

Article

Joint Overlapping Event Extraction Model via Role Pre-Judgment with Trigger and Context Embeddings

Qian Chen ¹, Kehan Yang ¹, Xin Guo ^{1,*}, Suge Wang ^{1,2} , Jian Liao ¹ and Jianxing Zheng ^{1,2} 

¹ School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; chenqian@sxu.edu.cn (Q.C.); 202222407057@email.sxu.edu.cn (K.Y.); wsg@sxu.edu.cn (S.W.); liaoj@sxu.edu.cn (J.L.); jxzheng@sxu.edu.cn (J.Z.)

² Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China

* Correspondence: guoxinjsj@sxu.edu.cn

Abstract: The objective of event extraction is to recognize event triggers and event categories within unstructured text and produce structured event arguments. However, there is a common phenomenon of triggers and arguments of different event types in a sentence that may be the same word elements, which poses new challenges to this task. In this article, a joint learning framework for overlapping event extraction (ROPEE) is proposed. In this framework, a role pre-judgment module is devised prior to argument extraction. It conducts role pre-judgment by leveraging the correlation between event types and roles, as well as trigger embeddings. Experiments on the FewFC show that the proposed model outperforms other baseline models in terms of Trigger Classification, Argument Identification, and Argument Classification by 0.4%, 0.9%, and 0.6%. In scenarios of trigger overlap and argument overlap, the proposed model outperforms the baseline models in terms of Argument Identification and Argument Classification by 0.9%, 1.2%, 0.7%, and 0.6%, respectively, indicating the effectiveness of ROPEE in solving overlapping events.

Keywords: overlapping event extraction; trigger overlap; argument overlap; joint learning; role pre-judgment



Citation: Chen, Q.; Yang, K.; Guo, X.; Wang, S.; Liao, J.; Zheng, J. Joint Overlapping Event Extraction Model via Role Pre-Judgment with Trigger and Context Embeddings. *Electronics* **2023**, *12*, 4688. <https://doi.org/10.3390/electronics12224688>

Academic Editor: Arkaitz Zubiaga

Received: 29 October 2023

Revised: 16 November 2023

Accepted: 16 November 2023

Published: 18 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Event extraction (EE) is a challenging task in natural language understanding that plays a crucial role [1–4]. EE is dedicated to extracting event information occurring in the real world from text, classifying them into predefined event types, identifying the trigger and event participants, etc. [5]. Event extraction can be widely used in information retrieval and summarization, knowledge graph construction [6], intelligence analysis [7], and other fields. EE usually includes four subtasks, i.e., trigger identification (TI), Trigger Classification (TC), Argument Identification (AI), and Argument Classification (AC) [8–10].

The objective of event extraction is to identify events and arguments from the text. The event extraction task is formally defined as follows:

Given an input sentence $x = \{x_1, x_2, \dots, x_N\}$ consisting of N words, an event-type collection (\mathcal{C}), and an argument role collection (\mathcal{R}). Let \mathcal{E}_x represent the gold set. \mathcal{E}_x includes all event types ($\mathcal{C}_x \subseteq \mathcal{C}$) in the sentence, all triggers ($\mathcal{T}_{x,c}$) in each event type ($c \in \mathcal{C}_x$), and different argument roles under each event type. $a_r \in \mathcal{A}_x$ is the argument corresponding to $r \in \mathcal{R}$.

However, a sentence may contain multiple events, and the arguments and triggers of these events have complex overlapping phenomena. We summarize this into three situations: (1) Different event types may be triggered by the same trigger. The trigger “减持” (reduced its holdings) marked in red in Figure 1 triggers both the shareholding reduction event and the share equity transfer event. (2) The same argument plays different roles in different event types. In Figure 1, “大族激光” (Han’s Laser) plays the

role of *obj* in the “股份股权转让” (share equity transfer) event, and also plays the *target-company* role in the share equity transfer, which is triggered by “减持” (reduced its holdings). (3) The same argument plays the same role in different events of the same event type. In Figure 1, the “股份股权转让” (share equity transfer) event appears twice in the sentence, both occurring in “10月” (October). In FewFC (a Chinese financial event extraction dataset) [11], about 13.5% of the sentences have trigger overlap problems, and 21.7% of the sentences have argument overlap problems. The same event type appears repeatedly in 8.3% of the sentences.

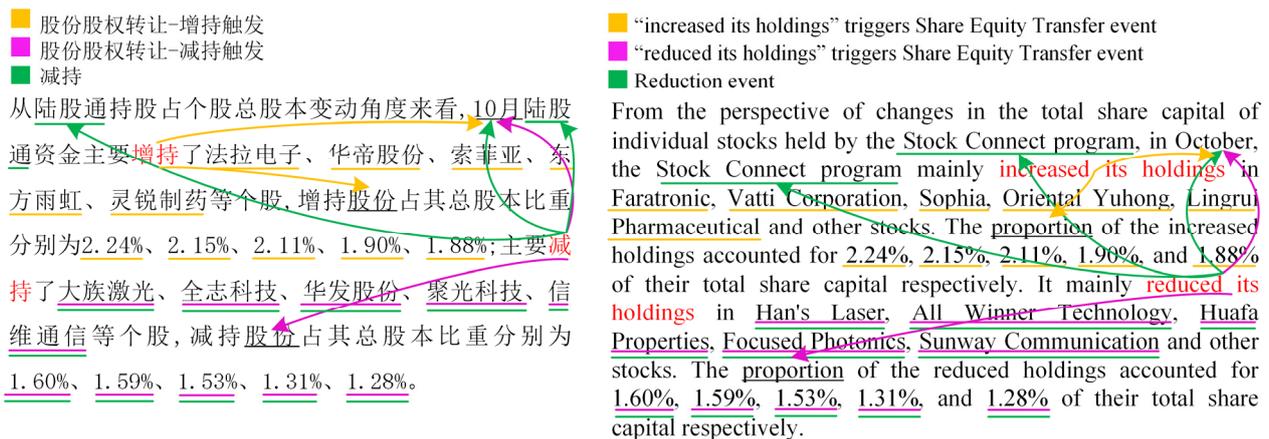


Figure 1. Examples of three overlapping phenomena.

