

Article

Not peer-reviewed version

Chinese Document-Level Emergency Event Extraction Dataset and Corresponding Methods

[Kongbin Chu](#) , [Wenzhong Yang](#) ^{*} , Fuyuan Wei , Jiangtao Shi

Posted Date: 15 May 2023

doi: 10.20944/preprints202305.0965.v1

Keywords: candidate entity extraction; document-level event extraction; role overlapping



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Chinese Document-Level Emergency Event Extraction Dataset and Corresponding Methods

Kongbin Chu ¹, Wenzhong Yang ^{1,2,*}, Fuyuan Wei ¹ and Jiangtao Shi ¹

¹ School of Information Science and Engineering, Xinjiang University, Urumqi 830017, China

² Xinjiang Key Laboratory of Multilingual Information Technology, Xinjiang University, Urumqi 830017, China

* Correspondence: yangwenzhong@xju.edu.cn

Abstract: Structured extraction of emergency event information can effectively enhance the ability to respond to emergency events. This article focuses on the extraction of Chinese document-level emergency events, which mainly faces two key issues in this field: first, related datasets; Secondly, existing DEE (document-level event extraction) studies mostly use sequence annotation to extract candidate entities in the subtask of candidate entity extraction without considering the problem of role overlapping between candidate entities. On the one hand, this article constructs a Chinese document-level emergency extraction dataset, CDEEE, which first annotates the issues of argument scattering, multiple events, and role overlapping. On the other hand, this article proposes a model RODEE for the problem of role overlapping in DEE tasks. This model first uses two independent modules to represent the head and tail positions of candidate entities, then uses a multiplication attention mechanism to interact with the two to obtain a scoring matrix. Finally, role-overlapping candidate entities are predicted to assist in completing DEE tasks. Experiments were conducted on our manually annotated dataset, CDEEE, and the results showed that RODEE can effectively solve the problem of role overlapping in candidate entities and improve the performance of the DEE model.

Keywords: candidate entity extraction; document-level event extraction; role overlapping

1. Introduction

In the real world, emergency events[1] such as traffic accidents, fires, forest fires, earthquakes, and health safety seriously threaten the safety of human life and property, and therefore are widely concerned by people from all walks of life. In the context of the Internet information era, the rapid spread and fermentation of information about emergency events through online media will breed public opinion events and affect social public safety. Accurate and efficient access to structured information of emergency events can help relevant staff to achieve early detection and early treatment of emergency events, curb the generation of public opinion events, and maintain social public safety.

The goal of event extraction is to identify pre-specified types of events and the corresponding event arguments from plain text. A large number of previous studies have focused on sentence-level event extraction[2–4], and most of these studies were based on the evaluation of ACE[5]. These approaches based on sentence-level event extraction make predictions within a sentence and are unable to extract events between sentences. In the real world, the information of event arguments cannot be fully obtained from a single sentence, as shown in Figure 1, the event argument of the "Accident" event "重庆永川吊水洞煤矿" and "12 月 4 日 17 时许" are distributed in two different sentences, s1 and s2, so we devoted ourselves to the study of DEE. In order to extract structured event information from documents, researchers have proposed a large number of model methods and datasets for model training and validation in previous work, which are presented below in terms of DEE datasets and DEE models and methods.

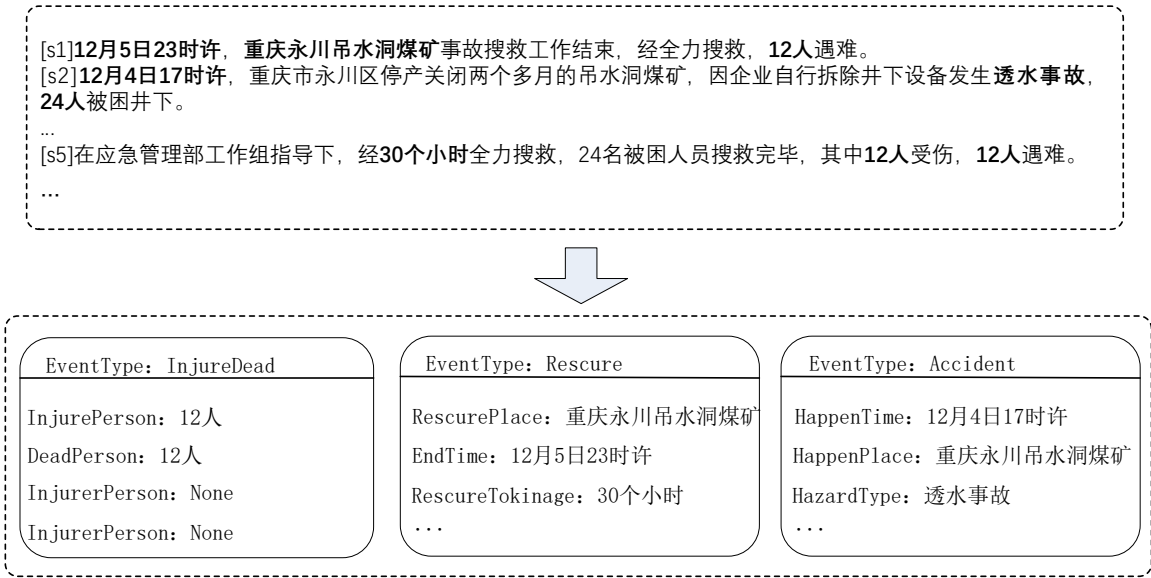


Figure 1. Event extraction instance.

A wide range of research scholars have done a lot of work on DEE datasets to be able to train and validate DEE model methods, constructing a large number of DEE data, such as the MUC-4[6] dataset, which consists of 1,700 documents annotated using an associated role-population template; Twitter dataset, which was constructed by collecting and annotating English tweets posted in December 2010, including 20 event types and 1,000 tweets; WIKIEVENTS dataset, published by Li et al.[7] as a document-level benchmark dataset, uses English Wikipedia articles as the data source; Yang et al.[8] conducted experiments on four types of financial events, namely, equity freeze events, equity pledge events, equity repurchase events, and equity increase events, and a total of 2,976 announcements were tagged. Although, a large number of DEE datasets have been constructed by domestic and foreign researchers in previous works, on the one hand, most of these datasets are English datasets which cannot be trained and validated for Chinese DEE model methods, and on the other hand, there is no DEE dataset constructed for the field of emergency events. Therefore, it is the current priority to construct Chinese document-level emergency event extraction datasets and solve the problem of missing datasets.

In terms of DEE models and methods, a large number of scholars have focused on the two challenges of argument scattering and multiple events. In particular, argument scattering refers to the fact that the event arguments of an event are scattered in multiple sentences of a document, for example, in Figure 1, the event arguments of the "Rescure" event, "重庆永川吊水洞煤矿" and "30 个小时", are distributed in two different sentences, s1 and s5 of the document; multiple events means that a document may contain several different events, for example, Figure 1 contains "InjureDead", "Rescure" and "Accident " three different events.

In a previous study, Yang et al.[8] proposed that the DCFEE model first extracts trigger words and arguments in a sentence-by-sentence manner, and then uses a convolutional neural network to classify each sentence to determine whether it is a key sentence. Meanwhile, in order to obtain the complete event arguments, an argument complementation strategy is proposed to obtain arguments from the surrounding sentences of the sentence in which the key event is located for complementation. Zheng et al.[9] redesigned the DEE task to treat the DEE task as a table-filling task using a trigger-word-free approach to populate candidate entities into a predefined event table. Specifically, they modeled DEE as a continuous prediction paradigm in which arguments are predicted in a predefined role order and multiple events are also extracted in a predefined event order. The method accomplishes the DEE task goal using trigger-word-free extraction, but since the

arguments are predicted in a predefined role order during the process of argument identification, the former argument identification results do not take into account the latter argument identification results, which leads to error propagation problems. Yang et al.[10] proposed an end-to-end model in which multiple events as well as event arguments information are extracted simultaneously from the document in a parallel manner using a multi-grain decoder after the overall document representation is obtained using multiple encoders. Based on some previous work, we can divide the DEE task into three subtasks: candidate entity extraction, event type detection and argument identification. Among them, candidate entity extraction is to extract entities related to events from the text; event type detection is to determine the types of events present in the text; and argument identification is to identify the arguments belonging to an event among the candidate entities. Candidate entity extraction, as the first subtask of DEE, affects the effectiveness of the two subsequent subtasks. Previous work has been devoted to solving the arguments scattering and multiple events problems, while ignoring the role overlapping that exist in the first subtask, which greatly affects the performance of the two subsequent subtasks as well as the overall DEE task. Role overlapping refer to the phenomenon of candidate entities playing multiple roles in the same event or in multiple different events. For example, in Figure 1, the entity "12 人" plays the role of "InjureDead" and "DeadPerson" in the event of "InjureDead"; the entity "重庆永川吊水洞煤矿" plays the role of "RescurePlace" in the "Rescure" event and "HappenPlace" role in the "Accident" event.

