Event Detection as Graph Parsing

Jianye Xie[1], Haotong Sun[1], Junsheng Zhou[1]*, Weiguang Qu[1], Xinyu Dai[2]
[1]School of Computer and Electronic Information, Nanjing Normal University, China
[2]National Key Laboratory for Novel Software Technology, Nanjing University, China
{zhoujs,wgqu}@njnu.edu.cn,daixinyu@nju.edu.cn
{xiejy,sunht}@nnu.edu.cn

Abstract

Event detection is a fundamental task in information extraction. Most previous approaches typically view event detection as a trigger-based classification problem, focusing on using syntactic dependency structure or external knowledge to boost the classification performance. To overcome the inherent issues with existing trigger classification based models, we propose a novel approach to event detection by formulating it as a graph parsing problem, which can explicitly model the multiple event correlations and naturally utilize the rich information conveyed by event type and subtype. Furthermore, to cope with data sparsity, we employ a pretrained sequence-to-sequence (seq2seq) model to transduce an input sentence into an accurate event graph without the need for trigger words. Extensive experimental results on the public ACE2005 dataset show that, our approach outperforms all previous state-of-the-art models for event detection by a large margin, obtaining an improvement of 4.2% F1 score. The result is very encouraging since we achieve this with a conceptually simple seq2seq model; moreover, by extending the graph structure, this proposed architecture can be flexibly applied to more information extraction problems for sentences.

1 Introduction

Event Detection (ED) is an important task in Information Extraction (IE) that seeks to identify instances of specified types of events in text (Ji and Grishman, 2008; Li et al., 2013). For example, for the input sentence shown in Figure 1, the ED model aims to predict three event types expressed by this sentence, each of which consists of an event type label and its subtype label, according to the ACE2005 Guidelines.

Sentence: *British found resistance softer than expected, picked up reports that the local Baath Party leadership was crumbling and fought into the core, losing at least three soldiers.*
Event types: *Business: End-Org, Conflict: Attack, Life: Die*

Figure 1: An example of event detection

ED is an actively studied task in IE where deep learning models have been the dominant approach to deliver the state-of-the-art performance (Chen et al., 2015; Nguyen et al., 2016; Sha et al., 2018; Chen et al., 2018). The last few years witness the success of graph convolutional neural networks for ED (Nguyen and Grishman, 2018; Liu et al., 2018; Yan et al., 2019; Lai et al., 2020) where the dependency trees are employed to boost the performance. Also, another line of research focused on exploiting external knowledge to improve classification (Lu et al., 2019; Liu et al., 2019a; Tong et al., 2020).

Nevertheless, most previous work typically treats ED as the identification and classification of trigger words, focusing on using various syntactic dependency structure or external knowledge to boost classification performance. Methodologically speaking, this type of trigger-based ED models suffer from the following inherent drawbacks.

Firstly, most previous approaches depend heavily on the trigger word. On the one hand, triggers are nonessential to event detection (Liu et al., 2019b); On the other hand, the identification and classification of trigger words may, to some extent, hinder the accurate recognition of the events, due to the fact that some events may be expressed by multiple discontinuous words or phrases in one sentence (See more illustrations in Section 5.4). Particularly, the trigger-based ED models are prone to suffer from the long tail issue (Tong et al., 2020). Literatures available show that, (Liu et al., 2019b)

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1630–1640

is the only work for ED without using trigger words. However, in the absence of trigger, (Liu et al., 2019b) simply cast ED as a multi-label classification problem for input sentences, which cannot address the inherent issues with the trigger-based approaches, as illustrated below.

Secondly, current ED models cannot explicitly model the correlations between multiple events in one sentence. In fact, the multiple event phenomenon has been investigated by some existing works (Chen et al., 2018; Liu et al., 2018), which explore to aggregate more contextual information from surrounding words to generate a powerful representing vector for each candidate trigger by employing a self-attention mechanism or a hierarchical tagging scheme, then respectively predict the trigger label. However, note that modeling the associations between triggers is not equivalent to modeling the correlations between events. That is to say, the current models cannot explicitly model the correlations between multiple events.

Lastly, the existing approaches cannot leverage the hierarchical structure information of event type and subtype. The two kinds of event type labels contain the information with different granularity. Intuitively, the event type-level information can be used to guide the subtype-level classification. Furthermore, the event type or subtype label itself also conveys explicit semantic information that may be conducive to event prediction. However, the rich information is neglected by the existing approaches.

