A Trigger-Aware Multi-Task Learning for Chinese Event Entity Recognition

Yangxiao Xiang
School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, China
351261717@qq.com

Chenliang Li
School of Cyber Science and Engineering, Wuhan University, China
cllee@whu.edu.cn

Abstract

This paper tackles a new task for event entity recognition (EER). Different from named entity recognizing (NER) task, it only identifies the named entities which are related to a specific event type. Currently, there is no specific model to directly deal with the EER task. Previous named entity recognition methods that combine both relation extraction and argument role classification (named NER+TD+ARC) can be adapted for the task, by utilizing the relation extraction component for event trigger detection (TD). However, these technical alternatives heavily rely on the efficacy of the event trigger detection, which have to require the tedious yet expensive human labeling of the event triggers, especially for languages where triggers contain multiple tokens and have numerous synonymous expressions (such as Chinese). In this paper, a novel **t**rigger-**a**ware **m**ulti-task learning framework (TAM), which jointly performs both trigger detection and event entity recognition, is proposed to tackle Chinese EER task. We conduct extensive experiments on a real-world Chinese EER dataset. Compared with the previous methods, TAM outperforms the existing technical alternatives in terms of $F1$ measure. Besides, TAM can accurately identify the synonymous expressions that are not included in the trigger dictionary. Morover, TAM can obtain a robust performance when only a few labeled triggers are available.

1 Introduction

In this paper, we introduce a variant of named entity recognition (NER) task, which is event entity recognition (EER). Different from NER task, EER aims at identifying the named entities corresponding to a specific event type. Figure 1 demonstrates some examples of EER. In example a, text contains both the event type '交易违规' (illegal trading) and '涉嫌传销' (illegal pyramid selling). Obviously, the corresponding entity of event type '交易违规' (illegal trading) is the 'organization B'. Hence, the 'organization B' is defined as the event entity of the query event '交易违规' (illegal trading). Similarly, the event entity of query event '涉嫌传销' (illegal pyramid selling) is 'organization C'. Clearly, EER can be seen as a selective NER task, which treats the 'text' and the 'query event' as the input, and identify the 'event entity' as output. EER is widely useful in many semantic applications such as public opinion analysis in financial domain, disease extraction in medical domain, etc.

Currently, however, there is no specific model to deal with EER task. To the best of our knowledge, mainly three types of techniques can be used to possibly achieve EER: vanilla NER models, question-answer (QA) models and NER+TD+ARC models. For the vanilla NER models, solving EER task may be difficult due to the lack of event information. QA models can effectively incorporate the event information in the form of 'question' input. However, how to derive the appropriate question with respect to EER is still unknown. The NER methods that incorporate both the relation extraction and argument role classification are the most relevant solutions to tackle EER. Specifically, we could adapt the relation extraction component for event trigger detection. These solutions extract all the named entities and the event triggers (*i.e.,* trigger detection (TD)) in a sentence, and then identify the relation between each named entity and the target event trigger respectively (*i.e.,* argument role classification (ARC)). Here, an event trigger is a word indicating the event type mentioned in the text. There are various forms of NER+TD+ARC models. (Chen et al., 2015) proposed a pipelined DMCNN model to perform TD and

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.

Example a:公司A举报公司B违规减持公司C债券,而公司C法人此前曾因卷入传销案被捕入狱
Company B was reported by Company A due to the illegal reduction of the bonds of Company C, the legal person of the Company C was previously arrested for Pyramid selling case.

Example b:公司A董事长提前获取了一些信息，并精准抛售了公司B1000万股
The chairman of Company A got some inside information in advance, and accurately sold 10 million shares of the Company B

Example c:公司A执行董事被指控违法
The executive director of Company A was accused by breaking the laws.

Figure 1: The examples of EER task.

ARC tasks separately. However, the pipelined method shows poor performance for both TD and ARC, because it cannot capture the inner dependency between TD and ARC tasks. Afterwards, some joint models (JRNN (Nguyen et al., 2016), JMEE (Liu et al., 2018), etc), which jointly perform TD and ARC tasks, are proposed. These models significantly improve the performance of the TD and ARC tasks. In addition, models which jointly perform NER, TD and ARC tasks are also proposed, which further improves performance (Nguyen and Nguyen, 2019). Although NER+TD+ARC methods are applicable for EER tasks, however, these existing solutions have the following three defects.