To cope with the above-mentioned problems of missing datasets and role overlapping, we have done the following two things. On the one hand, in order to cope with the lack of datasets, we define a framework for unexpected event extraction by analyzing and summarizing information of Chinese emergency events, and construct a Chinese document-level emergency event extraction dataset CDEEE. We defined 4 event types and 19 role types in this dataset and annotated each of the three problems of argument scattering, multiple events, and role overlapping. Finally, we annotated the CDEEE dataset consisting of 5,000 documents and 10,755 events. On the other hand, to cope with the role overlapping problem, we propose the DEE model RODEE for the role overlapping problem. In this model, we first use the pre-trained language model RoBERTa[11] to embed the text representation and then encode it using Transformer to obtain the text representation, which gives us an overall understanding of the text. Specifically, we design two separate models to represent the start position information and end position information of the candidate entities, and use multiplicative attention to interact the two to obtain the scoring matrix, so as to predict the candidate entities and assist in the event extraction task.

Overall, our main contributions are in the following three areas:

- We constructed a Chinese document-level emergency event extraction dataset CDEEE using manual annotation. In the annotation process of the dataset, we annotated the role overlapping problem in addition to the arguments scattering problem and the multiple events problem.
- We propose RODEE, a DEE model for the role overlapping problem, which first uses two independent matrices to represent the start position information and end position information of candidate entities, and then uses multiplicative attention to obtain the score matrix for prediction of candidate entities with the role overlapping problem, and finally assists in the event extraction task.
- We compare the RODEE model approach with the existing DEE model approach on the CDEEE dataset, and the experimental results show that the RODEE model approach outperforms the existing DEE model approach.

2. Related work

2.1. Document-level event extraction dataset

Event extraction datasets are the cornerstone of event extraction research, and we investigate the existing DEE datasets.

MUC-4: MUC-4 was presented at the Fourth Conference on Message Comprehension The dataset consists of 1,700 documents, where five types of events are annotated using an associated role-populated template. The dataset is mainly dedicated to the task challenge of argument scattering.

WIKIEVENTS: WIKIEVENTS was published by Li et al.[7] as a document-level benchmark dataset. The dataset is derived from English Wikipedia articles describing real-world events.

Google: The Google dataset is a subset of the GDELT Event Database1 with event-related word searches containing documents for 30 event types from 11,909 news articles.

Twitter: The Twitter dataset was collected from tweets released in December 2010 applying the Twitter Stream API, and includes 20 event types and 1,000 tweets.

NO.ANN, NO.POS, NO.NEG[8]: the researchers studied four types of financial events, equity freeze events, equity pledge events, equity repurchase events and equity increase events, and labeled 2976 announcements. NO.ANN represents the number of announcements that can be automatically labeled for each event type. NO.POS represents the total number of positive case mentions. Conversely, NO.NEG represents the number of negative mentions.

ChFinAnn[9]: using financial reports as the data source, we use the event knowledge base for event annotation in a remote-supervisory-based manner. Based on NO.ANN, NO.POS, NO.NEG, the Chinese DEE dataset in finance is further enriched and extended to include 32040 documents and 5 event types including equity freeze, equity repurchase, equity reduction, equity increase and equity pledge.

Roles Across Multiple Sentences (RAMS): RAMS was annotated by Eber et al.[12] using news as the data source for the dataset. The dataset contains 3,194 documents with 139 event types totaling 9124 events.

We performed preliminary statistics on the available DEE datasets as shown in Table 1.

Table 1. Document-level event extraction dataset.

Dataset	Doc	Event-Type	Language
MUC-4	1700	5	English
Google	11909	30	English
Twitter	1000	20	English
NO.ANN, NO.POS, NO.NEG	2976	4	Chinese
ChFinAnn	32040	5	Chinese
RAMS	3194	139	English
WIKIEVENTS	246		English

The existing DEE datasets are mostly annotated in English, which is very helpful for DEE in English. However, the available datasets for Chinese DEE tasks are relatively sparse and do not support DEE tasks in the field of emergency events.

2.2. Document-level event extraction

DEE can extract information about event arguments of interest to users directly from documents, and thus has received a great deal of attention from scholars. In some studies[13,14], the document-level event argument extraction task is considered as a populated paradigm that follows the MUC-4 task setting and is dedicated to extracting event arguments scattered in documents. In addition, Yang et al.[8], Huang et al.[15] and Li et al.[7] follow the approach of first detecting the event type and then performing event arguments extraction. Specific event trigger words are first identified to determine the event type, and then event arguments beyond the sentence boundaries are extracted. However, the trigger word-based event extraction approach does not work well in the DEE task because there are often events in documents with obscure trigger words or without trigger words. Therefore, researchers[9,10,16,17] attempted to perform DEE in a triggerless manner, where the event type is directly determined based on the document semantics. These approaches have addressed to some extent the problem of argument scattering and multiple events in the DEE task and have achieved good results. However, the role overlapping problem is ignored, and the candidate entities with role

overlapping problem cannot be extracted accurately, which further affects the overall performance of the event extraction task.

2.3. Event role overlapping

The traditional joint methods[18–22] for the role overlapping problem perform both trigger word and candidate entity extraction. They solve the problem in a sequential annotation manner and extract trigger words and candidate entities by marking sentences only once. However, these methods cannot solve overlapping event extraction because overlapping characters can lead to label conflicts when forcing to have multiple labels. In addition, some scholars[23–26], perform trigger word and candidate entity extraction in different stages. Although such pipelined approaches have the potential ability to resolve role overlapping, they usually lack explicit dependencies between trigger words and candidate entities and also suffer from error propagation. Among these studies, Yang et al.[27] and Xu et al.[28] used the construction of multiple classifiers to solve the role overlapping problem, which achieved some results in sentence-level event extraction tasks. The above studies, all of which have proposed their own methods and insights for the role overlapping problem, have alleviated the role overlapping problem to some extent. However, since the above studies are based on sentence-level event extraction, they cannot be directly transferred to DEE tasks. Therefore, the role overlapping problem in DEE tasks is still a problem that needs to be studied.

3. Chinese document-level emergency event extraction dataset

3.1. Data choice

To generate high-quality corpus data, we collected 13 types of common emergency events news reports, including traffic accidents, roofing accidents, earthquakes, collapse accidents, plane crashes, landslides, landslides, fires, mudslides, explosions, permeable accidents, forest fires, and other types of accidents, from the National Emergency Alert Information Release Network (<http://www.12379.cn/html/gzaq/fmytplz/index.shtml>; (accessed on 10 November 2022)), China News (<https://www.chinanews.com.cn/>; (accessed on 10 November 2022)), Sohu News (<https://news.sohu.com/>; (accessed on 10 November 2022)), Sina News (<https://news.sina.com.cn/>; (accessed on 10 November 2022)), People's Daily (<http://www.people.com.cn/>; accessed on 10 November 2022), Tencent News (<https://news.qq.com/>; (accessed on 10 November 2022)), and Today's Headlines (<https://www.toutiao.com/>; (accessed on 10 November 2022)). From the collected information, we screened out news reports that described the overall situation of the emergency in detail as our data to be labeled, and after screening we collected a total of 5000 news reports that met the requirements. Figure 2 counts the number of information collected on various types of emergency events.