To address the three problems stated above simultaneously, we take a fresh look at this problem and formulate ED, for the first time, as a graph parsing problem. By regarding the multiple events expressed by one sentence as a whole, we argue that the goal of ED task is to output an event graph, as shown in Figure 2. On the one hand, the event graph is constructed to model the potential interactions between the multiple events; on the other hand, this graph structure can flexibly integrate more event type information. Furthermore, under the graph parsing formulation, we employ a seq2seq model for event graph parsing, without the need for the identification of the trigger words. In particular, to cope with the data sparsity problem with the ED task, we adopt a pretrained seq2seq model, BART (Lewis et al., 2020), to accurately generate the event graph. Experimental results demonstrate that our method substantially outperforms all previous state-of-the-art models on the public dataset ACE2005.

To sum up, this paper makes the following contributions:

1. To the best of our knowledge, it is the first time to formulate the event detection task as graph parsing, which delivers some typical benefits compared to the existing ED models. First, this graph parsing formulation can explicitly model the correlations between multiple events in one sentence; second, the event graph can be flexibly constructed to reflect the hierarchical structure of event type and utilize the semantic representations of the event type labels.

2. We further propose a novel generation-based approach to predict the event graph via a seq2seq model. The proposed transducer can directly derive the events from the global contextual information in the input sentences, without being limited to the representations of trigger words. Particularly, we employ a pretrained model, BART (Lewis et al., 2020), to generate a linearized event graph, thereby addressing the data sparsity.

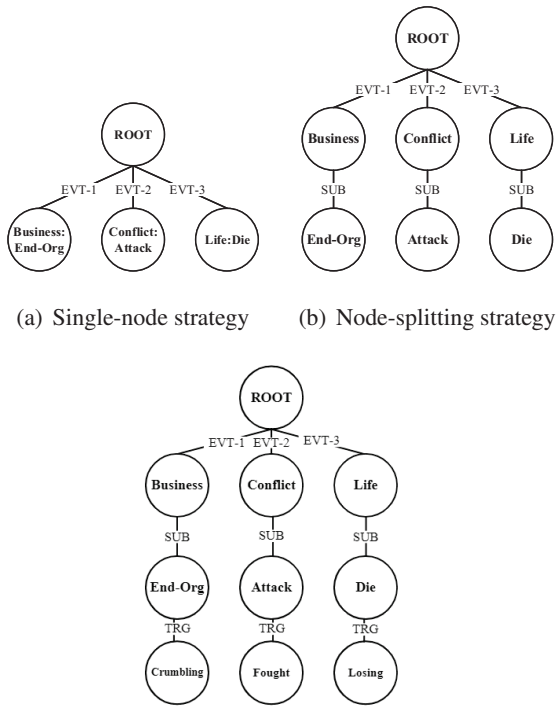
3. The extensive experiments over the public dataset ACE2005 demonstrate that our approach outperforms the previous state-of-the-art models for ED by a large margin, without using the syntactic dependency information and any external trigger knowledge.

Encouragingly, the proposed graph parsing paradigm is not limited to the ED task. By modifying the graph structure according to the target task, this graph parsing formulation can be flexibly applied to more information extraction problems, such as event extraction, relation extraction and so on.

2 A Novel View of Event Detection

2.1 Task Description

Event detection task requires that certain specified types of events, which are mentioned in the annotated data, to be detected. The most common used benchmark dataset in previous work is ACE 2005 corpus. The task defines 8 event types such as Life, Business and so on, and 33 subtypes such as Attack, End-Position, etc. By the ACE annotation guideline, every event mention is annotated with a



(a) Single-node strategy (b) Node-splitting strategy

(c) Trigger generation strategy

Figure 2: Three different event graphs constructed using the three different strategies, for the instance in Figure 1.

trigger that is a key word or phrase that most clearly expresses the event occurrence.

2.2 Formulating ED as Graph Parsing

Traditionally, the goal of ED consists of identifying trigger words (trigger identification) and classifying them for the event types of interest (event classification) (Nguyen and Grishman, 2018; Liu et al., 2018; Lai et al., 2020; Tong et al., 2020). However, as pointed out by (Liu et al., 2019b), triggers are nonessential to event detection. *More importantly*, some events may be triggered by multiple discontinuous words or phrases in one sentence, not by a single word or single phrase.

