary Learning. In recent years, many auxiliary learning techniques have been proposed for better representation learning. Self-supervised learning learns representation by designing auxiliary tasks rather than using manually labeled data. Examples include colorization in vision tasks (Doersch and Zisserman, 2017), language modeling in text tasks (Rei, 2017). Adversarial learning attempts to fool models through malicious input (Kurakin et al., 2016), it has been broadly used in many scenarios, e.g., domain adaptation (Zeng et al., 2018), knowledge distillation (Qin et al., 2017) and attribute cleaning (Elazar and Goldberg, 2018).

Some adversarial-based techniques have been used for event detection. Hong et al. (2018) overcomes spurious features during training via self-regularization. Liu et al. (2019) distills extra knowledge from external NLP resources using a teacher-student network. This paper employs ad-

versarial Δ-learning algorithm to eliminate lexical information in event representation so that both discrimination and generalization knowledge can be incrementally distilled.

6 Conclusions

This paper proposes a new representation learning framework – Δ-learning, which can distill both discrimination and generalization knowledge for event detection. Specifically, two effective Δ-learning algorithms are proposed to distill discrimination and generalization knowledge independently, and a lexical gate mechanism is designed to fuse different knowledge adaptively. Experimental results demonstrate the effectiveness of our method. Representation learning is a fundamental technique for NLP tasks, especially for resolving the ambiguity and the diversity problem of natural language expressions. For future work, we plan to investigate new auxiliary Δ-learning algorithms using our Δ-learning framework.

Acknowledgments

We sincerely thank the reviewers for their valuable comments. Moreover, this work is supported by the National Key R&D Program of China under Grant 2018YFB1005100; the National Natural Science Foundation of China under Grants no. 61433015, 61572477 and 61772505; and the Young Elite Scientists Sponsorship Program no. YESS20160177.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

James Allan. 2012. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797. Association for Computational Linguistics.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

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, Beijing, China. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Carl Doersch and Andrew Zisserman. 2017. Multi-task self-supervised visual learning. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2051–2060.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 352–361. Asian Federation of Natural Language Processing.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21. Association for Computational Linguistics.

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. Association for Computational Linguistics.

James Ferguson, Colin Lockard, Daniel 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)*, NAACL '2018, pages 359–364. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2:*

Short Papers), pages 369–373, Berlin, Germany. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. Curran Associates, Inc.

Yu Hong, Wenxuan Zhou, Jingli Zhang, Qiaoming Zhu, and Guodong Zhou. 2018. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 515–526. Association for Computational Linguistics.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *CoRR*, abs/1611.01236.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. 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. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018a. Adaptive scaling for sparse detection in information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1033–1043. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018b. Nugget proposal networks for chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1565–1574. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Kang Liu. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI '2019. Association for the Advancement of Artificial Intelligence.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '2018, pages 4865–4872, New York, NY, USA. Association for the Advancement of Artificial Intelligence.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018b. Exploiting contextual information via dynamic memory network for event detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1030–1035. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016a. Leveraging framenet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2143, Berlin, Germany. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Shulin Liu, Kang Liu, Shizhu He, and Jun Zhao. 2016b. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2993–2999. AAAI Press.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018c. 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. Association for Computational Linguistics.

Weiyi Lu and Thien Huu Nguyen. 2018. Similar but not the same - word sense disambiguation improves event detection via neural representation matching. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4822–4828. Association for Computational Linguistics.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track. In *TAC*.

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, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with 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 2: Short Papers), pages 365–371, Beijing, China. Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '2018, pages 5900–5907, New York, NY, USA. Association for the Advancement of Artificial Intelligence.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004. Association for Computational Linguistics.

Matthew 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, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

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 *The Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '2018, pages 5916–5923, New York, NY, USA. Association for the Advancement of Artificial Intelligence.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446. Association for Computational Linguistics.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.