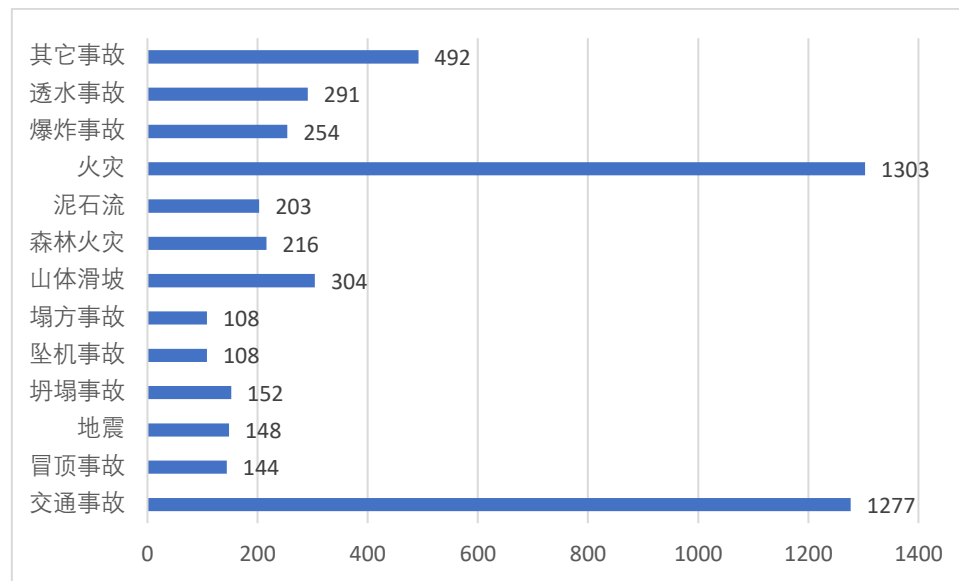


Figure 2. Statistics on the number of various emergency events.

3.2. Dataset construction

The construction of the event extraction dataset can be divided into two parts: the development of the event extraction structure and the event extraction data annotation. The event extraction structure, as the basis of the event extraction task, specifies the structured information to be annotated and extracted. Therefore, the characteristics of the information to be annotated and the characteristics of the event extraction task should be considered in the process of developing the event extraction structure.

On the one hand, we analyzed and summarized the 5000 news reports collected, and found that there were too many types of emergency events to enumerate them all. Finally, we classified the emergencies based on whether the emergencies were caused by natural factors, and the original type of emergencies became a role attribute of the emergencies "HazardType". At the same time, from the emergency event itself, the casualties caused by the emergency event and the rescue organization's rescue work for the emergency event, four types of events are defined: "InjureHead", "Rescure", "Accident" and "NaturalHazard", and each event is described in Appendix A.

On the other hand, in the process of developing event extraction structure, the verb that causes the change of things or states is generally used as the trigger word, and the time and place of the event as well as the participants are used as the key factors of the event. However, in the research related to DEE, it is found that there are often events in a chapter that do not have obvious trigger words or do not contain trigger words, so the trigger word-based event extraction method does not work well in DEE. Therefore, Zheng et al.[9] redesigned the DEE task and constructed ChFinAnn, a DEE dataset without trigger words, and validated its performance. The experimental results showed that this trigger word-free construction method met the requirements of the DEE task and greatly improved the efficiency of the data annotation work. Since our study is also based on document-level tasks, we adopt this trigger-free approach to formulate the event extraction structure.

Considering the above two aspects together, we define 19 event role types to describe event information, and in Appendix A we introduce each role and explain the event type to which it belongs.

After the development of the event extraction structure, we used manual annotation to annotate the 5000 emergency event documents we collected according to the developed event extraction structure, and annotated more than 10000 event information in total, and divided the annotated data into training set, validation set and test set according to the ratio of 8:1:1. Figure 3 shows the statistical information about the number of events in each category after dividing the dataset.

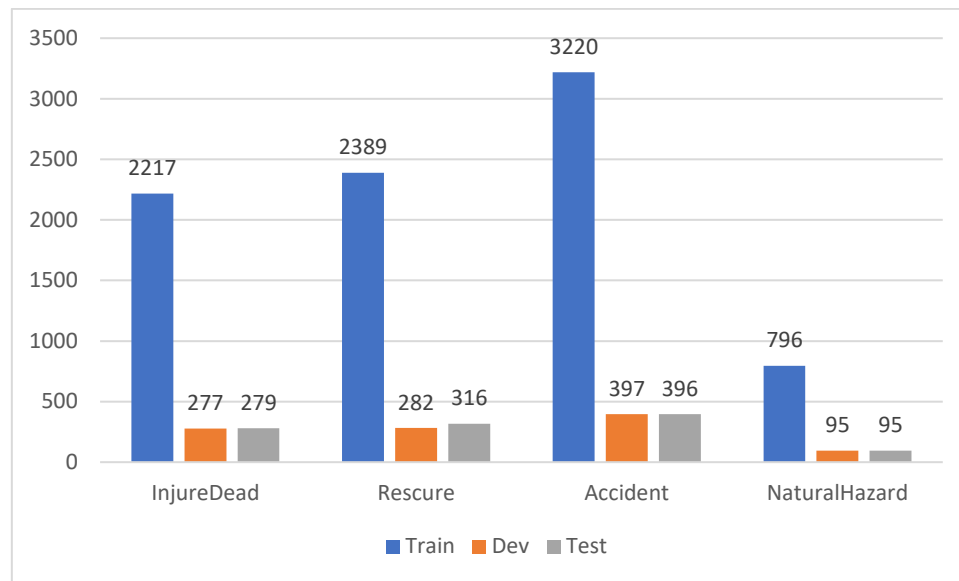


Figure 3. Statistics on the number of event types in the CDFEE dataset.

In the process of data annotation, we make the following arrangements in the annotation process for the reliability and accuracy of the annotated data: the annotated data of annotator A will be given to annotator B for validation, the annotated data of annotator B will be given to annotator C for validation, and so on the data of annotator C will be validated by annotator A. And, in order to further improve the verification work, we will adopt the voting method to choose when there is a disagreement in the verification process. After the manual verification, we further verify the quality of the dataset we use the existing DEE model for the final verification, which is described in the experimental section.

3.3. Dataset features

We not only annotate the DEE task with the unique problem of argument scattering and multiple events, but also with the problem of role overlapping, which is easily overlooked in the DEE task. Specifically, first, we annotate the event elements not only in a single sentence but also in the whole document, and in Figure 4 we count the number of sentences involved in the event element information. We can see from the figure that although some events can find all event elements in a single sentence, for most events the event elements need to be found in multiple sentences, which further demonstrates the prevalence of argument scattering in the DEE task.

