Improving Event Detection via Open-domain Trigger Knowledge

Meihan Tong[1], **Bin Xu**[1], **Shuai Wang**[2], **Yixin Cao**[3]*, **Lei Hou**[1], **Juanzi Li**[1] **and Jun Xie**[4]
[1]Knowledge Engineering Laboratory, Tsinghua University, Beijing, China
[2]SLP Group, AI Technology Department, JOYY Inc, China
[3]National University of Singapore, Singapore
[4]SPPD Group, Tencent Inc, China
tongmeihan@gmail.com, xubin@tsinghua.edu.cn
wangshuai1@yy.com, caoyixin2011@gmail.com
greener2009@gmail.com, lijuanzi@tsinghua.edu.cn
stiffxie@tencent.com

Abstract

Event Detection (ED) is a fundamental task in automatically structuring texts. Due to the small scale of training data, previous methods perform poorly on unseen/sparsely labeled trigger words and are prone to overfitting densely labeled trigger words. To address the issue, we propose a novel Enrichment Knowledge Distillation (EKD) model to leverage external open-domain trigger knowledge to reduce the in-built biases to frequent trigger words in annotations. Experiments on benchmark ACE2005 show that our model outperforms nine strong baselines, is especially effective for unseen/sparsely labeled trigger words. The source code is released on https://github.com/shuaiwa16/ekd.git.

1 Introduction

Event Detection (ED) aims at detecting trigger words in sentences and classifying them into predefined event types, which shall benefit numerous applications, such as summarization (Li et al., 2019) and reading comprehension (Huang et al., 2019). For instance, in S1 of Figure 1, ED aims to identify the word *fire* as the event trigger and classify its event type as *Attack*. Mainstream researches (Chen et al., 2015; Liu et al., 2017, 2018b; Liao and Grishman, 2010b; Zhao et al., 2018; Liu et al., 2018a) focus on the second step event type disambiguation via lexical and contextual features. However, it is also crucial to identify trigger words correctly as the preliminary step.

Trigger word identification is a non-trivial task, which suffers from the long tail issue. Take the benchmark ACE2005 as an example: trigger words with frequency less than 5 account for 78.2% of the total. The long tail issue makes supervised methods (Li et al., 2013; Yang et al., 2019) prone to overfitting and perform poorly on unseen/sparsely labeled triggers (Lu et al., 2019). Automatically generating more training instances seems to be a solution: expanding more instances by bootstrapping (Ferguson et al., 2018; Zhang et al., 2019; Cao et al., 2019) and expending more data from distantly supervised methods (Chen et al., 2017; Wang et al., 2019a). However, the performance of these methods on unseen/sparsely labeled trigger words is still unsatisfied, as shown in Table 1. We argue that these methods either lead to the homogeneity of the generated corpus, or subject to the low coverage of knowledge base. More importantly, the expanded data itself is unevenly distributed, and we cannot expect to alleviate the long tail problem with built-in bias data.

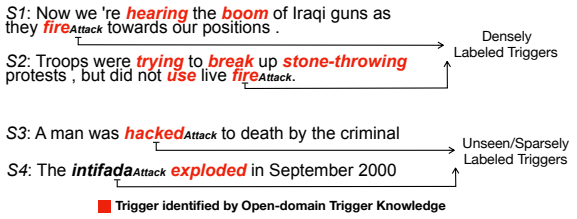

Figure 1: Examples of ED. *fire* is the densely labeled trigger for *Attack* event in ACE2005. *Hacked* and *intifada* are the unseen/sparsely labeled triggers in the training corpus. The red ones illustrate the triggers identified by open-domain trigger knowledge.

In the paper, we empower the model with external knowledge called **Open-Domain Trigger Knowledge** to provides extra semantic support on unseen/sparsely labeled trigger words and improve trigger identification. Open-Domain Trig-

*Corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5887–5897

Table 1: F score on unseen/sparsely and densely labeled triggers. DMBERT (Chen et al., 2015) refers to a supervised-only model with dynamic multi-pooling to capture contextual features; BOOTSTRAP (He and Sun, 2017) expands training data via bootstrapping. DGBERT expands training data with Freebase (Chen et al., 2017).

Method	Unseen	Sparse	Dense
DMBERT$_{sup-only}$	54.4	72.5	84.1
BOOTSTRAP$_{semi-sup}$	56.6	73.6	86.9
DGBERT$_{distant-sup}$	54.7	72.8	84.3

ger Knowledge is defined as a prior that specifies which words can trigger events without subject to pre-defined event types and the domain of texts. As shown in S1 of Figure 1, open-domain trigger knowledge can identify that *hearing* and *fire* as event triggers, even if *hearing* does not fit into any pre-defined event types in ACE2005. With open-domain trigger knowledge, we are able to discover unseen/sparsely triggers from the large-scale unlabeled corpus, which will improve the recall in trigger words identification. However, it is challenging to incorporate open-domain trigger knowledge into ED: Triggers identified by open-domain trigger knowledge do not always fit well with in-domain labels, and thus can not be directly adopted as the trigger identification result. For example in S4 of Figure 1, open-domain trigger knowledge argues that *exploded* is the trigger word, while under the labeling rules of ACE2005, *intifada* is the trigger word.

