Adversarial Training for Weakly Supervised Event Detection

Xiaozhi Wang[1*], **Xu Han**[1*], **Zhiyuan Liu**[1†], **Maosong Sun**[1], **Peng Li**[2]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
[2]Pattern Recognition Center, WeChat, Tencent Inc, China
{xz-wang16, hanxu17}@mails.tsinghua.edu.cn
{liuzy, sms}@tsinghua.edu.cn
patrickpli@tencent.com

Abstract

Modern weakly supervised methods for event detection (ED) avoid time-consuming human annotation and achieve promising results by learning from auto-labeled data. However, these methods typically rely on sophisticated pre-defined rules as well as existing instances in knowledge bases for automatic annotation and thus suffer from low coverage, topic bias, and data noise. To address these issues, we build a large event-related candidate set with good coverage and then apply an adversarial training mechanism to iteratively identify those informative instances from the candidate set and filter out those noisy ones. The experiments on two real-world datasets show that our candidate selection and adversarial training can cooperate together to obtain more diverse and accurate training data for ED, and significantly outperform the state-of-the-art methods in various weakly supervised scenarios. The datasets and source code can be obtained from https://github.com/thunlp/Adv-ED.

1 Introduction

Event detection (ED) aims at detecting event triggers, which are often words or phrases evoking events in instances, and then identifying their specific event types. For example, we can extract the trigger "married" of the event "Marry" from the text "Mark Twain and Olivia Langdon married in 1870". Detecting and identifying events is an important subtask of event extraction and also beneficial for various downstream NLP applications, such as question answering (Yang et al., 2003), information retrieval (Basile et al., 2014), and reading comprehension (Cheng and Erk, 2018). Hence, many efforts have been devoted to detecting event triggers and types.

Most prior methods for ED are based on feature engineering, such as the token-level features (Ahn, 2006; Ji and Grishman, 2008) and the structured features (Li et al., 2013; Araki and Mitamura, 2015). As the rapid development of neural networks, various neural models have been proposed to directly embed textual semantic information into a low-dimensional space and then detect event triggers based on those feature vectors (Chen et al., 2015; Nguyen and Grishman, 2015; Nguyen et al., 2016; Ghaeini et al., 2016; Feng et al., 2016). These methods follow a supervised learning approach to train models on human-annotated data, and their requirement of human-annotated data is a bottleneck in practice. Considering weak supervision is widely adopted to take full advantages of large-scale raw data, especially some specific work for information extraction (Mintz et al., 2009; Riedel et al., 2010; Zeng et al., 2015; Cao et al., 2018), weak supervision has been explored to automatically label training data for ED (Chen et al., 2017; Zeng et al., 2018; Yang et al., 2018; Araki and Mitamura, 2018). Compared with those supervised ED methods, the weakly supervised methods can be generalized to real-world ED applications efficiently without intensive labor.

Although promising results have been achieved by these weakly supervised methods, there are still some severe problems for these weakly supervised ED models: (1) Weakly supervised methods naturally suffer from the inevitable noise in data. (2) Current weakly supervised ED models adopt sophisticated pre-defined rules and incomplete knowledge bases to automatically obtain data, which results in the auto-labeled data with low coverage and topic bias.

In order to construct a large-scale dataset with better coverage and reduce topic bias, we avoid adopting sophisticated pre-defined rules and heavy toolkits for semantic component analysis. Instead,

[*] indicates equal contribution
[†] Corresponding author: Z.Liu(liuzy@tsinghua.edu.cn)

Proceedings of NAACL-HLT 2019, pages 998–1008
Minneapolis, Minnesota, June 2 - June 7, 2019.

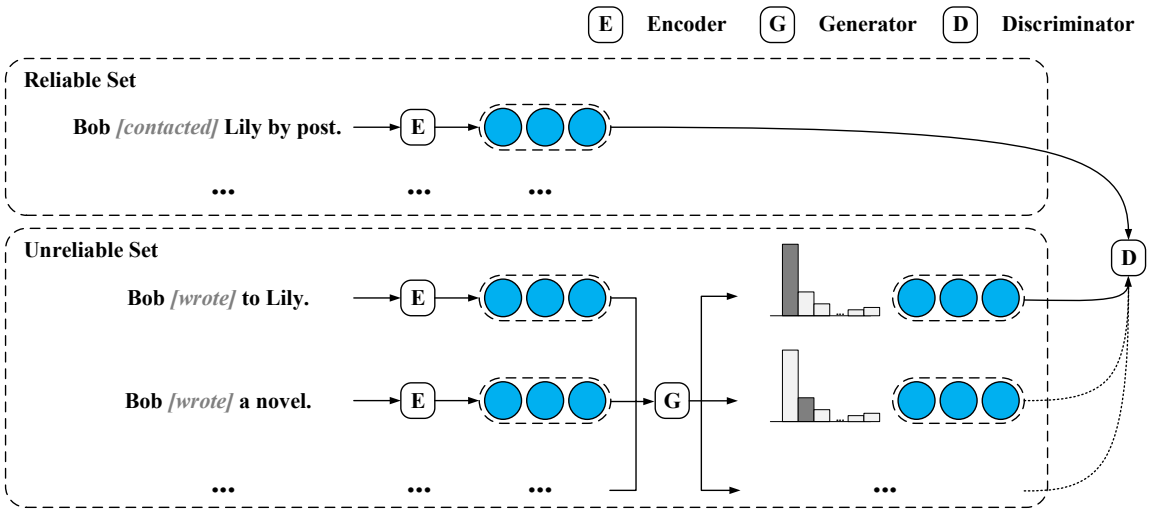

Figure 1: The overall architecture of adversarial training method for ED. The event type is `Contact`.