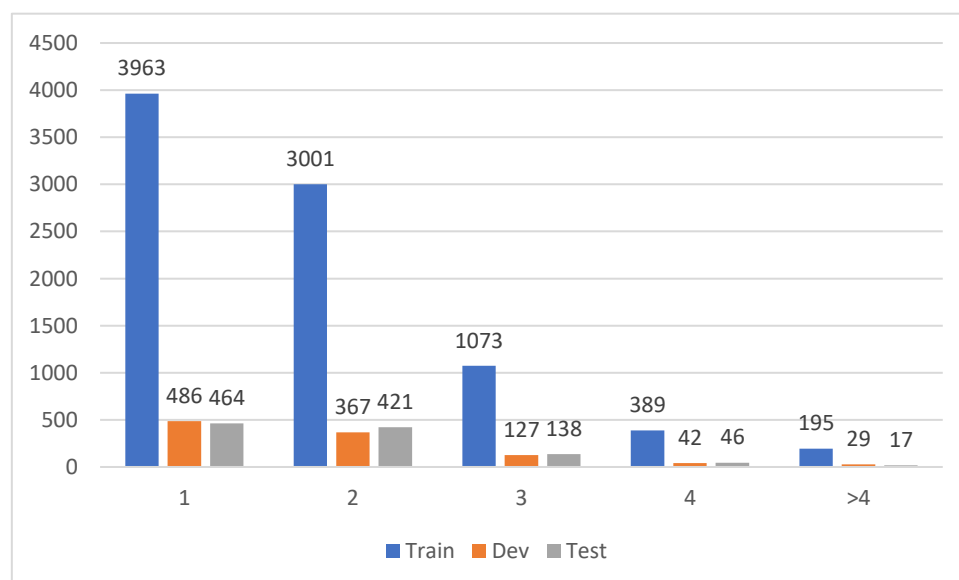
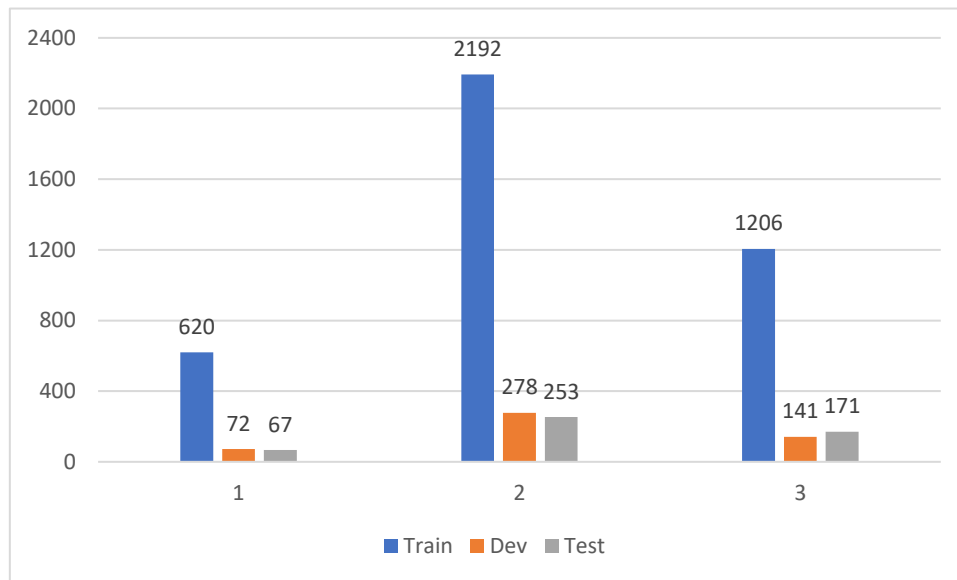
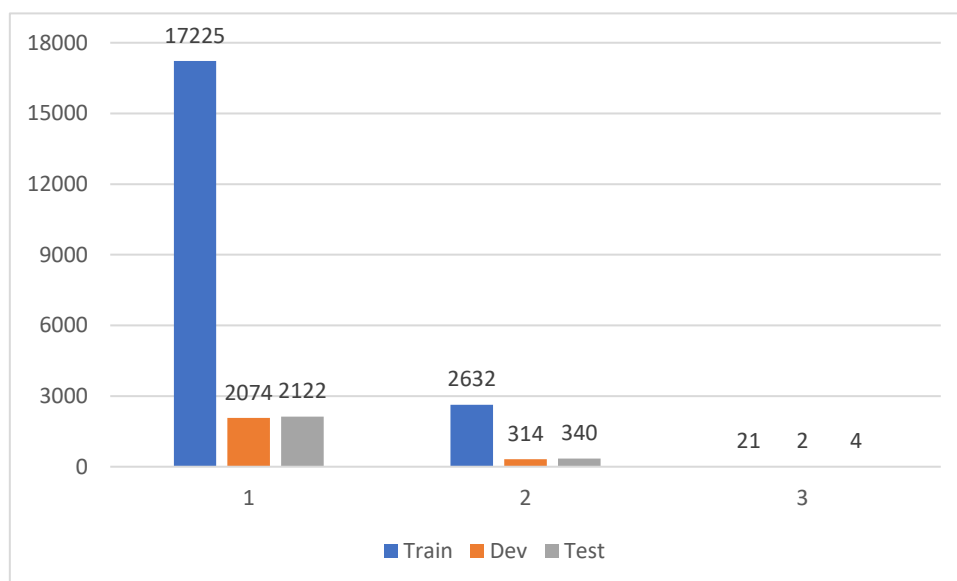


Figure 4. Statistics of the number of sentences involved in event arguments.

Second, we mark several different events in the same document, and in Figure 5 we have a count of the number of events contained in the document. We can see that most of the documents contain two or three events, except for a few documents that contain only one event. This is a good reflection of the fact that there are often multiple events in a document.

**Figure 5.** Document contains event count statistics.

Third, we label the role overlapping problem where a candidate entity plays different roles in one or more events, and in Figure 6 we count the number of roles played by the candidate entities. We can see from the figure that although in most cases a candidate entity plays only one role, there are still a significant number of candidate entities that play two or three different roles at the same time.

**Figure 6.** Statistics on the number of roles played by candidate entities.

3.4. Dataset summary

Compared with existing partial event extraction datasets, our proposed dataset CDEEE has the following advantages. First, we solve the problem of missing datasets in the field of document-level

emergency event extraction in Chinese, which allows the relevant models to be trained and validated. Second, our dataset is manually annotated with more documents than most of the existing DEE datasets at a size of 5000 documents. Third, we have annotated the role overlapping problem, which is more realistic to the real environment and reflects the text complexity.

4. Proposed method

The model in this paper consists of three main parts: candidate entity extraction, event type detection and arguments identification as three sub-tasks. First, the text embedding and text representation are obtained by pre-trained language models RoBERTa and Transformer[29], and then the head and tail position information of candidate entities are obtained using two independent modules, and the head and tail position information of candidate entities are interacted with each other with the help of multiplicative attention mechanism to obtain the score matrix for candidate entity prediction. Then, the candidate entity representation is fused with the sentence representation and the document representation is obtained for event type detection using Transformer. Finally, we decode the event and role information using a multi-granularity decoder for the recognition and prediction of argument elements. The structure of the proposed model is shown in Figure 7.

DEE task can be described as extracting one or more structured events $Y = \{y_i\}_{i=1}^{N_k}$ from an input document $D = \{S_i\}_{i=1}^{N_s}$ consisting of N_s sentences, where N_k is the number of events contained in the document. Each event extracted from the document contains event types and their associated roles, and we denote the set of all event types by T and the set of all role types by $Role$. For the structured event information extracted from the document, we denote it by y_i^t , where $t \in T$ denotes the event type, and the n role information corresponding to each event type and the arguments used to populate the role information are denoted by $\{r_i^1, r_i^2, \dots, r_i^n\} \in Role$ and $\{\alpha_i^1, \alpha_i^2, \dots, \alpha_i^n\}$, respectively.

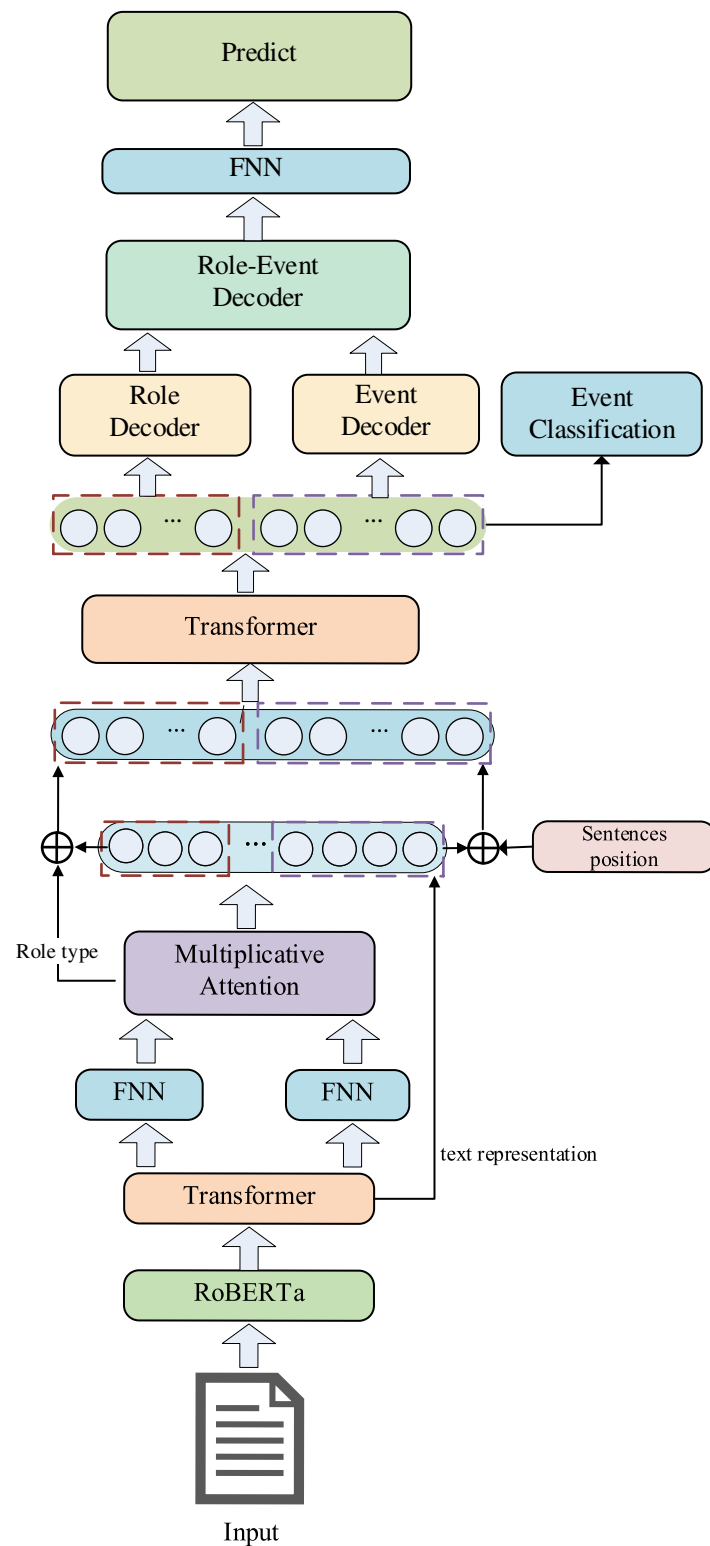


Figure 7. Overall Model Structure of RODEE.