Take a concrete example in ACE 2005 dataset to illustrate: *She lost her seat in the 1997 election.* In this sentence, an event type (*Personnel:Elect*) is mentioned, and its gold trigger was labelled as the word *lost*. In effect, to correctly recognize the event type (*Personnel:Elect*) from this sentence, we should comprehensively consider both the phrase *lost her seat* and the word *election* in the sentence (See more cases in Section 5.4). Therefore, it does not seem plausible that the problem of predicting an event from a whole sentence is reduced to the representation learning of the single trigger word for trigger classification or sequence labelling.

In this paper, we look at the ED task from a new perspective. Given an input sentence, ED aims to recognize and predict the mentioned event types. Intuitively, the multiple events derived from the same sentence should have a certain degree of correlations between them. Therefore, to model the correlations, we can view the multiple events corresponding to a sentence as a whole, by linking them together as a single graph, as shown in Figure 2(a). To be specific, we first introduce a special node as the root, and then attach each event type node as a child of the root. It is worth to note that *the root of this event graph is not a virtual node*. The root can take two possible values: *EVTS* and *NA*. While the input sentence does not contain any event, the root is assigned the value *NA*; otherwise it is assigned the value *EVTS*. Therefore, the prediction of root value is to judge whether the input sentence expresses some events or not.

In addition to facilitating modeling multiple event correlations, our graph parsing formulation for ED also allows for the straightforward inclusion of other types of graph-structured features. Particularly, the other typical benefits of graph parsing formulation are as following:

- **Flexibly leveraging the hierarchical structure of event type and subtype.** Previous approaches to ED treat the event subtypes simply as 33 separate event types and ignore the hierarchical structure between the event type and subtype. Obviously, the two kinds of type labels contain the information with different granularity, both of which are indicative for event detection. Fortunately, our graph parsing framework allows for reflecting the hierarchical structure information by simply extending the event graph. A natural *node-splitting strategy* can be adopted by decomposing one event node into two nodes: one event type node and one subtype node, and linking the subtype node as a child of the type node, as shown in Figure 2(b). Compared to the *single-node strategy* for event representing, as shown in Figure 2(a), the *node-splitting* strategy indeed introduces an effective *coarse-to-fine* way for event type prediction, and therefore it is theoretically superior to the *single-node* strategy.

- **Naturally utilizing the semantic representation of event type label.** Most previous

classification-based approaches to ED generally view each event type as a specific class, omitting the semantic information conveyed by the event type label. In fact, the event type label itself, such as *Divorce*, *Injure*, etc, is informative to the learning of ED models. In our graph parsing formulation, it is straightforward to incorporate the semantic representation of type label into the model. Specifically, during decoding, we can encode every previously generated node with the corresponding type label embedding to assist the prediction of later nodes. For example, in Figure 2(b), once the event type node *Conflict* is generated, we can leverage the semantic representation of the type *Conflict* to help to predict the subtype node *Attack* and next type node *Life*.

- **Reversely generating the event triggers.** If the event triggers are needed in some cases, it is very flexible and convenient for our graph parsing framework to generate the trigger words by simply extending the event graph to yield the triggers for every predicted events. Specifically, we can append the trigger nodes as the children of event type nodes in the event graph as shown in Figure 2(c), and thus generate the trigger in a reverse direction. That is to say, unlike the traditional fashions in the existing trigger classification based models, we first predict the event types from the input text, and then output the corresponding triggers for previously predicted event types.

3 Event Graph Parsing via a Pretrained Seq2seq Model

Under our graph parsing formulation, the ED task is to transduce an input sentence into an event graph, as illustrated in Section 2. To achieve this, we choose to predict nodes sequentially rather than simultaneously, because (1) we believe the previous node generation is informative to the current node prediction; (2) variants of efficient seq2seq models (Bahdanau et al., 2014; See et al., 2017) can be employed to model this process.

Theoretically, the advantages of applying a seq2seq model to event graph parsing are two-fold. First, there is no need to use trigger words for event detection. Second, when predicting an event type node during decoding, the global contextual information in the input sentence can be taken into consideration by the cross-attention mechanism between the decoder and the encoder.

However, in the preliminary research experiments, the generic seq2seq event detection approaches did not obtain satisfactory performance. The main reason may be that the seq2seq-based approaches are generally data-hungry. Especially, for the ED task based on the benchmark dataset ACE 2005, data sparsity may be the significant challenge.

