# A Novel Joint Framework for Multiple Chinese Events Extraction

**Nuo Xu** [1,2], **Haihua Xie** [2], **Dongyan Zhao** [1]
[1] Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China
[2] State Key Laboratory of Digital Publishing Technology,
Peking University Founder Group Co., Ltd., Beijing 100871, China
xunuo2019@pku.edu.cn, xiehh@founder.com, zhaodongyan@pku.edu.cn

## Abstract

Event extraction is an essential yet challenging task in information extraction. Previous approaches have paid little attention to the problem of roles overlap which is a common phenomenon in practice. To solve this problem, this paper defines event relation triple to explicitly represent relations among triggers, arguments and roles which are incorporated into the model to learn their inter-dependencies. A novel joint framework for multiple Chinese events extraction is proposed which jointly performs predictions for event triggers and arguments based on shared feature representations from pre-trained language model. Experimental comparison with state-of-the-art baselines on ACE 2005 dataset shows the superiority of the proposed method in both trigger classification and argument classification.

## 1 Introduction

Event extraction (EE) is of utility and challenge task in natural language processing (NLP). It aims to identify event triggers of specified types and their arguments in text. As defined in Automatic Content Extraction (ACE) program, the event extraction task is divided into two subtasks, i.e., trigger extraction (identifying and classifying event triggers) and argument extraction (identifying arguments and labeling their roles).

Chinese event extraction is a more difficult task because of language specific issue in Chinese (Chen and Ji, 2009). Since Chinese does not have delimiters between words, segmentation is usually a necessary step for further processing, leading to word-trigger mismatch problem (Lin et al., 2018). The approaches based on word-wise classification paradigm commonly suffer from this. For instance, two characters in one word "打死" (hit and die) trigger two different events: an "Attack" event triggered by "打" (hit) and a "Die" event triggered by "死" (die). It is hard to extract accurately when a trigger is part of a word or cross multiple words. To avoid this issue, we formulate Chinese event extraction as a character-based classification task. In addition, another interesting issue in event extraction which is rarely followed requires more efforts. It is the roles overlap problem that we concern in this paper, including the problems of either roles sharing the same argument or arguments overlapping on some words. There are

Proceedings of the 19th China National Conference on Computational Linguistics, pages 950-961,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

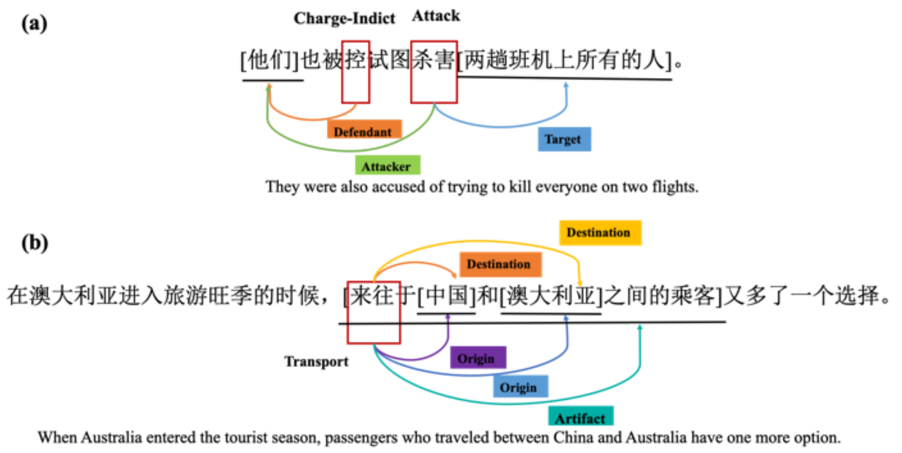
(a)
Charge-Indict
Attack
[他们]也被控试图杀害[两趟班机上所有的人]。
Defendant
Target
Attacker
They were also accused of trying to kill everyone on two flights.
(b)
Destination
Destination
在澳大利亚进入旅游旺季的时候，[来往于[中国]和[澳大利亚]之间的乘客]又多了一个选择。
Transport
Origin
Origin
Artifact
When Australia entered the tourist season, passengers who traveled between China and Australia have one more option.