- The three subtasks (NER, TD, and ARC) require enormous efforts for human labeling. In detail, NER requires to label all the entities in the text. TD requires to label event triggers as much as possible, and ARC requires to label the relationship between each entity and each event trigger.
- The ARC subtask is severely dependent on the event trigger detection. However in practice, event triggers are not always identifiable. On the one hand, event triggers do not always exist in the texts. In some cases, the event is expressed implicitly. Looking at example b demonstrated in Figure 1, this sentence expresses an event '交易违规' (illegal trading) with no explicit event trigger. On the other hand, for some languages (such as Chinese) where some event triggers are typically composed by multiple tokens and have numerous synonymous expressions, it is challenging to compile all the synonymous expressions of these event triggers with limited human efforts. For example, there are many synonymous expressions for an event trigger '犯罪' (crime), such as '犯了罪' (crime), '犯下罪行' (crime), '犯法' (crime), etc. Last but not least, some triggers can be associated with multiple events. Takeing example c in Figure 1 as an example, trigger word '违法' (break the law) indicates both the event '涉嫌违法' (crime) and the event '高管负面' (negative news of the executive) happen. Therefore, multi-label classification for event triggers is needed, which is mainly overlooked in the existing works.
- Previous methods are not directly aimed at dealing with EER task, some redundant process may cause unnecessary training cost and result in poor performance, such as all the named entities have to be recognized.

To this end, this paper proposes a novel trigget-aware multi-task learning framework (named TAM) to achieve EER task. TAM joints two networks to deal with the two subtasks. In the first stage of the TAM, a trigger detection network is proposed to perform the Chinese TD. It uses a local attention mechanism to catch the context information of each token, and then perform multi-label classificion of each single token. The trigger detection network effectively improves the performance in intractable Chinese event trigger identification. On the one hand, the trigger detection network is capable of identifying the synonymous expressions of event triggers with limited human-labeling. On the other hand, it effectively solves the multi-label classification of the event triggers, which is rarely considered in the previous TD works. In the second stage, we introduce an event-featured transformer-CRF network to identify the event entities directly. It discards the previous entity-trigger-relation classification workflow. Instead, it

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

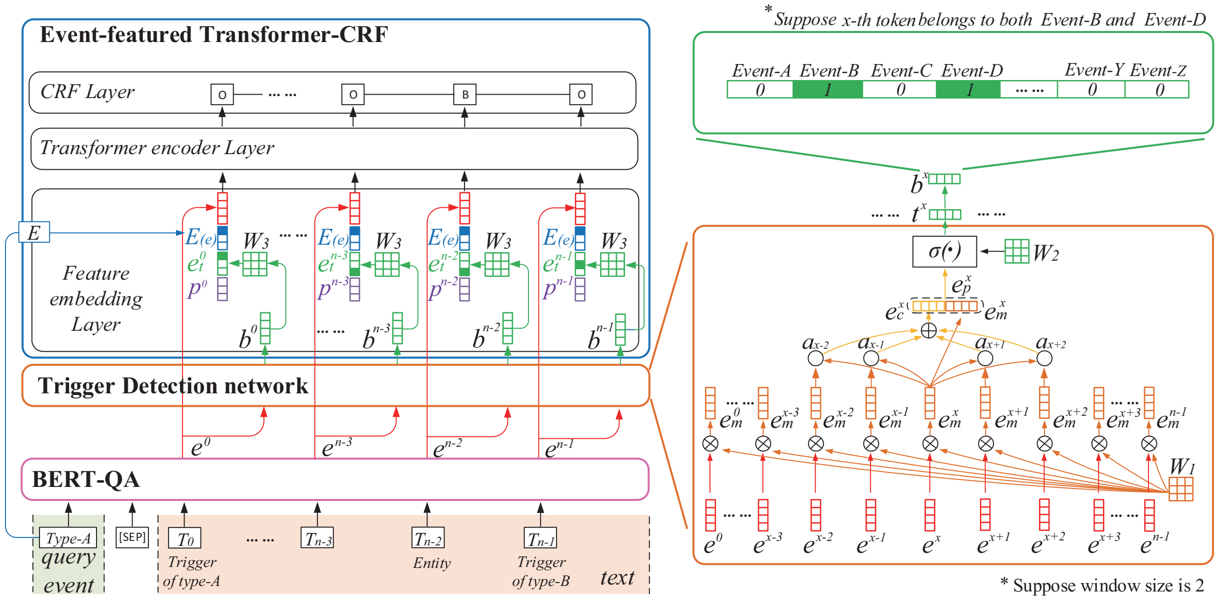


Figure 2: The network architecture of TAM.