Most previous studies partially addressed overlapping issues and did not cover all the above situations. In 2019, Yang et al. [12] utilized a staged pipeline approach for event trigger and argument extraction. Nevertheless, this method overlooked the challenge of trigger overlap. In 2020, Xu et al. [13] used a joint extraction framework to solve the role overlapping problem. Xu defined event relationship triples to represent the relationship between triggers, arguments, and roles, thereby converting the argument classification problem into a relationship extraction problem. In 2020, Huang et al. [14] proposed using a hierarchical knowledge structure graph containing conceptual and semantic reasoning paths to represent knowledge. They employed GEANet to encode intricate knowledge, addressing the issue of trigger extraction in nested structures within the biomedical domain [14]. In 2022, Zhang et al. [15] designed a two-stage pipeline model in which the trigger is identified using a sequence annotation approach, and overlapping arguments are identified through multiple sets of role binary classification networks. In 2023, Yang et al. [16] used a multi-task learning model to extract entity relationships and events, in which a multi-label classification method is used to settle the overlapping role problem shared by these two tasks.

The contributions of this paper are summarized as follows:

- (1) A role pre-judgment module is proposed to predict roles based on the correspondence between event types and roles, text embeddings, and trigger embeddings, which can significantly improve the recall rate of each subtask and provide a basis for extracting overlapping arguments.
- (2) ROPEE adopts a joint learning framework, and the designed loss function includes the losses of four modules, event-type detection, trigger extraction, role pre-judgment, and argument extraction, so as to effectively learn the interactive relationship between modules during training. Thus, error propagation issues can be alleviated in the prediction stage.
- (3) ROPEE outperforms the baseline model by 0.4%, 0.9%, and 0.6% in terms of F1 over TC, AI, and AC on the FewFC dataset. For sentences with overlapping triggers, ROPEE outperforms the baseline model by 0.9% and 1.2% in terms of F1 over AI and AC, respectively. In the case of overlapping arguments, ROPEE demonstrates

superior performance compared to the baseline model, with improvements of 0.7% and 0.6%. This highlights the effectiveness of our suggested approach in managing overlapping occurrences of event phenomena.

The remainder of this paper is organized as follows: In Section 2, related studies are given. In Section 3, the details of the ROPEE model are introduced. Comparative experiments are performed and experimental results are analyzed in Section 4. Section 5 concludes this work.

2. Related Studies

Event extraction is one of the most challenging tasks in information extraction research [17]. Existing paradigms related to event extraction include pipeline methods and joint learning methods [18].

The pipeline-based method handles these four subtasks of EE separately. Each subtask has its own objective function and loss. In 2015, Chen et al. [8] developed a dynamic multi-pooling convolutional neural network (DMCNN). This network utilizes a dynamic multi-pooling layer based on event triggers and arguments to retain essential information, combining both the sentence-level and lexical-level details from the raw text without the need for extensive preprocessing. Most deep learning supervised methods for event extraction require lots of labeled data for training. Annotating large amounts of data is very laborious and hard to get. To gain more insights from limited training data, Yang et al. [12] combined the extraction model and event generation method in 2019 and improved the performance of the argument extractor through a weighted loss function based on various role importance. The above two methods cannot explicitly model the semantics between events and roles, nor can they capture the interaction between them. In 2020, Li et al. [19] devised a multi-stage QA framework to represent event extraction as reading comprehension issues and captured the dynamic connection between each subtask by integrating previous answers into questions. The generative event extraction model proposed by Paolini et al. [20] in 2021 solves the encoding problem of label semantics and other weak supervision signals in a pipeline manner and can improve the performance in few-sample scenarios. Since the loss function in the pipeline-based method is calculated after the argument extraction, error propagation problems may occur.

Joint learning methods integrate the loss in both the trigger extraction stage and argument extraction stage into the final loss function, treating triggers and arguments equally, and the two can mutually promote each other's extraction effects [18]. In 2021, Sheng et al. [9] first covered all event overlap issues through a unified framework with a cascading decoder to perform TC, TI, and argument extraction in sequence, and F1 reached 71.4% on the FewFC dataset. In order to further extract inter-word relationships in overlapping sentences in parallel, Cao et al. [10] proposed a single-stage framework based on inter-word relationships by jointly extracting the intra-word and cross-word pair relationships of triggers and arguments. The above two methods focus on the event extraction task itself and do not introduce additional information or other tasks of joint information extraction. In 2022, Hsu et al. [21] converted event extraction into a conditional generation problem, and extracted triggers and arguments end-to-end through additional prompts and weak supervision information. In 2022, Van Nguyen et al. [22] used an edge weighting mechanism to learn the dependency graph between task instances and jointly complete the information extraction task. In addition to introducing additional prompt information in document-level event extraction, remote dependencies can also be used to improve extraction performance. In 2023, Liu et al. [23] proposed a chain reasoning paradigm for document-level event argument extraction, which represented argument queries by constructing first-order logic rules and T-Norm fuzzy logic, which is used for end-to-end learning. We propose a joint overlapping event extraction model ROPEE for the event overlapping phenomenon. It uses the correspondence between event types and roles and trigger embeddings to predict roles, which not only effectively alleviates error propagation, but also further improves the accuracy of event extraction.

3. ROPEE Model

The overall framework of ROPEE is illustrated in Figure 2. ROPEE includes four modules: event detection, trigger identification, role pre-judgment, and argument extraction. Specifically, type detection predicts potential event types and extracts overlapping triggers by calculating the similarity between sentence representations and event-type embeddings. The role pre-judgment module comprehensively considers text embeddings and trigger embeddings, pre-judgment roles based on the correspondence table between event types and roles, which can assist in the extraction of overlapping arguments. Trigger extraction and argument extraction are based on text representation that incorporates specific event types of information and specific role information, and binary classifiers are adopted to predict the starting and ending positions of triggers or arguments. To minimize error propagation, all modules are jointly learned during training.

