Chinese Event Trigger Recommendation Model for High-Accuracy Applications

Jiashun Duan and Xin Zhang *

Posted Date: 14 September 2024

doi: 10.20944/preprints202409.1129.v1

Keywords: event detection; event trigger recommendation; human–machine collaboration; span model; boundary smoothing; natural language processing

Article

Chinese Event Trigger Recommendation Model for High-Accuracy Applications

Jiashun Duan and Xin Zhang *

National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha 410073, China

* Correspondence: shinezhang_nudt@163.com

Featured Application: Authors are encouraged to provide a concise description of the specific application or a potential application of the work. This section is not mandatory.

Abstract: Event detection provides a crucial foundation for intelligent text analysis and supports numerous downstream applications. However, existing detection models often fail to meet the high accuracy requirements and typically require a human–machine collaborative detection mode involving "machine detection followed by manual audit and correction." Analysis reveals that the recall rate of machine detection is a major factor limiting the efficiency of human–machine collaboration. To address this at its root and ensure high recall, an event trigger recommendation task is introduced, and a span-based recommendation model for Chinese, SpETR (Span-based Event Trigger Recommender), is developed. This model integrates confidence information from both trigger identification and event classification stages to recommend candidate triggers and their event types for each event. To enhance recall, a semantic boundary smoothing method is proposed, which assigns soft labels to spans around gold triggers based on semantic completeness and part-of-speech, combined with a modified focal loss function to make the model predictions smoother. Experiments on DuEE and a self-built dataset show that SpETR is most effective when the number of recommendations is set to 3, achieving recall rates of 97.76% and 98.70%, respectively, comprising improvements of 4.30% and 5.22% over the optimal baseline models. In terms of human–machine collaborative detection, SpETR saved 30.6% and 25.8% of time compared to traditional detection models on two datasets, and the average F1 score of the annotation results was also improved by 5.35% and 3.48%, respectively.

Keywords: event detection; event trigger recommendation; human–machine collaboration; span model; boundary smoothing; natural language processing

MSC: 68T30; 68T50

1. Introduction

Event detection is the process by which a computer automatically identifies the event triggers [1] (the word or phrase that best represents the core meaning of an event) contained in text and determines the type of event. It is an important basis for intelligent text analysis, which can support downstream applications such as information retrieval, intelligent question answering, and knowledge graph construction. Within the field of event detection, the judicial event detection and political event monitoring tasks involved in information retrieval have high accuracy requirements (almost no error). In this paper, this kind of application is called a "high-accuracy downstream application," or "high-accuracy application" for short.

Due to its huge application value, event detection technology has attracted widespread attention and is a long-term research hotspot in the field of artificial intelligence and natural language processing. The introduction of deep learning approaches, especially pre-trained language models [2], has significantly improved detection performance over the past decade. However, due to the widespread existence of complex language phenomena such as the coexistence of multiple events,

 doi:10.20944/preprints202409.1129.v1

multiple triggers referring to the same event, and the traditional "either 1 or 0" hard label [3] easily leading to overfitting and other problems, the average accuracy (always using the F1 score, which harmonizes recall and precision, to measure performance on the test set) of the existing methods on closed datasets is only about 85% [4–7]. This will be lower, potentially even much lower than 80%, in the open corpus environment of practical applications. This cannot directly meet the needs of these high-accuracy applications. Therefore, in practice, professionals need to audit and correct the machine output, through the human–machine collaborative detection mode of "machine detection—manual audit and correction," in order to ensure that the final results are exactly correct. In the process, for the error of machine output, the auditor only needs to judge the labeling criteria, and adjust or delete annotations locally, and the time taken is usually less than the full manual annotation (without machine participation). For the omission of machine output, the auditor needs to re-read the text, then manually mark the missing triggers and their type according to the context, and the time taken is even longer than that for full manual annotation. Therefore, the recall rate of the machine output results is the key to affecting the efficiency of human–machine cooperative detection. However, existing detection methods usually only output a single result for each detected event, pursuing as high an F1 score as possible, and the recall rate has not received enough attention, resulting in a correspondingly high missed detection rate. Therefore, the overall efficiency of human–machine collaborative detection still has a lot of room for improvement.

In view of the above problems, we take improving the recall rate of machine output results as our first goal, aiming to improve the overall efficiency of human–machine collaborative detection. To address this, we approach it from three aspects: task modeling, model construction and inference, and model training (1) In terms of task modeling, we introduce an event trigger recommendation task that requires the machine to identify all potential events for each preferred recommended candidate trigger and event type with high recall, in order to ensure the auditor can focus on the machine output audit correction and avoid having to re-read text and complete missing parts, thus reducing the time needed. (2) In model construction and inference, we propose the Chinese-oriented span-based event trigger recommendation model SpETR (Span-based Event Trigger Recommender). This classifies the text span as the minimal processing unit directly (such models are referred to as "span models" hereafter), avoiding cumbersome decoding processes and improving recall rates. It achieves event trigger identification and classification through a two-stage process. We design a candidate recommendation algorithm, integrating the confidence levels from two stages and considering multiple events, recommend candidate triggers and their event types for each potential event, further ensuring high recall rates. (3) In model training, we propose a semantic-based boundary smoothing method. For the spans around gold triggers, considering the overlap with the gold trigger, semantic integrity and part-of-speech collocation, we assign them soft labels, forming a smooth area around the gold triggers. Combined with the modified focal loss function, the joint confidence of the model output is smoother, thus further improving the recall rate.

SpETR is integrated into the event annotation system through designing a system interface and optimizing the interaction mode to improve the efficiency of human–machine collaboration. Due to the lack of political events in the public Chinese event detection dataset, we also build a political event detection dataset containing nearly 1000 samples, hereinafter referred to as the "self-built dataset."

In summary, the main contributions of this study are as followes:

1. We introduce a new task: the event trigger recommendation task for high-accuracy applications. By identifying potential events as much as possible and recommending candidate triggers and their event types, we aim to reduce the proportion of cases where auditors need to re-read text to detect the omissions, thereby improving the efficiency of human–machine collaborative detection.
2. We construct an event trigger recommendation model, SpETR, which combines the confidence generated in the two stages of trigger identification and classification, and design the recommendation algorithm considering multiple events to output candidate triggers and event types. For the model training, the boundary smoothing method based on semantic integrity is proposed, and the spans around the hard labeled triggers are assigned soft labels according to

 doi:10.20944/preprints202409.1129.v1

the hard labeled ones. With the modified focal loss, the confidence of the model output is smoother and the quality of candidate results is improved.

3. The experimental results show the following: (1) The recall rate of SpETR with different number of recommendations K is significantly better than other baseline models. When $K = 3$, it can well meet the requirements of high recall rate and does not cause an excessive burden. At this time, the recall rate on the DuEE and self-built datasets reaches 97.76% and 98.70%, respectively, which is 4.30% and 5.22% higher compared with the optimal baseline model. (2) Ten testers were invited to conduct human–machine collaboration detection on the "competition" and "politics" domain. Compared with the traditional model, the average time consumption of SpETR decreased by 30.6% and 25.8%, respectively, and the average F1 score of the annotated results increased by 5.35% and 3.48%, respectively.

2. Related Work

The research fields related to this study include event detection, span models, and human–machine collaborative information extraction. This section briefly describes the current relevant work in these fields.

2.1. Event Detection

Deep learning-based event detection methods use neural networks to extract deep semantic features. The emergence of pre-trained language models such as BERT [2] turns the event detection method into the "pre-training + fine-tuning" paradigm, which further improves the model performance. Existing methods can be broadly divided into three categories: span-based classification, sequence labeling, and pointer network.

2.1.1. Span-Based Classification

This method directly classifies the text spans to determine the type of events; some limit the span length to 1 and then degenerate into word-level classification (character-level for Chinese). The DMCNN model was proposed by Chen et al. [8]. To extract the semantic features of each word using a convolutional neural network and a dynamic multi-pooling layer, Wang et al. [4,9] used enhanced word representation with event-type prompts. Guan et al. [7] used trigger theory element interaction information to better predict triggers; He et al.[10] utilized graph convolutional networks to learn directed graph relationships between entities and to get node embeddings, thereby enhancing the word representations. Liu et al. [11] and Wei et al. [12] studied incremental learning scenarios; Guzman-Nateras et al. [13] and Yue et al. [14] studied small-sample scenarios; and Liu et al. [15] studied the sample partial annotation scenario. The above studies modeled the decoding process as a word-level classification. Xu et al. [5] took span as the minimum processing unit of the event detection task, using the trigger–argument interaction attention-enhanced span, showing excellent performance in the event detection task. Liu et al. [16] in an incremental learning scenario used a soft prompt into the span to improve the event detection task performance. Zhang et al. [17], on the basis of a proposed HAet framework, modeled the decoding process as span classification.