identifies the event entities by full-text understanding with the help of the three event type relevant features. This alteration effectively increases the robustness of the model with different amount of trigger labeling, and significantly outperforms the NER-TD-ARC models in EER task where event triggers are not explicitly mentioned or more than one event are included in the sentence. The main contribution of this paper can be summarized as follows:

- We propose a novel trigger-aware multi-task learning framework for event entity recognition. TAM jointly combine a trigger detection network and an event-featured transformer-CRF to achieve the trigger detection and event entity recognition simultaneously.
- Experiments demonstrate that TAM significantly outperforms the existing technical alternatives for EER tasks with single event, multiple events and with no explicit event triggers. Besides, TAM also is effective at identifying the synonymous expressions of a Chinese event trigger, and is robust against the availability of the labeled triggers.

2 The Proposed Algorithm

The proposed TAM mainly consists of two components: *trigger detection network* for event trigger detection and *event-featured transformer-CRF* for event entity recognition. The architecture of the proposed TAM is illustrated in Figure 2. At first, we briefly describe the process to build the event trigger dictionary.

2.1 Event Trigger Dictionary Construction

Formally, an event trigger is a word which indicates a specific event mentioned in a text. For the complex sentence involving multiple events and multiple entities (ref. Figure 1), the corresponding event triggers naturally provide semantic segmentation between the sub-sentences related to different event types. Therefore, event trigger detection is very helpful to solve EER task for the sentences with multiple events and multiple entities.

However, a Chinese event trigger is typically composed by several tokens and would have many synonymous expressions. Therefore, labeling Chinese event triggers is a tedious yet expensive task. We first utilize a simple discriminative measure based on the frequency statistics to obtain the trigger candidates. Specifically, we first perform Chinese word segmentation with an external Chinese NLP toolkit Jieba [1].

[1] https://github.com/fxsjy/jieba

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.

Then, for each word w, we calculate the relatedness $\beta_e(w)$ towards an event type e as follows:

$$\beta_e(w) = \frac{c(w,e)}{c(e)} \frac{c(\neg e)}{c(w,\neg e)} \tag{1}$$

where $c(w,e)$ is the number of sentences containing word w and belonging to event type e, $c(e)$ is the number of sentences of event type e. $c(\neg e)$ is the number of sentences of the other event types, $c(w,\neg e)$ is the number of the sentences containing word w but not belonging to event type e. We manually identify the words as the triggers for event type e by examining the words with high $\beta_e(w)$. Note that our trigger labeling is not exhaustive due to the numerous synonymous expressions of event triggers. That is, the built event trigger dictionary is limited on its coverage.

2.2 Trigger Detection Network

BERT is one of the most popular pretrained language models recently, the resultant contextual embeddings have achieved the state-of-the-art performance in various NLP tasks. Here, we utilize a Chinese character based BERT-QA to obtain the contextual word embedding of each token in the given sentence. The structure of the BERT-QA is shown as the pink block in Figure 2. Specifically, the sentence and query event are fed into BERT-QA, where text and query are separated by a special token '[SEP]'. Then, the contextual token embedding $\mathbf{e}^x \in R^{d_1}$ of x-th token in the sentence is obtained through the pretrained Chinese Simplified character based BERT model[2], where d_1 is the dimension size. Note that our trigger labeling is not exhaustive due to the numerous synonymous expressions of event triggers. That is, the built event trigger dictionary is limited on its coverage. In the proposed TAM, we first learn a trigger detection network to identify the plausible triggers which will be utilized in entity recognition for the target event.

We apply a linear mapping by transfering each token embedding $\mathbf{e}^x$ into a feature vecotr $\mathbf{e}_m^x$ with a learnable weigth matrix $\mathbf{W}_1 \in R^{d_2 \times d_1}$, in order to better adapt to the event trigger detection task. Note that a Chinese event trigger often contains several tokens. Here, we utilize a context window of length w centering at each token, $[\mathbf{e}_m^{x-w}, \cdots, \mathbf{e}_m^{x+w}]$, to help identify whether the token is a constituent of a trigger. This local context window could help us judge whether the token belongs to an event trigger and what event type it is. In detail, we utilize an attention network to help $\mathbf{e}_m^x$ find the most related contextual information:

$$\alpha_k = \frac{\exp(\mathbf{e}_m^{x\top} \mathbf{e}_m^k)}{\sum_{j=x-w}^{x-1} \exp(\mathbf{e}_m^{x\top} \cdot \mathbf{e}_m^j) + \sum_{j=x+1}^{x+w} \exp(\mathbf{e}_m^{x\top} \cdot \mathbf{e}_m^j)} \tag{2}$$

$$\mathbf{e}_c^x = \sum_{j=x-w}^{x-1} \alpha_j \cdot \mathbf{e}_m^j + \sum_{j=x+1}^{x+w} \alpha_j \cdot \mathbf{e}_m^j \tag{3}$$

where $\mathbf{e}_c^x$ denotes the contextual information around x-th token. Then, the contextual information is concatenated with $\mathbf{e}_m^x$ to form the phrase embedding $\mathbf{e}_p^x$: $\mathbf{e}_p^x = \mathbf{e}_m^x \oplus \mathbf{e}_c^x$, where $\oplus$ is the concatenation operaton.

With the phrase embedding, we then perform a multi-label classification to idenitfy the event types of the current token. The structure of the event mapping layer is shown as the orange block in Figure 2. Let the total number of the event types is d_3, the classification is performed as a regression:

$$\hat{\mathbf{t}}^x = \sigma(\mathbf{W}_2 \mathbf{e}_p^x + \mathbf{b}) \tag{4}$$

where $\sigma(\cdot)$ is the nonlinear sigmoid function, $\mathbf{W}_2 \in R^{d_3 \times 2 \cdot d_2}$ is a learnable weight matrix. Each dimension of the $\hat{\mathbf{t}}^x \in R^{d_3}$ is the probability of the x-th token belonging to the corresponding event type, which is exhibited in the green block of Figure 2. Finally, by using the event trigger dictionary as the

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

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.

supervision, the loss of the trigger detection networks is calculate as the sum of the cross-entropy loss of each token:

$$\mathcal{J}_{TD} = -\sum_{i=1}^{d_3}\sum_{j=1}^{n} \mathbf{t}^j(i)\log(\hat{\mathbf{t}}^j(i)) \tag{5}$$

where $\mathbf{t}^j \in R^{d_3}$ is the groundtruth multi-hot vector of j-th token, n is the sentence length.

2.3 Event-Featured Transformer-CRF

In order to reduce the strong dependency between TD and ARC subtasks existed in the current technical alternatives, here, we introduce an event-featured transformer-CRF network (shown as the blue block in Figure 2) to directly identify the event entities. Specifically, besides the contextual token embeddings produced by BERT-QA, the event-featured transformer-CRF incorporates three additional features (ie event type feature, trigger feature and position feature) related to the target event type to perform event entity recognition.

Event Type Feature. We utilize another learnable event type embedding matrix $\mathbf{E} \in R^{d_3 \times d_4}$ to encode the features related to each event type. Given the target event type e, we can adopt look-up operation to extract the corresponding feature embedding $\mathbf{E}(e)$. The utilization of the $\mathbf{E}(e)$ could enhance the model's awareness towards the target event type.

Trigger Feature. Since the trigger information provides both the information of what events occur in the sentence, we utilize two sets of trigger feature embeddings to inject the trigger information detected by the trigger detection network mentioned above. We first binarize the predicted event type vector $\hat{\mathbf{t}}^x$ of each token into vector $\mathbf{b}^x$ by utilizing 0.5 as the threshold. When $\hat{\mathbf{t}}^x(i) \geq 0.5$, the corresponding value $\mathbf{b}^x(i)$ is set to 1; otherwise, $\mathbf{b}^x(i)$ is set to 0. Then, we transform the $\mathbf{b}^x$ into trigger embedding $\mathbf{e}_t^x \in R^{d_5}$ by looking up a learnable matrix $\mathbf{W}_3$: $\mathbf{e}_t^x = \mathbf{W}_3\mathbf{b}^x$. Here, $\mathbf{e}_t^x$ encodes the event type information of the x-th token in the sentence.

Position Feature. The position embeddings have been widely adopted in many relevant NLP tasks, such as relation extraction (Chen et al., 2015). Here, we further encode the position of each token as a d_6 dimensional embedding $\mathbf{p}^x$, using the work presented in (Devlin et al., 2019). Under the combination of both position embeddings and trigger embeddings, the information of semantic segmentation between sub-sentences of different events could be extracted by the transformer-CRF network.