we propose a simple trigger-based latent instance discovery strategy, by applying an assumption that if a given word [1] serves as the trigger in a known event instance, all instances mentioning this word may also express an event. As compared with the sophisticated rules, this strategy is less restrictive in the correlation among words, triggers and event types. Hence, our strategy can obtain a candidate set covering more topics and instances without any manual design.

We further propose an adversarial training mechanism like Goodfellow et al. (2014); Radford et al. (2016), which can not only distill those informative instances from the candidate set but also improve the performance of ED model on a noisy scenario such as distant supervision. As shown in Figure 1, we split the dataset into a reliable set and an unreliable set respectively, and design a discriminator and a generator. The discriminator is applied to judge whether a given instance is informative and annotated correctly, and the generator is used to select the most confusing instances from raw data to fool the discriminator. The discriminator is trained with the reliable data as positive instances and the data selected by the generator as negative ones. Meanwhile, the generator is trained to select data to fool the discriminator. During the training process, the generator can provide large amounts of latent noisy data to enhance the discriminator, and the discriminator can influence the generator to select those more informative data. Since noisy data makes no effect on optimizing both the generator and the discriminator, when the generator and the discriminator reach a balance, the discriminator can boost resistance to noise and better categorize events, and the generator can effectively select informative instances to the discriminator.

We conduct experiments on both semi-supervised and distantly supervised scenarios. The experimental results demonstrate that our trigger-based latent instance discovery strategy and adversarial training method can cooperate to obtain more diverse and accurate training data as well as reduce the side effect of the noise problem, and thus significantly outperform the state-of-the-art ED models.

2 Related Work

ED has attracted wide attention recently. Traditional feature-based methods (Ahn, 2006; Ji and Grishman, 2008; Gupta and Ji, 2009; Riedel et al., 2010; Hong et al., 2011; McClosky et al., 2011; Huang and Riloff, 2012a,b; Araki and Mitamura, 2015; Li et al., 2013; Yang and Mitchell, 2016; Liu et al., 2016b) rely on manually designed features to detect the event triggers and event types. With the development of neural networks, various neural methods have also been proposed (Chen et al., 2015; Nguyen and Grishman, 2015; Nguyen et al., 2016; Duan et al., 2017; Nguyen et al., 2016; Ghaeini et al., 2016; Lin et al., 2018).

Furthermore, some efforts have been made to improve the performance of ED systems with external knowledge (Liu et al., 2016a, 2017), contextual information (Liu et al., 2018b), document-level information (Duan et al., 2017; Zhao et al.,

[1] We treat phrases as words in this paper.

2018) and multimodal integration (Zhang et al., 2017). Some advanced architectures have also been applied, such as attention mechanism (Liu et al., 2017, 2018a), graph convolutional networks (Nguyen and Grishman, 2018) and generative adversarial networks (Hong et al., 2018).

All the supervised methods above rely on human-annotated data, and the data is often restricted to a small scale due to the expensive human annotation. Hence, unsupervised methods (Huang et al., 2016; Yuan et al., 2018) and various weakly supervised methods on ED are proposed. Muis et al. (2018) adopt distant supervision to create training instances for low-resource language. Araki and Mitamura (2018) adopt WordNet (Miller et al., 1990) and rule-based methods to generate open-domain data without event-type labels. Chen et al. (2017) and Zeng et al. (2018) use distant supervision to generate large-scale data from existing structured event knowledge in knowledge bases. Liao and Grishman (2010a), Huang and Riloff (2012a) and Ferguson et al. (2018) conduct semi-supervised ED with bootstrapping. Nevertheless, due to the low coverage of existing knowledge bases as well as lack of advanced denoising mechanism, those weakly supervised methods still suffer from the problem of low coverage and noisy data.

Inspired by Szegedy et al. (2013) and Goodfellow et al. (2014), adversarial training has been explored for several NLP applications recently to resist noise, such as text classification (Miyato et al., 2016) and text generation (Xie et al., 2017; Chen et al., 2018). Adversarial training has also been adopted for information extraction (Wu et al., 2017; Hong et al., 2018; Qin et al., 2018; Wang et al., 2018; Han et al., 2018). These adversarial information extraction methods either generate adversarial instances by adding simple noise perturbation to embeddings (Wu et al., 2017; Hong et al., 2018), or mainly adopt models to denoise data and neglect to discover more training instances from raw data (Qin et al., 2018; Han et al., 2018). Compared with these methods, our adversarial method samples adversarial examples from the real-world data rather than generating pseudo noisy perturbations. Furthermore, our method not only denoises auto-labeled data but also labels unlabeled instances to extend datasets for higher coverage. Hence, our method can effectively alleviate low coverage, topic bias, and noise problem in ED.