Specifically, we propose an Enrichment Knowledge Distillation (EKD) model to efficiently distill open-domain trigger knowledge from both labeled and abundant unlabeled corpora. We first apply a light-weight pipeline to equipment unlabeled sentences with trigger knowledge from WordNet. The method is not limited to specific domains, and thus can guarantee the coverage of trigger words. Then, given the knowledge enhanced data as well as ED annotations, we train a teacher model for better performance; meanwhile, a student model is trained to mimic teacher's outputs using data without knowledge enhancement, which conforms to the distribution during inference. We further promote the generalization of the model by adding noise to the inputs of the student model.

We evaluate our model on the ACE2005 ED benchmark. Our method surpasses nine strong baselines, and is especially effective for unseen/sparsely labeled triggers word. Experiments also show that the proposed EKD architecture is very flexible, and can be conveniently adapted to distill other knowledge, such as entity, syntactic and argument.

Our contributions can be summarized as:

- To the best of our knowledge, we are the first to leverage the wealth of the open-domain trigger knowledge to improve ED.
- We propose a novel teacher-student model (EKD) that can learn from both labeled and unlabeled data, so as to improve ED performance by reducing the in-built biases in annotations.
- Experiments on benchmark ACE2005 show that our method surpasses nine strong baselines which are also enhanced with knowledge. Detailed studies show that our method can be conveniently adapted to distill other knowledge, such as entities.

2 Related Work

2.1 Event Detection

Traditional feature-based methods exploit both lexical and global features to detect events (Li et al., 2013). As neural networks become popular in NLP (Cao et al., 2018), data-driven methods use various superior DMCNN, DLRNN and PLMEE model (Duan et al., 2017; Nguyen and Grishman, 2018; Yang et al., 2019) for end-to-end event detection. Recently, weakly-supervised methods (Judea and Strube, 2016; Huang et al., 2017; Zeng et al., 2018; Yang et al., 2018) has been proposed to generate more labeled data. (Gabbard et al., 2018) identifies informative snippets of text as expending annotated data via curated training. (Liao and Grishman, 2010a; Ferguson et al., 2018) rely on sophisticated pre-defined rules to bootstrap from the paralleling news streams. (Wang et al., 2019a) limits the data range of adversarial learning to trigger words appearing in labeled data. Due to the long tail issue of labeled data and the homogeneity of the generated data, previous methods perform badly on unseen/sparsely labeled data and turn to overfitting densely labeled data. With open-domain trigger knowledge, our model is able to perceive the unseen/sparsely labeled trigger words from abundant unlabeled data, and thus successfully improve the recall of the trigger words.

2.2 Knowledge Distillation

Knowledge Distillation, initially proposed by (Hinton et al., 2015), has been widely adopted in NLP to distill external knowledge into the model (Laine and Aila, 2016; Saito et al., 2017; Ruder and Plank, 2018). The main idea is to adopt a student model to learn from a robust pre-trained teacher model. (Lee et al., 2018; Gong et al., 2018) reinforces the connection between teacher and student model by singular value decomposition and the laplacian regularized least squares. (Tarvainen and Valpola, 2017; Huang et al., 2018) stabilize the teacher model by a lazy-updated mechanism to enable student model not susceptible to external disturbances. (Liu et al., 2019) uses an adversarial imitation approach to enhance the learning procedure. Unlike previous methods that relied on golden annotations, our method is able to learn from pseudo labels and effectively extract knowledge from both labeled and unlabeled corpus.

3 Methodology

In the section, we introduce the proposed Enrichment Knowledge Distillation (EKD) model, which leverages open-domain trigger knowledge to improve ED. In general, we have a teacher model and a student model. The teacher is fully aware of open-domain trigger knowledge, while the student is not equipped with open-domain trigger knowledge. We make the student model to imitate the teacher's prediction to distill the open-domain trigger knowledge to our model. Figure 2 illustrates the architecture of the proposed EKD model. During training, we first pre-train the teacher model on labeled data, and then force the student model, under the knowledge-absent situation, to generate pseudo labels as good as the teacher model on both labeled and unlabeled data. By increasing the cognitive gap between teacher and student model, the student model has to learn harder.