To deal with sparsity, we explore to employ transfer learning by exploiting BART (Lewis et al., 2020) to generate a linearized event graph incrementally. BART is a Transformer-based encoder-decoder model which is pretrained through a denoising self-supervised task, and has shown significant improvements in conditioned generation tasks, especially gaining state-of-the-art results on summarization. For given input sentence, our event graph generation is, to some extent, similar to the abstractive summarization task. We therefore hypothesize that BART's denoising pretraining should be suitable for this task.

While applying BART to event graph parsing, we first need to create the reference node list by a depth-first traversal over the gold event graph for training. In the preliminary experiments, we found that the labels of edges (e.g. *EVT-1*, *SUB*, etc.) are not informative to event detection; we therefore omit these labels while linearizing the event graph. For example, in Figure 2(b), the linearization order contains *EVTS*, *Business*, *End-Org*, *Conflict*, *Attack*, *Life*, *Die*.

Furthermore, we augment the vocabulary of BART with some special symbols in order to make it suitable for event detection. For example, we first expand the vocabulary by adding the root value (*EVTS* or *NA*) in the graph; then for some special event subtype, such as *Start-Org* and *End-Position*, we need to append these subtype labels to the vocabulary. Additionally, we need to expand the embedding matrices of encoder and decoder to include the average of the subword constituents of each special label.

For decoding in testing phase, we design a constrained beam search algorithm to generate the node sequence incrementally, as illustrated in Algorithm 1. Note that each particular event subtype is subject to the event type constraints. Apparently, this constrained decoding algorithm performs the prediction of event type nodes in a coarse-to-fine fashion, and can ensure to generate a valid event graph.

Algorithm 1: Constrained beam search

```
score = 0, Y = {BOS};
beam = {Y, score};
for i = 1 to maxlen do
    new_beam = {};
    {Y, score} = beam.pop();
    if i==1 then
        if P(EVTS)<P(NA) then
            break;
    else if i==2 then
        for v_i in type_set do
            Y = Y ∪ v_i, score+ = P(v_i);
            new_beam.push({Y, score});
    else
        last_token = Y[i-1];
        if last_token==EOS then
            new_beam.push({Y, score});
            continue;
        else if last_token in type_set then
            for v_i in constrained_subtype_set do
                Y = Y ∪ v_i, score+ = P(v_i);
                new_beam.push({Y, score});
        else
            set = type_set + {EOS};
            for v_i in set do
                Y = Y ∪ v_i, score+ = P(v_i);
                new_beam.push({Y, score});
    beam = new_beam.topK();
{Y, score} ← beam.topK(k = 1);
```

4 Experiments

4.1 Dataset and Evaluation Metrics

We utilized the ACE 2005 corpus as our dataset. For comparison, as the same as previous work (Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Li et al., 2013), we used the same test set with 40 newswire articles and the same development set with 30 other documents randomly selected from different genres and the rest 529 documents are used for training.

Also, following previous work (Liao and Grishman, 2010; Li et al., 2013; Chen et al., 2015; Liu et al., 2019a; Tong et al., 2020), we use the following criteria to evaluate the results:

Precision: the proportion of correctly predicted events in total predicted events.

Recall: the proportion of correctly predicted events in total gold events of the dataset.

F1-measure: $\frac{2*P*R}{P+R}$

4.2 Hyperparameters

The hyperparameters are tuned on the validation set. For all the experiments, we use the same model hyperparameters as BART-Large, as defined in Huggingface's transformers library. The models are trained using cross-entropy with RAdam as optimizer and a learning rate of $5 * 10^{-5}$. Gradient is accumulated for 10 batches. Dropout is set to 0.25. Our models are trained for 80 epochs, the batch size in our training experiments is set to 300. For decoding, we set beam size to 5 and the constant maxlen to 20.

4.3 Overall Performance

In this section, we comprehensively compare our performance with the following state-of-the-art methods:

JRNN proposes a joint event extraction model based on recurrent neural network to improve ED (Nguyen et al., 2016).

DLRNN exploits document information via recurrent neural networks (Duan et al., 2017).

TBNNAM is the first work on detecting events without triggers (Liu et al., 2019b).

GCN-ED uses an argument pooling mechanism for event detection based on GCN (Nguyen and Grishman, 2018).

JMEE uses GCN with highway network and self-attention (Liu et al., 2018).

MOGANED is an advanced graph neural network (GNN) model. It proposes a multi-order graph attention network to effectively model the multi-order syntactic relations in dependency trees and improve ED (Yan et al., 2019).