4.1. Candidate entity extraction

Candidate entity extraction, as the first subtask of the DEE task, has a huge impact on the performance of the two subsequent subtasks of entity type detection and argument identification. However, in previous candidate entity extraction tasks, candidate entities are usually considered as flat entities and the candidate entity extraction task is accomplished using sequence annotation. Although this sequence labeling-based approach achieves better results in the flat entity extraction task, it ignores the role overlapping problem and cannot perform accurate extraction of candidate

entities with multiple roles. And to solve this problem, we firstly use two different matrices to represent the head position information and tail position information of candidate entities respectively in this stage, and then use multiplicative attention to make the two interact to mine the deep information, obtain the score matrix and complete the candidate entity extraction according to the score matrix.

Specifically, given a document $D = \{S_i\}_{i=1}^{N_s}$, for each sentence $S_i = [x_i^1, x_i^2, \dots, x_i^m]$, we first perform an embedding representation $S_i = [w_i^1, w_i^2, \dots, w_i^m]$ using the pre-trained language model RoBERTa, where m is the sentence length. Then, to obtain the textual representation, we encode the embedded representation of the text using the Transformer encoder, and eventually we can obtain the text representation H_i for each sentence S_i as follows:

$$[w_i^1, w_i^2, \dots, w_i^m] = \text{RoBERTa}([x_i^1, x_i^2, \dots, x_i^m]) \quad (1)$$

$$[h_i^1, h_i^2, \dots, h_i^m] = \text{Transformer}([w_i^1, w_i^2, \dots, w_i^m]) \quad (2)$$

where $w_i^j \in R^d$, $h_i^j \in R^d$, d is the hidden layer size.

Finally, in order to accurately extract the overlapping candidate entities of characters, we use two FNNs networks to generate two different matrices, H_s and H_e , respectively, to represent the head position information and tail position information of candidate entities, i.e., the contextual information of target characters. By using different matrices to represent the head position information and tail position information of the candidate entities and training them, the start position and end position of the candidate entities can be identified, respectively. Since the contexts of the start and end positions of the candidate entities are different, using different matrices to represent the head position information and tail position information of the candidate entities, respectively, greatly improves task accuracy compared with using the output of Transformer directly. On top of this, we do what is shown in Figure 8, i.e., we use multiplicative attention to let the start position information and end position information interact to generate a score matrix $C_i \in R^{head \times m \times m}$ for candidate entity prediction, where *head* is the number of predefined role types plus one (i.e., non-predefined role types). We obtain the score matrix in the following way:

$$H_s = \text{FFNNs}([h_i^1, h_i^2, \dots, h_i^m]) \quad (3)$$

$$H_e = \text{FFNNs}([h_i^1, h_i^2, \dots, h_i^m]) \quad (4)$$

$$C_i = h_{s, \text{begin}, r_u} h_{s, \text{end}, r_u}^T \quad (5)$$

where $H_s \in R^{m \times head \times o}$, $H_e \in R^{m \times head \times o}$, o are hyperparameters indicating the window size in obtaining the contextual embedding of the target character, and we take the value of 64 here to indicate that 64 characters before and after the target character are obtained as the contextual representation of the target character, i.e., the start position or end position representation information, $h_{s, \text{begin}, r_u}, h_{s, \text{end}, r_u}^T$ indicates the vector representation of the start position information and end position information of the candidate entity with the role type r_u entity span (begin, end), $r_u \in \text{Role}$.

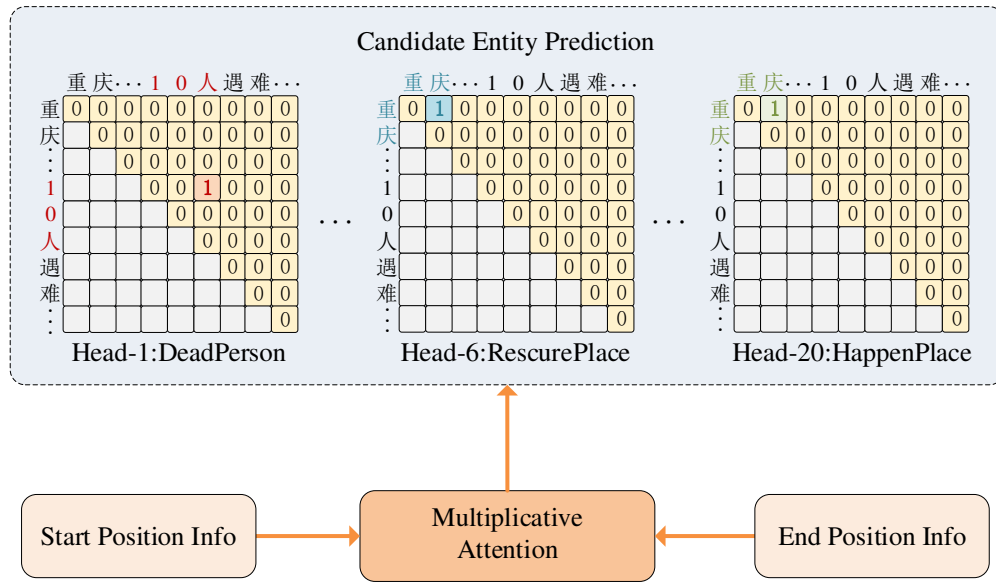


Figure 8. Candidate entity prediction.

After obtaining the score matrix C_i , we perform candidate entity prediction according to the following equation:

$$P^e = \text{Sigmoid}(W_0 C_i) \quad (6)$$

where $W_0 \in R^{\text{head} \times \text{head}}$, $P^e \in R^{\text{head} \times m \times m}$. Finally, after the transformation, we obtain the prediction results as shown in Figure 8 for the start position, end position, and role type of the candidate entities. For the candidate entities extracted from each sentence, we denote them by the triplet (e^s, e^e, e^r) , where e^s is the start position of the candidate entity, e^e is the end position of the candidate entity, and e^r is the role type of the candidate entity. For all the candidate entities extracted from the whole document, we denote them by the set $E = \{e_i\}_{i=1}^{N_e}$, and for each candidate entity, we denote them by the quadruplet $e_i = (e^{\text{sent}}, e^s, e^e, e^r)$, where e^{sent} denotes the sentence index of the candidate entity.

Regarding the loss function in this section, the cross-entropy loss function is used as follows:

$$L_1 = \sum_{i=1}^{N_S} \sum_{0 \leq j_s, j_t < m} -(entity_{j_s, j_t}^i) \cdot \log P_{j_s, j_t}^i \quad (7)$$

4.2. Event type detection

Before performing event type detection, our model needs to understand the document as a whole, i.e., obtain document-level contextual encoding information. To obtain a holistic representation of the document, we use the Transformer encoder to allow all sentence information to interact with the candidate entity information. Specifically, to obtain comprehensive document encoding information, we first MaxPooling the textual representation of candidate entities $H_i[e^s:e^e]$ and each sentence textual representation H_i so that the candidate entities textual representation has the same dimension as the sentence textual representation, and then facilitate the interaction between the two. Then, the sentence information and the candidate entity information are embedded separately in two aspects: on the one hand, the sentence position information is fused with the sentence text representation after the MaxPooling operation using the sentence position encoder. On the other hand, we use the entity type encoder to embed the role type e^r of the candidate entity into the text representation of the candidate entity after the same MaxPooling operation, in which we

embed multiple role types separately for the candidate entity with multiple role types and fuse multiple embeddings. Finally, the completed embedded sentence text representation $C^S \in R^{N_s \times d}$ and the candidate entity text representation $C^E \in R^{N_E \times d}$ are fed to the Transformer encoder, and the whole document representation is obtained by the interaction between them. Specifically, as in the following equation:

$$C^E = \text{Position} - \text{Encoder}(\text{MaxPooling}(H_i)) \quad (8)$$

$$C^S = \text{Role} - \text{Encoder}(\text{MaxPooling}(H_i[e^s; e^e])) \quad (9)$$

$$[H^S; H^E] = \text{Transformer}([C^S; C^E]) \quad (10)$$

where $H^S \in R^{N_s \times d}$ is the sentence representation of the document representation, $H^E \in R^{N_E \times d}$ is the candidate entity representation of the document representation, and N_E is the number of candidate entities extracted from the document.