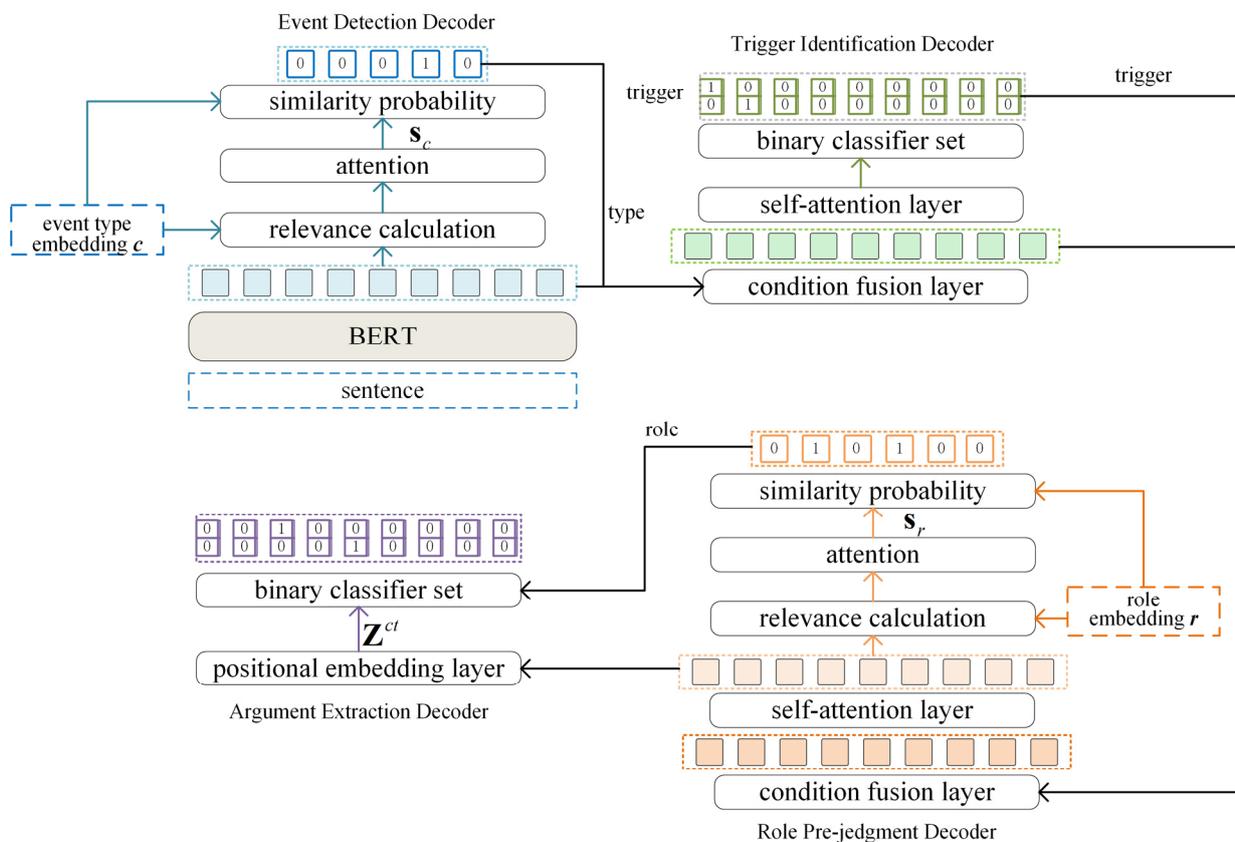


Figure 2. The overall framework of ROPEE.

3.1. Encoder

BERT [24] is utilized as an encoder. Sentence $x = \{x_1, x_2, \dots, x_N\}$ treats each Chinese character (x_i) as a token and is fed into the *bert-base-Chinese* module. The embedding of the sentence is obtained by $H = \text{BERT}(x_1, x_2, \dots, x_N) = \{h_1, h_2, \dots, h_N\} \in \mathbb{R}^{N \times d}$, where d is the dimension of the embeddings.

3.2. Event Detection Decoder

The event detection decoder is shown in the upper left corner of Figure 2, which is used to predict potential event types in sentences by calculating the correlation between sentence representations that imply type features and event-type embeddings. Specifically, event-type embeddings can be denoted by a randomly initialized matrix ($C \in \mathbb{R}^{|C| \times d}$). We apply *rel* to calculate the relevance between each token embedding (h_i) and the potential event type ($c \in C$), see Equation (1), and then a sentence representation (s_c) adaptive to the

event type is obtained, see Equation (2). Thus, the similarity probability between s_c and c is generated using a normalization σ operation for each event, see Equation (3).

$$rel(c, h_i) = v^T \times tanh(W_{rel}^1[c; h_i; |c - h_i|; c \odot h_i]) \tag{1}$$

$$s_c = \sum_{i=1}^N \frac{exp(rel(c, h_i))}{\sum_{j=1}^N exp(rel(c, h_j))} h_i \tag{2}$$

$$\hat{c} = \sigma(rel(c, s_c)) \tag{3}$$

where $W_{rel}^1 \in \mathbb{R}^{4d \times 4d}$ and $v \in \mathbb{R}^{4d \times 1}$ are parameters of relevance calculation. $|\cdot|$ denotes the element-wise subtraction operation. \odot denotes the element-wise multiplication. $[\cdot; \cdot]$ represents the concatenating operation. σ represents the sigmoid function. Types satisfying $\hat{c} > \zeta_1$ are selected as potential event types, where ζ_1 is a threshold hyperparameter between 0 and 1. All potential event types hidden in sentence x constitute the set of event types \mathcal{C}_x . The decoder can learn the parameters $\theta_{td} = \{W_{rel}^1, v, C\}$.

3.3. Trigger Identification Decoder

A large number of experiments demonstrate that trigger information can enhance the ability of argument extraction. The trigger identification decoder is used to identify triggers according to a specific event type ($c \in \mathcal{C}_x$). The decoder includes a conditional layer normalization (CLN) [25], a self-attention layer [26], and a binary trigger tagging classifier pair.