EE-GCN simultaneously exploits syntactic structure and typed dependency label information to perform ED (Cui et al., 2020).

GatedGCN proposes a novel gating mechanism to filter noisy information in the hidden vectors of the GCN models for ED (Lai et al., 2020).

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., 2019a).

EKD leverages the wealth of the open-domain trigger knowledge to improve ED (Tong et al., 2020).

Table 1 shows the overall performance comparison between our best system and the above state-of-the-art models. In Table 1, we roughly divide the state-of-the-art methods into three groups: data-driven methods, syntactic dependency enhanced

Method	Precision	Recall	F1
JRNN	66.0	73.0	69.3
DLRNN	77.2	64.9	70.5
TBNNAM*	76.2	64.5	69.9
GCN-ED†	77.9	68.8	73.1
JMEE†	76.3	71.3	73.7
MOGANED†	79.5	72.3	75.7
EE-GCN†	76.7	78.6	77.6
GatedGCN†	78.8	76.3	77.6
Lu's DISTILL∧	76.3	71.9	74.0
TS-DISTILL∧	76.8	72.9	74.8
EKD†∧	79.1	78.0	78.6
Ours	**83.2**	**82.4**	**82.8**

Table 1: Overall Performance on ACE2005 dataset (%). The results of baselines are adapted from their original papers. † indicates that the method uses dependency structures, ∧ indicates that the method uses external knowledge and resources, * indicates that the method does not need triggers.

methods and knowledge enhanced methods. Additionally, TBNNAM is the only work that does not use trigger for ED, similarly to our work.

From Table 1, we can see that our approach that adopts the *node-splitting strategy* to construct the event graph achieves the best Precision, Recall and F1 score among all the compared methods. It is worth noting that *our model simultaneously significantly improves both Precision and Recall without using any additional information including the POS tags, the syntactic dependency and external knowledge*, which shows the superiority of the proposed graph parsing formulation for ED.

5 Analysis

5.1 Effect of Exploiting the Event Type Information

Unlike previous models, our graph parsing formulation can flexibly incorporate the rich information conveyed by the event types into the event graph, including the hierarchical structure of event type and the semantic representation of the type label. In this section, we prove the effect of exploiting the event type information by the ablation test.

First, we compare the two different strategies for constructing the event graph, i.e. the *single-node strategy* and *node-splitting strategy* (as illustrated in Section 2). As expected, the *node-splitting strategy* that uses the hierarchical structure information significantly outperform its counterpart, as shown in Table 2.

Next, we verify the effect of the semantic representation of the type label by treating every label of event type or subtype as a special symbol, not using their word embedding learned in the pretrained language model. From the results in Table 2 we can observe that, ignoring the semantic representation of type label leads to a significant performance drop, especially for the *node-splitting strategy*.

Method	P	R	F1
Single-node strategy(-)	80.0	76.0	78.0
Single-node strategy	76.8	81.0	78.8
Node-splitting strategy(-)	79.0	78.8	78.9
Node-splitting strategy	**83.2**	**82.4**	**82.8**

Table 2: Performance comparison of differently structured event graphs with or without label embeddings. The symbol - indicates that the method does not use type label embeddings.

5.2 Effect of Multiple Event Recognition

Compared to the existing work, our ED approach provides a more natural formulation to model the multiple event correlations. To evaluate the effect of our approach to the multiple event recognition, we divide the test data into two parts (**1/1** and **1/N**) following previous work (Chen et al., 2015; Nguyen et al., 2016), and perform evaluations separately. **1/1** means that one sentence only has one event; otherwise, **1/N** is used. Table 3 illustrates the performance (F1 scores) of **DMCNN** (Chen et al., 2015), **JRNN** (Nguyen et al., 2016), **JMEE** (Liu et al., 2018) and **HBTNGMA** (Chen et al., 2018), the four baseline models and our model for ED task. As shown in Table 3, our model significantly outperforms all the other methods. In the **1/N** data split, our method is **9.8%** better than the best baseline, **JMEE**. The experimental results demonstrate that our method works well on the task of multiple event recognition.