3 Methodology

In this section, we introduce the overall framework of our proposed models for weakly supervised ED.

3.1 Framework

As shown in Figure 1, the overall framework consists of three modules, including instance encoder, adversarial training strategy, and their adaption for various weakly supervised ED scenarios.

The instance encoder is applied to encode the instances into its corresponding embeddings to provide semantic features for the other modules of our models. Given an instance $x = \{w_1, \ldots, t, \ldots, w_n\}$ consisting of n words and its candidate trigger t, we adopt several effective neural models to represent the semantic features of the instance x with the embedding $\mathbf{x}$. Details of the instance encoder are shown in Section 3.2.

After representing instances into their embeddings by the instance encoder, an adversarial training strategy is applied, which aims at highlighting those informative instances and filtering out those noisy instances from a large-scale unreliable dataset $\mathcal{U}$ under the guidance of another reliable dataset $\mathcal{R}$. The adversarial training strategy is the core module of our framework, and we will introduce its details in Section 3.3.

Each instance x in $\mathcal{U}$ and $\mathcal{R}$ will be labeled with a trigger word t and an event type $e \in \mathcal{E}$. If an instance does not have a trigger and cannot express any definite events, it will be labeled with a special event `NA`, which indicates that the event of this instance is not available. Before applying adversarial training, $\mathcal{U}$ and $\mathcal{R}$ are automatically labeled. The details of splitting and automatically labeling $\mathcal{U}$ and $\mathcal{R}$ for various weakly supervised ED scenarios, as well as utilizing adversarial training strategy to extend datasets, will be introduced in Section 3.4.

3.2 Instance Encoder

In this paper, we select CNN (Chen et al., 2015) and BERT (Devlin et al., 2018) as representative encoders to encode the given instances.

CNN After representing all words in the instance x into their input embeddings, including both word embeddings and position embeddings which encode the relative position to candidate triggers, CNN slides a convolution kernel over the input embeddings to get hidden embeddings as

follows,

$$\{\mathbf{h}_1, \ldots, \mathbf{h}_n\} = \text{CNN}\big(w_1, \ldots, t, \ldots, w_n\big). \quad (1)$$

BERT Similar to CNN, after summing word piece (Wu et al., 2016), segment and position embeddings of all words in the instance x as input embeddings, BERT adopt a multi-layer bidirectional transformer encoder (Vaswani et al., 2017) to get hidden embeddings as follows,

$$\{\mathbf{h}_1, \ldots, \mathbf{h}_n\} = \text{BERT}\big(w_1, \ldots, t, \ldots, w_n\big). \quad (2)$$

Because the candidate trigger t splits the instance x into two parts, we follow Chen et al. (2015) to adopt a dynamic multi-pooling operation over the hidden embeddings to achieve the instance embedding $\mathbf{x}$,

$$\begin{aligned} [\overleftarrow{\mathbf{x}}]_j &= \max\{[\mathbf{h}_1]_j, \ldots, [\mathbf{h}_i]_j\}, \\ [\overrightarrow{\mathbf{x}}]_j &= \max\{[\mathbf{h}_{i+1}]_j, \ldots, [\mathbf{h}_n]_j\}, \\ \mathbf{x} &= [\overleftarrow{\mathbf{x}}; \overrightarrow{\mathbf{x}}], \end{aligned} \quad (3)$$

where $[\cdot]_j$ is the j-th value of a vector and i is the position of the trigger t. As CNN and BERT adopt a dynamic multi-pooling operation, we name them "**DMCNN**" and "**DMBERT**" in this paper.

3.3 Adversarial Training

As shown in Figure 1, the overall framework of our adversarial strategy consists of a discriminator and a generator. The discriminator is adopted to detect event triggers and identify event types for each instance in datasets. When given a noisy instance, the discriminator is also expected to resist noise and explicitly point out that there are no triggers and events. The generator is used to select instances from the unreliable dataset $\mathcal{U}$ to confuse the discriminator as much as possible.

Each instance $x \in \mathcal{R}$ is assumed to explicitly express its labeled trigger t and event type e. In contrast, each instance $x \in \mathcal{U}$ is assumed to be untrustworthy during the adversarial training, i.e., there is a certain probability that it is labeled incorrectly. Hence, we design the discriminator to judge whether a given instance can expose its labeled event type, which aims at maximizing the conditional probability $P(e|x,t), x \in \mathcal{R}$ and $1 - P(e|x,t), x \in \mathcal{U}$. The generator is trained to select the most confusing instances from $\mathcal{U}$ to fool the discriminator, i.e., selecting the instances by $P(e|x,t), x \in \mathcal{U}$. The training process is an adversarial min-max game as follows,

$$\begin{aligned} \phi_D = \max\big(&E_{x\sim P_R}\big[\log\big(P(e|x,t)\big)\big] \\ &+E_{x\sim P_\mathcal{U}}\big[\log\big(1-P(e|x,t)\big)\big]\big), \\ \phi_G = \max&E_{x\sim P_\mathcal{U}}\big[\log\big(P(e|x,t)\big)\big], \end{aligned} \quad (4)$$

where $P_\mathcal{R}$ is the reliable data distribution, and the generator samples adversarial examples from the unreliable data according to the probability distribution $P_\mathcal{U}$. Although ϕ_D and ϕ_G are conflicting, noisy data in $\mathcal{U}$ has the side effect for both ϕ_D and ϕ_G. Hence, when the generator and the discriminator reaching a balance after sufficient training, the generator tends to select those informative instances with a higher probability compared with those noisy ones, and the discriminator boosts resistance to noise and can better categorize events.