We first introduce how to collect the open-domain trigger knowledge in Knowledge Collection. We then illustrate how to exploit the labeled data to pre-train the teacher model in Feature Extraction and Event Prediction. Finally, we elaborate on how to force the student model to learn from the teacher model in Knowledge Distillation.

3.1 Notation

Given the labeled corpus $L = \{(S_i, Y_i)\}_{i=1}^{N_L}$ and abundant unlabeled corpus $U = \{(S_k)\}_{k=N_L+1}^{N_T}$, our goal is to jointly optimize two objections: 1) maximize the prediction probability $P(Y_i|S_i)$ on labeled corpus L, 2) minimize the prediction probability discrepancy between the teacher $P(Y_k^{'}|S_k^{+})$ and student model $P(Y_k^{'}|S_k^{-})$ on both L and U, where N_T stand for the total number of sentences in both labeled and unlabeled data. S^+ and S^- stand for the enhanced and weakened variant of the raw sentence S, we will explain them in detail in the Section 3.5. $Y = \{y_1, y_2, \ldots, y_n\}$ stands for the golden event type label, where each $y \in Y$ belongs to the 33 event types pre-defined in ACE and a "NEGATIVE" event type (Chen et al., 2015; Nguyen et al., 2016; Feng et al., 2018). $Y^{'}$ is the pseudo label proposed by pre-trained teacher model.

3.2 Knowledge Collection

Open-domain trigger knowledge elaborates whether a word triggers an event from the perspective of word sense. Whether the trigger is densely labeled or unseen/sparsely labeled, open-domain trigger knowledge will identify them without distinction. For instance in S3 in Figure 1, although *hacked* is a rare word and has not been labeled, judging from word sense, open-domain trigger knowledge successfully identifies *hacked* as a trigger word.

We adopt a light-weight pipeline method, called Trigger From WordNet (TFW), to collect open-domain trigger knowledge (Araki and Mitamura, 2018).

$$S^+ = TFW(S) \tag{1}$$

TFW uses WordNet as the intermediary. It has two steps, 1) disambiguate word into WordNet sense, 2) determine whether a sense triggers an event. For the first step, we adopt IMS (Zhong and Ng, 2010) to disambiguate word into word sense in WordNet (Miller et al., 1990). We obtain the input features by POS tagger and dependency parser in Stanford CoreNLP (Manning et al., 2014). For the second step, we adopt the simple dictionary-lookup approach proposed in (Araki and Mitamura, 2018) to determine whether a sense triggers an event. TFW is not limited to particular domains, which is able to provide unlimited candidate triggers. With the support of the lexical database, TFW has high efficiency and can be applied to large-scale knowledge collection.

Finally, we obtain a total of 733,848 annotated sentences from New York Times (Sandhaus, 2008)

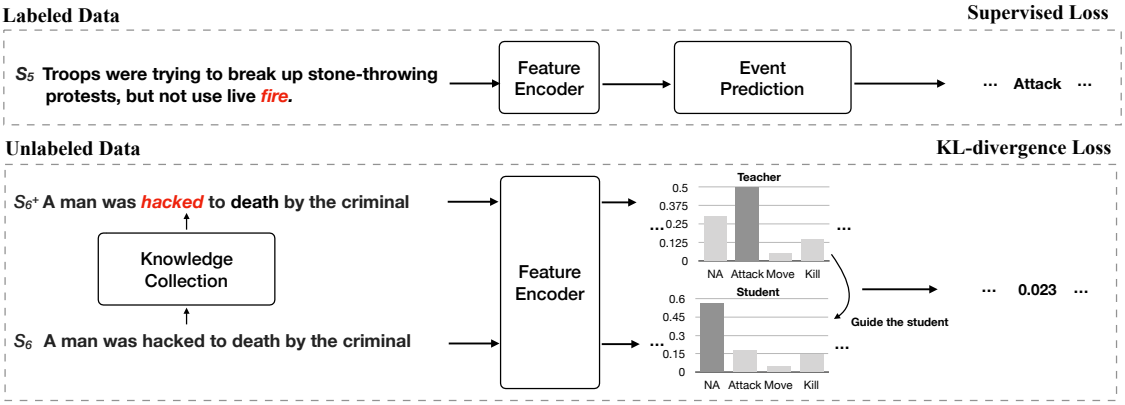

Figure 2: The architecture of the proposed EKD model. Besides the supervised signals, EKD exploits abundant unlabeled data by ensuring the prediction consistency of raw sentence and knowledge-attending sentence.

corpus in the first half of 2007. The total number of triggers is 2.65 million, with an average of 3.6 triggers per sentence.

3.3 Feature Extraction

We adopt BERT to obtain the hidden representation for both labeled and unlabeled sentences. BERT is a pre-trained language representation model, and BERT has achieved SOTA performance on a wide range of tasks, such as question answering and language inference. The powerful capability of BERT has also been demonstrated in ED scenario (Wang et al., 2019a).