After obtaining the overall document representation, we can use the sentence representation H^S of the document representation for event type detection. Specifically, we bifurcate each event type by performing the MaxPolling operation on H^S , i.e., for each event type, as follows:

$$P^t = \text{Softmax}(\text{FNN}(\text{MaxPolling}(H^S))) \quad (11)$$

where P_i^t denotes the probability that the i -th event is an event of type t .

Regarding the loss function in this section, the cross-entropy loss function is used as follows:

$$L_2 = \sum_{t=1}^{|T|} -(y^t) \cdot \log P^t \quad (12)$$

where $|T|$ indicates the number of defined event types.

4.3. Argument identification

In this stage, we need to match and fill in the arguments and event roles for the existing events. Following Yang et al.[10], we use a multi-granularity decoder to extract events in a parallel manner. This method consists of three parts: an event decoder, a role decoder, and an event-to-role decoder.

The event decoder is designed to support parallel extraction of all events and is used to model interactions between events. A learnable query matrix $Q^{event} \in R^{k \times d}$ is generated for event extraction, where k is a hyperparameter representing the number of events contained in the document. The event query matrix Q^{event} is then fed into a non-autoregressive decoder composed of multiple identical stacks of Transformer layers. In each layer, there is a multi-head self-attention mechanism to simulate interactions between events, and a multi-head cross-attention mechanism to integrate the document-aware representation H^S into the event query Q^{event} :

$$H^{event} = \text{Event} - \text{Decoder}(Q^{event}; H^S) \quad (13)$$

where $H^{event} \in R^{k \times d}$.

The role decoder is designed in a similar way to support parallel filling of all roles in the event and modeling interactions between roles. A learnable query matrix $Q^{role} \in R^{n \times d}$ is generated for event extraction, where n is the number of roles for the corresponding event. Then, the role query matrix Q^{role} is fed into a decoder with the same architecture as the event decoder. Specifically, the self-attention mechanism can model relationships between roles, and the cross-attention mechanism can integrate candidate entity representations H^E from the document representation.

$$H^{role} = \text{Role} - \text{Decoder}(Q^{role}; H^E) \quad (14)$$

where $H^{role} \in R^{n \times d}$.

In order to generate different events and their associated roles, we designed an event-to-role decoder to simulate the interaction between event information and role information.

$$H^{event2Role} = Event2Role - Decoder(H^{event}, H^{Role}) \quad (15)$$

where $H^{event2Role} \in R^{k \times n \times d}$.

Finally, after decoding with a multi-granularity decoder, we transform k event queries and n role queries into k predicted events and their corresponding n predicted roles. To filter out false events, we assess whether each predicted event is non-empty. Specifically, predicted events can be obtained through the following approach:

$$p^{event} = softmax(H^{event}W_e) \quad (16)$$

where $W_e \in R^{d \times 2}$ is learnable matrix.

Afterwards, for each predicted event with predefined roles, we decode the predicted arguments by filling the candidate indices or null values with an $N_E + 1$ class classifier.

$$p^{role} = softmax(tanh(H^{event2Role}W_1 + H^E W_2)v) \quad (17)$$

where $W_1 \in R^{d \times d}$, $W_2 \in R^{d \times d}$, $v \in R^d$ are learnable matrices, $p^{role} \in R^{k \times n \times (N_E + 1)}$.

So far, we have obtained k predicted events, $\hat{Y} = \hat{y}_{i=1}^k$, and the candidate entities for each role corresponding to each event, $\hat{y}_i^f = (P_i^1, P_i^2, \dots, P_i^n)$. This completes the event extraction and the identification, matching, and filling of the corresponding arguments.

Regarding the loss function for this section, we first use the assignment problem in operations research[30,31] to find the optimal assignment between the predicted events \hat{Y} and the ground truth events Y :

$$\hat{\sigma} = \underset{\sigma \in \Pi(k)}{argmax} \sum_i^k \mathcal{C}_{match}(\hat{Y}_{\sigma(i)}, Y_i) \quad (18)$$

where $\Pi(k)$ is the permutation space with a length of k , and $\mathcal{C}_{match}(\hat{Y}_{\sigma(i)}, Y_i)$ is the pairwise matching cost between the ground truth data Y_i and the predicted data $\hat{Y}_{\sigma(i)}$ with the index $\sigma(i)$. By considering all the predicted cases of the roles in the event, we define $\mathcal{C}_{match}(\hat{Y}_{\sigma(i)}, Y_i)$ as follows:

$$\mathcal{C}_{match}(\hat{Y}_{\sigma(i)}, Y_i) = -\mathbb{I}_{judge_i \neq \phi} \sum_{j=1}^n \mathbf{P}_{\sigma(i)}^j(r_i^j) \quad (19)$$

where $judge_i \neq \phi$ indicates that the event is not empty. The optimal assignment $\sigma(i)$ can be effectively calculated using the Hungarian algorithm[30]. Then, based on all optimal assignments, we define a loss function with negative log-likelihood:

$$L_3 = \sum_{i=1}^m -\mathbb{I}_{judge_i \neq \phi} \left[\sum_{j=1}^n -\log \mathbf{P}_{\hat{\sigma}(i)}^j(r_i^j) \right] \quad (20)$$

Finally, our overall loss function considers the candidate entity extraction loss L_1 , the event type detection loss L_2 , and the event argument recognition loss L_3 , which involves filling the entity-role pairs, as shown below:

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \quad (21)$$

where λ_1 , λ_2 , and λ_3 are hyperparameters.

5. Experiments

5.1. Experimental Setting

We use our labeled Chinese document-level unexpected event extraction dataset CDEEE as our experimental data. Our dataset contains a total of 5000 documents, including four event types: "InjureDead", "Rescure", "Accident", and "NaturalHazard".

Regarding the evaluation metrics, this paper adopts the evaluation criteria used in the Doc2EDAG model. Specifically, for all golden events in each chapter, the predicted events with the

same event type and the highest number of correct roles and thesis elements are found using a non-replayback approach, and the precision (P), recall (R), and F1 measure (F1 score) are calculated as the prediction results of the model. Since event types usually include multiple roles, the Micro-F1 value at the role level is calculated as the final metric. Information about the experimental setting and hyperparameters is described in detail in the Appendix A.

5.2. Comparison experiment and result analysis

Given that our dataset follows the concept of triggerless word annotation, we utilize the following model as a baseline model for comparison experiments as well as a quality validation model for the dataset:

- Doc2EDAG: An end-to-end model that converts DEE into a table-population task, directly populating event tables with entity-based paths for extensions.
- GreedyDec: This model is a baseline model in the Doc2EDAG model that uses a greedy strategy to populate against an event table.
- DE-PPN: This model uses multi-granularity decoders for parallel extraction of events to improve the speed of event extraction while effectively addressing the challenges of multiple events and argument scattering of document-level events.

Based on the experimental setup in Section 5.1, we use a manual approach to extract event information and analyze it against the results of the baseline model on the CDEEE dataset on the one hand; on the other hand, we train our proposed model RODEE and analyze the results of RODEE against the baseline model experiments under the same experimental conditions.

Tables 2 and 3 show the results obtained by each model on the CDEEE dataset with respect to each event type and the overall experimental results. We can observe that the scores achieved by humans on the CDEEE dataset are much higher than those of the existing DEE models. On the one hand, this indicates the high quality of our labeled dataset, and on the other hand, it also indicates that there is still more room for improvement in the DEE task.

Table 2. Experimental results of DEE models for various events in the CDFEE dataset.