The above three kinds of event type relevant features are concatenated together with the contextual token embeddings produced by the BERT-QA to form the new token embeddings. Then, these updated token embeddings are then fed into a transformer-CRF layer for event entity recognition. It includes two sub-layers. The first layer is a transformer encoder model (Devlin et al., 2019), which is a powerful feature extractor various NLP tasks. In the second layer, a conditional random field (CRF) is utilized to learn the correlation between the tag correlations. As for the labeling, we utilize the BIO scheme in CRF. The maximum conditional likelihood loss is used to update the parameters as follows:

$$\mathcal{J}_{EER} = -\sum_{i=x}^{n} \log p(y_x|t_x) \tag{6}$$

where $p(y_x|t_x)$ is the probability of assigning label y_x to token t_x, n is the length of the sentence, and y_x is the groundtruth label for token t_x.

2.4 Loss Function

Since TAM is devised with a multi-task learning paradigm, for model training, we minimize the joint negative log-likelihood loss function by combining $\mathcal{J}_{EER}$ and $\mathcal{J}_{TD}$ as:

$$\mathcal{J}_{total} = \mathcal{J}_{EER} + \lambda\mathcal{J}_{TD} \tag{7}$$

where λ is the hyper-parameter which controls the training weight of the two subtasks. We use the Adam algorithm to optimize the parameters with minibatches. The gradients are computed with back-propagation.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

3 Experiments

3.1 Experimental Setup

Dataset. We evaluate the proposed TAM on the CCKS2019-task4 (named CCK19) dataset[3]. CCKS19 dataset contains in total $19,500$ sentences from the financial news. After excluding the overlapping sentences in both training and test sets, There are $16,000$ sentences in training set and $2,400$ sentences in test set. It defines 22 negative event types (including type 'None'). For each item in training set, the event entity and the corresponding event type are annotated. For each item in test set, only query event type is used as an input, and the event entity are used for performance evaluation. It is worth mentioning that since the event triggers are not provided in CCKS19 dataset, we utilize the event trigger dictionary constructed in Section 2.1.

Baseline Methods. Currently, there is no direct model to deal with EER task. We compare TAM with the following related methods:

- BiLSTM+CRF was proposed by (Huang et al., 2015). It achieves a good performance on vanilla NER tasks. We use the sentence and the annotated event entities while training, and only use the sentence while testing.
- BERT-QA was proposed by (Devlin et al., 2019). It achieves the state-of-the-art performance in question answer tasks. We use the sentence (paragraph) and the event type (question) as the input of the BERT-QA, and treat the event entity as the answer.
- DMCNN was proposed by (Chen et al., 2015). It is a typical pipelined relation extraction method (TD-ARC). In our settings, The NER is realized by BiLSTM+CRF, TD and ARC are realized by DMCNN.
- JRNN was proposed by (Nguyen et al., 2016). It is a typical joint relation extraction method (TD-ARC). In our settings, The NER is realized by BiLSTM+CRF, TD and ARC are realized by JRNN.
- Joint3EE was proposed by (Nguyen and Nguyen, 2019). It is a typical joint relation extraction method which combines three subtasks (*i.e.,* NER-TD-ARC).

We further introduce several variants of TAM by excluding the trigger detection network (named TAM / TD), event type feature (named TAM / EE), position feature (named TAM / PE), both event type feature and position feature (named TAM / EE&PE), both trigger detection network and position feature (named TAM / TD&PE), both trigger detection network and event type feature (named TAM / TD&EE) to validate their impacts in an individual and combination manner. Note that when trigger detection network is excluded, trigger embedding $\mathbf{e}_t^x$ is equivalent to the zero vector.

Parameter Settings. For BiLSTM+CRF, DMCNN, JRNN and Joint3EE, word embeddings are obtained using vanilla BERT with a dimension size 768 for a fair comparison. The same dimension size applies for BERT-QA and TAM. For the trigger detection network used in TAM, the window size is set to 5. For the event-featured transformer-CRF network, the dimension size of each feature embedding is set to 48 (*i.e.,* $d_4 = d_5 = d_6 = 48$). The transformer encoding module utilizes 3 layers and 6 attention heads. Loss weight λ in Equation 7 is set to 0.05. The dropout rate is 0.5, and we apply the Adam optimizer for parameter update. Three metrics: precision (P), recall (R) and $F1$, are utilized for performance comparison. We report the optimal $F1$ performance of each method on the test set.