2.1.2. Sequence Labeling

This method designs the annotation scheme to annotate the text sequence to detect the triggers. Nguyen et al. [18] proposed the JRNN model, using two recurrent neural networks to learn sentence representation and make better use of global information. Using the conditional random field as the decoding layer, Wei et al. [19] enhanced the sentence semantic representation using the design dialogue, and all of them used the B-I-O sequence annotation strategy to identify event triggers. Zheng et al. [20] extended the B-I-O label scheme to accommodate more complex scenarios. However, they are all plagued by the problem of label ambiguity. Tian et al. [21] used a joint model based on the bidirectional long short-term memory network to detect events in the financial field dataset, and proposed a new sequence annotation scheme that helps to solve the multi-event problem. Cao et al. [22] designed a word pair annotation scheme in the OneEE model with the event extraction task as

 doi:10.20944/preprints202409.1129.v1

the recognition of word pair relationships and used this label to jointly decode event triggers and arguments. Li et al. [6] combined sequence annotation with the text generation and attempted to inject external knowledge through prompts.

2.1.3. Pointer Networks

Such methods can solve the overlapping and nesting problems by indicating the text spans through a network composed of one or more sets of head-and-tail pointers. Yang et al. [23] proposed the PLMEE model, which designed a multi-layer pointer network to determine the start and end positions of different types of event triggers. Du et al. [24], Chen et al. [25], and Li et al. [26] modeled the event detection task as a question-and-answer task, and used a set of head-and-tail pointers to locate the triggers. Later, many models based on machine reading comprehension (MRC) followed this method.

2.1.4. Others

And some methods model event detection as sentence-level classification tasks. Wan et al. [27] captured the overall semantics of a sentence with a bidirectional attention mechanism without detecting the specific location of the trigger word. Yan et al. [28] Utilized the external knowledge graph to build a type hierarchy to enhance the semantic representation of event types, and divided the input text into word-level and context-level representations with different levels of features. They only determine the type of event that exists without detecting the corresponding trigger word, which does not support downstream tasks well.

In summary, all the above methods aim for the highest possible F1 score. However, due to the widespread occurrence of complex language phenomena such as multiple events, multiple trigger referring to the same event and son on, the F1 score typically peaks around 85% across different datasets, with recall rates also being relatively low and difficult to improve significantly. Additionally, these methods often perform worse on open-domain corpus in practical applications, failing to meet the needs of human-machine collaborative detection in high-accuracy applications.

2.2. Span Model

In recent years, the span model has been shown to effectively solve the problems of span nesting and overlap due to its intuitive modeling method, and has attracted increasing attention in information extraction, especially in the task of named entity recognition.

Dixit and Al-Onaizan [29] modeled all possible spans for the named entity recognition and relationship extraction tasks, and Sohrab and Miwa [30] built representations and classified spans using a shared underlying long short-term memory network to solve the entity nesting problem. Due to the presence of a large number of irrelevant spans, these methods have high computational costs, and Eberts et al. [31] filtered out a large number of negative samples based on the span representation and rule. Zhu et al. [32] proposed the DSpERT framework that uses low-level span representations as queries, from bottom to general word-level representations aggregated into keys and values to enhance the representation of long spans. Shen et al. [33] targeted the serious positive and negative case imbalance and hard boundary problems caused by the traditional hard label labeling mode, and balanced the positive and negative cases and alleviated the hard boundary through constructing soft positive cases. Zhu et al. [3] proposed a regularization technique for span boundary smoothing, which assigned the hard label value equally to the surrounding spans and made the model output smoother. Peng et al. [34] reassigned the span confidence according to the Gaussian distribution, and dynamically focused on the semantics around the span, rather than the whole sequence, in the span representation learning. Zheng et al. [35] used contrast learning to enhance interactions and contrast relationships between adjacent spans.

Some recent studies have tried to solve the problem of positive and negative sample imbalance and hard boundaries in the span model by constructing soft positive examples and boundary smoothing, and have achieved certain results. However, most of these methods only consider the

 doi:10.20944/preprints202409.1129.v1

positional relationship between the spans and the original label, ignoring information such as the semantic integrity and part-of-speech collocation of the spans, and fail to fully model them combined with the task characteristics.

2.3. Human–Machine Collaborative Information Extraction

Event detection is a sub-task in the field of information extraction, and many studies in other information extraction tasks also adopt the method of human–machine collaboration.

Kristjansson et al. [36] developed a system for automatically filling in structured contact information, using a constrained conditional random field to propose a mechanism to estimate the confidence of each field, and integrated the results into a human–computer interaction interface to make manual filling more intuitive and faster. Dalvi et al. [37] proposed an interactive knowledge extraction tool, which can provide a high-quality template for target relationship extraction through fast and interactive guidance. The creation of a relationship table can be much faster than when using the pure manual mode, and is more accurate and reliable than when using the fully automatic mode. Pafilis et al. [38] developed an interactive extraction tool, EXTRACT, to help administrators more effectively identify named entities such as environmental descriptors, organisms, tissues, and diseases in the text of the medical environment domain. Blankemeier et al. [39] proposed the KRISS-Search method for the medical text naming entity task. In the span recommendation module, manual feedback was divided into three categories, completely correct, partially overlapping and completely non-overlapping, and the model was improved by collecting manual feedback. Duan et al. [40] proposed an interactive event detection model based on machine reading comprehension, using the predictive confidence of head-to-tail pointers and event type classifiers to output candidate candidates for manual screening to improve human–machine collaboration efficiency. In [41], in the task of automatic entity extraction from the Chinese ancient language corpus, the authors used a multi-label automatic aggregation module to integrate the annotation results of different annotation personnel, and integrated the three functional modules of automatic marking, manual marking, and annotation query into the annotation interface, in order to improve the efficiency of human–machine collaborative annotation. Wei et al. [42] built a human–machine collaborative information extraction toolkit assisted by a large language model, and implemented pre-annotation based on model prediction.

At present, the research on human–machine collaborative information extraction mainly focuses on annotation tools and system design, and there are few studies on the extraction model itself; almost all of these are for named entity identification and relationship extraction; except for our previous study [40] of this research group, there is little research on event detection tasks. However, [40] only put forward the idea of a recommendation task and did not provide an accurate task definition. Moreover, the corresponding method is limited by the head-to-end pointer decoding mechanism, such that the quality of the recommended candidate results is not high enough, and the recall rate still has room for improvement.

3. Task Definition

The event trigger recommendation task is oriented to the "machine detection—manual audit correction" human–machine collaborative event detection mode, requiring the machine to identify as many potential events as possible, and recommending the candidate triggers and their event types for each event.

3.1. Formulation

Given a text sequence $X = \{x_1, x_2, \ldots, x_l\}$ composed of l words (characters for Chinese). Assuming that there are C event types of interest and X contains M events of interest, the task is recommending K candidate triggers and event types for each event. This can be described as the following transformation f:

 doi:10.20944/preprints202409.1129.v1

$$f: X \xrightarrow{ETR} \{(s,c)_m^k\}_{m=1,\dots,M'}^{k=1,\dots,K} \quad (1)$$

Specifically, we output an $M' \times K$ combinations of span and its event type , denoted as (s,c), where M' denotes the number of predicted events, $s = \{x_{st},\dots,x_{ed}\}$ is a span of X, briefly recorded as $s = (st, ed)$, and $c \in \{1,\dots,C\}$ denotes the event type s belongs to. Further, let the output set $\{(s,c)_m^k\}_{m=1,\dots,M'}^{k=1,\dots,K}$ be Y, Equation (1) can be simplified as follows:

$$f: X \xrightarrow{ETR} Y \quad (2)$$

3.2. Characteristics

Based on practical application experience and manual annotation logic, the recommendation task of event triggers has the following features:

- In the ideal case, (1) $M' = M$; (2) $K = 1$; (3) all spans' boundaries in Y are correct; (4) corresponding event types c are also correct. The judgment of (3) and (4) requires the gold triggers and their event types, which are generated by manual prior annotation. At this time, the recall and precision are 100%.
- Taking a step back, a high recall rate can be obtained as long as the combination of the real trigger and its type (s,c) exists in the machine output set Y, and the auditor can directly select the real result from Y.
- Taking another step back, from the perspective of saving the auditor's time in checking the omissions, the overall efficiency of the human–machine collaborative detection should be improved, as long as each gold trigger has a span s overlapping or even close to it in Y. It can help the auditor to quickly locate events and save time.
- Generally speaking, increasing K helps to improve the recall rate, but too large a K will produce too many candidate results requiring manual audit, which will reduce the efficiency.

3.3. Connection and Distinction

It is worth noting that when K is set to 1, the event trigger recommendation task degenerates into a traditional event detection task. The difference between them is that traditional event detection task is designed for scenarios where the machine performs event detection independently, outputting only one trigger for each predicted event and aiming for a high F1 score. In contrast, recommendation task is designed for human-machine collaborative detection, recommending candidate triggers for each predicted event and aiming for a high recall rate. Therefore, targeted improvements and optimizations need to be made at both the model training and inference levels to meet the requirements of the recommendation task.