Formally, given the raw sentence S and knowledge-attending sentence S_+, we feed them into BERT respectively, and adopt the sequence output of the last layer as the hidden representation for each word in S and S^+.

$$\begin{aligned} H &= BERT(S) \\ H_+ &= BERT(S_+) \end{aligned} \tag{2}$$

3.4 Event Prediction

After obtaining the hidden representation of sentencen S, we adopt a full-connected layer to determine the event type Y for each word in sentence S.

We use $S_{(i)}$ and $Y_{(i)}$ to denote the i-th training sentence and its event type in labeled corpus L. We first transform the hidden representation H obtained from Section 3.3 to a result vector O, where O_{ijc} represents the probability that the j-th word in S_i belongs to the c-th event class. And then we normalize O by the softmax function to obtain the conditional probability.

$$p(Y_{(i)}|S_{(i)},\theta) = \sum_{j=1}^{n} \frac{exp(O_{ijc})}{\sum_{c=1}^{C} exp(O_{ijc})}/n \tag{3}$$

Given the labeled corpus $L = \{S_i, Y_i\}|_{i=1}^{N_L}$, the optimization object is defined as:

$$J_L(\theta) = -\sum_{i=1}^{N_L} \log p(Y_{(i)}|S_{(i)},\theta) \tag{4}$$

3.5 Knowledge Distillation

In this section, we distill open-domain trigger knowledge into our model. The main idea is to force the student model, with only raw texts as the input, to generate as good pseudo labels as the teacher model on both labeled and unlabeled data.

Formally, given golden event type Y, the objective is:

$$p(Y|S^+\theta) = p(Y|S^-,\theta) \tag{5}$$

where $p(Y|S^+\theta)$ and $p(Y|S^-,\theta)$ are the predictions from the teacher and student model respectively.

We share the parameters of the teacher and student model. The input of teacher model S^+ is aware of the open-domain trigger knowledge, and the input of student model S^- does not know. We give the detailed construction process of S^+ and S^- below.

Knowledge-attending Sentences (S^+) We embed the open-domain trigger knowledge into the sentence by *Marking Mechanism*. Specifically, we introduce two symbols, named *B-TRI* and *E-TRI* to mark the beginning and ending boundary of triggers identified by open-domain trigger knowledge. Formally, given the raw sentence $S = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$ and trigger w_i identified by open-domain trigger knowledge, the knowledge-attending sentence is $S^+ =$

$\{w_1, w_2, \ldots, \textit{B-TRI}, w_i, \textit{E-TRI}, \ldots, w_n\}$. Marking mechanism works well for our feature extractor BERT (Soares et al., 2019), which is very flexible in embedding knowledge, and can be conveniently adapted to other types of knowledge without heavily-engineered work.

Note that the newly added symbols are lack of pre-trained embedding in BERT. Random initialization undermines the semantic meaning of the introduced symbols, where *B-TRI* indicates the beginning of a trigger, and *E-TRI* means the ending. We address the issue by fine-tuning BERT on the annotation sentences in Section 3.2. Specifically, we adopt Masked LM task (Devlin et al., 2018) to exploit surrounding words to learn the semantic representation of the introduced symbols (*B-TRI* and *E-TRI*) based on the Harris distributional hypothesis (Harris, 1954). The mask word rate is set to 0.15 and the accuracy of masked words achieves 92.3% after fine-tune.

Knowledge-absent Sentences (S^-) To make the student model learn harder from the teacher model, we further disturb the input of student model by randomly masking out triggers identified by open-domain trigger knowledge. In this way, the student model has to judge the event type of trigger word solely based on the surrounding context. Formally, given the raw sentence $S = \{w_1, w_2, \ldots, w_i, \ldots, w_n\}$ and trigger w_i identified by open-domain trigger knowledge, the knowledge-absent sentence is $S^- = \{w_1, w_2, \ldots, \textit{[MASK]}, \ldots, w_n\}$. The mask words are not randomly selected, but among triggers determined by open-domain trigger knowledge, avoiding the model is optimized only for the non-trigger negative class.

KL-divergence Loss We move the added symbols to the end of the sentence to ensure strict alignment of words in S^+ and S^-, and then we minimize the discrepancy between conditional probability $p(Y|S^-, \theta)$ and $p(Y|S^+\theta)$ with KL-divergence loss. Given the collection of labeled and unlabeled corpus $T = \{(S_k)\}_{k=1}^{N_L+N_U}$, the KL-divergence loss is:

$$\begin{aligned} J_T(\theta) &= \mathbf{KL}(p(Y|S^+, \theta)||p(Y|S^-, \theta)) \\ &= \sum_{k=1}^{N_L+N_U} p(Y_{(k)}|S^+_{(k)}, \theta) \frac{p(Y_{(k)}|S^+_{(k)}, \theta)}{p(Y_{(k)}|S^-_{(k)}, \theta)} \end{aligned} \tag{6}$$

KL divergence is asymmetric in the two distributions. We treat predictions from knowledge-absent inputs as approximate distributions and predictions from knowledge-attending inputs as approximated distributions. If we reverse the direction of approximation, the experimental results decline significantly. The reason may be that we should ensure the low-confidence predictions approximate the high-confidence predictions.