5.3 Effect of Pretrained Seq2seq Model

In this section, we inspect the effect caused by the pretrained seq2seq model, BART. For this end, we implemented a traditional seq2seq event graph parsing system as baseline, by adopting the pointer-generator network (See et al., 2017), and using BERT embeddings (Devlin et al., 2019) to encode

Method	1/1	1/N	all
DMCNN	74.3	50.9	69.1
JRNN	75.6	64.8	69.3
HBTNGMA	78.4	59.5	73.3
JMEE	75.2	72.7	73.7
Ours	**83.1**	**82.5**	**82.8**

Table 3: Performance comparison on single event sentences (1/1) and multiple event sentences (1/N).

the words. And the experimental results are illustrated in Table 6. We can see that, compared to most state-of-the-art models as shown in Table 1, this baseline obtains a competitive result. However, BART-based model can further substantially improve F1 score by 6.6%, which indicates the importance of the pretrained seq2seq for tackling data sparsity.

Method	P	R	F1
Pointer-generator-based	81.2	71.8	76.2
BART-based	**83.2**	**82.4**	**82.8**

Table 4: Performance of test set with or without pretrained seq2seq model.

5.4 Analysis of Cross-Attention Mechanism

In the absence of trigger words, can our Transformer-based seq2seq event detection framework capture the key clues in the source sentence that express the target event type? In this section, we answer this question by the case study.

Figure 3 presents several examples of the attention distributions learned by our model. In the first case, the target event type is *Life:Die* and the gold trigger is the word *killed*. We can see that when predicting this event type, our attention not only successfully attend the trigger word *killed*, but also attend another strongly indicative phrase *two people* with higher score. In the second case, the target event type is *Conflict:Attack*, and the gold trigger is the word *strike*. It can be observed that, the three words: *destroyed*, *houses* and *killed* are assigned with higher attention scores than the trigger *strike*, which seems plausible for this target type prediction. In the third case, the target event type is *Personnel:Elect*, and the gold trigger is the word *lost*. For this target type, there are relatively strong connections with the phrase *lost her seat* and another indicative word *election*.

These cases demonstrate that, though the triggers are not used in our model, the cross-attention mechanism between the decoder and encoder can learn to automatically and accurately capture the correlation between the target event type and multiple indicative words or phrases in the source sentence.

5.5 Can Our Approach Alleviate the Long Tail Issue?

The trigger-based event detection models generally suffer from the long tail issue (Lu et al., 2019; Tong et al., 2020). Taking 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 the trigger-based models perform poorly on unseen/sparsely labeled trigger words. In this section, we evaluate whether our approach could cope with the long tail issue.

Following previous work (Tong et al., 2020), we divide the event instances in the test set into three categories: *Unseen*, *Sparsely-Labeled* and *Densely-Labeled*, according to their trigger frequency in the training set. Specifically, the frequency of Sparsely-Labeled is less than 5 and the frequency of Densely-Labeled is more than 30. Also, following the work (Tong et al., 2020), we choose the following

Figure 3: Visualization of cross-attention scores of sample instances learned by our model.

baselines for comparison: (1) **DMBERT** (Chen et al., 2015), (2) **DGBERT** (Chen et al., 2017), (3) **BOOTSTRAP** (He and Sun, 2017), and (4) the method **EKD** (Tong et al., 2020).

As shown in Table 5, our approach substantially outperforms all baselines in all three settings, especially on unseen (**+12.9%**) and sparsely-labeled settings (**+4.8%**). Why can our approach effectively mitigate the long tail issue? Besides the better generalization endowed by our seq2seq event graph parsing formulation, an important possible reason is that since our approach adopts a trigger-free way to detect the events, the event types corresponding to the unseen or sparsely-labeled triggers can also be expressed with other different triggers and thus appear many times in the training set, thereby alleviating the long tail problem. The experimental results clearly indicate that, *the trigger-free ED approach may be a better alternative to the traditional trigger-based models*.

5.6 Effect of the Generation of Event Triggers

In this section, we investigate the performance of the trigger generation. As mentioned in Section 2.2, our graph parsing framework can also conveniently generate the event triggers by simply appending the trigger nodes as the children of event type nodes in the event graph, as shown is Figure 2(c). The experimental results of trigger generation are illustrated in Table 6.

	P	R	F1
event type evaluation	80.6	84.5	82.5
event trigger evaluation	78.1	81.8	80.0

Table 6: Performance of event trigger generation.

We can observe that the F1-score of event type evaluation achieves 82.5%, remaining almost unchanged as the result of 82.8% shown in Table 1, and that the F1-score of trigger recognition achieves 80.0%, significantly outperforming the existing trigger classification based models.