CLN fuses the two features and filters out unnecessary information. Here, the event-type information is encoded into the token representation, and the event-typed token representation (g_i^c) is obtained:

$$g_i^c = CLN(c, h_i) = \gamma_c \odot \left(\frac{h_i - \mu}{\sigma} \right) + \beta_c \tag{4}$$

where type embedding c is used as the condition for $\gamma_c = W_\gamma c + b_\gamma$ and $\beta_c = W_\beta c + b_\beta$ in CLN. $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ are regarded as the average and deviation of h_i :

$$\mu = \frac{1}{d} \sum_{k=1}^d h_{ik}, \quad \sigma = \sqrt{\frac{1}{d} \sum_{k=1}^d (h_{ik} - \mu)^2} \tag{5}$$

where h_{ik} represents the k -th dimension of h_i .

In order to fully consider the contextual connection in the sentence, a self-attention layer [26] is used on the event-typed token representation:

$$Z^c = SelfAttention(G^c) \tag{6}$$

where $G^c = \{g_1^c, g_2^c, \dots, g_N^c\}$, $G^c \in \mathbb{R}^{N \times d}$.

For each token, the binary classifier pair can mark the beginning and end position of a trigger span:

$$\hat{t}_i^{sc} = p(t_s | x_i, c) = \sigma(w_{t_s}^T z_i^c + b_{t_s}), \quad \hat{t}_i^{ec} = p(t_e | x_i, c) = \sigma(w_{t_e}^T z_i^c + b_{t_e}) \tag{7}$$

where z_i^c stands for the i -th token embedding in Z^c . We select the token satisfying $\hat{t}_i^{sc} > \zeta_2$ as the start position, and the one satisfying $\hat{t}_i^{ec} > \zeta_3$ as the end. ζ_2 and ζ_3 are threshold hyperparameters. To acquire trigger t , each starting position is enumerated and the nearest subsequent ending position is searched in the sentence. A token span from the start position to the end constitutes a complete trigger. The corresponding triggers are extracted at different stages according to the potential event type. Thus, the trigger overlapping prob-

lem can be solved naturally. The set $\mathcal{T}_{c,x}$ contains all predicted triggers (t) under event type (c) in sentence (x). θ_{te} is used to denote all parameters in the trigger identification decoder module.

3.4. Role Pre-Judgment Decoder

Since not all roles appear in a sentence under a specific event type, we designed a role pre-judgment decoder. Based on the predicted event type, it predicts the roles appearing in the sentence based on the corresponding list of event types and roles, providing a basis for extracting overlapping arguments. The decoder consists of three parts: the conditional fusion layer, the self-attention layer, and a role similarity detection function.

In order to obtain richer semantic information, we use CLN to fully integrate trigger embeddings and token representation with event-type knowledge to obtain a new token representation (\mathbf{g}_i^{ct}), see Equation (8). Here, trigger embedding t is calculated by the average pooling of token embeddings in the trigger span.

$$\mathbf{g}_i^{ct} = \text{CLN}(\mathbf{g}_i^c, t) \quad (8)$$

The self-attention layer then reinforces the contextual relationships and the sentence representation \mathbf{Z}^{ct} is obtained:

$$\mathbf{Z}^{ct} = \text{SelfAttention}(\mathbf{G}^{ct}) \quad (9)$$

where $\mathbf{G}^{ct} = \{\mathbf{g}_1^{ct}, \mathbf{g}_2^{ct}, \dots, \mathbf{g}_N^{ct}\}$, $\mathbf{G}^{ct} \in \mathbb{R}^{N \times d}$.

The role similarity detection function predicts potential event-type-specific roles in sentences by calculating the correlation between role embeddings and sentence representations fused with role feature information. Specifically, a randomly initialized matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}| \times d}$ is used as role embeddings. We apply rel to calculate the relevance between each token embedding (\mathbf{z}_i^{ct}) and the potential role ($\mathbf{r} \in \mathbf{R}$), see Equation (10), and then a sentence representation (\mathbf{s}_r) adaptive to the role is obtained, see Equation (11). Thus, the similarity probability between \mathbf{s}_r and \mathbf{r} is generated, see Equation (12), based on which normalization operation is performed to obtain the predicted probabilities of all roles under a specific event type.

$$rel(\mathbf{r}, \mathbf{z}_i^{ct}) = v^T \times \tanh\left(\mathbf{W}_{rel}^2[\mathbf{r}; \mathbf{z}_i^{ct}; |\mathbf{r} - \mathbf{z}_i^{ct}|; \mathbf{r} \odot \mathbf{z}_i^{ct}]\right) \quad (10)$$

$$\mathbf{s}_r = \sum_{i=1}^N \frac{\exp(rel(\mathbf{r}, \mathbf{z}_i^{ct}))}{\sum_{j=1}^N \exp(rel(\mathbf{r}, \mathbf{z}_j^{ct}))} \mathbf{z}_i^{ct} \quad (11)$$

$$\hat{p}^{ct} = p(r|x_i, c, t) = \sigma(rel(\mathbf{r}, \mathbf{s}_r)) \quad (12)$$

Select the role that satisfies $\hat{p}^{ct} > \zeta_4$ as the potential role type, and ζ_4 is the threshold. The role type set $\mathcal{R}_{t,c,x}$ contains all potential roles whose trigger is t under event type c in sentence x . θ_{re} is used to denote all parameters in the role pre-judgment decoder module.

3.5. Argument Extraction Decoder

An argument extraction decoder is composed of a positional embedding layer (PEL) and role-aware binary classifier pairs for argument tagging.

The relative position of a token to the trigger in the text is beneficial for argument extraction [9,27]. Here, relative position embeddings [8] imply the relative distance information between the current token and the trigger boundary token. Relative positional embeddings are incorporated into the sentence representation (\mathbf{Z}^{ct}) using a concatenation operation:

$$\mathbf{Z}^{ct} = [\mathbf{Z}^{ct}; \mathbf{P}] \quad (13)$$

where $\mathbf{P} \in \mathbb{R}^{N \times d_p}$ is the relative position embeddings, and d_p is the dimension of position embeddings.