Figure 1: Examples of roles overlap problem

multiple events existing in the one sentence, which commonly causes the roles overlap problem and is easy to overlook (Yang et al., 2019). Fig. 1(a) shows example of roles sharing the same argument in ACE 2005 dataset. "控" (accuse) triggers a Charge-Indict event and "杀害" (kill) triggers an Attack event, while argument "他们" (them) plays the role "Defendant" as well as the role "Attacker" at the same time. Fig. 1(b) shows example of arguments overlapping on some words in ACE 2005 dataset. "来往" (traveled between) triggers a Transport event, while argument "中国" (China) plays not only the role "Origin" but "Destination" and argument "来往于中国和澳大利亚之间的乘客" (passengers who traveled between China and Australia) plays the role "Artifact". We observe that the above two arguments overlap on word "中国" (China), which is more challenging for traditional methods to simultaneously identify these two arguments, especially for those being long noun phrases. Research shows that there exist about 10% events in ACE 2005 dataset (Doddington et al., 2004) having the roles overlap problem (Yang et al., 2019). Moreover, the results of event extraction could affect the effectiveness of many other NLP tasks, such as the construction of knowledge graph. If there exist roles overlap problems in events, the model identities accurately when it predicts any one argument or role, which leads to omission and incompleteness of information for knowledge graph construction and is obviously far from real-world applications. Therefore, the roles overlap problem is of great importance and needs to be seriously addressed.

It is thus appealing to design a single architecture to solve the problem. Although there exist prior studies that mention the roles overlap problem on ACE 2005 dataset, they share the limitations that include either depending on elaborate engineering features (i.e, hand-crafted features (He and Duan, 2019), dependency paths (Liu et al., 2018), etc.) or following the pipelined approach (Yang et al., 2019).

To overcome the issues of such prior works, in this paper, we propose a single framework to jointly extract triggers and arguments. Inspired by the effectiveness of pre-trained language models, we adopt bidirectional encoder representation from transformer (BERT) as the encoder to obtain the shared feature representations. Specifically, the relations among triggers ($t$),

Proceedings of the 19th China National Conference on Computational Linguistics, pages 950-961,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

arguments ($a$) and roles ($r$) are defined as event relation triples $< t, r, a >$ where $r$ represents the dependencies of $a$ on $t$ in the event triggered by $t$. The event sentence of Fig. 1(b) could be represented by event relation triples as $<$ 来往, Origin, 中国 $>$, $<$ 来往, Destination, 中国 $>$, $<$ 来往, Origin, 澳大利亚 $>$, $<$ 来往, Destination, 澳大利亚 $>$, $<$ 来往, Artifact, 来往于中国和澳大利亚之间的乘客 $>$. As is seen, event relation triples could explicitly describe relations among the three items. The key contribution of this paper is to design a novel joint extraction framework which jointly conducts trigger and argument extraction with incorporating the event relations defined. The task of argument classification is converted to relation extraction. Specially, to extract multiple events and relation triples, we utilize multiple sets of binary classifiers to determine the spans (each span includes a start and an end). By this approach, not only roles overlap problem but also word-trigger mismatch and word boundary problems in Chinese language are solved. Our framework avoids human involvements and elaborate engineering features in event extraction, but yields better performance over prior works.

This paper is organized as follows: Section 2 presents the related work for EE. Section 3 introduces our approach to tackle problems of roles overlap. Extensive experiments are conducted to evaluate the effectiveness of the proposed model on widely-used dataset ACE 2005 in Section 4. Besides, more rigorous evaluation criteria are adopted in experiments. Conclusions and future work are given in Section 5.

## 2 Related Work

EE is an important task which has attracted many attentions. There are two main paradigms for EE: a) the joint approach that predicts event triggers and arguments jointly, and b) the pipelined approach that first identifies trigger and then identifies arguments in separate stages (Nguyen et al., 2016). The advantages of such a joint system are twofold: (1) mitigating the error propagation from the upstream component (trigger extraction) to the downstream classifier (argument extraction), and (2) benefiting from the inter-dependencies among event triggers and argument roles (Nguyen and Nguyen, 2019). Traditional methods that rely heavily on hand-craft features are hard to transfer among languages and annotation standards (Chen and Ng, 2012; Liao and Grishman , 2010; Li et al., 2013). The neural network based methods that are able to learn features automatically (Chen et al., 2015; Feng et al., 2016; Nguyen et al., 2016; Nguyen and Grishman, 2016; Zeng et al., 2016) have achieved significant progress. Most of them have followed the pipelined approach. Some improvements have been made by jointly predicting triggers and arguments (Liu et al., 2018; Nguyen et al., 2016; Nguyen and Nguyen, 2019) and introducing more complicated architectures to capture larger scale of contexts. These methods have achieved promising results in EE.