4. Methodology

The overall structure and recommendation flow of the SpETR model are illustrated in Figure 1. First, in the trigger identification stage, spans are initially filtered using combined rules. After constructing span representations, a binary classifier is used to generate trigger confidence, and by setting a threshold, candidate triggers set is obtained. Next, in the trigger classification stage, a mult-classifier generates event type confidence. Finally, the joint confidence is assessed by considering confidences from two stages, and the candidate recommendation algorithm considering multiple events is designed to recommend candidate triggers and event types for potential events.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 September 2024 doi:10.20944/preprints202409.1129.v1

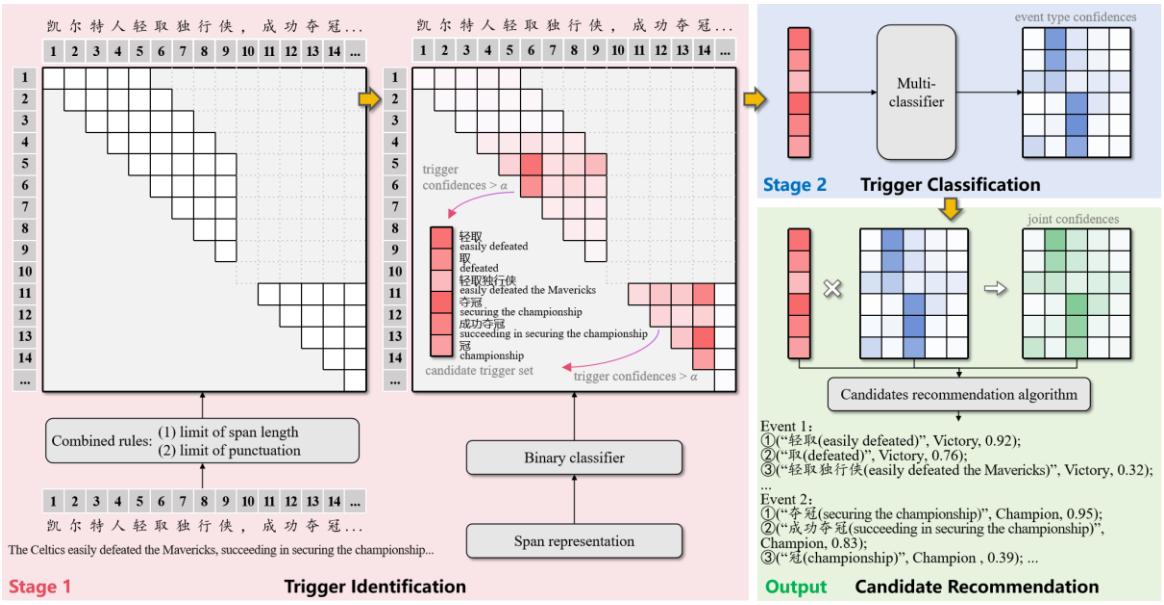


Figure 1. The structure of SpETR. Where each square represents a span, red represents trigger confidence, blue represents event type confidence, and green represents two-stage joint confidence.

4.1. Trigger Identification

At this stage, spans are quickly filtered based on combined rules after text input to reduce computational load. A pretrained model is used to construct span representations, which are then input to the binary classifier to obtain the trigger confidence. Finally, by setting a threshold, the candidate triggers set is derived.

4.1.1. Combined Rules Filtering

The traditional span model enumerates all spans in the text sequence X for classification, which will not only have a large computational burden but also cause a serious imbalance between positive and negative cases. In order to improve the efficiency of model training and inference, we use the combined rules to filter the spans and obtain filtered spans set $S = \{s_1, \ldots, s_i, \ldots, s_{N_1}\}$, where N_1 denotes the number of spans obtained, $s_i = (st_i, ed_i)$ denotes the i-th span, and st_i and ed_i denote the start and end positions of s_i in X, respectively.

Specifically, through the statistical analysis of public event detection datasets and the self-built dataset, the following applies with respect to the combined rules of span length and punctuation: when the span length is less than the predefined maximum length limit L and the span does not contain punctuation, such as "，。！？" (Chinese punctuation), it may be a trigger.

4.1.2. Span Representation

The text sequence X is input into the pre-trained model to obtain the text representation $H = \{h_1, h_2, \ldots, h_l\}$ containing the contextual semantics, where $h_t \in \mathbb{R}^d$ denotes the representation of the t-th character and d denotes the dimension. For span $s_i = (st_i, ed_i)$, concatenate the representation of its start and end character, and length embedding $h_i^{length} \in \mathbb{R}^{d_w}$ to obtain its span representation h_i^{span}:

$$h_i^{span} = [h_{st_i}; h_{ed_i}; h_i^{length}], \tag{3}$$

where d_w denotes the dimension of the length embedding and [;] denotes a concatenation operation.

4.1.3. Binary Classification

 doi:10.20944/preprints202409.1129.v1

The representation h_i^{span} of s_i is input into a binary classifier MLP^1 composed of a multi-layer perceptron and sigmoid activation function. The confidence $p_i^{trigger}$ of s_i playing the candidate trigger is calculated and output:

$$p_i^{trigger} = Sigmoid(MLP^1(h_i^{span})) \tag{4}$$

When $p_i^{trigger}$ is greater than the trigger identification threshold α, s_i is considered a candidate trigger. This produces the candidate triggers set $S' = \{s'_1, \dots, s'_i, \dots, s'_{N_2}\}$ involved in the next stage trigger classification, and N_2 is the number of candidate triggers.

4.2. Trigger Classification

After identifying the span corresponding to the candidate trigger, it is also necessary to judge its event type. For the candidate triggers set S' obtained above, their confidence vectors belonging to each predefined event types are output via a multiclass classifier.

Specifically, the representation $h_i'^{span}$ of the candidate trigger s'_i is input into a multiclass classifier composed of multi-layer perceptron MLP^2 and a softmax activation function to obtain the event type confidence vector $\boldsymbol{p}_i^{event\_type} \in \mathbb{R}^{n+1}$, following Equation (5):

$$\boldsymbol{p}_i^{event\_type} = softmax(MLP^2(h_i'^{span})), \tag{5}$$

where n denotes the number of predefined event types and "+ 1" is to include the case where the candidate trigger does not belong to any predefined class, thus filtering out unreasonable triggers or non-concerned events.

4.3. Candidate Recommendation

The joint confidence vector $\boldsymbol{p}_i^{joint} \in \mathbb{R}^{n+1}$ is calculated based on the two-stage output of trigger identification and classification,

$$\boldsymbol{p}_i^{joint} = 2(\boldsymbol{p}_i^{trigger} \circ \boldsymbol{p}_i^{event\_type})/(\boldsymbol{p}_i^{trigger} + \boldsymbol{p}_i^{event\_type}), \tag{6}$$

where $\boldsymbol{p}_i^{trigger} \in \mathbb{R}^{n+1}$ is the vector extended by $p_i^{trigger}$, and "$\circ$" and "/", respectively, denote the multiplication and division of the corresponding position elements of the vector.

Let the confidence tuple corresponding to s'_i be $\boldsymbol{p}_i$:

$$\boldsymbol{p}_i = (p_i^{trigger}, \boldsymbol{p}_i^{event\_type}, \boldsymbol{p}_i^{joint}). \tag{7}$$

We design a candidate trigger recommendation algorithm considering multiple events to recommend candidate triggers and their event types for potential events. First, judge the potential events in the input text according to the trigger confidence $p_i^{trigger}$ and spatial distribution, and then recommend K candidate triggers and event types for the event according to the joint confidence $p_{i,j}^{joint}$, as shown in Algorithm 1.

Algorithm 1. Candidate recommendation algorithm considering multiple events

Input: candidate trigger set $S' = \{s'_1, \dots, s'_i, \dots, s'_{N_2}\}$,
candidate trigger confidence tuple set $P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_i, \dots, \boldsymbol{p}_{N_2}\}$,
where $\boldsymbol{p}_i = (p_i^{trigger}, \boldsymbol{p}_i^{event\_type}, \boldsymbol{p}_i^{joint})$

Output: candidate triggers and event type for all potential events Y

```
Y ← {}    # Initialize the model output
Arrange the S' in descending order according to trigger confidence p_i^trigger
Q ← {}    # The set of candidate triggers representing potential events
for s'_i in S' do
  if s'_i does not overlap with any q_j ∈ Q do
    Q ← Q ∪ {s'_i}
  end if
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 September 2024 doi:10.20944/preprints202409.1129.v1