For each token, a binary classifier sequence is employed to mark the boundary position of an argument under $r \in \mathcal{R}_{t,c,x}$:

$$\begin{aligned} \hat{a}_i^{sctr} &= p(a_r^s | x_i, c, t, r) = \hat{r}^{ct} \sigma(w_{r_s}^T z_i^{ct} + b_{r_s}) \\ \hat{a}_i^{ectr} &= p(a_r^e | x_i, c, t, r) = \hat{r}^{ct} \sigma(w_{r_e}^T z_i^{ct} + b_{r_e}) \end{aligned} \tag{14}$$

where z_i^{ct} represents the i -th token in \mathbf{Z}^{ct} . For each role (r), select the token that satisfies $\hat{a}_i^{sctr} > \zeta_5$ as the starting position and the one that satisfies $\hat{a}_i^{ectr} > \zeta_6$ as the end. $\zeta_5, \zeta_6 \in [0, 1]$ are thresholds. In order to extract the boundary of argument (a_r) with role (r), all starting positions are enumerated and the nearest subsequent ending position is searched in the sentence. Tokens between the starting and the ending position constitute a complete argument. In this way, only arguments of a specific role (r) under a specific trigger (t) and specific event type (c) in a sentence are extracted at each prediction stage. Thereby, the argument overlapping problem can be solved naturally. All candidate arguments (a_r) form a set $\mathcal{A}_{r,t,c,x}$, and θ_{ae} denotes the set of all parameters of the PEL and argument classifier.

3.6. Model Training

The loss of four modules is integrated during the training process, so the total loss function is designed as follows:

$$\text{Loss}_{\text{all}} = - \sum_{x \in \mathcal{D}} [\sum_{c \in \mathcal{C}_x} \log p_{\theta_1}(c|x) + \sum_{t \in \mathcal{T}_{x,c}} \log p_{\theta_2}(t|x,c) + \sum_{r \in \mathcal{R}_{x,c,t}} \log p_{\theta_3}(r|x,c,t) + \sum_{a_r \in \mathcal{A}_{x,c,t,r}} \log p_{\theta_4}(a_r|x,c,t,r)] \tag{15}$$

where $\Theta \triangleq \{\theta_1, \theta_2, \theta_3, \theta_4\}$. The first two subtasks, $p_{\theta_1}(c|x)$ and $p_{\theta_2}(t|x,c)$, are adopted from [9]. We decomposed the argument extraction loss to $p_{\theta_3}(r|x,c,t)$ and $p_{\theta_4}(a_r|x,c,t,r)$ and formulated it as:

$$\begin{aligned} p_{\theta_1}(c|x) &= (\hat{c})^{\bar{c}} (1 - \hat{c})^{(1-\bar{c})} \\ p_{\theta_2}(t|x,c) &= \prod_{z \in \{s,e\}} \prod_{i=1}^N (\hat{t}_i^{zc})^{\bar{t}_i^{zc}} (1 - \hat{t}_i^{zc})^{(1-\bar{t}_i^{zc})} \\ p_{\theta_3}(r|x,c,t) &= (\hat{r}^{ct})^{\bar{r}^{ct}} (1 - \hat{r}^{ct})^{(1-\bar{r}^{ct})} \\ p_{\theta_4}(a_r|x,c,t,r) &= \prod_{z \in \{s,e\}} \prod_{i=1}^N (\hat{a}_i^{zctr})^{\bar{a}_i^{zctr}} (1 - \hat{a}_i^{zctr})^{(1-\bar{a}_i^{zctr})} \end{aligned} \tag{16}$$

where \hat{c} , \hat{t}_i^{sc} , \hat{t}_i^{ec} , \hat{r}^{ct} , \hat{a}_i^{sctr} , and \hat{a}_i^{ectr} are the predicted probabilities of the event type, starting and ending positions of triggers, role types, and starting and ending positions of arguments, respectively, which can be calculated according to Formulas (3), (7), (12), and (14). \bar{c} , \bar{t}_i^{sc} , \bar{t}_i^{ec} , \bar{r}^{ct} , \bar{a}_i^{sctr} , and \bar{a}_i^{ectr} are real labels in the training data. $\theta_1 \triangleq \{\theta_{bert}, \theta_{td}\}$, $\theta_2 \triangleq \{\theta_{bert}, \theta_{te}\}$, $\theta_3 \triangleq \{\theta_{bert}, \theta_{re}\}$, $\theta_4 \triangleq \{\theta_{bert}, \theta_{ae}\}$, and θ_{bert} , θ_{td} , θ_{te} , θ_{re} , θ_{ae} denote the parameters from BERT, event detection, trigger identification, role pre-judgment, and argument extraction, respectively. We choose Adam [28] over shuffled mini-batches to minimize Loss_{all} .

4. Experiments and Analysis

4.1. Datasets

We use the FewFC dataset [11] to conduct comparative experiments. The reason for choosing FewFC is that other datasets do not completely cover the three overlapping situations mentioned in Section 1 like FewFC. For example, only 10% of events in the mainstream ACE2005 dataset have overlapping arguments, and there are no samples with overlapping triggers [12]. FewFC is a benchmark dataset in the Chinese financial field extracted for overlapping events, in which a total of 10 event types and 18 roles are annotated, and about 22% of the sentences contain overlapping events. Regardless of whether the event types are the same, the test set in FewFC contains 168 samples with overlapping triggers

and 209 samples with overlapping arguments. Note that there is an intersection between overlapping samples. The dataset is split into training, validation, and testing by 80%, 10%, and 10%. See Table 1 for more detail.

Table 1. Statistics of FewFC. Each column represents the size of each subset in terms of overlapped triggers, arguments, samples, and events.

	Trigger Overlap	Argument Overlap	Samples	Event
Training	1314	1541	7185	10,277
Validation	168	203	899	1281
Testing	168	209	898	1332
All	1650	1953	8982	12,890

4.2. Implementation Details

We choose PyTorch for code implementation, and NVIDIA A100-PCIE-40GB for training. ROPEE uses the *bert-base-Chinese* model with a starting learning rate of 2×10^{-5} , a decoder learning rate of 1×10^{-4} , and a decoder dropout rate of 0.3. The batch size is eight. The hidden layer size (d) is 768, the size of the position embeddings (d_p) is 64, and the epochs for training are 20. All hyperparameters are turned on for the validation set. The event type embeddings and role embeddings are trained from scratch with random initialization.