Unfortunately, roles overlap problem has been put forward (He and Duan, 2019; Yang et al., 2019), but there are only few works in the literature to study this. He and Duan (2019) construct a multi-task learning with CRF enhanced model to jointly learn sub-events. However, their method relies on hand-crafted features and patterns, which makes them difficult to be integrated into recent neural models. The similar work to ours is Yang et al.(2019) that adopts a two-stage event extraction by adding multiple sets of binary classifiers to solve roles overlap problem. But this work needs to detect triggers and arguments separately which suffers from error propagation. It does not employ shared feature representations as we do in this work.

In recent years, pre-trained language models are successful in capturing words semantic information dynamically by considering their context. McCann et al.(2017) pre-train a deep LSTM encoder from an attentional sequence-to-sequence model for machine translation (MT) to contextualize word vectors. ELMo (Embeddings from Language Models) improve 6 challenging NLP problems by learning the internal states of the stacked bidirectional LSTM (Long Short-Term Memory) (Peters et al., 2018). Open AI GPT (Generative Pre-Training) improves the state-of-the-art in 9 of 12 tasks (Radford et al., 2018). BERT obtains new state-of-the-art results on 11 NLP tasks (Devlin et al., 2018).

## 3 Extraction Model

This section describes our approach that is designed to extract events occurring in plain text. We now define the scope of our work. The task of argument extraction is defined as automatically extracting event relation triples defined. In our model, instead of treating entity mentions as being provided by human annotators, only event label types and argument role types are utilized as training data for both trigger and argument extraction.

We propose a pre-trained language model based joint multiple Chinese event extractor (JMCEE). Let $s = \{c_1, c_2, ..., c_n\}$ be annotated sentence $s$ with $n$ as the number of characters and $c_i$ as the $i$th character. Given the set of event relation triples $E = \{< t, r, a >\}$ in $s$, the goal of our framework is to perform the task of trigger extraction $T$ and argument extraction $A$ jointly:

$$P(A, T|s) = P(A|T, s) \times P(T|s) = \prod_{(r,a)\in E|t} p((r, a)|t, s) \prod_{t\in E} p(l, t|s) \tag{1}$$

Here $(r, a) \in E|t$ denotes an argument and role pair $(r, a)$ in the event triples $E$ triggered by $t$ and $l$ denotes the event label type. Based on Eq. (1), we first predict all possible triggers and their label types in a sentence; then for each trigger, we integrate information of predicted trigger word to extract event relation triple $< t, r, a >$ by simultaneously predicting all possible roles and arguments, as illustrated in Fig. 2. We employ a pre-trained BERT encoder to learn the representation for each character in one sentence, then feed it into downstream modules. The input of our joint extractor follows the BERT, i.e. the sum of three types of embeddings, including WordPiece embedding, Position embedding and Segment embedding. Token [CLS] and [SEP] are placed at the start and end of the sentence. Multiple sets of binary classifiers are added on the top of the BERT encoder to implement predictions for multiple events and relation triples. For trigger extraction, we need to predict the start and end of event type $l$ for $c_i \in s$ ($l$ could be "Other" type to indicate that there is no word triggering any event) with each set of binary classifiers severing for an event type to determine the starts and ends of all triggers. For argument extraction, we need to extract event relation triple $< t, r, a >$ by predicting the start and end of role type $r$ for $c_i$ in sentence $s$ based on predicted triggers ($r$ is set to "Other"