```
end for
for q_j in Q do
    I ← {}    # Candidate triggers and their confidence tuple for one potential event
    for s'_i in S' do
        if s'_i overlaps with q_j do
            I ← I ∪ {(s'_i, p_i)}
        end if
    end for
    Y ← Y ∪ {TopK(I, p_{i,j}^{joint})}    # For each potential event, the maximum K joint confidences are found, and their corresponding tuples, like (candidate trigger, event type, joint confidence), are output as recommendations
end for
```

4.4. Model Training

SpETR recommends candidate triggers according to confidence to improve recall rate. However, the "either 1 or 0" traditional hard label produces an obvious distinction between adjacent spans, as shown in Figure 2a. In the trigger identification stage, only the hard label corresponding span "轻取 (easily defeated)" and "夺冠 (securing the championship)" are considered as positive, while the semantically similar spans around them, such as "取 (defeated)", "轻取独行侠 (defeated the Mavericks)", "成功夺冠 (succeeding in securing the championship)" and "冠 (championship)" are all considered as negative, which easily leads to overfitting. This causes the output confidence to deviate from the real situation, which is not conducive to recommending candidates according to the confidence.

We propose a semantic-based boundary smoothing method, with the help of Chinese part-of-speech tagging toolkit pkuseg [43] to perform word segmentation and part-of-speech tagging on the input text. Through incorporating span integrity and part-of-speech collocation information, assigns soft labels are assigned to the spans around hard labels in the trigger identification stage, as shown in Figure 2b. In the classification stage, event type labels are assigned based on these soft labels.

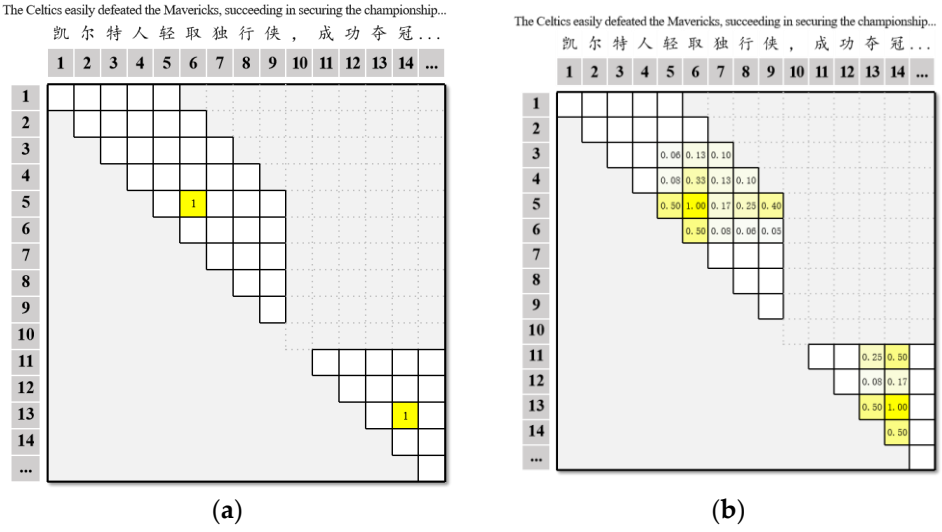


Figure 2. Instances of different label assignment methods. (**a**) Traditional hard label. (**b**) Semantic-based boundary smoothing.

4.4.1. Semantic-Based Boundary Smoothing

Let t_i be the trigger nearest to s_i and overlapping with it, and $\tilde{y}_i \in [0,1]$ be the soft label of s_i as trigger. In order to make the output confidence of the trigger identification stage closer to the real

 doi:10.20944/preprints202409.1129.v1

situation, this information needs to be reflected in the soft label. However, no one knows exactly the "real situation," so we model it from the perspective of trigger annotation.

First, consider the overlap degree between s_i and t_i. Generally, spans overlapping with t_i are semantically close to t_i, which are likely to be annotated as triggers. Define the overlap score between s_i and t_i as f_1:

$$f_1(s_i, t_i) = \frac{s_i \cap t_i}{s_i \cup t_i} \tag{8}$$

Second, the triggers should have semantic integrity. For example, in the four spans of "成功夺冠 (succeeding in securing the championship)," "夺冠 (securing the championship)," "成功夺 (none)," and "功夺冠 (none)," where "none" means that Chinese span has no direct equivalent in English , the first two have semantic integrity and are more likely to be triggers, while the last two are obviously less likely to be triggers. With the help of the Chinese part-of-speech annotation toolkit pkuseg [43], preprocess the input text and obtain its segmentation and part-of-speech information, define the semantic integrity score f_2 as follows:

$$f_2(s_i) = \begin{cases} 1, & s_i \text{ is in segmentation results} \\ 0.5, & s_i \text{ is not in segmentation results} \end{cases} \tag{9}$$

Third, consider the part-of-speech collocation of triggers. Usually, a single word trigger is mostly a verb or a noun, while the word group as the trigger is mostly a "verb + auxiliary," "preposition + verb," and so on. Reasonable part-of-speech collocation can be obtained from the historical annotation information, so similar collocation in the dataset is reasonable; otherwise, it is regarded as unreasonable. The part-of-speech score f_3 is calculated as follows:

$$f_3(s_i) = \begin{cases} 1, & \text{the part of speech of } s_i \text{ is reasonable} \\ 0.5, & \text{the part of speech of } s_i \text{ is not reasonable} \end{cases} \tag{10}$$

Considering the above three aspects, the soft label $\tilde{y}_i$ of s_i as the trigger is calculated as follows:

$$\tilde{y}_i = f_1(s_i, t_i) \cdot f_2(s_i) \cdot f_3(s_i), \tag{11}$$

4.4.2. Loss Function

In order to help the model better learn the information contained in the soft label, the weight-adaptive focal loss is used to measure the training loss of the trigger identification stage. For the filtered span set $S = \{s_1, \ldots, s_i, \ldots, s_{N_1}\}$, according to the soft label $\tilde{y}_i$ of each span s_i, and then $L_{trigger}$ is calculated as follows:

$$w_i = \begin{cases} \tilde{y}_i, & \tilde{y}_i \geq \alpha \\ 1 - \tilde{y}_i, & \tilde{y}_i < \alpha' \end{cases} \tag{12}$$

$$L_{trigger} = \sum_{i}^{N_1} \mathbf{1}_{[\tilde{y}_i \geq \alpha]} w_i \left(1 - p_i^{trigger}\right)^{\gamma} log\, p_i^{trigger} + \mathbf{1}_{[\tilde{y}_i < \alpha]} w_i p_i^{trigger^{\gamma}} log(1 - p_i^{trigger}), \tag{13}$$

where $\mathbf{1}_{[\cdot]}$ is the indicative function, taking 1 when satisfying the condition in parentheses and 0 otherwise.

In the trigger classification stage, for the candidate trigger set $S' = \{s_1', \ldots, s_i', \ldots, s_{N_2}'\}$, use multi-classification cross-entropy as the loss function L_{event_type}:

$$L_{event\_type} = \sum_{i}^{N_2} CE(y_i^{event\_type}, p_i^{event\_type}), \tag{14}$$

where $CE(\cdot)$ is the multi-classification cross-entropy function, $y_i^{event\_type} \in \mathbb{R}^{n+1}$ is the event type one-hot label of the candidate trigger s_i', 1 at the subordinate event type and 0 at the rest. In particular, when the s_i' soft label $\tilde{y}_i > \alpha$, s_i' is considered to have the same event type as t_i, otherwise considered as the "other" class.

 doi:10.20944/preprints202409.1129.v1

The trigger identification and classification are weighted by λ_1 and λ_2, respectively, and the joint loss L_{joint} is calculated as follows to train the model:

$$L_{joint} = \lambda_1 L_{trigger} + \lambda_2 L_{event\_type} \quad (15)$$

5. Experiments

5.1. Experimental Setup

We conducted five experiments or analyses comprising the following: (1) comparative experiment; (2) ablation study; (3) case study; (4) parameter analysis; (5) human–machine collaborative detection experiment.

Experiment (1) demonstrates the superiority of SpETR and its effectiveness compared to the baseline model. Experiment (2) analyzes the role of each component of the model by removing them. Study (3) visually illustrates the advantages of the model by providing case studies. Analysis (4) explores the sensitivity of the main parameters and their impacts. Experiment (5) involves inviting testers to perform event annotations to verify the improvements that SpETR brings to human–machine collaborative detection.

5.1.1. Datasets

To verify the model performance of SpETR, experiments were performed on the DuEE dataset [44] and a self-built dataset. The former is the largest Chinese event detection dataset at present, which is jointly built by Baidu, the China Computer Society, the Chinese Information Society, and data resource developers from universities and enterprises. DuEE defines 9 categories and 65 subcategories of events. The training set contains 12,000 sentence-level samples and the test set contains 1498 samples. Many of the texts in the dataset are from Baidu Information. Compared with traditional news and information, they have a higher freedom of expression, cover a wider range of knowledge and cause more difficulties.

Due to the lack of political event types in the DuEE dataset, which is not enough to support the training and validating of the model in the political field, we constructed a Chinese event detection dataset in the political field according to the standard of DuEE, hereinafter referred to as the self-built dataset. Specifically, a politically related body of data was taken from the Xinhua News Agency, Phoenix News, and Sina News Website. After cleaning and sentence pretreatment, relying on the event annotation system, we built nine events including meeting, speech, and deployment, creating a self-built dataset including nearly 1000 sentence-level samples. Similarly to the division of DuEE, we randomly selected 80% for the training set and used the remaining 20% as the valid and test set. The statistics of both datasets are shown in Table 1.