3.6 Joint Training

The final optimization objection is the integration of the supervised loss from labeled dataset and KL-divergence loss from unlabeled dataset defined in Equation 4 and 6.

$$J(\theta) = J_L(\theta) + \lambda * J_T(\theta) \tag{7}$$

We stop the gradient descent of teacher model when calculating J_T to ensure that the learning is from teacher to student.

Since unlabeled data is much larger than the labeled data, joint training leads the model quickly overfitting the limited labeled data while still underfitting the unlabeled data. To handle the issue, we adopt the Training Signal Annealing (TSA) technique proposed in (Xie et al., 2019) to linearly release the 'training signals' of the labeled examples as training progresses.

4 Experiment

4.1 Experiment Setup

Datasets For the labeled corpus, we adopt dataset ACE2005 to evaluate the overall performance. ACE2005 contains 13,672 labeled sentences distributed in 599 articles. Besides the pre-defined 33 event types, we incorporate an extra "Negative" event type for non-trigger words. Following (Chen et al., 2015), we split ACE2005 into 529/30/40 for train/dev/test respectively.

Evaluation We report the Precision, Recall and micro-averaged F1 scores in the form of percentage over all 33 events. A trigger is considered correct if both its type and offsets match the annotation.

Hyperparameters For feature extraction, we adopt BERT as our backbone, which has 24 16-head attention layers and 1024 hidden embedding dimension. For the batch size, The batch size of labeled data is 32, and we set the proportion of labeled and unlabeled data to 1:6. For most of our experiments, we set the learning rate 3e-5, the maximum sequence length 128 and the λ in joint training 1. Our model trains on one V100 for a

half day. The best result appears around 12,500 epochs. Balancing the performance and training efficiency, we actually use 40,236 unlabeled data for knowledge distillation unless otherwise stated. All reported results are the average results of ten runs. We use Adam as the gradient descent optimizer.

Baselines As our methods incorporate open-domain trigger knowledge, for fair competition, we compare our methods with two data-driven methods and five state-of-the-art knowledge-enhanced methods, including: **DMCNN** proposes a dynamic multi-pooling layer above CNN model to improve event detection (Chen et al., 2015). **DLRNN** exploits document information via recurrent neural networks (Duan et al., 2017). **ANN-S2** exploits argument information to improve ED via supervised attention mechanisms (Liu et al., 2017).**GMLATT** adopts a gated cross-lingual attention to exploit the complement information conveyed by multilingual data (Liu et al., 2018a). **GCN-ED** exploits structure dependency tree information via graph convolutions networks and entity mention-guided pooling (Nguyen and Grishman, 2018). **Lu's DISTILL** proposes a -learning approach to distill generalization knowledge to handle overfitting (Lu et al., 2019). **TS-DISTILL** exploits the entity ground-truth and uses an adversarial imitation based knowledge distillation approach for ED (Liu et al., 2019). **AD-DMBERT** adopts an adversarial imitation model to expend more training data (Wang et al., 2019b). **DRMM** employs an alternative dual attention mechanism to effectively integrate image information into ED (Tong et al., 2020). The last two baselines both use BERT as feature extractor.

4.2 Overall Performance

Table 2: Overall Performance on ACE2005 dataset (%). The results of baselines are adapted from their original papers.

Method	Precision	Recall	F1
DMCNN	75.6	63.6	69.1
DLRNN	77.2	64.9	70.5
ANN-S2	78.0	66.3	71.7
GMLATT	78.9	66.9	72.4
GCN-ED	77.9	68.8	73.1
Lu's DISTILL	76.3	71.9	74.0
TS-DISTILL	76.8	72.9	74.8
AD-DMBERT	77.9	72.5	75.1
DRMM	77.9	74.8	76.3
EKD (Ours)	**79.1**	**78.0**	**78.6**

Table 2 presents the overall performance of the proposed approach on ACE2005. As shown in Table 2, EKD (our) outperforms various state-of-the-art models, showing the superiority of open-domain trigger knowledge and the effectiveness of the proposed teacher-student model. BERT-based models AD-DMBERT, DRMM and EKD (ours) significantly outperform the CNN-based or LSTM-based models, which is due to the ability to capture contextual information as well as large scale pre-training of BERT. Compared to these BERT-based models, our methods consistently improves the F score by 3.5% and 2.3%, which shows the superiority of our method even if the encoder is powerful enough.