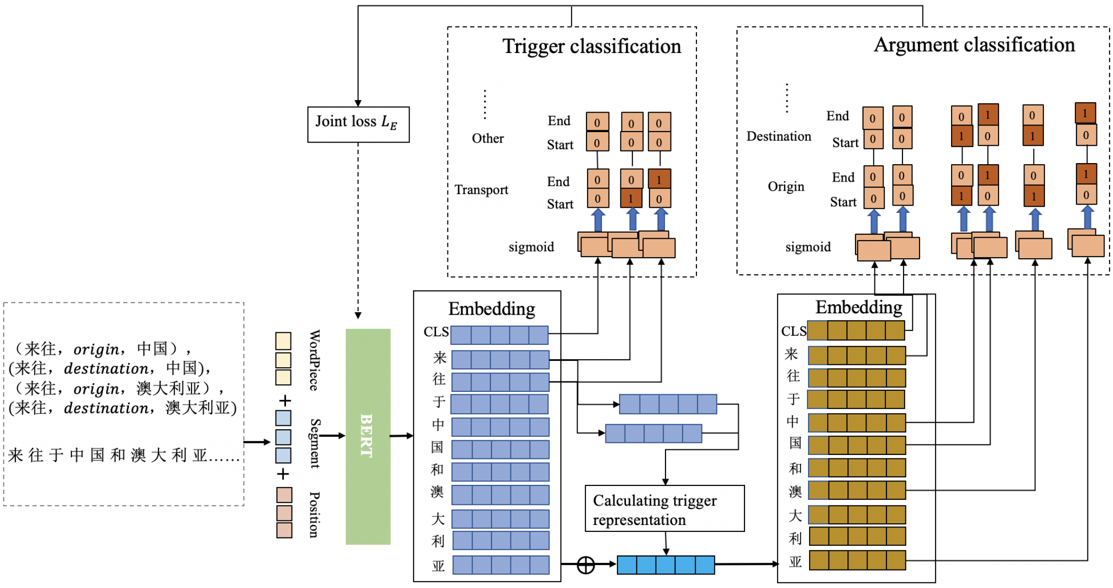

Figure 2: The framework of JMCEE, including the trigger extract component and the argument extract component. The extraction procedure of the event instance is shown.

if there is no word triggering any event as well) with each set of binary classifiers severing for a role to determine the starts and ends of all arguments that play it. The roles overlap problem could be solved since the prediction could belong to different arguments and roles. Besides, our JMCEE enables to identify those arguments being long noun phrases like “来往于中国和澳大利亚之间的乘客” (passengers who traveled between China and Australia), which tackles the word boundary problem often encountered in Chinese. Compared with sentence-level sequential modeling methods, our approach also avoids suffering low efficiency in capturing very long-range dependencies in previous works (Sha et al., 2018; Liu et al., 2018).

### 3.1 Trigger Extraction

Trigger extraction aims to predict whether a token is a start or an end of a trigger for type label $l$. A token $c_i$ is predicted as the start of a trigger with probability for type label $l$ through feeding it into a fully-connected layer with sigmoid activation function:

$$P^l_{Ts}(c_i) = \sigma(W^l_{Ts}\beta(c_i) + b^l_{Ts}) \tag{2}$$

while as the end with probability:

$$P^l_{Te}(c_i) = \sigma(W^l_{Te}\beta(c_i) + b^l_{Te}) \tag{3}$$

where we utilize subscript “s” to denote “start” and subscript “e” to denote “end”. $W_{Ts}$ and $b_{Ts}$ are respectively the trainable weights and bias of binary classifier that targets to detect starts of triggers’ labels, while $W_{Te}$ and $b_{Te}$ are respectively the trainable weights and bias of another binary classifier that targets to detect ends of triggers’ labels. $\beta$ is the BERT embedding. Set thresholds of detecting starts and ends as $\delta^l = \{\delta^l_s, \delta^l_e\}$, $\delta^l_s$ and $\delta^l_e$ are respectively the thresholds of binary classifiers that targets to detect starts and ends of triggers' labels. If $P^l_{Ts}(c_i) > \delta^l_s$, token $c_i$ is identified as the start of type label $l$. if $P^l_{Te}(c_i) > \delta^l_e$, token $c_i$ is identified as the end of type label $l$.