Table 1. Statistics of DuEE and self-built datasets.

Datasets	Domain	Event Type	Statistic Specification	Training Set	Valid & Test Set
DuEE	9	65	Sentences	11,958	1498
			Events	13,915	1790
Self-built	1	9	Sentences	767	182
			Events	939	230

5.1.2. Baselines

In order to verify the effectiveness and advancement of SpETR in the event trigger recommendation task, the recommendation model [40] based on machine reading comprehension (MRC-based) was first selected as the baseline model. There is no other published trigger recommendation model or method. Therefore, recent span models with soft positive examples and

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 September 2024 doi:10.20944/preprints202409.1129.v1

boundary smoothing ideas were also selected to transform them into trigger recommendation models for comparison.

- MRC-based [40]. Prepend a description of the event type to the text as a query, determine the position of a trigger using head and tail pointers, then identify its event type. Based on the trigger and event type confidence to recommend candidate results.
- Locate and Label [33] (LL). Using the two-stage method of determining the span position and then conducting multiclass classification, the soft positive case is constructed according to the degree of overlap between the span and the positive case, and the model is trained with focal loss. As the model is oriented to the named entity recognition task, for fair comparison in the event trigger recommendation task, the same implementation as SpETR is adopted on the span presentation.
- Boundary Smooth [3] (BS). Using the one-stage method of directly classifying the type of spans, the proposed boundary smoothing method constructs soft labels by evenly assigning the confidence of hard labels equally to the surrounding spans. As the model is oriented to the named entity recognition task, for fair comparison in the event trigger recommendation task, the same implementation as SpETR is adopted on the span presentation.
- Boundary Smooth* [3] (BS*). Similar to BS, except the span representation adopts the double affine structure proposed in their paper. To ensure consistency and fairness in the comparison experiments, LL, BS, and BS* used the same recommendation algorithm as SpETR.

5.1.3. Implementation Details

Both SpETR and the baseline models used the RoBERTa pre-trained model as the encoder to acquiring the text sequence. During the training process, the AdamW optimizer was used to train the model. In order to make the model training more stable, the learning rate warm-up and attenuation strategy were also adopted. The experimental environment configuration is shown in Table 2, and the relevant parameter settings and implications of SpETR are shown in Table 3.

Table 2. Experimental environment.

Item	Configuration
Operating system	Ubuntu 20.04.3 LTS x86_64
CPU	40 vCPU Intel(R) Xeon(R) Platinum 8457C
GPU	L20 (48 GB) * 2
Memory	200 GB
Python	3.8.10
Framework	Pytorch (1.10.0) + Transformers (4.38.2) + Cuda (12.4)

Table 3. Parameter setting.

Parameter	Description	Value	
		DuEE	Self-Built
	Main parameters		
dropout_rate	dropout rate of pre-trained model	0.1	0.45
α	trigger identification threshold	0.2	0.2
γ	focusing parameter of modified focal loss	2	2
L	limit of span length	10	6
	Optimizer parameters		
learning_rate	initial learning rate	1×10^{-5}	1×10^{-5}
warm_up_steps	warm-up steps	500	500
lr_decay_steps	decay steps	3000	3000
min_lr_rate	minimum learning rate	1×10^{-6}	1×10^{-6}

 doi:10.20944/preprints202409.1129.v1

5.1.4. Evaluation Metrics

The strict matching standard was adopted; that is, a result is considered correct only if both the boundary of the trigger and the event type are all correct. $Recall@K$ on the test set was used to evaluate the model, where K denotes the number of candidates.

$$Recall@K = \frac{recall\_n@K}{gold\_n}, \tag{16}$$

where $gold\_n$ denotes the total number of correct results and $recall\_n@K$ denotes the number of recalled correct results when K candidates are recommended.

For the human–machine collaborative detection experiment in Section 5.6, the F1 score was used to comprehensively evaluate the quality of the manual annotated results; this is calculated as follows.

$$Recall = \frac{recall\_n}{gold\_n}, \tag{17}$$

$$Precision = \frac{recall\_n}{pred\_n}, \tag{18}$$

$$F1 = 2\frac{Recall * Precision}{Recall + Precision}, \tag{19}$$

where $recall\_n$ denotes the number of correct results in the manual annotation results and $pred\_n$ denotes the total number of annotation results.

5.2. Comparative Experiment

$Recall@K$ for SpETR and the baseline models with the number of recommendations K ranging from 1 to 5 on the DuEE and self-built datasets is shown in Table 4. The best performances are highlighted in **bold** and the second-best are underlined.

Table 4. $Recall@K$ of different models on the two datasets.

Model	DuEE					Self-Built				
	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$
SpETR	**92.23**	**96.59**	**97.76**	**97.93**	**97.98**	**92.17**	**96.96**	**98.70**	**98.70**	**99.13**
MRC-based	85.47	90.28	92.46	93.80	94.47	82.61	86.09	86.96	87.39	87.83
LL	87.87	90.39	91.95	93.63	94.13	89.57	90.87	93.48	93.91	93.91
BS	91.62	92.68	93.46	94.24	94.41	86.09	87.93	90.00	90.44	90.87
BS*	91.56	92.57	93.46	94.19	94.52	84.78	89.13	90.87	90.87	91.30

The results indicate the following:

- The recall of SpETR significantly reduces at $K > 3$. Through error analysis, it was found that, at this point, the vast majority of correct results have already been recalled. The small portion of results that are not recalled consists of ambiguous cases or errors due to incorrect annotations in the dataset. Even with an increased number of recommendations, it is difficult to improve the recall rate further. Excessive recommendations may also cause interference for the auditors. Therefore, in practical applications, $K = 3$ is typically the most efficient.
- SpETR has a significant advantage in recall rate. SpETR shows a clear advantage in recall rate over the other baseline models on both the DuEE dataset and the self-built dataset. When $K = 3$, it improves by 5.30% and 11.74% compared to the MRC-based models.
- SpETR meets the high recall rate requirements. When the number of recommendations is set to 3, the recall rates on the DuEE and self-built datasets reach 97.76% and 98.70%, respectively. Given the presence of annotation errors in the datasets, this indicates that auditors can, in most cases, directly select the correct results from the recommendations without re-reading the text and manually annotating.

 doi:10.20944/preprints202409.1129.v1

In addition, all span models performed better than the MRC-based ones, indicating that the models based on direct span classification are better than those based on pointer network in terms of recall rate. In the DuEE dataset, the one-stage BS performs better, while the two-stage LL performs better on the self-built dataset. This is because the two-stage model can use all class training samples to update the classifier during the trigger identification stage, making it relatively advantageous in low-resource scenarios.

5.3. Ablation Study

To further investigate the specific effects of the two-stage design, semantic boundary smoothing, and the modified focal loss on SpETR, ablation experiments were conducted. The detailed results are shown in Table 5.

Table 5. *Recall@K* of SpETR on the two datasets in the ablation study.

Model	DuEE					Self-Built				
	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$
SpETR	**92.23**	**96.59**	**97.76**	**97.93**	**97.98**	**92.17**	**96.95**	**98.69**	**98.69**	**99.13**
w/o two-stage design	85.58	87.54	87.87	88.04	88.10	83.04	84.34	85.21	85.21	85.65