3.2 Results and Discussion

Event Entity Recognition. The results of different models on EER task are reported in Table 1. In order to better demonstrate the traits of each kind of model, the test set are divided into three subsets: 1) sentences with a single event $(1,366$ instances); 2) sentences with multiple explicit events $(1,005$ instances) and sentences with no event trigger (29 instances).

[3] http://www.ccks2019.cn/?page_id=62

Method	With Trigger						Without Trigger			Total		
	Single Event			Multiple Event								
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
BiLSTM-CRF	0.935	0.937	0.936	0.850	0.862	0.856	0.750	0.750	0.750	0.896	0.902	0.899
BERT-QA	0.949	0.945	0.947	0.920	0.923	0.922	0.848	**0.875**	**0.862**	0.935	0.935	0.935
DMCNN	0.952	0.949	0.951	0.895	**0.955**	0.924	**1.00**	0.094	0.171	0.926	0.940	0.933
JRNN	0.954	0.938	0.946	0.890	0.949	0.919	0.591	0.406	0.481	0.922	0.936	0.929
Joint3EE	**0.960**	0.940	0.951	0.890	0.948	0.918	0.647	0.344	0.449	0.926	0.936	0.931
TAM	0.958	**0.958**	**0.958**	**0.935**	0.941	**0.938**	0.848	**0.875**	**0.862**	**0.947**	**0.950**	**0.948**
TAM / TD	0.948	0.946	0.947	0.919	0.929	0.924	0.848	**0.875**	**0.862**	0.947	0.950	0.948
TAM / TD&PE	0.949	0.945	0.947	0.921	0.931	0.926	0.848	**0.875**	**0.862**	0.935	0.938	0.937
TAM / TD&EE	0.949	0.947	0.948	0.918	0.924	0.921	0.844	0.844	0.844	0.935	0.936	0.935
TAM / EE&PE	0.957	0.943	0.955	0.926	0.930	0.928	0.897	0.813	0.852	0.943	0.941	0.942
TAM / EE	0.955	0.952	0.953	0.927	0.936	0.931	0.867	0.813	0.839	0.942	0.944	0.943
TAM / PE	0.956	0.956	0.956	0.932	0.936	0.934	0.871	0.844	0.857	0.945	0.946	0.945

Table 1: The performance of different methods in EER task. The best results are highlighted in boldface.

Method	Trigger Detection		
	P	R	$F1$
DMCNN	0.929	0.840	0.882
JRNN	0.929	0.844	0.884
Joint3EE	0.928	0.844	0.884
TAM	**0.930**	**0.865**	**0.895**

Table 2: The performance of different methods in TD task.

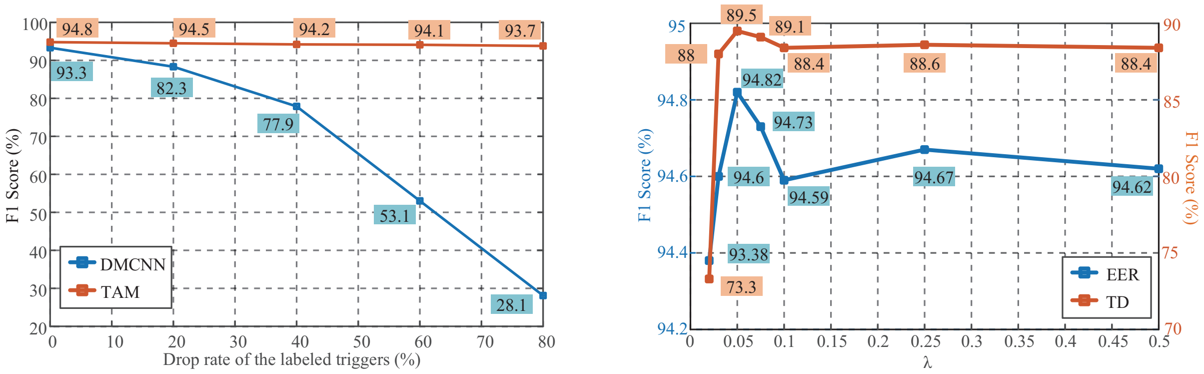
According to Table 1, it can be seen that TAM outperforms the other baselines in all the three subsets. The BiLSTM-CRF model for vanilla NER delivers the worst performance in dealing with EER task. It is reasonable since no event information is incorporated.