4.3. Evaluation Metric

The evaluation metric includes four parts [8–10]: (1) Trigger Identification (TI): The trigger is considered correctly recognized if the predicted trigger span aligns with the ground truth label. (2) Trigger Classification* (TC): The trigger is deemed correctly classified when it is both accurately identified and assigned to the right event type. (3) Argument Identification (AI): The argument is considered correctly identified if the event type is accurately recognized and the predicted argument span aligns with the gold span. (4) Argument Classification (AC): The argument is considered correctly classified when it is both accurately identified and the predicted role matches the gold role. Each of these parts is evaluated by three metrics: micro precision (P), micro recall (R), and micro F1-measure (F1). The specific formula is as follows:

$$\begin{aligned}
 \text{Precision}_{micro} &= \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i} \\
 \text{Recall}_{micro} &= \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i} \\
 \text{F1}_{micro} &= 2 \cdot \frac{\text{Precision}_{micro} \cdot \text{Recall}_{micro}}{\text{Precision}_{micro} + \text{Recall}_{micro}}
 \end{aligned} \tag{17}$$

where Precision_{micro} and Recall_{micro} represent the average precision and recall across all categories.

4.4. Baselines

The following baseline models are chosen to compare with ROPEE over the FewFC dataset:

- (1) BERT-softmax [24]: It uses BERT to obtain the feature representation of words for the classification of both trigger and event arguments.
- (2) BERT-CRF [29]: It uses a CRF module based on BERT to catch the transfer rules between adjacent tags.
- (3) BERT-CRF-joint: It extends the classic BIO labeling scheme by merging tags of event types and roles for sequence annotation, such as BIO-type roles [30].

These methods convert the EE task into a sequence labeling task by attaching a label to each token. The flattened sequence labeling approach cannot address the overlapping problem due to label conflicts.

- (4) PLMEE [12]: It extracts triggers and arguments in a pipeline manner, and alleviates the argument overlapping issue by extracting role-aware arguments.
- (5) MQAEE: It is extended based on Li et al. [15]. It first predicts overlapping triggers through question and answer, and then predicts overlapping arguments based on typed triggers.
- (6) CasEE [9]: It performs all four subtasks sequentially with a cascade decoder based on the specific previous predictions.

The above three methods are multi-stage methods for overlapping event extraction.

4.5. Main Results

Table 2 shows experimental results comparing ROPEE with baseline models over FewFC. It can be revealed from the table below.

Table 2. Model comparative results of EE on four subtasks over FewFC. Highest scores are in bold.

	TI (%)			TC (%)			AI (%)			AC (%)		
	P	R	F1									
BERT-softmax	89.8	79.0	84.0	80.2	61.8	69.8	74.6	62.8	68.2	72.5	60.2	65.8
BERT-CRF	90.8	80.8	85.5	81.7	63.6	71.5	75.1	64.3	69.3	72.9	61.8	66.9
BERT-CRF-joint	89.5	79.8	84.4	80.7	63.0	70.8	76.1	63.5	69.2	74.2	61.2	67.1
PLMEE	83.7	85.8	84.7	75.6	74.5	75.1	74.3	67.3	70.6	72.5	65.5	68.8
MQAEE	89.1	85.5	87.4	79.7	76.1	77.8	70.3	68.3	69.3	68.2	66.5	67.3
CasEE	89.4	87.7	88.6	77.9	78.5	78.2	72.8	73.1	72.9	71.3	71.5	71.4
ROPEE	88.8	88.2	88.5	74.7	82.8	78.6	69.5	78.6	73.8	67.6	76.9	72.0

- (1) In contrast to the flattened sequence labeling methods, ROPEE achieves superior recall and F1 scores. Specifically, ROPEE outperforms BERT-CRF-joint by 15.7% and 4.9% on recall and the F1 score of AC. ROPEE also achieves significantly better than the sequence labeling method on recall because the sequence labeling method can only solve the flat event extraction problem, which can cause label conflicts. This shows that ROPEE can effectively solve the problem of overlapping event extraction.
- (2) Compared with the multi-stage methods for overlapping event extraction, the F1 scores of ROPEE in AI and AC are greater than that of CasEE by 0.9% and 0.6%, respectively. We believe that the role pre-judgment decoder in the model provides good help for argument extraction. In particular, ROPEE outperforms CasEE on the recall score of all four subtasks, especially AI and AC by 5.5% and 5.4%, respectively. This shows that the ROPEE model can better recall arguments that match the role type in the training of role pre-judgment. Overall, ROPEE outperforms all multi-stage methods for overlapping event extraction.
- (3) We also conducted comparative experiments on the large language model ChatGLM2-6B [31,32]. Some parameters of ChatGLM were fine-tuned using the P-Tuning-v2 method, but the final result was not ideal. ChatGLM is more accurate in extracting core arguments and triggers, but the span position it extracts is seriously inconsistent with the original text, indicating that ChatGLM fails to understand the boundary meaning represented by span.

4.6. Results of Overlapped EE

Table 3 shows the comparative performance of ROPEE and CasEE in two overlapping situations. For sentences with overlapping triggers, ROPEE achieves improvements of 0.9% and 1.2% over CasEE on the F1 score of AI and AC, respectively. For overlapping arguments, ROPEE outperforms CasEE in all four subtasks. Experimental results illustrate the superiority of ROPEE in solving overlapping problems, and we argue that the role pre-judgment decoder predicts potential roles and promotes the performance of extraction in overlapping arguments.

Table 3. Comparative results of ROPEE and CasEE in two overlapping situations. Highest scores are in bold.