### 3.2 Argument Extraction

Once the triggers and their type labels have been identified, we come to the argument extraction component. Argument classification is converted to event relation extraction for triple $< t, r, a >$. Note that when the sentence is identified as "Other" type, we simply skip the following operation for argument role extraction. To better learn the inter-dependencies among the multiple events appearing in one sentence, we randomly pick one of predicted triggers in a sentence during the training phase, while in the evaluation phase, all the predicted triggers are picked in turn to predict corresponding arguments and roles played in the triggering events. We integrate information of predicted trigger word to argument extraction component. In ACE corpus, more than 98.5% triggers contain no more than 3 characters, so we simply pick the embedding vectors of start $\beta_s(c_i)$ and end $\beta_e(c_j)$ of one predicted trigger word $t$, and then generate representation of trigger word $\beta(t)$ by averaging these two vectors.

$$\beta(t) = \frac{(\beta_s(c_i) + \beta_e(c_j))}{2} \tag{4}$$

When obtain representations of trigger words $\beta(t)$, we add original embedding generated by BERT and $\beta(t)$ together:

$$\beta^{'}(s) = \beta(s) + \beta(t) \tag{5}$$

After integrate information of predicted trigger word to BERT sentence encoding, feed $\beta^{'}(s)$ into a full-connected layer with sigmoid activation function. A token $c_k$ is predicted as the start of an argument triggered by word $t$ which plays role $r$ with probability:

$$P_{As}(c_k, r|t) = \sigma(W^r_{As}\beta^{'}(c_k) + b^r_{As}) \tag{6}$$

while as the end triggered by word t with probability:

$$P_{Ae}(c_k, r|t) = \sigma(W^r_{Ae}\beta^{'}(c_k) + b^r_{Ae}) \tag{7}$$

where $W_{Ae}$ and $b_{As}$ are respectively the trainable weights and bias of binary classifier that targets to detect starts of arguments' roles, while $W_{Ae}$ and $b_{Ae}$ are respectively the trainable weights of the other binary classifier that detects ends of arguments' roles. Set thresholds of detecting starts and ends as $\varepsilon^r = \{\varepsilon^r_s, \varepsilon^r_e\}$, $\varepsilon^r_s$ and $\varepsilon^r_e$ are respectively the thresholds of binary classifiers that target to detect starts and ends of triggers' labels. If $P_{As}(c_k, r|t) > \varepsilon^r_s$ ,token $c_k$ is identified as the start of argument role $r$. if $P_{Ae}(c_k, r|t) > \varepsilon^r_e$, token $c_k$ is identified as the end of argument role $r$. Algorithm 1 is utilized to detect each token to determine triggers, types, arguments and roles.

### 3.3 Model Training

We train the joint model and define $L_T$ as the loss function of all binary classifiers that are responsible for detecting triggers, shown as follows:

$$L_T = \frac{1}{m \times n}(\sum_{l=0}^{m}\sum_{i=0}^{n} -logP_{Ts}^{l}(c_i) + \sum_{l=0}^{m}\sum_{i=0}^{n} -logP_{Te}^{l}(c_i)) \tag{8}$$

$L_T$ denotes the average of cross entropy of output probabilities of all binary classifiers which detect starts and ends of triggers on each type label. In the same way, we define $L_A$ as the loss function of all binary classifiers that are responsible for detecting event relation triples:

$$L_A = \frac{1}{m \times n}(\sum_{r=0}^{m}\sum_{i=0}^{n} -logP_{As}(c_k, r|t) + \sum_{r=0}^{m}\sum_{i=0}^{n} -logP_{Ae}(c_k, r|t)) \tag{9}$$

Where $m$ denotes the sum of event label types and argument role types. $L_A$ denotes the average of cross entropy of output probabilities of all binary classifiers which detect starts and ends of arguments on each role. The final loss function $L_E = L_T + L_A$. We minimize the final loss function to optimize the parameters of the model.

**Algorithm 1** trigger and argument identification

```
Input: $P_{Ts}^l$,$P_{Te}^l$,$P_{As}$, $P_{Ae}$,predicted trigger matrix $TP$,predicted argument matrix $AP$, sentence $s$, label list $L$
Output: predicted trigger list $L_T$, length of $L_T$ $l$, predicted argument list $L_A$
Take out matrix $S_t$ of ids and labels of starts that satisfy $P_{Ts}^l > \delta_s^l$ from $TP$ and matrix $E_t$ of ids and labels of ends that satisfy $P_{Te}^l > \delta_e^l$ from $AP$
for each $(id_s$,$l_s)$ in $S_t$ do
    for each $(id_e$,$l_e)$ in $E_t$ do
        if $id_s < id_e \& l_s == l_e$ then
            $trigger \leftarrow s[id_s - 1, id_e]$
            $label \leftarrow L[l_e]$
            $Append[trigger, label]toL_T$
            $break$
        end if
    end for
end for
return $L_t$
if $L_T$ then
    for $i = 0 \rightarrow l$ do
        Take out matrix $S_{ai}$ of ids and labels of starts that satisfy $P_{As} > \varepsilon_s^r$ from $AP$ and matrix $E_{ai}$ of ids and labels of ends that satisfy $P_{Ae} > \varepsilon_e^r$ for $i$th trigger from $AP$
        for each $(id_{si}$,$r_{si})$ in $S_{ai}$ do
            for each $(id_{ei}$,$r_{ei})$ in $E_{ai}$ do
                if $id_{si} < id_{ei} \& r_{si} == r_{ei}$ then
                    $argument \leftarrow s[id_{si} - 1, id_{ei}]$
                    $role \leftarrow L[r_{ei}]$
                    $Append[argument, role]toL_A$
                    $break$
                end if
            end for
        end for
    end for
end if
return $L_A$
```