w/o semantic-based boundary smoothing	89.77	91.84	92.29	93.18	93.46	90.00	92.17	93.47	94.34	95.21
w/o modified focal loss	87.87	91.56	92.51	92.90	93.12	85.21	92.17	92.60	93.04	93.04
w/o semantic-based boundary smoothing and modified focal loss	88.60	91.40	91.68	91.73	91.73	84.34	90.00	90.86	91.30	92.17

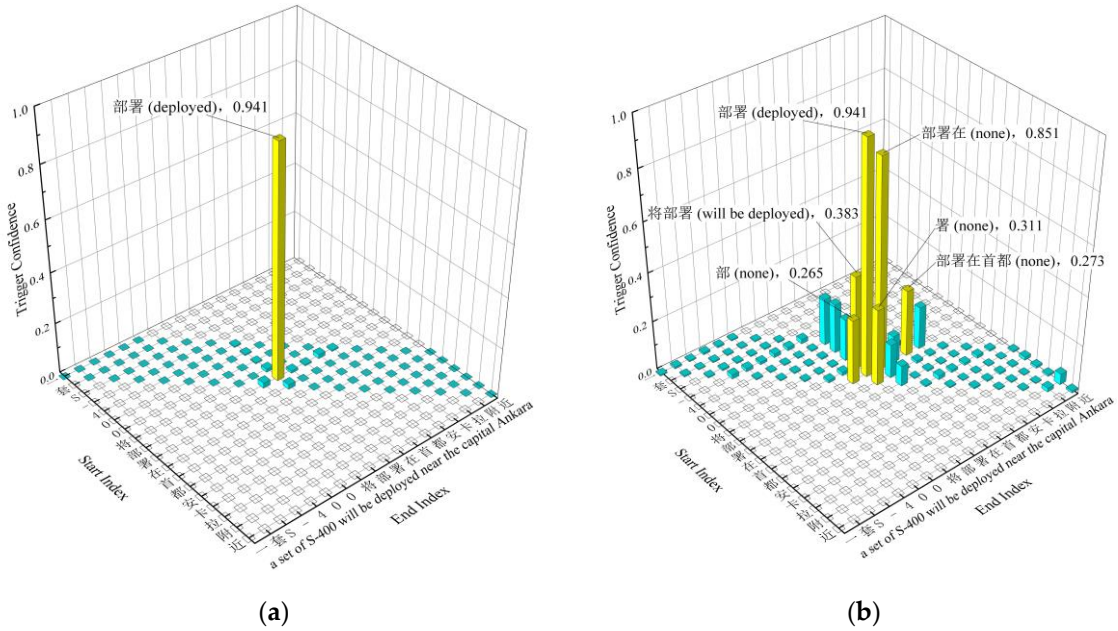
- Ablation of the two-stage design. When this part is removed, trigger identification and event classification are completed in one stage. This is simple and direct, but cannot reflect the trigger boundary and event type information in the confidence degree of the multi-classifier output. The results show that *Recall*@3 decreased by 9.89% and 13.48% on DuEE and the self-built dataset, respectively, indicating that the two-stage design of trigger identification and then trigger classification produced higher-quality candidate results.
- Ablation of semantic-based boundary smoothing methods. When this part is removed, traditional hard labels are used for training in the trigger identification stage; the model is prone to overfitting, which reduces the reliability of the output confidence. The results showed that, after removing the semantic-based boundary smoothing method, *Recall*@3 decreased by 4.75% and 5.22% on DuEE and the self-built dataset, respectively, indicating that the semantic-based boundary smoothing method can improve the candidate result quality and recall through a more refined soft label assignment method.
- Ablation of the modified focal loss function. When this part is removed, the ordinary binary cross-entropy function is adopted in the trigger identification stage during the model training. The results showed that *Recall*@3 decreased by 5.25% and 6.09% in DuEE and the self-built dataset, respectively, indicating that the modified focal loss can adjust the sample weight adaptively to alleviate the imbalance of positive and negative cases in the span model, while the common binary cross-entropy loss function does not have this characteristic, so it is easy to bias the negative cases, leading to reducing the recall rate.
- Simultaneous ablation of semantic-based boundary smoothing and modified focal loss. When these two parts are removed, the traditional hard label labeling method and binary cross-entropy loss are used in the trigger identification stage during the model training. The results showed that *Recall*@3 decreased by 6.08% and 7.83% on the DuEE and self-built datasets, respectively. This was higher than when removing either of them alone, indicating that the two parts cooperate and promote each other.

5.4. Case Study

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 September 2024 doi:10.20944/preprints202409.1129.v1

In the experiment described in Section 5.3, where both semantic boundary smoothing and modified focal loss are ablated, a text from the self-built dataset, "据土媒体报道，一套 S-400 将部署在首都安卡拉附近… (According to local media reports, a set of S-400 will be deployed near the capital Ankara…)", is used as a case study. The confidence of the trigger output by SpETR trained with traditional hard labels and cross-entropy loss is shown in Figure 3a. It is evident that the model exhibits overconfidence—showing high confidence for "部署 (deployed)" but very low confidence for surrounding semantically similar spans. This significant miscalibration of confidence is detrimental to models that rely on confidence to recommend candidates.

The visualization of trigger confidence output by complete SpETR is shown in Figure 3b. It can be seen that the confidence for "部署 (deployed)" as a trigger remains the highest, while surrounding spans such as "部署在 (none)," "将部署 (will be deployed)," "署 (none)," and "部署在首都 (none)," also had certain confidence due to their semantic similarity. The output of the model was smoother and more aligned with the actual situation, which helps in recommending higher-quality candidates based on confidence degrees.



(**a**) (**b**)

Figure 3. Trigger confidences of SpETR under different training methods. (**a**) Traditional hard label. (**b**) Semantic-based boundary smoothing. Where "none" means that Chinese span has no direct equivalent in English

5.5. Parameter Analysis

A sensitivity analysis of the four main parameters, drop_out_rate, α, γ, and L, are performed to verify their effect on SpETR. Using single-parameter sensitivity analysis, according to the specific parameter setting in Section 5.1.3, one parameter was changed and fixed at a time, and the recommended number of results K was 3. The results are shown in Figure 4. The specific analyses are as follows:

- Dropout rate of the pre-trained model (dropout_rate). This parameter represents the dropout rate of the output representation of the pre-trained model, which helps to avoid overfitting during model training. When the parameters change in the range of [0.00, 0.45], the recall rate in the DuEE dataset and self-built dataset varies little, and the overall range is relatively flat, indicating that the model has good robustness to this parameter, which is due to the simple and efficient model structure of SpETR.
- Trigger identification threshold (α). In trigger identification stage, spans with confidence greater than the parameter α are regarded as candidate triggers and are involved in the subsequent inference and training, which has a great impact on the recall rate. When α increases, it means

 doi:10.20944/preprints202409.1129.v1

that the requirement for triggers becomes "strict" and the recall rate will decrease in the DuEE and self-built datasets, the former trend is flat, indicating that the correct trigger confidence is less and evenly distributed within [0.05–0.5]; the latter decreases more and there are two abnormal points, 0.15 and 0.40, which may be due to the instability caused by the small amount of dataset. These show that the model can still show a good recall rate with increasing α when the training data is sufficient while, in the low-resource scenario, the recall rate will decrease relatively faster.

- The coefficient for the modified focal loss (γ). This adjusts the loss weight for difficult classification samples, with more ambiguous samples receiving a higher loss. As γ increases, the adjustment for difficult samples becomes stronger, causing the model to focus more on hard-to-recognize triggers, thereby increasing the recall rate. The results from the two datasets show that γ is generally positively correlated with the model's recall rate. However, an excessively large γ does not significantly improve the recall rate and may actually reduce precision, impacting the audit efficiency in practical applications.
- The limit of span length (L). This defines the maximum length of spans considered during training and inference. When L is set to 4, the model shows a noticeable increase in recall rate across both datasets, indicating that most triggers are shorter than 4. Increasing L further continues to improve the recall rate up to a point; however, it begins to decline when L exceeds 10 on the DuEE dataset and 6 on the self-built dataset. This is because, while a larger L recalls longer spans, it also introduces more negative examples, causing the model to favor negatives and thereby reducing the recall rate. Thus, L should be selected based on prior knowledge of trigger lengths in different domains.

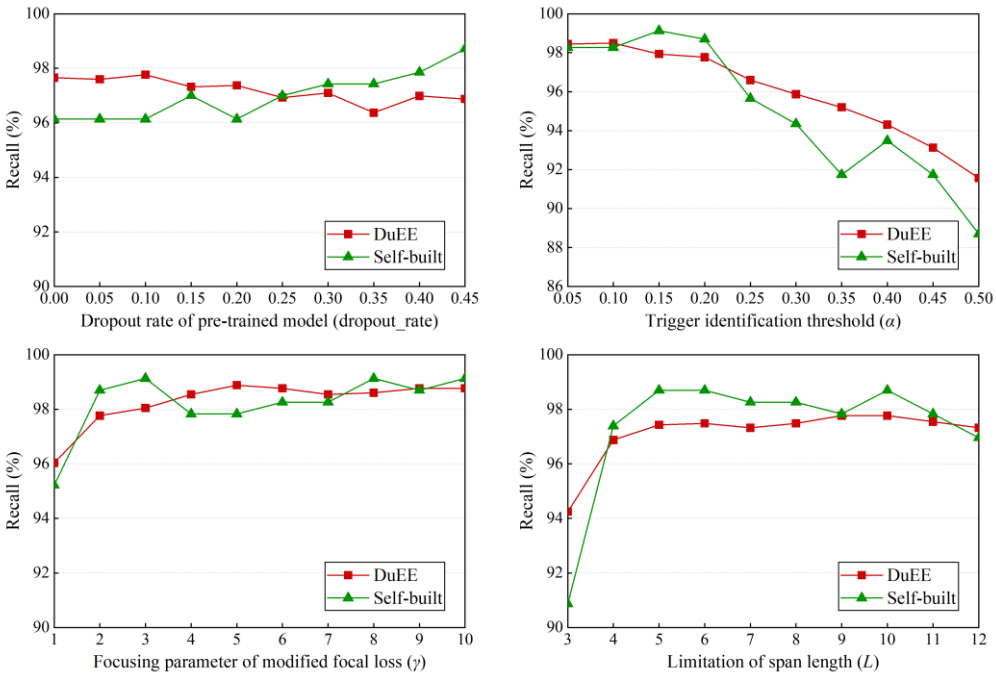


Figure 4. Results of parameter analysis.

5.6. Human–Machine Collaborative Detection

The above experiments highlight superior recall rate of SpETR. To further validate its impact on human–machine collaborative detection, we integrated SpETR into the event annotation system and invited testers for empirical analysis.

5.6.1. Event Annotation System Integration

To maximize SpETR's effectiveness in practical applications, it was integrated into the our team's existing event annotation system. The system interface was optimized based on the characteristics of

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 14 September 2024 doi:10.20944/preprints202409.1129.v1