Discriminator

Given an instance x and its labeled trigger t and event type e, the discriminator is responsible for judging whether the given instance exposes its labeled trigger and event type. After representing the instance x with its embedding $\mathbf{x}$, we implement the discriminator as follows,

$$\begin{aligned} D(e|x,t) &= \mathbf{e} \cdot \mathbf{x}, \\ P(e|x,t) &= \frac{\exp\big(D(e|x,t)\big)}{\sum_{\hat{e}\in\mathcal{E}} \exp\big(D(\hat{e}|x,t)\big)}, \end{aligned} \quad (5)$$

where $\mathbf{e}$ is the embedding of the event type $e \in \mathcal{E}$.

An optimized discriminator will assign high scores to those instances in $\mathcal{R}$, and meanwhile distrust those instances and their labels in $\mathcal{U}$. Hence, in practice, we formalize the loss function to optimize the discriminator as follows,

$$\begin{aligned} \mathcal{L}_D = &-\sum_{x\in\mathcal{R}} \frac{1}{|\mathcal{R}|} \log\big(P(e|x,t)\big) \\ &-\sum_{x\in\mathcal{U}} P_\mathcal{U}(x) \log\big(1-P(e|x,t)\big). \end{aligned} \quad (6)$$

When optimizing the discriminator, we regard the component of the encoder and $D(e|x,t)$ as parameters for updating. This loss function $\mathcal{L}_D$ is corresponding to ϕ_D in Eq. (4).

Generator

The generator aims at selecting the most confusing instances from $\mathcal{U}$ to cheat the discriminator. We design the generator to optimize the probability distribution $P_\mathcal{U}$ to select instances. The generator computes confusing scores for all instances in

$\mathcal{U}$ to evaluate their perplexity and further computes the confusing probability $P_{\mathcal{U}}$ as follows,

$$f(x) = \mathbf{W} \cdot \mathbf{x} + \mathbf{b},$$
$$P_{\mathcal{U}}(x) = \frac{\exp\big(f(x)\big)}{\sum_{\hat{x}\in\mathcal{U}} \exp\big(f(\hat{x})\big)}. \quad (7)$$

where $\mathbf{x}$ is the embedding of the instance x computed by the encoder. $\mathbf{W}$ and $\mathbf{b}$ are parameters for a separating hyperplane.

We regard that the higher scores computed by the discriminator the instances have, the more confusing the instances are, because they are more likely to fool the discriminator to make a wrong decision. We expect that an optimized generator pays more attention to those most confusing instances. Hence, given an instance $x \in \mathcal{U}$ and its unreliable-labeled trigger t and event type e, we formalize the loss function to optimize the generator as follows,

$$\mathcal{L}_G = -\sum_{x\in\mathcal{U}} P_{\mathcal{U}}(x) \log\big(P(e|x,t)\big), \quad (8)$$

where $P(e|x,t)$ is computed by the discriminator. When optimizing the generator, we regard the component to compute $P_{\mathcal{U}}(x)$ as parameters for updating. This loss function $\mathcal{L}_G$ is corresponding to ϕ_G in Eq. (4).

There may be some instances in $\mathcal{U}$ labeled `NA` and these instances are always wrongly predicted into some other events. Thus we specifically use the average scores over all feasible events to replace their $P(e|x,t)$ in Eq. (8) as follows,

$$P(\text{NA}|x,t) = \frac{1}{|\mathcal{E}|-1} \sum_{e\in\mathcal{E}, e\neq\text{NA}} P(e|x,t), \quad (9)$$

where $\mathcal{E}$ indicates the set of event types.

Training and Implementation Details

Because there may be large amounts of instances in $\mathcal{R}$ and $\mathcal{U}$, directly computing $\mathcal{L}_D$ and $\mathcal{L}_G$ is time-consuming, and frequently traversing the whole dataset of $\mathcal{R}$ and $\mathcal{U}$ also accordingly becomes difficult. For improving training efficiency, we sample subsets of $\mathcal{R}$ and $\mathcal{U}$ to approximate the essential probability distribution, and formalize a new loss function for optimization,