By comparing TAM with TAM / TD and BERT-QA, we can see that since TD effectively provides the event type information and trigger position information, such as the three event type relevant features for transformer-CRF, incorporating the trigger detection network and the additional event type relevant features is notably beneficial for the downstream EER task with explicit event triggers, especially for the sentences with multiple events. However, for the sentences with no trigger, the utilization of TD shows trivial effect, BERT-QA, TAM / TD and TAM all show good performance in sentences with no event trigger. Note that we take the target event type as input into BERT-QA. The observation suggests that explicitly modeling event type information is very useful for EER task.

The three NER-TD-ARC based models (DMCNN, JRNN and Joint3EE) all show good performance in dealing sentence with single event and multiple events, because these models are all relying on event trigger detection. However, TAM shows better performance, especially in EER task with multiple events, due to its good performance in TD task. Since NER-TD-ARC based models are strongly dependent on event trigger detection, for the sentences where event is expressed implicitly (no event trigger exists), these models perform much worse instead. Since DMCNN performs relatively better overall, we take this model as a reference to examine the impact of different amounts of labeled triggers. Figure 3(a) plot the performance patterns of DMCNN and TAM with different rates of dropping out the labeled event triggers for supervision. The results shows that with few trigger labeling, the performance of DMCNN model experiences a large deterioration. In contrast, TAM experience only a very small performance decrease, which is much desired for real-world applications.

As to the several variants of TAM, we can clearly see that excluding each feature or their combinations lead to performance deterioration to some extent. Seperately applying the position feature (*i.e.,* TAM / TD&EE) or event type feature (*i.e.,* TAM / TD&PE) shows little effect on EER task.However, when solely applying TD, the performance of EER is much improved (*i.e.,* TAM / EE&PE), which verifies that the upstream TD task is helpful for the downstream EER task. In addition, it is observed that the inclusion of the event type feature and position feature both effectively enhance the performance of the TD based EER task, especially for sentence with multiple events (*i.e.,* TAM / EE and TAM / PE). It

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.



(a) The performance pattern of different numbers of labled triggers.

(b) The performance pattern of varying λ values.

Figure 3: Ratios at different ranks (a), and performance with λ values (b).

Example a:

中兴通讯(000063)财务弄虚作假疑问重重 夏草指其信息披露避重就轻

There exists many doubts for the financial fraud of ZTE(000063), Xiacao accused ZTE for intencially evading the crucial point of information disclosure

TAM：财务弄虚作假(Finacial Fraud) Others：无 (None)

Example b:

北京恒昌柯桥分公司涉及受害人134人，合计追回非法吸储资金3300万元

134 victims are affected by Beijing Hengchang Keqiao branch, and totally 33 million yuan of illegal absorption of public deposits is recovered

TAM：非法吸储(Illegal absorption of public deposits) Others：无 (None)

Figure 4: Examples of trigger detection by TAM.

may because that the event type feature helps the model know which event trigger should be attentioned and the position feature offers the position information between entity candidates and event triggers.

Trigger Detection. To evaluate the performance of the trigger detection network, we randomly pick 240 sentences from the test set and check the detected event trigger manually. Some event triggers are not existing in the constructed trigger dictionary because they rarely appear in the training set, but they are the synonymous expressions of the triggers covered by the dictionary. The classification performance are listed in Table 2. As can be seen, the trigger detection network outperforms the other methods in terms of $F1$. Note that the synonymous expressions of the event triggers covered by the dictionary comprise only a small proportion of the test set. Hence, the improvement over the baseline methods for TD task is significant. Figure 4 gives two examples in which our model outperforms. It can be seen that TAM can effectively identify the synonymous expressions of the trigger but other methods fail. By including '财务造假' (financial fraud) and '非法吸收公众存款'(illegal absorption of public deposits) in the trigger dictionary, TAM successfully predicts their synonymous expression '财务弄虚作假' (financial fraud) and '非法吸储'(illegal absorption of public deposits) as triggers. Furthermore, the results also show that TAM precisely predicts the boundary of trigger phrases. These abilities are very useful for Chinese trigger detection, where numerous synonymous expressions exist and these triggers usually contains multiple tokens.

It is worth mentioning that the loss weight parameter λ has a vital influence on the performance of TAM. As can be seen in Figure 3, the performance of the downstream task EER is highly related with the upstream TD task. When λ is too large, TAM losts its generation capability and becomes the same as the other TD models. Also, if λ is set too small, nothing will be learned. In our experiment, the trigger detection network shows generation capability among $[0.03, 0.1]$.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.