Compared to data-driven methods DMCNN and DLRNN, knowledge enhanced methods Lu's DISTILL, TS-DISTILL and EKD (ours) improve the recall by a large margin. Due to the small scale of ACE2005, it is quite tricky to disambiguate triggers solely based on the surrounding context. Enhanced by external knowledge, these methods have a stand-by commonsense to depend on, which prevents from overfitting densely labeled trigger words and thus can discover more trigger words. Among them, our model achieves the best performance, which may be caused by two reasons: 1) The superiority of open-domain trigger knowledge. Compared to general linguistic knowledge used in Lu's DISTILL and entity type knowledge used in TS-DISTILL, open-domain trigger knowledge is more task-related, which directly provides trigger candidates for trigger identification, and thus is more informative. 2) The superiority of the proposed teacher-student model. Our method is able to learn open-domain trigger knowledge from unlimited unlabeled data, while Lu's DISTILL and TS-DISTILL can only learn from labeled data.

It is worth noting that our model simultaneously improves precision. Unseen/sparsely labeled trigger words are usually rare words, which are typically monosemous and exhibiting a single clearly defined meaning. These words are easier for the model to distinguish, thereby resulting in the improvement of the overall precision.

To evaluate whether EKD has distilled knowledge into model, we report the performance of EKD in the test set with and without knowledge. As illustrated in Table 3, whether the input data masters the open-domain knowledge or not, the performance makes no big difference (78.4% vs 78.6%), which shows EKD (our) already distills

the knowledge into the model. During testing, our model needs no more engineering work for knowledge collection.

Table 3: Performance of test set with or without open-domain trigger knowledge

Test Set	P	R	F
without knowledge	78.8	78.1	78.4
with knowledge	79.1	78.0	78.6

4.3 Domain Adaption Scenario

We use ACE2005 to simulate a domain adaption scenario. ACE2005 is a multi-domain dataset, with six domains: broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and webblogs (wl). Following the common practice (Plank and Moschitti, 2013; Nguyen and Grishman, 2014), we adopt the union of bc and nw as source domains, and bc, ct, wl as three target domains. The event types and vocabulary distribution are quite different between the source and target domains (Plank and Moschitti, 2013). For evaluation, we split source domain data into train/test 4:1 and report the average results on ten runs as the final result. For baselines, MaxEnt and Joint (Li et al., 2013) are two feature-enriched methods, exploiting both lexical and global features to enhance the domain adaption ability. Nguyen's CNN (Nguyen and Grishman, 2015) integrates the feature and neural approaches and proposes a joint CNN for domain adaption. We also compare with supervised SOTA PLMEE (Yang et al., 2019), which exploits the pre-trained language model BERT for event extraction.

As illustrated in Table 4, our method achieves the best adaptation performance on both bc and wl target domains and achieve comparable performance on cts target domain. The superior of domain adaption may come from the open-domain trigger knowledge. The open-domain trigger knowledge is not subject to specific domains, which will detect all the event-oriented trigger words and cover the event type from both the source and the target domains. Armed with open-domain trigger knowledge, our model reinforces associations between source and target data, and thus has superior performance in domain adaption.

4.4 Various Labeling Frequencies

In the section, we answer the question whether our model can address the long tail problem. According to the frequency in the training set, we divide trigger words into three categories: *Unseen*, *Sparsely-Labeled* and *Densely-Labeled*. The frequency of *Sparsely-Labeled* is less than 5 and the frequency of *Densely-Labeled* is more than 30. The baselines are 1) supervised-only method DMBERT (Chen et al., 2015), 2) distant-supervised method DGBERT (Chen et al., 2017) and 3) semi-supervised method BOOTSTRAP (He and Sun, 2017). We replace the encoders in the three baselines to more powerful BERT to make the baseline stronger.

As illustrated in Table 5, all the three baselines show a significant performance degradation in unseen/sparsely labeled scenarios due to the limited training data. Our method surpasses the baselines in all three settings. Especially, our method gains more improvement on unseen (+6.1%) and sparsely-labeled settings (+2.8%). Open-domain trigger knowledge allows us to discover unseen/sparsely triggers from the large-scale unlabeled corpus, which increases the frequency at which the model sees unseen/sparsely triggers.