$$\tilde{\mathcal{L}}_D = -\sum_{x\in\tilde{\mathcal{R}}} \frac{1}{|\tilde{\mathcal{R}}|} \log\big(P(e|x,t)\big) - \sum_{x\in\tilde{\mathcal{U}}} P_{\tilde{\mathcal{U}}}(x) \log\big(1 - P(e|x,t)\big), \quad (10)$$
$$\tilde{\mathcal{L}}_G = -\sum_{x\in\tilde{\mathcal{U}}} P_{\tilde{\mathcal{U}}}(x) \log\big(P(e|x,t)\big),$$

where $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{U}}$ are the subsets sampled from $\mathcal{R}$ and $\mathcal{U}$, and $P_{\tilde{\mathcal{U}}}$ is the approximation to Eq. (7),

$$P_{\tilde{\mathcal{U}}}(x) = \frac{\exp\big(f(x)^\alpha\big)}{\sum_{\hat{x}\in\tilde{\mathcal{U}}} \exp\big(f(\hat{x})^\alpha\big)}. \quad (11)$$

α is a hyperparameter that controls the sharpness of the probability distribution to avoid the weights concentrating on some specific instances. Finally, the overall optimization function is,

$$\mathcal{L} = \tilde{\mathcal{L}}_D + \lambda \tilde{\mathcal{L}}_G, \quad (12)$$

where λ is a harmonic factor. In practice, $\tilde{\mathcal{L}}_D$ and $\tilde{\mathcal{L}}_G$ in adversarial training are optimized alternately, and λ is also integrated into the learning rate of $\tilde{\mathcal{L}}_G$ to avoid adjusting λ additionally.

3.4 Adaption for Weakly Supervised Scenarios

In this section, we introduce the adaption of adversarial training strategy for various weakly supervised ED scenarios (semi-supervised scenarios and distantly supervised scenarios), as well as the method to automatically label and split the reliable set and unreliable set used for adversarial training.

Trigger-based Latent Instance Discovery

To utilize unlabeled data, we propose a simple trigger-based latent instance discovery strategy, which can automatically label trigger words and event types for raw data. The trigger-based strategy is based on a heuristic assumption that if a given word serves as the trigger in a known instance, all other instances mentioning this word in raw data are latent instances and may also express an event. For example, the word "`married`" serves as the trigger in the instance "Mark Twain and Olivia Langdon `married` in 1870" to expose the event "`Marry`", and then all instances in unlabeled data containing the word "`married`" will be picked up and added into a latent instance candidate set. As compared with the sophisticated

rules used in existing weakly supervised ED models, our trigger-based latent instance discovery is simple, without the need of considering the correlation among words, triggers, and event types. Because our strategy is less restrictive, it is effective and efficient to obtain a large-scale candidate set without any special manual design. Meanwhile, the candidate set can cover much more instances and topics than the existing strategies.

Semi-supervised Scenarios

When adapting our adversarial training strategy for semi-supervised scenarios, we first use the small-scale labeled data to pretrain the encoder and discriminator to let them gain the ability to detect event triggers and identify event types to a certain extent. Then, we construct a large-scale latent candidate set based on our instance discovery strategy with the trigger words in the labeled data as heuristic seeds. We use the pretrained encoder and discriminator to automatically label triggers and event types for all instances in the candidate set to build noisy large-scale data. With the small-scale labeled data as the reliable set $\mathcal{R}$ and the large-scale auto-labeled data as the unreliable set $\mathcal{U}$, we can optimize the encoder, discriminator, and generator together to carry out adversarial training. During the adversarial training, when the discriminator and generator reach a balance after certain training epochs, all instances from the unreliable set $\mathcal{U}$ recommended by the generator and regarded as being labeled correctly by the discriminator will be adjusted from $\mathcal{U}$ to $\mathcal{R}$. Conducting adversarial training iteratively can identify informative instances and filter out noisy instances in $\mathcal{U}$, and accomplish utilizing large-scale unlabeled data to enrich small-scale labeled data.

Distantly Supervised Scenarios

The adaption for distantly supervised scenarios is similar to the adaption for semi-supervised scenarios. We first use the whole auto-labeled data to pretrain the encoder and discriminator. Then, the encoder and discriminator are used to compute confident scores for all instances in the auto-labeled set. By setting a particular threshold, we can split the whole auto-labeled set into two parts. The instances with scores higher than the threshold will be added into the reliable set $\mathcal{R}$, and the other instances with lower scores will be added into the unreliable set $\mathcal{U}$. After the whole auto-labeled set being split into $\mathcal{R}$ and $\mathcal{U}$, we can conduct adversarial training to reduce the side effect of those noise in $\mathcal{U}$ and enhance the discriminator for better identifying events. Intuitively, the reliable set $\mathcal{R}$ isolated from the auto-labeled set can be used as seeds to utilize more raw data in a similar way applied in semi-supervised scenarios.

4 Experiments

We evaluate our models on both semi-supervised and distantly supervised scenarios. Before introducing the detailed experimental settings and results, we list the hyperparameters first.

4.1 Hyperparameter Settings

For DMCNN, following the settings of previous work, we use the pre-trained word embeddings learned by Skip-Gram (Mikolov et al., 2013) as the initial word embeddings. We implement DMCNN by ourselves and follow the same hyperparameters used in Chen et al. (2015) for fair comparisons. For DMBERT, we follow the same hyperparameters used for $\text{BERT}_{\text{BASE}}$ in Devlin et al. (2018) and apply the pre-trained model[2] to initialize the parameters. We list the essential hyperparameters of the discriminator and the generator for adversarial training in Table 1.