		TI (%)	TC (%)	AI (%)	AC (%)
Trigger Overlap	CasEE	92.5	82.8	75.5	74.2
	ROPEE	92.0	82.4	76.4	75.4
Argument Overlap	CasEE	88.6	78.2	74.2	72.8
	ROPEE	89.2	78.7	74.9	73.4

4.7. Ablation Studies

To verify the rationality of the role pre-judgment decoder, we conducted ablation experiments. In Table 4, *ROP-text_emb* only uses the text representation containing event-type embeddings as the predicted role type. *type_emb* denotes models that pre-judge roles using only event-type embeddings. As shown from this table, the *type_emb* used in the CasEE model performs worse than ROPEE on the F1 of TC, AI, and AC. It can be seen that event-type embeddings alone are not as efficient as the text embeddings containing event-type embeddings when used as the input for the role pre-judgment decoder, e.g., *ROP-text_emb*. ROPEE performs role pre-judgment by fusing the information of text representation and trigger embeddings, which outperforms *ROP-text_emb* in the F1 score of all four subtasks. It can be seen that trigger embeddings serve as supplementary information, allowing the model to grasp the knowledge hidden in the text.

Table 4. Comparative results on F1 using different role classifier strategies. Highest scores are in bold.

	TI (%)	TC (%)	AI (%)	AC (%)
type_emb	88.6	78.2	72.9	71.4
ROP-text_emb	88.2	78.4	73.4	71.7
ROPEE	88.5	78.6	73.8	72.0

In order to check the effectiveness of the role pre-judgment decoder, we adopt two training strategies while retaining the role pre-judgment decoder. The experimental results are given in Table 5. We use the correct role type for training and validation, such as *Role-x* (x can be 1, 2, 3, 4, or 5, representing the role pre-judgment decoder in the loss function after the correct role is input to the model weight value). With hyperparameters unchanged, the *Role-1* model converged quickly. It stopped iterating at 11 epochs, and its effects on four subtasks were not as good as ROPEE. To enable the model to learn more fully and pay attention to the learning of role types, we gradually increase the weight of the loss function in the role prediction part with the other hyperparameters remaining unchanged. An obvious trend of first increasing and then decreasing can be seen from *Role-x* in Table 5, and the best performance is achieved in *Role-3*, which is only 0.3% lower than ROPEE in both of AI and AC. It can be seen that the training strategy of using the correct role type requires adjusting parameters on the original basis to enable the model to focus on role-type learning. However, the performance of this training strategy is not as good as ROPEE. We argue that the correct role type information can greatly reduce the difficulty of the task. The rapid convergence of the model during training leads to insufficient learning and over-fitting.

Table 6 shows the comparative results between a model with trigger extraction and one without trigger extraction. In the stage of data annotation, if no triggers are annotated, the event extraction task is completed using only event types, roles, and arguments. Since AI and AC tasks need only event-type information, we use these two to measure the performance of ROPEE. According to Table 6, 20 epochs are also used for training. When the correct triggers are not provided, the F1 score is decreased by 1.8% and 1.5% respectively, in AI and AC. Due to the increased difficulty in model training, we increased the number

of training epochs to 50, and the result was increased by 0.2% in the F1 of AI, in contrast with 20 epoch w/o trigger. Thus, the lack of trigger annotation information will result in the inability to obtain the position embeddings of triggers during argument extraction, which also affects the model performance. Despite this, the model without trigger information achieves improvements than that of PLMEE and MQAEE, both of which are multi-stage methods with triggers involved.

Table 5. Comparative results on F1 using different training strategies. Highest scores are in bold.

	TI (%)	TC (%)	AI (%)	AC (%)
Role-1	88.1	77.6	71.7	69.4
Role-2	88.5	77.7	72.7	70.6
Role-3	88.4	78.1	73.5	71.7
Role-4	87.8	76.6	72.0	70.3
Role-5	88.2	77.2	72.4	70.7
ROPEE	88.5	78.6	73.8	72.0

Table 6. Comparative results on F1 with and without trigger annotation.

		P (%)	R (%)	F1 (%)
ROPEE	AI	69.5	78.6	73.8
	AC	67.6	76.9	72.0
20 epoch w/o trigger	AI	71.4	72.6	72.0
	AC	70.0	71.0	70.5
50 epoch w/o trigger	AI	70.9	73.5	72.2
	AC	69.3	71.7	70.5

4.8. Case Analysis

Figure 3 shows a test case analysis on CasEE and ROPEE, two events both triggered by “reduce”. For the “减持” (Reduction) event, both CasEE and ROPEE predict “Great Wall Film and Television” as the role of “target-company”, and there is no such argument in the standard answer. This shows that these two models lack the ability to judge the semantic boundaries of events. For the “股份股权转让” (share equity transfer) event, CasEE failed to extract the remote argument “Great Wall Group” as the role *sub*, while ROPEE improved the recall of AC by adding a role pre-judgment decoder, and, thus, was able to correctly detect the role of *sub*.

- 事件类型：减持
- 事件类型：股份股权转让

- event type: Reduction
- event type: Share Equity Transfer

长城影视公告称, 因公司股价持续下跌, 长城集团的东北证券客户信用交易担保证券账户于10月29日至11月4日期间, 以集中竞价交易方式累计被动减持公司股份51.6万股 (占公司总股本的0.1%) 成交均价为3.0165元/股, 成交金额155.65万元。

Great Wall Movie and Television announced that due to the continuous decline in the company's stock price, the Northeast Securities client credit trading guarantee securities account of Greatwall Group was passively reduced by a total of 516,000 shares (0.1% of the total share capital of the company) through centralized bidding trading from October 29th to November 4th, with an average transaction price of 3.0165 yuan per share and a transaction amount of 1.5565 million yuan.

Figure 3. Golden data of test case analysis on CasEE and ROPEE.

5. Conclusions

We design a joint learning framework ROPEE for overlapping event extraction, in which the event detection decoder can identify potential event types and help extract overlapping triggers. Based on text embeddings and trigger embeddings, a role pre-judgment module is proposed to predict roles based on the corresponding relationship between event types and roles, thereby enhancing the extraction of overlapping arguments. ROPEE has a certain effectiveness in addressing the issue of trigger overlap in EE, as well as argument overlap. Unlike other pipeline models, we integrate the tasks of four modules in the loss optimization layer, avoiding the traditional problem of error propagation existing in other pipeline methods. On the FewFC dataset, which was compared with other flattened sequence labeling methods (such as BERT-softmax, BERT-CRF, and BERT-CRF-joint), ROPEE achieves excellent recall and F1 scores on all subtasks; compared with multi-stage methods for overlapping event extraction (such as PLMEE, MQAEE, and CasEE), and ROPEE is superior to the above methods in terms of F1 scores of TC, AI, and AC. The above results show the superiority of ROPEE in overlapping event extraction. The ablation experimental results show that our model can also be used in different training strategies and other task scenarios or datasets without trigger labeling. In the future, we plan to build one more overlapping datasets based on specific application scenarios to verify our model's performance. In addition, since trigger tagging requires additional manpower and material resources, the absence of trigger tagging will reduce the accuracy of the model. The scenario without trigger annotation is general (models should automatically find the core arguments in the sentence), and we plan to improve the model performance in this scenario using position embeddings relative to the core arguments. Finally, we implement joint learning by designing a joint loss of four modules in this article, and we plan to develop different joint strategies to strengthen the interaction between each subtask.