trigger recommendation task and annotators' habits to enhance the efficiency of the human–machine collaborative event detection.

As shown in Figure 5, the annotation interface consists of four main sections:

- Text Display and Annotation: Shows the text and trigger locations, with candidate triggers highlighted by colored blocks indicating their joint confidence levels. Annotators can manually select spans and annotate triggers and event types.
- Candidate Recommendation: Displays the model's recommended candidate triggers and event types for each event, along with joint confidence levels for reference. Annotators confirm recommendations by clicking, with the first candidate automatically annotated to save time.
- Annotation Confirmation: Structurally displays annotated triggers and event types for easy verification by annotators.
- Annotated Text List: Shows the annotation status of each sample, allowing for quick navigation to specific samples and monitoring overall annotation progress.

Figure 5. Event annotation system interface.

5.6.2. Empirical Results

To verify the improvement effect of SpETR on the efficiency of human–machine collaborative detection, we used the event annotation system to compare the time consumption and result quality of human–machine collaborative detection under three conditions: (1) full manual annotation, (2) traditional model, and (3) SpETR. The specific meanings of these conditions are presented in Table 6.

Table 6. Annotation conditions.

Annotation Condition	Description
Full manual	The tester completed the event detection task independently.
Traditional model	The traditional event detection model was used to output triggers and event types for automatic annotation, with testers adjusting and manually correcting errors and omissions.

 doi:10.20944/preprints202409.1129.v1

SpETR	SpETR was used to recommend candidate triggers and event types, and automatically annotate the first candidate. Testers prioritized selecting from candidate triggers to address errors and omissions.

Ten graduate students were invited as testers. They received a standardized training, including the introduction to the event detection task, system usage, and annotation standards. Each tester was assigned to annotate 40 randomly selected samples from both the DuEE competition dataset and self-built dataset under the three different annotation conditions. The annotation time was recorded and the final annotation results' F1 scores were calculated.

Figure 6 shows the time taken and result quality under different annotation conditions. It is evident that both the traditional event detection model and SpETR significantly reduced the annotation time and improved the quality compared to full manual annotation. Due to its high recall rate, SpETR allows testers to focus on the model's recommended candidates, often eliminating the need to re-read the entire text. This results in further reduction in annotation time compared to the traditional event detection model, with average time reductions of 30.6% and 25.8% on the DuEE competition dataset and self-built dataset, respectively. Additionally, the recommended candidates provide hints to testers, especially those without event annotation experience, helping them quickly identify potential events in the text, leading to F1 score improvements of 5.35% and 3.48%, respectively.

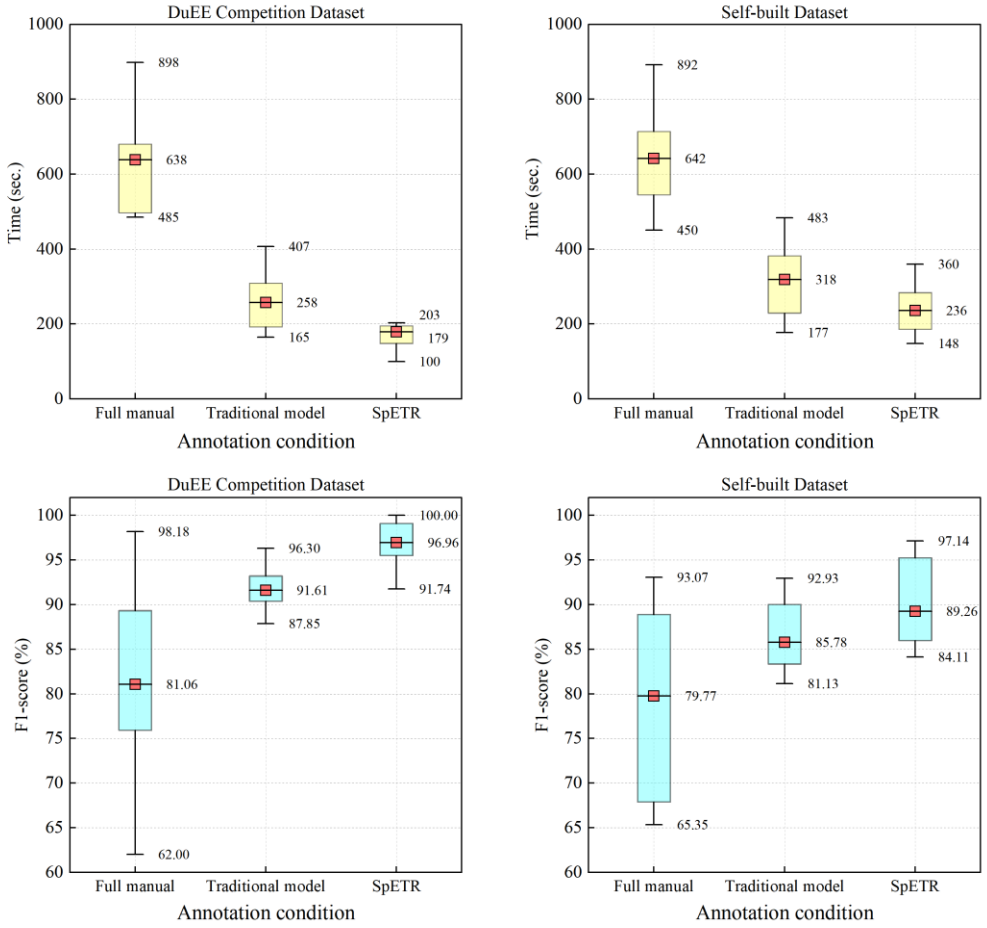


Figure 6. Time taken and quality of annotation under different conditions.

6. Conclusions

 doi:10.20944/preprints202409.1129.v1

Applications such as judicial event detection and political event monitoring demand extremely high precision of the event detection results, requiring a "machine detection—human audit and correction" collaborative mode. However, traditional models are constrained by their task modeling, leading to insufficient recall rates and requiring extensive manual audit to identify and correct omissions. To address this, we first introduced the event trigger recommendation task to fundamentally improve recall rates. Furthermore, we developed the SpETR for event trigger recommendation, treating spans as the smallest processing unit and employing a two-stage approach of trigger identification and classification. The model incorporates a multi-event candidate trigger recommendation algorithm to output candidate results. Additionally, a semantic-based boundary smoothing method is proposed which, combined with a modified focal loss function during the trigger identification stage, helps train the model to produce smoother confidence and recommend higher quality candidates. Moreover, the model is integrated into the event annotation system, optimizing the system's interface and interaction methods to leverage the advantages of the SpETR recommendation model in practical applications.

Experiments demonstrated that the two-stage design, semantic-based boundary smoothing, and modified focal loss effectively enhance the recall rate of the SpETR. With three recommended candidates, recall rates reached 97.76% and 98.70% on the DuEE and self-built datasets, respectively; meaning that, in most cases, correct results can be selected directly from the candidates without re-reading the text. The human–machine collaborative detection experiments indicate that the SpETR not only significantly saves time but also improves the annotation quality.

The limitations of our work include the following: (1) The event trigger recommendation task focuses on the recall rate. However, when the precision falls below a certain level, it can affect the efficiency of human–machine collaboration detection. At present, there is no suitable metric for evaluating this impact, and it can only be measured through annotation experiments. (2) The joint confidence output by SpETR is not comparable between samples, lacking a global reference standard for auditors. Future work should aim to achieve the following: (1) Design an evaluation metric specifically for trigger recommendation tasks to reflect the model's impact on human–machine collaboration efficiency; and (2) improve the confidence calculation method to provide a unified reference standard for auditors.

In addition, the human–machine collaboration mode is expected to persist across various fields and tasks for a considerable period into the future. This study investigated aspects such as task definition, model construction, and training within the scope of human–machine collaborative event detection. Theoretical research on human cognitive processes and underlying logic in the context of human–machine collaboration is crucial for guiding these aspects, and will be a key focus for future exploration and research.

Author Contributions: Conceptualization, X.Z. and J.D.; methodology, J.D. and X.Z.; validation, J.D.; investigation, J.D. and X.Z.; resources, X.Z.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, X.Z.; visualization, J.D.; supervision, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under the grants of NO.62102431, the National University of Defense Technology under the grant of NO.ZK21-32, and Science and Technology on Information Systems Engineering Laboratory under the grant of NO. 6142101220209.

Data Availability Statement: The data used in this study have been cited and referenced [44]. They are available at https://ai.baidu.com/broad/introduction (accessed on 18 August 2024). Other data available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Linguistic Data Consortium. ACE (Automatic Content Extraction) English Annotation Guidelines for Events Version 5.4.3. Available online: https://www.ldc.upenn.edu/ (accessed on 11 August 2024).
2. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; 2018; Volume abs/1810.04805.

3. Zhu, E.; Li, J. Boundary Smoothing for Named Entity Recognition. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 7096–7108.
4. Wang, S.; Yu, M.; Huang, L. The Art of Prompting: Event Detection based on Type Specific Prompts. *arXiv* **2022**, arXiv:2204.07241.