Model	InjureDead			Rescure			Accident			NaturalHazard		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
GreedyDec	79.5	66.0	72.2	62.2	26.2	36.9	79.5	70.4	74.7	73.7	71.2	72.4
GreedyDec*	86.8	69.1	77.0	65.5	24.3	35.4	80.6	73.6	76.9	76.4	72.6	74.4
Doc2EDAG	85.8	65.3	74.2	75.5	47.0	58.0	82.8	69.4	75.5	78.0	74.6	76.2
Doc2EDAG*	85.8	72.0	78.3	78.5	50.8	61.7	78.1	74.7	76.4	75.5	76.3	75.9
DE-PPN	87.4	46.5	60.7	72.0	31.9	44.2	89.8	65.9	76.0	84.4	77.6	80.8
DE-PPN*	85.8	46.2	60.0	69.0	32.1	43.8	89.6	66.5	76.4	85.1	78.6	81.7
RODEE	95.8	54.2	69.2	79.4	44.5	57.0	96.4	74.5	84.1	94.7	89.6	92.1
Human	99.9	95.2	97.4	95.4	94.5	94.9	98.5	94.8	96.6	98.7	96.6	97.6

Table 3. Overall experimental results of DEE in CDFEE dataset.

Model	Precision	Recall	F1
GreedyDec	76.7	60.3	67.5
GreedyDec*	79.4	62.1	69.7
Doc2EDAG	81.4	64.5	71.9
Doc2EDAG*	78.9	69.3	73.8
DE-PPN	86.2	56.9	68.6
DE-PPN*	85.4	57.3	68.6
RODEE	93.2	66.6	77.7
Human	98.5	95.2	96.8

Considering that the existing DEE model methods use sequence annotation to complete the candidate entity prediction task in the candidate entity extraction stage and embed the role types of candidate entities to assist in the DEE task, our CDEEE dataset is annotated for the candidate entity role overlapping problem. Therefore, we modify the baseline model by removing the role type embedding module from the baseline model and naming the modified baseline model as Doc2EDAG*, GreedyDec*, and DE-PPN*.

As can be seen from Table 2 and 3, in the CDEEE dataset labeled with the role overlapping problem, the model approach of embedding a single role type to assist in the DEE task has an overall lower performance compared to the DEE model approach without role type embedding. Therefore, we believe that the embedding of incomplete role type information not only does not contribute to the overall performance of the DEE task but also may have a negative impact on the performance of the DEE task to some extent.

We can also observe from Tables 2 and 3 that the overall performance of our proposed RODEE model is better than the existing DEE task model, with an improvement of 7.8 percentage points in the accuracy P of our model compared to the best performing DE-PPN* model among the baseline models, and compared to the best performing Dco2EDAG* model among the baseline models in terms of F1 values, our model's F1 value improved by 3.9 points. In addition, we also observe that the recall R and F1 values for event "InjureDead" are lower than those of the Doc2EDAG* model, which we attribute to the fact that event "InjureDead" has only four role types and a low role overlapping rate, and thus the performance of our proposed model for the candidate entity role overlapping problem is slightly lower than that of the Doc2EDAG* model for this class of events.

To further analyze the performance of RODEE, we also conducted experiments on the candidate entity extraction subtask, and the results are shown in Figure 9. As can be seen in Figure 9, RODEE not only outperforms the other models on the DEE task, but also on the candidate entity extraction subtask. Compared with other models, RODEE improves at least 11 percentage points on the F1 of the candidate entity extraction subtask. We believe this is due to the fact that more deep information about the text is obtained to improve the performance of the model for candidate entity extraction. This demonstrates that the performance of candidate entity extraction, as the first subtask of the DEE task, has a significant impact on the overall DEE task, and the improvement of the candidate entity extraction subtask has a positive effect on the DEE task.

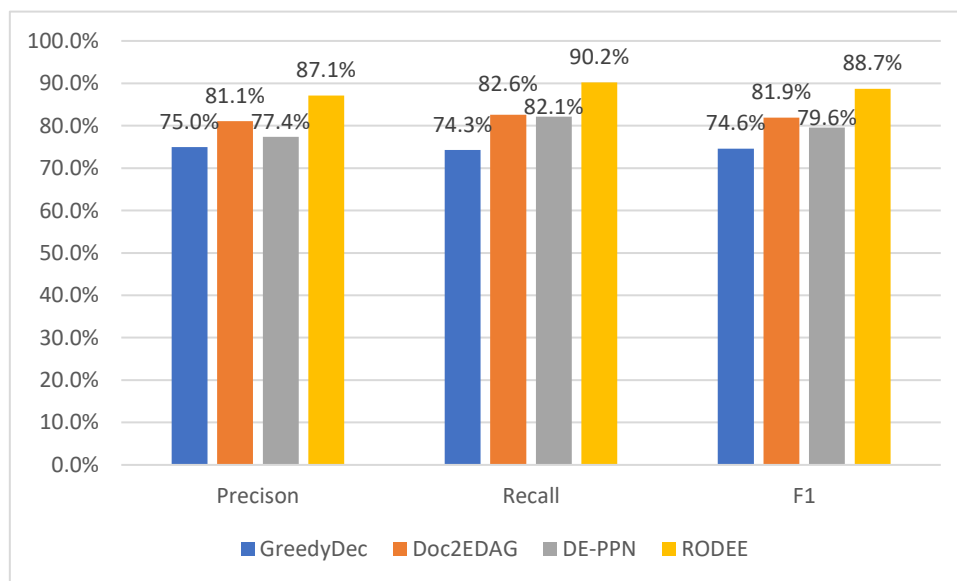


Figure 9. Comparison of candidate entity extraction results.

5.3. Ablation experiment

To verify the effectiveness of our work on model improvement, we conducted ablation experiments on some of the modules. First, we perform ablation experiments for the fusion of features

of role information for candidate entities. Since we have previously removed the role information features of the candidate entities in each baseline model and obtained better results than the source model, we also need to verify our role information features. Therefore, we also need to verify whether our fusion of role information features has a positive or negative impact on our overall task. Second, since we use the pre-trained language model RoBERTa, we need to verify whether the overall task performance improvement is entirely due to the pre-trained language model. It is well known that pre-trained language models have a large performance improvement for all natural language processing tasks, and we use a pre-trained language model that can capture more textual information due to the need to obtain more fine-grained information when solving the problem of overlapping role information of meta-entities and use the RoBERTa pre-trained language model as an important part of our model. For various reasons, we also performed ablation experiments on this part, but instead of removing the pretrained language model to support our need for finer-grained information, we added the pretrained language model to the baseline model, DE-PPN.

- -RoleType: remove the role type embedding module from the RODEE model.
- +BERT: add the RoBERTa pre-trained language model to the DE-PPN baseline model.

In Table 4, we present information about the model -RoleType after deleting the candidate entity role information features on our model and adding the same pre-trained language model +BERT as ours to the baseline model DE-PPN, and we can see that our model is still in an advantageous position compared to the above two models. Our model still has a 1.1-point F1 improvement compared to the model with the candidate role information features removed, thus demonstrating that incorporating the correct candidate role information is a boost to our DEE task and that incomplete or incorrect candidate role features pull down the overall performance of the document-level event extraction task, as shown in the experimental results in Section 5.1. And by comparing our model with the baseline model with the addition of the pre-trained language model, we can find that our model still has a performance improvement of 3.2 F1 points. We can conclude that although the powerful performance of the pre-trained language model is leveraged in our model, our proposed model improvements for the candidate entity role overlapping problem still contribute significantly to the performance of the DEE task.

Table 4. Results of ablation experiment.

Model	P	R	F1
-RoleType	92.6	65.2	76.6
+BERT	90.5	63.4	74.5
RODEE	93.2	66.6	77.7

5. Conclusions

In this paper, we propose and construct a Chinese document-level burst event extraction dataset for the problem of data scarcity in Chinese document-level emergency event extraction tasks and validate it using an existing DEE model. The feasibility and research value of our constructed dataset are demonstrated. In the process of constructing the dataset, we discovered the role overlapping problem of candidate entities, which is common in the real world and often overlooked in this field. To address this problem, we propose a DEE model to solve the problem and conduct experiments on our dataset. Our approach significantly improves the overall performance of the DEE task compared to the benchmark model. For future work in this area, we believe that the introduction of external knowledge bases such as knowledge graphs and factual knowledge graphs will greatly improve the model's ability to cope with the role overlapping problem and thus improve the overall performance of the DEE task.

Author Contributions: Conceptualization, K.C. and W.Y.; methodology, K.C.; validation, K.C., F.W. and J.S.; data curation, K.C., F.W. and J.S.; writing—original draft preparation, K.C.; writing—review and editing, K.C. W.Y. and F.W.; supervision, K.C. and W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China grant number 202204120017, the Autonomous Region Science and Technology Program grant number 2022B01008-2, and the Autonomous Region Science and Technology Program grant number 2020A02001-1.