Dropout Probability p	5×10^{-1}
Learning Rate α_{gc} for the generators (DMCNN)	5×10^{-3}
Learning Rate α_{dc} for the discriminators (DMCNN)	2×10^{-2}
Learning Rate α_{gb} for the generators (DMBERT)	2×10^{-5}
Learning Rate α_{db} for the discriminators (DMBERT)	1×10^{-4}

Table 1: Hyperparameter settings.

4.2 Distantly Supervised Scenarios

Dataset and Evaluation

For distantly supervised scenarios, we utilize the distantly supervised dataset developed by Chen et al. (2017) with FreeBase (Bollacker et al., 2008). The dataset contains $142,611$ labeled instances and 21 event types. Following previous work (Mintz et al., 2009; Chen et al., 2017), we evaluate our adversarial training mechanism by held-out evaluation. We report the precision-recall curves of recall under 0.7 since we mainly focus on the performance of those top-ranked results. To give a complete view of the overall performance, we also report the area under the curve (AUC).

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

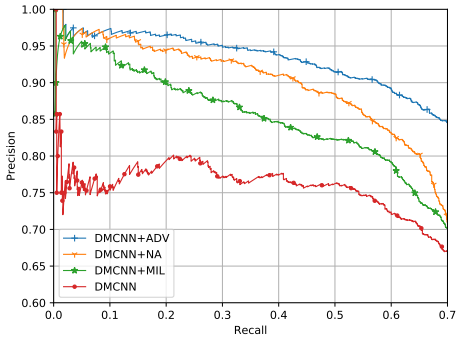

Figure 2: The aggregated precision-recall curves of DMCNN models.

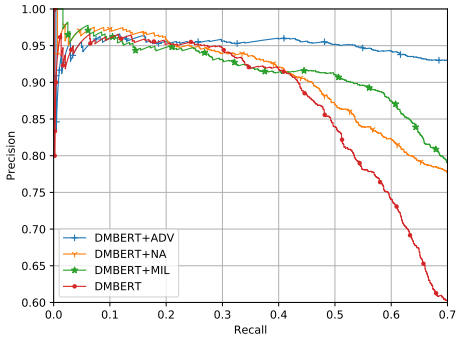

Figure 3: The aggregated precision-recall curves of DMBERT models.

To evaluate the effectiveness of our proposed models, we compare our adversarial training models (**DMCNN+ADV**, **DMBERT+ADV**) with various neural baselines, including: (1) **DMCNN** and **DMBERT** proposed in Chen et al. (2015) and this paper respectively, which are the basic models without any adaption to the noisy distant supervision. (2) **+MIL** models, which improve the basic models with multi-instance learning proposed in Chen et al. (2017) to alleviate the noise problem. (3) **+NA** models, which simply treat the instances in the unreliable set as negative instances with the label NA. This method could be regarded as a simplified version of our adversarial training to conduct ablation study. In this experiment, we separate the reliable and unreliable set by the confidence of the basic models following Section 3.4.

Overall Evaluation Results

The precision-recall curves of DMCNN models and DMBERT models are shown in Figure 2 and Figure 3, and the results of AUC are shown in Table 2. From the results, we can observe that: (1)

Method	AUC	
	Micro	Macro
DMCNN	67.6	38.7
DMCNN+MIL	75.7	43.3
DMCNN+NA	70.6	25.8
DMCNN+ADV	**85.5**	**50.7**
DMBERT	70.6	42.2
DMBERT+MIL	79.4	47.3
DMBERT+NA	74.0	38.6
DMBERT+ADV	**91.5**	**67.6**

Table 2: The AUC results (%) of various models.

BERT-based models significantly outperform the CNN-based models, which is due to the ability to capture contextual information as well as large-scale pre-training of BERT. And benefiting from the effective pre-trained parameters, the BERT-based models all have high precision when the recall is under 0.3. (2) The **+NA** models achieve similar performance with **+MIL** and even outperform them in low-recall range, but **+NA** models have the worst macro AUC. It indicates that the separation of reliable and unreliable set is effective but also have severe side effects, and our adversarial training method works well to overcome the side effect. (3) Our adversarial training method significantly outperforms all the baselines in every metric. This demonstrates the strong ability of our method to alleviate the noise problem on distantly supervised scenarios.

4.3 Semi-supervised Scenarios

Dataset and Evaluation

For semi-supervised scenarios, we conduct experiments on a widely-used benchmark dataset ACE-2005 (Walker et al., 2006) containing 599 documents annotated with 8 types and 33 subtypes of events. Following the previous work (Liao and Grishman, 2010b; Li et al., 2013; Chen et al., 2015), we use the same test set containing 40 newswire documents, development set with 30 randomly selected documents and training set with the remaining 529 documents.