## 4 Experiments

We evaluate JMCEE framework on the ACE 2005 dataset that contains 633 Chinese documents. We follow the same setup as (Chen and Ji, 2009; Lin et al., 2018; Zeng et al., 2016), in which 549/20/64 documents are used for training/development/test set. The proposed model is compared with the following state-of-the-art methods:

1) DMCNN (Chen et al., 2015) adopts dynamic multi-pooling CNN to extract sentence-level features automatically.

2) Rich-C (Chen and Ng, 2012) is a joint-learning, knowledge-rich approach including character-based features and discourse consistency features, which is the feature-based state-of-art system.

3) C-BiLSTM (Zeng et al., 2016) designs a convolutional Bi-LSTM model which conduct Chinese event extraction from perspective of a character-level sequential labeling paradigm.

4) NPNs (Lin et al., 2018) performs event extraction in a character-wise paradigm, where a hybrid representation for each character is learned to capture both structural and semantic information from both characters and words.

ACE 2005 dataset annotates 33 event subtypes and 35 role classes. The tasks of event trigger classification and argument classification in this paper are combined into a 70-category task along with "None" word and "Other" type. In order to evaluate the effectiveness of our proposed model, we evaluate models by micro-averaged Precision (P), Recall (R) and F1-score followed the computation measures of Chen and Ji (2009). The following criteria are utilized to evaluate the performance of predicted results:

1) A trigger prediction is correct only if its span and type match with the golden labels.

2) An argument prediction is correct only if its span, role, related trigger and trigger type match with the golden labels.

It is worth noting that all the predicted roles for an argument are required to match with the golden labels, instead of just one of them. We take a further step to see the impacts of pipelined model and joint model. The pipelined model called MCEE which identifies triggers and arguments in two separate stages based our classification algorithm. The highest F-score parameters on the development set are picked and listed in Table 1.

| Hyper-parameter | Trigger classification | Argument classification |
|---|---:|---:|
| character embedding | 768 | 768 |
| maximum length | 510 | 510 |
| batch size | 8 | 8 |
| learning rate of Adam | 0.0005 | 0.0005 |
| classification thresholds | [0.5,0.5,0.5,0.5] | [0.5,0.4,0.5,0.4] |

Table 1: Hyper-parameters for experiments.

### 4.1 Overall Results

Table 2 shows the results of trigger extraction on ACE 2005. As is seen, our JMCEE framework achieves the best F1 scores for trigger classification among all the compared methods.

Note that the results of Rich-C could obtain more accurate estimation of model performance since it performed 10-fold cross-validation experiments. However, our JMCEE gains at least 8% F1-score improvements on trigger classification task on ACE 2005, which steadily outperforms all baselines. The improvement on the trigger extraction is quite significant, with a sharp increase of near 10% on the F1 score compared with these conventional methods.

| Model | Trigger identification | | | Trigger classification | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| DMCNN | 66.6 | 63.6 | 65.1 | 61.6 | 58.8 | 60.2 |
| Rich-C | 62.2 | 71.9 | 66.7 | 58.9 | 68.1 | 63.2 |
| C-BiLSTM | 65.6 | 66.7 | 66.1 | 60.0 | 60.9 | 60.4 |
| NPNs | 75.9 | 61.2 | 67.8 | 73.8 | 59.6 | 65.9 |
| MCEE(BERT-Pipeline) | 82.5 | 78.0 | 80.2 | 72.6 | 68.2 | 70.3 |
| JMCEE(BERT-Joint) | **84.3** | **80.4** | **82.3** | **76.4** | **71.7** | **74.0** |

Table 2: Comparison of different methods on Chinese trigger extraction on ACE 2005 test set. Bold denotes the best result.