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

Appendix A

Regarding the Chinese document-level dataset for extracting unexpected events, the related event types and their corresponding role attributes are described as follows:

Table 5. Event type information description.

ID	EventType	Description
0	InjureDead	InjureDead, used to describe the situation of personnel casualties caused by unexpected events.
1	Rescure	Rescure, used to describe the rescue and aftermath situation of the relevant organization after an accident or disaster occurs.
2	Accident	Accident, used to describe the relevant information of sudden accidents caused by non-natural factors.
3	NaturalHazard	NaturalHazard, used to describe relevant information on unexpected events caused by natural factors.

Table 6. Event role type information description.

ID	RoleType	Description
0	InjurePerson	Used to describe the number of injured people in the InjureDead event.
1	DeadPerson	Used to describe the number of deaths in the InjureDead event.
2	InjurerPerson	Used to describe the relevant personnel who caused injuries to people in InjureDead events, such as the culprit in criminal cases.
3	InjurerArticle	Used to describe the related items that cause personal injuries in InjureDead events, such as the weapon used by the perpetrator in criminal cases.
4	RescureOrg1	To describe the organizations involved in rescuing operations in Rescue events, based on the analysis of the text information related to unexpected events, typically no more than three rescue organizations are involved.
5	RescureOrg2	
6	RescureOrg3	Therefore, we have established three role types: RescueOrg1, RescueOrg2, and RescueOrg3.
7	StartTime	This is used to describe the time when rescue organizations start rescue operations in a Rescue event, in order to illustrate the speed of the rescue organization's response time.
8	EndTime	To describe the Rescue event, the time taken by the rescue organization to complete the rescue operation can be compared with the start time to evaluate the efficiency of the rescue operation.
9	RescureTokenage	Used to describe the duration of rescue missions performed by rescue organizations in Rescue events, it can directly reflect the efficiency of rescue operations. However, since RescueTokenage, despite having similar functionality when combined with StartTime and EndTime, is not always paired with them.
10	RescurePlace	Used to describe the location where rescue organizations conduct rescue operations during Rescue events.
11	RescuredTarget	Used to describe the situation of individuals rescued by rescue organizations during a Rescue event.
12	HappenTime	Used to describe the time of occurrence of emergency events during Accident and Natural Hazard incidents.

13	HappenPlace	Used to describe the location of the incident that occurred during Accident and Natural Hazard events.
14	HazardType	Used to describe the type of emergency incidents in Accident and Natural Hazard events. In Accident events, these can include traffic accidents, roof collapses, fires, etc. In Natural Hazard events, these can include earthquakes, mudslides, landslides, etc.
15	AffectOject	Used to describe the relevant information about individuals affected by emergency events during Accidents and Natural Hazard events.
16	CauseObject1	Used to describe the main information caused by a emergency accident during an Accident event, for example, in a traffic accident, Car A and Car B that collided with each other. Similarly, through the analysis of a large amount of textual information, we have established three role types: CauseObject1, CauseObject2, and CauseObject3.
17	CauseObject2	
18	CauseObject3	

Regarding the experimental environment, we used NVIDIA GeForce RTX 3090 graphics card for model training and testing. The hyperparameter settings in the model are shown in Table 6:

Table 7. Super parameter setting.

Parameter name	Parameter value
Batch size	4
Epoch	100
number of generated events	4
Embedding size	768
Hidden size 768	768
Learning rate for Transformer	1e-5
Learning rate for RoBERTa	2e-5
Learning rate for Decoder	2e-5
Dropout	0.1

References

1. Hong, Y.; Kim, J.-S.; Xiong, L. Media Exposure and Individuals’ Emergency Preparedness Behaviors for Coping with Natural and Human-Made Disasters. *Journal of Environmental Psychology* **2019**, *63*, 82–91.

2. Ji, H.; Grishman, R. Refining Event Extraction through Cross-Document Inference. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus Ohio, USA, 15-20 June 2008.

3. Liao, S.; Grishman, R. Using Document Level Cross-Event Inference to Improve Event Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Uppsala, Sweden , July 2010.

4. Hong, Y.; Zhang, J.; Ma, B.; Yao, J.; Zhou, G.; Zhu, Q. Using Cross-Entity Inference to Improve Event Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011.

5. Doddington, G.; Mitchell, A.; Przybocki, M.; Ramshaw, L.; Strassel, S.; Weischedel, R. The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. *Proc Lrec* **2004**.

6. Grishman, R.; Sundheim, B. Message Understanding Conference- 6: A Brief History. In Proceedings of the 16th conference on Computational linguistics, 1996.

7. Li, S.; Ji, H.; Han, J. Document-Level Event Argument Extraction by Conditional Generation. **2021**.

8. Yang, H.; Chen, Y.; Liu, K.; Xiao, Y.; Zhao, J. DCFEE: A Document-Level Chinese Financial Event Extraction System Based on Automatically Labeled Training Data. In Proceedings of the Meeting of the Association for Computational Linguistics, 2018.

9. Zheng, S.; Cao, W.; Xu, W.; Bian, J. Doc2EDAG: An End-to-End Document-Level Framework for Chinese Financial Event Extraction. **2019**.

10. Yang, H.; Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; Wang, T. Document-Level Event Extraction via Parallel Prediction Networks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021.
11. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. **2019**.
12. Ebner, S.; Xia, P.; Culkin, R.; Rawlins, K.; Van Durme, B. Multi-Sentence Argument Linking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5-10 July 2020.
13. Du, X.; Cardie, C. Document-Level Event Role Filler Extraction Using Multi-Granularity Contextualized Encoding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5-10 July 2020.
14. Du, X.; Rush, A.; Cardie, C. GRIT: Generative Role-Filler Transformers for Document-Level Event Entity Extraction. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 19 - 23 April 2021.
15. Huang, K.-H.; Peng, N. Document-Level Event Extraction with Efficient End-to-End Learning of Cross-Event Dependencies. In Proceedings of the Third Workshop on Narrative Understanding, 2021.
16. Huang, Y.; Jia, W. Exploring Sentence Community for Document-Level Event Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 16-20 November 2021.
17. Xu, R.; Liu, T.; Li, L.; Chang, B. Document-Level Event Extraction via Heterogeneous Graph-Based Interaction Model with a Tracker. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 1-6 August 2021.
18. Nguyen, T.H.; Cho, K.; Grishman, R. Joint Event Extraction via Recurrent Neural Networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, 12-17 June 2016.
19. Li, Q.; Ji, H.; Huang, L. Joint Event Extraction via Structured Prediction with Global Features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4-9 August 2013.
20. Nguyen, T.M.; Nguyen, T.H. One for All: Neural Joint Modeling of Entities and Events. In Proceedings of The Thirty-First Innovative Applications of Artificial Intelligence Conference, Honolulu Hawaii, USA, 27 January - 1 February 2019.
21. Sha, L.; Qian, F.; Chang, B.; Sui, Z. Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction. In Proceedings of the AAAI conference on artificial intelligence, 2018.
22. Liu, X.; Luo, Z.; Huang, H. Jointly Multiple Events Extraction via Attention-Based Graph Information Aggregation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October - 4 November 2018.
23. Wadden, D.; Wennberg, U.; Luan, Y.; Hajishirzi, H. Entity, Relation, and Event Extraction with Contextualized Span Representations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3-7 November 2019.
24. Li, F.; Peng, W.; Chen, Y.; Wang, Q.; Pan, L.; Lyu, Y.; Zhu, Y. Event Extraction as Multi-Turn Question Answering. In Proceedings of the Findings of the Association for Computational Linguistics, 16-20 November 2020.
25. Du, X.; Cardie, C. Event Extraction by Answering (Almost) Natural Questions. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 16-20 November 2020.
26. Chen, Y.; Chen, T.; Ebner, S.; White, A.S.; Van Durme, B. Reading the Manual: Event Extraction as Definition Comprehension. In Proceedings of the Fourth Workshop on Structured Prediction for NLP, 20 November 2020.
27. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring Pre-Trained Language Models for Event Extraction and Generation. In Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 28 July- 2 August 2019.

28. Xu, N.; Xie, H.; Zhao, D. A Novel Joint Framework for Multiple Chinese Events Extraction. In Proceedings of the Chinese Computational Linguistics - 19th China National Conference, Hainan, China, 30 October - 1 November 2020.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need, In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, USA, 4-9 December 2017.
30. Kuhn, H.W. The Hungarian Method for the Assignment Problem. In Proceedings of the Naval Res Logist Quart, 1995.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.