As described in Section 3.4, using existing triggers in ACE-2005 training set as heuristic seeds and our trigger-based latent instance discovery strategy, we construct a large-scale candidate set from the New York Times corpus (Sandhaus, 2008) and use our adversarial training strategy to filter out the noisy instances to build a new ACE-style dataset. We extend the ACE-2005 training set with the new dataset, and then test the models trained on the extended training set on the orig-

inal test set. Our models trained on the original training set are named **DMCNN** and **DMBERT**, and our bootstrapped models trained on the extended dataset are named **DMCNN+Boot** and **DMBERT+Boot**.

We compare our bootstrapped models with various state-of-the-art methods on the ACE-2005 dataset, including: (1) The feature-based models. We select **Li's joint** (Li et al., 2013) as the representative, which achieves the best performance among feature-based models. (2) The vanilla neural network models, including the **DMCNN** (Chen et al., 2015) and **JRNN** (Nguyen et al., 2016). (3) The neural network models with external information, including **ANN-FN** (Liu et al., 2016a) leveraging the information of FrameNet (Baker et al., 1998), **DLRNN** (Duan et al., 2017) using document-level information, **GMLATT** (Liu et al., 2018a) utilizing multilingual attentions, and the bootstrapped model **DMCNN+Chen's DS** (Chen et al., 2017) trained with additional data distantly supervised by FreeBase (Bollacker et al., 2008). (4) The neural network models with advanced architecture, including: **Bi-LSTM+GAN** (Hong et al., 2018) utilizing GAN to conduct self-regulation, **GCN-ED** (Nguyen and Grishman, 2018) utilizing graph convolutional network to model dependency trees.

Overall Evaluation Results

The results are shown in Table 3. From the results, we have the following observations: (1) As compared with the basic **DMCNN** and **DMBERT**, the bootstrapped models achieve significant improvement ($+1.7\%$ and $+0.5\%$). Furthermore, our **DMCNN+Boot** model achieves similar performance with the **ANN-FN** and **DLRNN** which design complex architectures to utilize the additional information. These results indicate that our methods can construct high-quality dataset without sophisticated rules and large-scale knowledge bases, and can effectively collect diverse instances which will benefit training models. (2) **DMBERT** and **DMBERT+Boot** achieve the best performance among all the models. This is benefiting from the effective architecture and the large-scale pre-training information of BERT, as well as the dynamic multi-pooling mechanism for ED. Our methods augment the training data to further enhance BERT, which achieve better performance and demonstrate the effectiveness of our models.

Method	Trigger Identification +Classification		
	P	R	F1
Li's Joint (Li et al., 2013)	73.7	62.3	67.5
DMCNN (Chen et al., 2015)	75.6	63.6	69.1
JRNN (Nguyen et al., 2016)	66.0	**73.0**	69.3
ANN-FN (Liu et al., 2016a)	77.6	65.2	70.7
DLRNN (Duan et al., 2017)	77.2	64.9	70.5
GMLATT (Liu et al., 2018a)	78.9	66.9	72.4
DMCNN+Chen's DS (Chen et al., 2017)	75.7	66.0	70.5
Bi-LSTM+GAN (Hong et al., 2018)	71.3	74.7	73.0
GCN-ED (Nguyen and Grishman, 2018)	**77.9**	68.8	73.1
DMBERT	77.6	71.8	74.6
DMCNN+Boot	77.7	65.1	70.8
DMBERT+Boot	**77.9**	72.5	**75.1**

Table 3: The overall performance (%) of different models on ACE-2005.

Method	Average Precision	Fleiss's Kappa
Chen et al. (2017)	88.9	–
Zeng et al. (2018)	91.0	–
Our First Iteration	91.7	61.3
Our Second Iteration	87.5	52.0

Table 4: The human evaluation results (%) of auto-labeled data in different iterations.

Manual Evaluation

To perform a fine-grained evaluation for the quality of the dataset constructed with our trigger-based instance discovery strategy and adversarial training strategy, we manually evaluate the precision of the constructed dataset. To be specific, we randomly select 150 instances from the newly constructed dataset and recruit four well-trained annotators to annotate the instances independently. We ask the annotators to label an instance as correct if and only if the trigger and event-type are both correct. We use the Fleiss' kappa (Fleiss, 1971) to measure the annotation consistency among these annotators. The results of the data distilled in different iterations during adversarial training are shown in Table 4. From the results, we can observe that the precision of the dataset constructed with our models is comparable to existing distant supervision methods (Chen et al., 2017; Zeng et al., 2018) using sophisticated human-designed rules and knowledge bases, and even outperforms them in the first iteration. It indicates that our models can distill informative instances with high precision.

Case Study

To further show the effectiveness of our models to improve the coverage of the dataset, we give an example in Table 5. The instance in the "In ACE-2005" row is a typical instance of the `Sue`

	Event-Type: `Justice` Subtype: `Sue`
In ACE-2005	Dell **sued** for "bait and switch" and false promises.
Discovered	1. The lawyers for the four former state officials who have been **sued** told the jurors . . . 2. But **litigation** held up the project until . . . .