4.5 Knowledge-Agnostic

Then, to evaluate whether EKD (ours) can distill other knowledge types, we conduct experiments on the three most commonly used knowledge in ED scenario: 1) Entity knowledge. Entity type is an important feature for trigger disambiguation in ED (Zhang et al., 2007). We compare with (Liu et al., 2019), which distills ground-truth entity type knowledge via an adversarial teacher-student model. 2) Syntactic knowledge. Syntactic knowledge is implied in the dependency parse tree. The closer in tree, the more important of the word for the trigger (McClosky et al., 2011). Our baseline (Nguyen and Grishman, 2018) is the best syntactic knowledge enhanced model, which exploits structure dependency tree information via graph convolutions networks. 3) Argument knowledge. Event arguments play an important role in ED. Our baseline ANN-S2 (Liu et al., 2017) designs a supervised attention to leverage the event argument knowledge.

For the adaption of our model, we obtain entity annotations by Stanford CoreNLP, syntactic by NLP-Cube(Boro et al., 2018) and argument by CAMR (Wang et al., 2015). The marking contents are: 1) For entity, we tag three basic entity types, *People*, *Location* and *Organization*. 2) For

Table 4: Performance on domain adaption. We train our model on two source domains bn and nw, and test our model on three target domains bc, cts and wl.

Methods	In-Domain (bn+nw)			bc			cts			wl		
	P	R	F	P	R	F	P	R	F	P	R	F
MaxEnt	74.5	59.4	66.0	70.1	54.5	61.3	66.4	49.9	56.9	59.4	34.9	43.9
Joint	73.5	62.7	67.7	70.3	57.2	63.1	64.9	50.8	57.0	59.5	38.4	46.7
Nguyen's CNN	69.2	67.0	68.0	70.2	65.2	67.6	68.3	58.2	62.8	54.8	42.0	47.5
PLMEE	77.1	65.7	70.1	72.9	67.1	69.9	70.8	64.0	67.2	62.6	51.9	56.7
EKD (ours)	**77.8**	**76.1**	**76.9**	**80.8**	65.1	**72.1**	**71.7**	61.3	66.1	**69.0**	49.9	**57.9**

Table 5: Performance of our method on various labeling frequencies trigger words.

Methods	Unseen			Sparsely Labeled			Densely Labeled		
	P	R	F	P	R	F	P	R	F
DMBERT$_{supervised-only}$	66.7	45.9	54.4	74.4	70.7	72.5	84.8	83.5	84.1
DGBERT$_{distant-supervised}$	76.5	42.6	54.7	75.7	70.1	72.8	85.9	83.8	84.3
BOOTSTRAP$_{semi-supervised}$	73.7	45.9	56.6	76.0	71.3	73.6	90.6	83.5	86.9
EKD (ours)	**79.0**	**52.0**	**62.7**	**80.8**	**72.4**	**76.4**	**92.5**	82.2	**87.1**

syntactic, we take the first-order neighbor of trigger word on dependency parse tree. We consider neighbors in both directions. 3) For argument, we focus on the words played as the ARG0-4 roles of the trigger in AMR parser following (Huang et al., 2017). As we do not know trigger words on unlabeled data, we use pseudo labels generated by pre-trained BERT instead. We encode the entity, syntactic and argument knowledge into sentences with the same *Marking Mechanism* in Section 3.2. To prevent information leakage, we only use that knowledge in the training procedure.

As illustrated in Table 7, Our three adaption models, EKD-Ent, EKD-Syn and EKD-Arg, consistently outperform baselines on the F score, proving that the effectiveness of EKD is independent to specific knowledge type. EKD increases the cognitive gap between teacher model and student model to maximize knowledge utilization, and the idea universally works for all types of knowledge distillation. If we compare the performances from the perspective of knowledge type, the results show that open-domain trigger knowledge (EKD) is better than the argument knowledge (EKD-Arg), and they are both superior to the entity knowledge (EKD-Ent) and syntactic knowledge (EKD-Syn). The reason might be the more task-related of the knowledge, the more informative of the knowledge. Since open-domain trigger knowledge and event argument knowledge consider the important words directly from the event sides, they are more valuable than the entity and syntactic knowledge in ED.

4.6 Case Study

We answer the question of how and when the open-domain trigger knowledge enhances the understanding of event triggers. Table 6 gives examples about how open-domain trigger knowledge affects predictions of ED. In S1, since *trek* is a rare word that never shows up in the training procedure, supervised-only method fails to recognize it. Open-domain trigger knowledge provides the priory that *trek* should be an event trigger. Coupled with pre-trained information that *trek* is similar to densely-labeled trigger words such as *move*, our model successfully recalls it. In S3, *be* is a very ambiguous word, and in most cases, *be* is not used as a trigger word in the labeled data. Supervised-only method is prone to overfitting the labeled data and fails to recognize it. Open-domain trigger knowledge owns word sense disambiguation ability, which knows that *be* here belongs to the word sense 'occupy a certain position' instead of the common word sense 'have the quality of being', and thus can successfully identify *be* as the trigger for event *Start-Position*.