Table 3 shows results of argument extraction. Compared with these baselines, our JMCEE is at least 3% higher over other models on F1-score on argument classification task. While the improvement in argument extraction is not so obvious comparing to trigger extraction. This is probably due to the rigorous evaluation metric we have taken and the difficulty of argument extraction. Note that by our approach we identify 89% overlap roles in test set. Moreover, results show that our joint model substantially outperforms the pipelined model whether on trigger classification or argument classification. It is seen that joint model enables to capture the dependencies and interactions between the two subtasks and communicate deeper information between them, and thus improves the overall performance.

| Model | Argument identification | | | Argument classification | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| Rich-C | 43.6 | **57.3** | 49.5 | 39.2 | **51.6** | 44.6 |
| C-BiLSTM | 53.0 | 52.2 | 52.6 | 47.3 | 46.6 | 46.9 |
| MCEE(BERT-Pipeline) | 59.5 | 40.4 | 48.1 | 51.9 | 37.5 | 43.6 |
| JMCEE(BERT-Joint) | **66.3** | 45.2 | **53.7** | **53.7** | 46.7 | **50.0** |

Table 3: Comparison of different methods on Chinese argument extraction on ACE 2005 test set. Bold denotes the best result.

### 4.2 The Effect of Classification Thresholds

The effectiveness of thresholds settings for the trigger and argument classification is studies in this subsection. Table 4 lists the results of thresholds settings of the starts and ends of both two tasks. Specially, we tune two set of thresholds of starts and ends of trigger and arguments through setting $\delta^l$ to be 0.5, 0.5 and setting $\varepsilon^r$ ranging from 0.5 to 0.4. Then, set $\delta^l$ to be 0.5, 0.4 and set $\varepsilon^r$ ranging from 0.5 to 0.4. By analyzing the results, we find that the best performance of JMCEE on trigger extraction is achieved with parameters 0.5, 0.5, 0.5, 0.5, while the best performance of JMCEE on argument extraction is achieved with parameters 0.5, 0.4, 0.5, 0.4.

It suggests that when the ends of thresholds of both trigger and argument classification are set to be 0.4 could identify more candidate triggers and arguments. More candidate triggers could contribute to identifying arguments as we incorporate inter-dependencies between event triggers and argument roles in our joint extraction architecture, while the increased triggers could bring more noise to trigger classification with decreasing on the F1 score.

| $\delta_l$ | | $\varepsilon_r$ | | Trigger classification | | | Argument classification | | |
|---|---|---|---|---|---|---|---|---|---|
| Start | End | Start | End | P | R | F1 | P | R | F1 |
| 0.5 | 0.5 | 0.5 | 0.5 | **76.4** | **71.7** | **74.0** | 53.4 | 43.7 | 48.0 |
| 0.5 | 0.5 | 0.5 | 0.4 | 71.2 | 68.9 | 70.0 | 50.3 | 44.9 | 47.5 |
| 0.5 | 0.5 | 0.4 | 0.5 | 74.1 | 69.6 | 71.8 | 52.6 | 45.7 | 48.9 |
| 0.5 | 0.4 | 0.5 | 0.5 | 74.6 | 69.2 | 71.8 | 49.5 | 44.2 | 46.7 |
| 0.5 | 0.4 | 0.5 | 0.4 | 73.8 | 71.4 | 72.6 | **53.7** | **46.7** | **50.0** |
| 0.5 | 0.4 | 0.4 | 0.5 | 72.0 | 70.7 | 71.3 | 47.8 | 47.5 | 47.7 |

Table 4: Results of thresholds settings for the start and end of trigger and argument classification. Bold denotes the best result

Overall, the experimental results are remarkable facts given that our framework achieves better performance without any external and manually-generated features. We consider this as a strong promise toward our proposed joint framework which could be used as a good starting point.

## 5 Conclusions

In this paper, we propose a simple yet effective joint Chinese multiple events extraction framework which jointly extracts triggers and arguments by adopting a pre-trained BERT encoder without elaborate engineering features. Our contribution in this work is as follows:

1) Event relation triple is defined and incorporated into our framework to learn inter-dependencies among event triggers, arguments and arguments roles, which solves the roles overlap problem.