5. Xu, C.; Zeng, Z.; Duan, J.; Qi, Q.; Zhang, X. Extracting Events Using Spans Enhanced with Trigger-Argument Interaction. In Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering, Fuzhou, China, 17–19 November 2023. https://doi.org/10.1109/iske60036.2023.10480953.
6. Li, H.; Mo, T.; Fan, H.; Wang, J.; Wang, J.; Zhang, F.; Li, W. KiPT: Knowledge-injected Prompt Tuning for Event Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 1943–1952.
7. Guan, Y.; Chen, J.; Lecue, F.; Pan, J.; Li, J.; Li, R. Trigger-Argument based Explanation for Event Detection. In *Findings of the Association for Computational Linguistics: ACL*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023.
8. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. *Event Extraction Via Dynamic Multi-Pooling Convolutional Neural Networks*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; Volume P15-1, pp. 167–176.
9. Wang, S.; Yu, M.; Chang, S.; Sun, L.; Huang, L. Query and Extract: Refining Event Extraction as Type-oriented Binary Decoding. In *Findings of the Association for Computational Linguistics: ACL*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 169–182. https://doi.org/10.18653/v1/2022.findings-acl.16.
10. He, L.; Meng, Q.; Zhang, Q.; Duan, J.; Wang, H. Event detection using a self-constructed dependency and graph convolution network. *Applied Sciences* **2023**, *13*, 3919.
11. Liu, M.; Huang, L. Teamwork Is Not Always Good: An Empirical Study of Classifier Drift in Class-incremental Information Extraction. *arXiv* **2023**, arXiv:2305.16559.
12. Wei, K.; Zhang, Z.; Jin, L.; Guo, Z.; Li, S.; Wang, W.; Lv, J. HEFT: A History-Enhanced Feature Transfer framework for incremental event detection. *Knowl.-Based Syst.* **2022**, *254*, 109601. https://doi.org/10.1016/j.knosys.2022.109601.
13. Nateras, L.G.; Dernoncourt, F.; Nguyen, T. Hybrid Knowledge Transfer for Improved Cross-Lingual Event Detection via Hierarchical Sample Selection. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, USA, 9–14 July 2023.
14. Yue, Z.; Zeng, H.; Lan, M.; Ji, H.; Wang, D. Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning. *arXiv* **2023**; arXiv:2305.17373.
15. Liu, J.; Sui, D.; Liu, K.; Liu, H.; Zhao, Z. Learning with Partial Annotations for Event Detection. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, USA, 9–14 July 2023.
16. Liu, M.; Chang, S.; Huang, L. Incremental Prompting: Episodic Memory Prompt for Lifelong Event Detection. *arXiv* **2022**, arXiv:2204.07275. https://doi.org/10.48550/arXiv.2204.07275.
17. Zhang, C.; Cao, P.; Chen, Y.; Liu, K.; Zhang, Z.; Sun, M.; Zhao, J. Continual Few-shot Event Detection via Hierarchical Augmentation Networks. *arXiv* **2024**, arXiv:2403.17733.
18. Nguyen, T.H.; Cho, K.; Grishman, R. *Joint Event Extraction* via *Recurrent Neural Networks*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016.
19. Wei, Y.; Liu, S.; Lv, J.; Xi, X.; Yan, H.; Ye, W.; Mo, T.; Yang, F.; Wan, G. DESED: Dialogue-based Explanation for Sentence-level Event Detection. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2483–2493.
20. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction Of Entities And Relations Based on a Novel Tagging Scheme. *arXiv* **2017**, arXiv:1706.05075.
21. Tian, C.; Zhao, Y.; Ren, L. A Chinese Event Relation Extraction Model Based on BERT. In Proceedings of the 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 25–28 May 2019.
22. Cao, H.; Li, J.; Su, F.; Li, F.; Fei, H.; Wu, S.; Li, B.; Zhao, L.; Ji, D. OneEE: A One-Stage Framework for Fast Overlapping and Nested Event Extraction. *arXiv* **2022**, arXiv:2209.02693. https://doi.org/10.48550/arxiv.2209.02693.
23. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. *Exploring Pre-Trained Language Models For Event Extraction And Generation*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume P19-1, pp. 5284–5294.
24. Xinya, D.; Claire, C. *Event Extraction by Answering (Almost) Natural Questions*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 671–683.
25. Yunmo, C.; Tongfei, C.; Seth, E.; Benjamin, V.D. Reading the Manual: Event Extraction as Definition Comprehension. *arXiv* **2020**, arXiv:1912.01586. https://doi.org/10.18653/v1/2020.spnlp-1.9.

26. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. *Event Extraction as Multi-Turn Question Answering*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 829–838.
27. Wan, X.; Mao, Y.; Qi, R. Chinese Event Detection without Triggers Based on Dual Attention. *Applied Sciences* **2023**, *13*, 4523.
28. Yan, Y.; Liu, Z.; Gao, F.; Gu, J. Type Hierarchy Enhanced Event Detection without Triggers. *Applied Sciences* **2023**, *13*, 2296.
29. Dixit, K.; Al-Onaizan, Y. *Span-Level Model For Relation Extraction*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume P19-1, pp. 5308–5314.
30. Sohrab, M.G.; Miwa, M. *Deep Exhaustive Model for Nested Named Entity Recognition*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; Volume D18-1, pp. 2843–2849.
31. Markus, E.; Adrian, U. Span-Based Joint Entity And Relation Extraction with Transformer Pre-Training. *ECAI* **2020**, *325*, 2006–2013.
32. Zhu, E.; Liu, Y.; Li, J. Deep Span Representations for Named Entity Recognition. In *Findings of the Association for Computational Linguistics: ACL*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023.
33. Shen, Y.; Ma, X.; Tan, Z.; Zhang, S.; Wang, W.; Lu, W. Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. *arXiv* **2021**, arXiv:2105.06804. https://doi.org/10.18653/v1/2021.acl-long.216.
34. Peng, T.; Li, Z.; Zhang, L.; Du, B.; Zhao, H. FSUIE: A Novel Fuzzy Span Mechanism for Universal Information Extraction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, USA, 9–14 July 2023.
35. Zheng, Q.; Wu, Y.; Wang, G.; Chen, Y.; Wu, W.; Zhang, Z.; Shi, B.; Dong, B. Exploring Interactive and Contrastive Relations for Nested Named Entity Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 2899–2909. https://doi.org/10.1109/taslp.2023.3293047.
36. Kristjansson, T.T.; Culotta, A.; Viola, P.A.; McCallum, A. *Interactive Information Extraction with Constrained Conditional Random Fields*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; pp. 412–418.
37. Dalvi, B.; Bhakthavatsalam, S.; Clark, C.; Clark, P.; Etzioni, O.; Fader, A.; Groeneveld, D. IKE—An Interactive Tool for Knowledge Extraction. *Comput. Sci.* **2016**, 12–17. https://doi.org/10.18653/v1/w16-1303.
38. Pafilis, E.; Buttigieg, P.L.; Ferrell, B.; Pereira, E.; Schnetzer, J.; Arvanitidis, C.; Jensen, L.J. EXTRACT: Interactive extraction of environment metadata and term suggestion for metagenomic sample annotation. *Database J. Biol. Databases Curation* **2016**, *2016*, baw005. https://doi.org/10.1093/database/baw005.
39. Blankemeier, L.; Zhao, T.; Tinn, R.; Kiblawi, S.; Gu, Y.; Chaudhari, A.; Poon, H.; Zhang, S.; Wei, M.; Preston, J. Interactive Span Recommendation for Biomedical Text. In Proceedings of the 5th Clinical Natural Language Processing Workshop, Toronto, ON, Canada, 14 July 2023. https://doi.org/10.18653/v1/2023.clinicalnlp-1.40.
40. Duan, J.; Zhang, X.; Xu, C. LwF4IEE: An Incremental Learning Method for Interactive Event Extraction. In Proceedings of the 2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Suzhou, China, 14–16 October 2022. https://doi.org/10.1109/CyberC55534.2022.00026.
41. Chengxi, Y.; Xuemei, T.; Hao, Y.; Qi, S.; Jun, W. HanNER: A General Framework for the Automatic Extraction of Named Entities in Ancient Chinese Corpora. *J. China Soc. Sci. Tech. Inf.* **2023**, *42*, 203–216. https://doi.org/10.3772/j.issn.1000-0135.2023.02.007.
42. Wei, X.; Chen, Y.; Cheng, N.; Cui, X.; Xu, J.; Han, W. CollabKG: A Learnable Human-Machine-Cooperative Information Extraction Toolkit for (Event) Knowledge Graph Construction. *arXiv* **2023**, arXiv:2307.00769. https://doi.org/10.48550/arxiv.2307.00769.
43. Luo, R.; Xu, J.; Zhang, Y.; Ren, X.; Sun, X. PKUSEG: A Toolkit for Multi-Domain Chinese Word Segmentation. *arXiv* **2019**, arXiv:1906.11455.
44. Li, X.; Li, F.; Pan, L.; Chen, Y.; Peng, W.; Wang, Q.; Lyu, Y.; Zhu, Y. *DuEE: A Large-Scale Dataset for Chinese Event Extraction in Real-World Scenarios*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020.