5 Conclusion

We leverage the wealth of the open-domain trigger knowledge to address the long-tail issue in ACE2005. Specifically, we adopt a WordNet-based pipeline for efficient knowledge collection, and then we propose a teacher-student model, EKD, to distill open-domain trigger knowledge from both labeled and abundant unlabeled data. EKD forces the student model to learn open-domain trigger knowledge from teacher model by mimicking the

Table 6: Error analysis: How and When does the open-domain trigger knowledge improve ED? *GT* refers to the ground truth labels. On the unlabeled data, we use a majority vote of three humans as the ground truth.

Sentence	GT	Prediction	
		S	S^+
*S1: Mr. Caste leaves at 5 A.M. for a train **trek** to manhatten and does not return utill 6 P.M.*	**Transport**	**O**	**Transport**
*S2: Militants in the region escalate their attacks in the weeks leading up to the **inauguration** of Nigeria's president.*	**Start-Position**	**O**	**Start-Position**
*S3: Mr.Mason, who will **be** president of CBS radio, said that it would play to radio's strengths in delivering local news.*	**Start-Position**	**O**	**Start-Position**

Table 7: Knowledge-Agnostic.

Knowledge Type	Methods	Metrics		
		P	R	F
Entity	TS-DISTILL	76.8	72.9	74.8
	EKD-Ent	74.5	78.6	**76.5**
	improvement	-2.3	+4.7	+1.7
Syntactic	GCN-ED	77.9	68.8	73.1
	EKD-Syn	76.5	76.3	**76.4**
	improvement	-1.4	+7.5	+3.3
Argument	ANN-S2	78.0	66.3	71.7
	EKD-Arg	75.8	78.4	**77.1**
	improvement	-2.2	+23.1	+5.4

predicted results of the teacher model. Experiments show that our method surpasses seven strong knowledge-enhanced baselines, and is especially efficient for unseen/sparsely triggers identification.

6 Acknowledgments

This work is supported by the National Key Research and Development Program of China (2018YFB1005100 and 2018YFB1005101), NSFC Key Projects (U1736204, 61533018). It also got partial support from National Engineering Laboratory for Cyberlearning and Intelligent Technology, and Beijing Key Lab of Networked Multimedia. This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative.

References

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891.

Tiberiu Boro, Stefan Daniel Dumitrescu, and Ruxandra Burtica. 2018. NLP-cube: End-to-end raw text processing with neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 171–179.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *COLING*.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. In *EMNLP*.

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.

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)*, volume 1, pages 167–176.

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 the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 352–361.

Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. A language-independent neural network for event detection. *Science China Information Sciences*, 61(9):092106.

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)*, pages 359–364, New Orleans, Louisiana. Association for Computational Linguistics.

Ruan Gabbard, Jay DeYoung, and Marjorie Freedman. 2018. Events beyond ace: Curated training for events. *arXiv preprint arXiv:1809.05576*.

Chen Gong, Xiaojun Chang, Meng Fang, and Jian Yang. 2018. Teaching semi-supervised classifier via generalized distillation. In *IJCAI*, pages 2156–2162.

Zellig S. Harris. 1954. Distributional structure. *¡i¿WORD¡/i¿*, 10(2-3):146–162.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R Voss. 2017. Zero-shot transfer learning for event extraction. *arXiv preprint arXiv:1707.01066*.

Mingkun Huang, Yongbin You, Zhehuai Chen, Yanmin Qian, and Kai Yu. 2018. Knowledge distillation for sequence model. In *Interspeech*, pages 3703–3707.

Alex Judea and Michael Strube. 2016. Incremental global event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2279–2289.

Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. 2018. Self-supervised knowledge distillation using singular value decomposition. In *European Conference on Computer Vision*, pages 339–354. Springer.

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)*, volume 1, pages 73–82.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019. Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, 7:25001–25015.

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

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 789–797, Stroudsburg, PA, USA. 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6754–6761.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. *Statistics*, 1000:1250.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018b. Exploiting contextual information via dynamic memory network for event detection. *arXiv preprint arXiv:1810.03449*.

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)*, volume 1, pages 1789–1798.

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4366–4376.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

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

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.

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.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 68–74.

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. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1498–1507.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204.

Meihan Tong, Shuai Wang, Yixin Cao, Bin Xu, Juaizi Li, Lei Hou, and Tat-Seng Chua. 2020. Image enhanced event detection in news articles.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.

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

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019b. Adversarial training for weakly supervised event detection. In *NAACL*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

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 2018, System Demonstrations*, pages 50–55.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kuo Zhang, Juan Zi, and Li Gang Wu. 2007. New event detection based on indexing-tree and named entity. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 215–222. ACM.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1(2):99–120.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419.

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