4 Related Work

Event entity recognition (EER) is a novel task which has wide applications in various scenarios such as public opinion analysis, etc. Since our work jointly perform EER and TD, we will introduce the related works in these two tasks separately. For the TD task, various methods have been proposed to realize TD task by identifying the event trigger with supervision. In the early years, feature-based methods, which incorporated diverse semantic clues (such as lexical features, dependency features, etc) into feature vectors, have long been used for TD task (Ahn, 2006; Gupta and Ji, 2009; Liao and Grishman, 2010). Recently, representation based methods have achieved the state-of-art performance. These methods represented the event mentions into embeddings, and then be sent to neural networks to classify the event type (Chen et al., 2015; Nguyen and Grishman, 2015; Liu et al., 2016). However, current TD models are mainly used in English corpus. For some languages where event triggers are generally in the form of nugget (composed by several tokens), the trigger boundary mismatch problem severely deteriorates the effectiveness of the TD models (Chen and Ji, 2009). To solve that, (Li et al., 2012; Li and Zhou, 2012) defined manually character compositional patterns for Chinese event triggers. (Ghaeini et al., 2016) first applied a neural network based model FBRNNs to identify event trigger nugget. Besides, (Liu et al., 2019) proposed a nugget proposal network to realize Chinese TD by exploiting character compositional structure of Chinese event triggers. Although supervised TD models achieves a good performance in most TD tasks, these methods also require too many labeling labors. Some semi-supervised methods were proposed to relieve the cost of labeling (Liao and Grishman, 2011; Ferguson et al., 2018).

For the EER task, to the best of our knowledge, there is no previous works directly dealing with it. Previous NER methods are a clue for solving EER since they both aim at identifying named entities. However, NER method lacks event information. Previous NER+TD+ARC models are another clues for dealing with EER. Difference lies on that NER+TD+ARC aims at classifying the role of the arguments for an event type, but EER aims at recognizing the named entities that are related to a specified event. Previous methods mainly focus on the TD and ARC, and the named entities were considered as known. (Chen et al., 2015) proposed DMCNN based method to pipeline realize TD and ARC, (Nguyen et al., 2016; Liu et al., 2018; Sha et al., 2018) jointly realized ED+ARC to catch the inner relation between the two tasks. (Nguyen and Nguyen, 2019) pointed that only concerning the TD+ARC had neglected the error propagation from the NER, and further improved the model by jointly realizing NER, TD and ARC in one model.

5 Conclusion

This paper proposed a trigger-aware multi-task model (TAM) to deal with a novel event entity extraction (EER) task. TAM is composed by two subtasks:a trigger detection network for event trigger identification and an event-featured transformer-CRF for event entity recognition. The former effectively tackles the intractable Chinese event trigger detection, where triggers usually contain multiple tokens and have numerous syn-onymous expressions. Experiments show that the trigger detection network is effective for identifying the synonymous expressions of the labeled event triggers, which is useful to the downstream EER subtask, which is also robust against the availability of labeled triggers. The latter incorporates three kinds of event type relevant features and avoid the dependency modeling between the event trigger and ARC existed in the relevant techniques. The experimental results demonstrate the superiority of the proposed TAM against the existing SOTA technical alternatives. Future work will be dedicated in further improving the performance of trigger detection.

Acknowledgements

The acknowledgements should go immediately before the references. Do not number the acknowledgements section. Do not include this section when submitting your paper for review.

References

David Ahn. 2006. The stages of event extraction. In *ARTE*.

Zheng Chen and Heng Ji. 2009. Language specific issue and feature exploration in chinese event extraction. In *NAACL-HLT*, pages 209–212.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

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

Reza Ghaeini, Xiaoli Z. Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *ACL*.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *ACL-IJCNLP*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. volume abs/1508.01991.

Peifeng Li and Guodong Zhou. 2012. Employing morphological structures and sememes for chinese event extraction. In *COLING*, pages 1619–1634.

Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in chinese event extraction. In *EMNLP-CoNLL*, pages 1006–1016.

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

Shasha Liao and Ralph Grishman. 2011. Can document selection help semi-supervised learning? A case study on event extraction. In *ACL*, pages 260–265.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *ACL*.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019. Event detection without triggers. In *NAACL-HLT*, pages 735–744.

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

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *AAAI*, pages 6851–6858.

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

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 *AAAI*, pages 5916–5923.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1121-1130, Hohhot, China, Augest 13 - 15, 2021.