Author Contributions: Conceptualization, Q.C. and K.Y.; methodology, K.Y. and Q.C.; software, K.Y.; investigation, S.W.; resources, X.G.; writing—original draft preparation, K.Y.; writing—review and editing, J.Z.; supervision, X.G.; project administration, J.L.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by National Key Research and Development Program of China (grant NO. 2022QY0300-01), National Natural Science Foundation of China (grant NO. 62076158), Natural Science Foundation of Shanxi Province of China (grant NOs. 202203021221021, 20210302123468 and 202203021221001), and CCF-Zhipu AI Large Model Fund (grant NO. 202310).

Data Availability Statement: The dataset used for this work is the FewFC dataset published by Zhou et al. [11]. It can be found at <https://github.com/TimeBurningFish/FewFC> (accessed on 17 November 2023). The codes are at <https://github.com/yang-4074/ROPEE> (accessed on 17 November 2023).

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

References

1. Miwa, M.; Bansal, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv* **2016**, arXiv:1601.00770.
2. Katiyar, A.; Cardie, C. Investigating lstms for joint extraction of opinion entities and relations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 919–929.
3. Fei, H.; Zhang, M.; Ji, D. Cross-lingual semantic role labeling with high-quality translated training corpus. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7014–7026.
4. Li, J.; Xu, K.; Li, F.; Fei, H.; Ren, Y.; Ji, D. MRN: A locally and globally mention-based reasoning network for document-level relation extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1359–1370.
5. Liu, Z.; Li, Y.; Zhang, Y.; Weng, Y.; Yang, K.; Wang, C. Effective Event Extraction Method via Enhanced Graph Convolutional Network Indication with Hierarchical Argument Selection Strategy. *Electronics* **2023**, *12*, 2981. [[CrossRef](#)]
6. Bosselut, A.; Le Bras, R.; Choi, Y. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In Proceedings of the 35th AAAI conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 4923–4931.

7. Xiang, G.; Shi, C.; Zhang, Y. An APT Event Extraction Method Based on BERT-BiGRU-CRF for APT Attack Detection. *Electronics* **2023**, *12*, 3349. [[CrossRef](#)]
8. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 167–176.
9. Sheng, J.; Guo, S.; Yu, B.; Li, Q.; Hei, Y.; Wang, L.; Liu, T.; Xu, H. CasEE: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Cedarville, OH, USA, 2021; pp. 164–174.
10. 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. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 1953–1964.
11. Zhou, Y.; Chen, Y.; Zhao, J.; Wu, Y.; Xu, J.; Li, J. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In Proceedings of the 35th AAAI conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 14638–14646.
12. 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 Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5284–5294.
13. Xu, N.; Xie, H.; Zhao, D. A novel joint framework for multiple Chinese events extraction. In Proceedings of the China National Conference on Chinese Computational Linguistics, Hainan, China, 30 October–1 November 2020; pp. 174–183.
14. Huang, K.H.; Yang, M.; Peng, N. Biomedical event extraction with hierarchical knowledge graphs. *arXiv* **2020**, arXiv:2009.09335.
15. Zhang, X.; Zhu, Y.H.; OuYang, K.; Kong, L.W. Chinese Event Extraction Based on Role Separation. *J. Shanxi Univ.* **2022**, *45*, 936–946.
16. Yang, H.J.; Jin, X.Y. A general model for entity relationship and event extraction. *Comput. Eng.* **2023**, *49*, 143–149.
17. Zhu, M.; Mao, Y.C.; Cheng, Y.; Chen, C.J.; Wang, L.B. Event Extraction Method Based on Dual Attention Mechanism. *Ruan Jian Xue Bao/J. Softw.* **2023**, *34*, 3226–3240.
18. Li, Q.; Li, J.; Sheng, J.; Cui, S.; Wu, J.; Hei, Y.; Peng, H.; Guo, S.; Wang, L.; Beheshti, A.; et al. A survey on deep learning event extraction: Approaches and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *14*, 1–21. [[CrossRef](#)]
19. 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: EMNLP 2020, Online, 16–20 November 2020; pp. 829–838.
20. Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; Santos, C.N.d.; Xiang, B.; Soatto, S. Structured prediction as translation between augmented natural languages. In Proceedings of the Ninth International Conference on Learning Representations, Online, 3–7 May 2021.
21. Hsu, I.; Huang, K.-H.; Boschee, E.; Miller, S.; Natarajan, P.; Chang, K.-W.; Peng, N. DEGREE: A data-efficient generation-based event extraction model. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, DC, USA, 10–15 July 2022; pp. 1890–1908.
22. Van Nguyen, M.; Min, B.; Dernoncourt, F.; Nguyen, T. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, DC, USA, 10–15 July 2022; pp. 4363–4374.
23. Liu, J.; Liang, C.; Xu, J.; Liu, H.; Zhao, Z. Document-level event argument extraction with a chain reasoning paradigm. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; pp. 9570–9583.
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Yu, B.; Zhang, Z.; Sheng, J.; Liu, T.; Wang, Y.; Wang, Y.; Wang, B. Semi-open information extraction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1661–1672.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; 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, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv* **2021**, arXiv:2104.09864.
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. 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, Online, 5–10 July 2020; pp. 8010–8020.
30. 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.

31. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X. Glm-130b: An open bilingual pre-trained model. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
32. Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; Tang, J. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 320–335.

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.