Table 5: The examples with highlighting triggers.

events, and the two instances in the "Discovered" row are sampled from the dataset constructed with our methods. In the "Discovered" row, the first instance is with an existing trigger in ACE-2005 but different in syntax, and the second instance is with a newly discovered trigger which is not contained in ACE-2005. In our extended dataset, there are 1.2% of the triggers are newly discovered. This demonstrates that our methods can not only find new instances from the unlabeled data which is similar to those instances in the labeled data, but also discover new triggers and extend the coverage of datasets substantially.

5 Conclusion and Future Work

In this paper, we take advantages of adversarial training and propose an effective method for weakly supervised ED. To be specific, our method is able to denoise and enhance distantly supervised ED models, as well as automatically construct more diverse and accurate training data for semi-supervised ED models. The experiments on two real-world datasets show that our method achieves the state-of-the-art results on the settings of both distant supervision and semi-supervision. In the future, we plan to explore the following directions: (1) We will extend our method to further extract event arguments and perform event extraction. (2) We will develop a large-scale and clean dataset for ED based on our method, which will benefit further research in this field.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503), the National Natural Science Foundation of China (NSFC No. 61572273) and China Association for Science and Technology (2016QNRC001). Wang is also supported by the Tsinghua University Initiative Scientific Research Program.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of ACL Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of EMNLP*, pages 2074–2080.

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of COLING*, pages 878–891.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of COLING*, pages 86–90.

P Basile, A Caputo, G Semeraro, and L Siciliani. 2014. Extending an information retrieval system through time event extraction. In *Proceedings of DART*, pages 36–47.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, pages 1247–1250.

Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Chengjiang Li, Xu Chen, and Tiansi Dong. 2018. Joint representation learning of cross-lingual words and entities via attentive distant supervision. In *Proceedings of EMNLP*, pages 227–237.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In *Proceedings of ACL*, pages 2587–2597.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of ACL*, pages 409–419.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 167–176.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *Proceedings of ACL*, pages 831–840.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of IJCNLP*, pages 352–361.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Bing Qin, Heng Ji, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of ACL*, pages 66–71.

James Ferguson, Colin Lockard, Daniel Weld, and Hannaneh Hajishirzi. 2018. Semi-supervised event extraction with paraphrase clusters. In *Proceedings of NAACL*, pages 359–364.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of ACL*, pages 369–373.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In *Proceedings of ICLR*, pages 2574–2582.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *Proceedings of ACL-IJCNLP*, pages 369–372.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distant supervision for relation extraction via instance-level adversarial training. *arXiv preprint arXiv:1805.10959*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL-HLT*, pages 1127–1136.

Yu Hong, Wenxuan Zhou, Guodong Zhou, Qiaoming Zhu, et al. 2018. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of ACL*, pages 515–526.

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 ACL*, pages 258–268.

Ruihong Huang and Ellen Riloff. 2012a. Bootstrapped training of event extraction classifiers. In *Proceedings of EACL*, pages 286–295.

Ruihong Huang and Ellen Riloff. 2012b. Modeling textual cohesion for event extraction. In *Proceedings of AAAI*, pages 1664–1670.

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

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*, pages 73–82.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *Proceedings of COLING*, pages 680–688.

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of ACL*, pages 789–797.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Nugget proposal networks for Chinese event detection. In *Proceedings of ACL*, pages 1565–1574.

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

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018b. Exploiting contextual information via dynamic memory network for event detection. In *Proceedings of EMNLP*, pages 1030–1035.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016a. Leveraging Framenet to improve automatic event detection. In *Proceedings of ACL*, volume 1, pages 2134–2143.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of ACL*, pages 1789–1798.

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 AAAI*, pages 2993–2999.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2016. Distributional smoothing with virtual adversarial training. *Proceedings of ICLR*.

Aldrian Obaja Muis, Naoki Otani, Nidhi Vyas, Ruochen Xu, Yiming Yang, Teruko Mitamura, and Eduard Hovy. 2018. Low-resource cross-lingual event type detection via distant supervision with minimal effort. In *Proceedings of COLING*, pages 70–82.

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of AAAI*, pages 5900–5907.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 365–371.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. DSGAN: Generative adversarial training for distant supervision relation extraction. In *Proceedings of ACL*, pages 496–505.

Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of ICLR*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Evan Sandhaus. 2008. The New York Times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Xiaozhi Wang, Xu Han, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Adversarial multi-lingual neural relation extraction. In *Proceedings of COLING*, pages 1156–1166.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of EMNLP*, pages 1778–1783.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. *Proceedings of ICLR*.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of NAACL*, pages 289–299.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: a document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL*, pages 50–55.

Hui Yang, Tat-Seng Chua, Shuguang Wang, and Chun-Keat Koh. 2003. Structured use of external knowledge for event-based open domain question answering. In *Proceedings of SIGIR*, pages 33–40.

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 CIKM*, pages 587–596.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, pages 1753–1762.

Ying Zeng, Yansong Feng, Rong Ma, and Zheng Wang. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of AAAI*, pages 878–891.

Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. 2017. Improving event extraction via multimodal integration. In *Proceedings of MM*, pages 270–278.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of ACL*, pages 414–419.