6 Related Work

6.1 Event Detection

In earlier ED studies, the traditional feature-based methods focused on exploiting the lexical and global features to detect events (Ji and Grishman, 2008; Li et al., 2013). Most recent works have focused on using neural networks in this task and have achieved significant progress. We roughly divide the recent approaches into three categories as following:

Sequence-based models: This line of research operates on the word sequences using the deep neural networks. (Chen et al., 2015) devises a dynamic multi-pooling convolutional neural network to capture more information. (Nguyen et al., 2016) presents a joint model based on bidirectional RNN for event extraction. (Sha et al., 2018) adds dependency arcs with weight to BiLSTM to make use of tree structure and sequence structure simultaneously. (Chen et al., 2018) exploits a hierarchical and bias tagging networks to detect multiple events collectively.

GCN-based models: This line of research adopts the Graph Convolutional Network (GCN) over the dependency tree of a sentence to boost the performance. (Nguyen and Grishman, 2018) is the first attempt to use GCN in ED. (Liu et al., 2018) employs a syntactic GCN and a self-attention mechanism to model multiple events extraction. (Yan et al., 2019) improves GCN by combining multi-order word representation from different GCN layers.

Knowledge Distillation-based models: More recently, some approaches focus on leveraging various external knowledge to improve classification.

Method	Unseen			Sparsely Labeled			Densely Labeled		
	P	R	F1	P	R	F1	P	R	F1
DMBERT	66.7	45.9	54.4	74.4	70.7	72.5	84.8	83.5	84.1
DGBERT	76.5	42.6	54.7	75.7	70.1	72.8	85.9	83.8	84.3
BOOTSTRAP	73.7	45.9	56.6	76.0	71.3	73.6	90.6	83.5	86.9
EKD	79.0	52.0	62.7	80.8	72.4	76.4	**92.5**	82.2	87.1
Ours	**92.2**	**64.1**	**75.6**	**87.9**	**75.5**	**81.2**	85.7	**90.6**	**88.1**

Table 5: Performance comparison on the unseen, sparsely-labeled and densely-labeled settings.

(Lu et al., 2019) proposes a new representation learning framework to distill both discrimination and generalization knowledge for ED. (Liu et al., 2019a) uses an adversarial imitation based knowledge distillation approach for ED. (Tong et al., 2020) proposes an enrichment knowledge distillation model to leverage external open-domain trigger knowledge to address the long-tail issue.

Unlike the existing ED models based on trigger classification, we formulate ED as a novel graph parsing problem, therefore it can explicitly model the multiple event correlations and incorporate the rich information regarding the event types.

6.2 Prertained Seq2seq Models

Pre-training a universal model and then fine-tuning the model on a downstream task have recently become a popular strategy in the field of natural language processing (Devlin et al., 2019). Recent studies also propose approaches to pre-training seq2seq models, such as **MASS** (Song et al., 2019), **PoDA** (Wang et al., 2019), **PEGASUS** (Zhang et al., 2019), **BART** (Lewis et al., 2020), and **T5** (Raffel et al., 2019)

In this paper, our experiments only examine BART. We leave explorations of these models for future work.

7 Conclusion

This paper presents the first work to formulate ED as a graph parsing task, and to introduce a novel generation-based method to predict event graph by using a pretrained seq2seq model. Our approach is conceptually simple and does not use syntactic dependency information and any other extra knowledge; however, it significantly outperforms the traditional trigger classification-based encoder-only approaches, advancing the state of the art in event detection.

In future work, we will integrate the syntactic dependency structure and external knowledge into our model to further improve the ED performance; in particular, we will explore to extend our graph parsing architecture to more IE problems, such as event extraction, relation extraction and so on. For example, by adding the event argument nodes to the event graph and linking them as the children of event type nodes, the event extraction problem should be solved in a similar way.

Acknowledgments

We thank all reviewers for the valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61472191, No. 61976114, No. 61772278 and No. 61936012).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2329–2339.

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

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Hangfeng He and Xu Sun. 2017. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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 the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. 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.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019b. Event detection without triggers. 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 735–744, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Thien Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. *CoRR*, abs/1905.02450.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, Online. Association for Computational Linguistics.

Liang Wang, Wei Zhao, Ruoyu Jia, Sujian Li, and Jingming Liu. 2019. Denoising based sequence-to-sequence pre-training for text generation. *CoRR*, abs/1908.08206.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5766–5770, Hong Kong, China. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.