2) Our framework performs event extraction in a character-wise paradigm by utilizing multiple sets of binary classifiers to determine the spans, which allows to extract multiple events and relation triples and avoids Chinese language specific issues such as word-trigger mismatch and word boundary problem.

Experiments have shown that our method outperforms conventional methods. We believe our proposed framework could be applied to many other NLP tasks for exploiting inner composition structure during extraction, such as Entity Relation Extraction. Our future work will focus on data generation to enrich training data and try to extend our framework to the open domain.

## Acknowledgements

This work has been supported by National Key Research and Development Program(No.2019YFB1406302), China Postdoctoral Science Foundation(NO.2020M670057) and Beijing Postdoctoral Research Foundation(No.ZZ2019-92).

## References

Chen Chen and Vincent Ng. 2012. *Joint Modeling for Chinese Event Extraction with Rich Linguistic Features.* Proceedings of COLING 2012, 529–544.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng and Jun Zhao. 2015. *Event Extraction via Dynamic Multi-pooling Convolutional Neural Networks* . Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,vol.1,167–176.

Zheng Chen and Heng Ji. 2009. *Language Specific Issue and Feature Exploration in Chinese Event Extraction.* Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 209–212.

Jacob Devlin, Ming-W. Chang, Kenton Lee and Kristina Toutanova. 1972. *Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv preprint arXiv:1810.04805

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel and Ralph Weischedel. 2004. *The Automatic Content Extraction (ACE) Program-tasks, Data, and Evaluation.* LREC,vol.2

Xiaocheng Feng, Bing Qin and Ting Liu. 2016. *A Language-independent Neural Network for Event Detection..* Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol.2, 66–71.

Ruifang He and Shaoyang Duan. 2019. Joint Chinese Event Extraction based Multi-task Learning. *Journal of Software*, 30(4):1015–1030.

Qi Li, Heng Ji and Liang Huang. 2013. *Joint Event Extraction via Structured Prediction with Global Features.* Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, vol.1, 73–82.

Shasha Liao and Ralph Grishman. 2010. *Using Document Level Cross-event Inference to Improve Event Extraction.* Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 789–797.

Hongyu Lin, Yaojie Lu, Xianpei Han and Le Sun. 2018. *Joint Chinese Event Extraction based Multi-task Learning.* Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,1565–1574.

Jian Liu, Yubo Chen, Kang Liu and Jun Zhao. 2018. *Event Detection via Gated Multilingual Attention Mechanism.* Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 4865–4872.

Xiao Liu, Zhunchen Luo and Heyan Huang. 2018. *Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation.* arXiv preprint arXiv:1809.09078.

Bryan McCann, James Bradbury, Caiming Xiong and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. *In Advances in Neural Information Processing Systems*, 6294–6305.

Trung-M. Nguyen and Thien-H. Nguyen. 2019. *One for all: Neural Joint Modeling of Entities and Events.* Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 6851–6858.

Thien Huu Nguyen, Kyunghyun Cho and Ralph Grishman. 2016. *Joint Event Extraction via Recurrent Neural Networks.* Proceedings of NAACL-HLT 2016, 300–309.

Thien Huu Nguyen and Ralph Grishman. 2016. *Modeling Skip-grams for Event Detection with Convolutional Neural Networks.* Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 886–891.

Matthew E. Peters, Mark Neumann, Mohit Iyyer and Matt Gardner. 2018. *Deep Contextualized Word Representations.* Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, 2227–2237 .

Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever. 2018. *Improving Language Understanding by Generative Pre-training.* https://www.cs.ubc.ca/ amuham01/LING530/papers/radford2018improving.pdf.

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*. Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 5916–5923.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan and Dongsheng Li. 2019. *Exploring Pre-trained Language Models for Event Extraction and Generation*. Proceedings of the 57th Annual Meeting of the Association for Computa-tional Linguistics, 5284–5294.

Ying Zeng, Honghui Yang, Yansong Feng and Dongyan Zhao. 2016. *A Convolution Bilstm Neural Network Model for Chinese Event Extraction*. In: Lin CY., Xue N., Zhao D., Huang X., Feng Y. (eds) Natural Language Understanding and Intelligent Applications. ICCPOL 2016, NLPCC 2016. LNCS, vol. 10102, 275–287.