Hindawi
Mobile Information Systems
Volume 2022, Article ID 2742796, 8 pages
https://doi.org/10.1155/2022/2742796

*Research Article*

# DEERE: Document-Level Event Extraction as Relation Extraction

**Jian Li**,[1] **Ruijuan Hu**,[2] **Keliang Zhang**,[1] **Haiyan Liu**,[2] **and Yanzhou Ma**[1]

[1]*PLA Strategic Support Force Information Engineering University, Luoyang 471003, China*
[2]*PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China*

Correspondence should be addressed to Jian Li; maomaotfntfn@163.com

Received 18 July 2022; Accepted 10 August 2022; Published 26 August 2022

Academic Editor: Chia-Huei Wu

The descriptions of complex events usually span sentences, so we need to extract complete event information from the whole document. To address the challenges of document-level event extraction, we propose a novel framework named Document-level Event Extraction as Relation Extraction (DEERE), which is suitable for document-level event extraction tasks without trigger-word labelling. By well-designed task transformation, DEERE remodels event extraction as single-stage relation extraction, which can mitigate error propagation. A long text supported encoder is adopted in the relation extraction model to aware the global context effectively. A fault-tolerant event integration algorithm is designed to improve the prediction accuracy. Experimental results show that our approach advances the SOTA for the ChFinAnn dataset by an average F1-score of 3.7. The code and data are available at https://github.com/maomaotfntfn/DEERE.

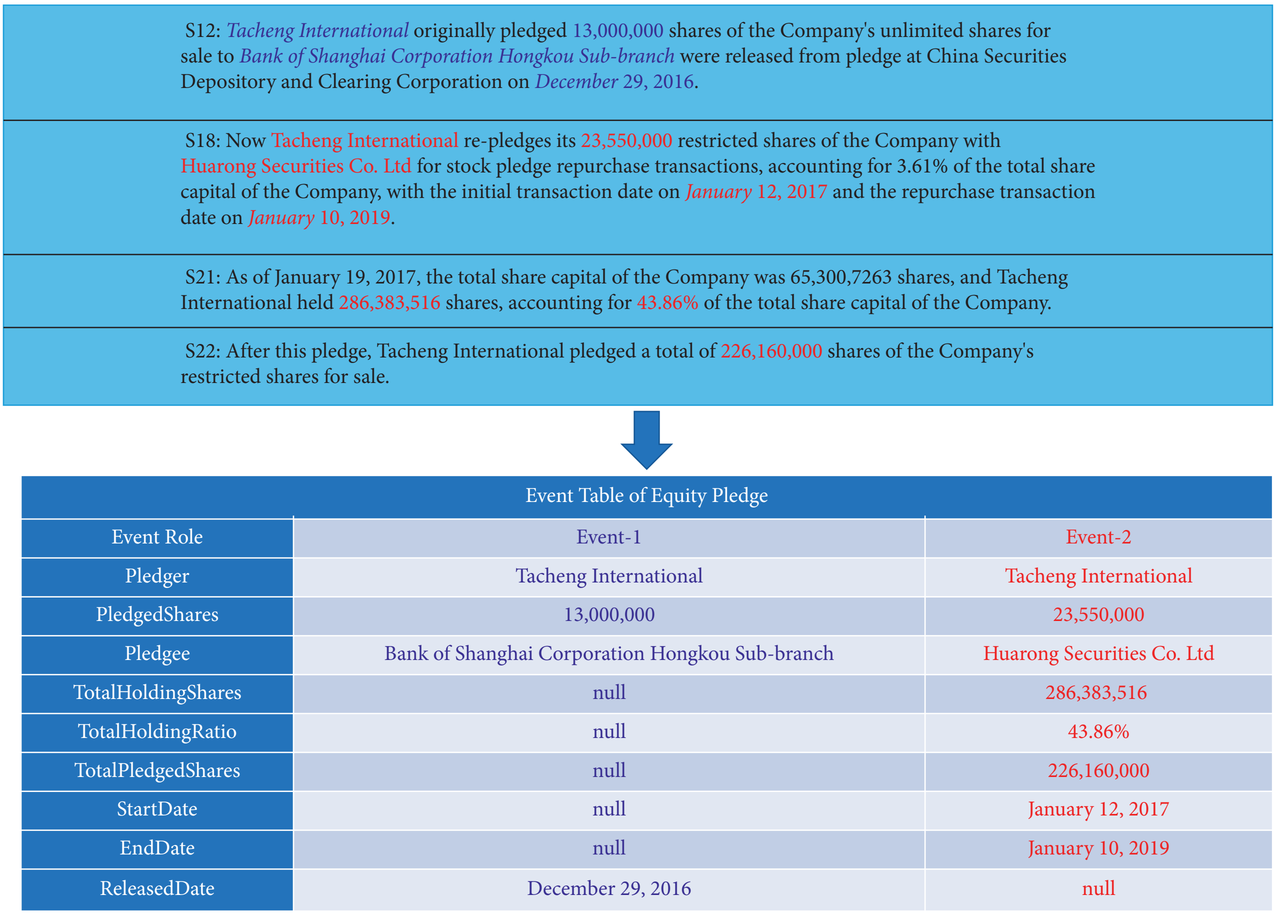
## 1. Introduction

The aim of the Event Extraction (EE) task is to extract structured event information from unstructured text [1]. EE can be divided into Sentence-level Event Extraction (SEE) and Document-level Event Extraction (DEE). Previous research has focused on SEE, but the description of a complex event usually involves multiple sentences, so we need to extract more complete event information from the whole document. SEE no longer meets our needs, and its methods are ill-suited for DEE tasks.

The two main challenges of DEE are argument-scattering and multievents. Argument-scattering means that the arguments of an event are scattered in multiple sentences. As shown in Figure 1, the arguments of Event-2 are scattered in S18, S21, and S22. Multievents mean that a document includes multiple events, in which there may be arguments that overlap. Depending on the degree of overlap, the relationship between two events can be classified as (1) no arguments overlap, (2) arguments overlap between events of different types, and (3) arguments overlap between events of the same type (e.g., both Event-1 and Event-2 contain the argument *Tacheng International*). In addition, we will also encounter the DEE task without trigger-word labelling, which can be regarded as another challenge.

To address the above-given challenges, the most recent SOTA method, DE-PPN [2], designed an end-to-end model, where a document-level encoder is used to obtain the text representations, and a multigranularity decoder is used to generate events in parallel. DE-PPN encodes each sentence separately and concatenates them into document encoding after max pooling, which does not consider the interaction between sentences and cannot be fully aware of the global context. The extraction process of DE-PPN includes candidate argument identification, event prediction, and role filling. This kind of multi-stage structure is prone to error propagation.

In this paper, we propose an event extraction framework named DEERE (short for Document-level Event Extraction as Relation Extraction). The key idea is to transform the complex DEE task into a relatively simple relation extraction task, which can deal with both challenges of argument-scattering and multievents. DEERE adopts a single-stage entity-relation joint extraction model to mitigate error propagation. A long text supported Transformer is used as the text encoder, which can effectively aware of the global context.



S12: *Tacheng International* originally pledged 13,000,000 shares of the Company's unlimited shares for sale to *Bank of Shanghai Corporation Hongkou Sub-branch* were released from pledge at China Securities Depository and Clearing Corporation on *December* 29, 2016.

S18: Now Tacheng International re-pledges its 23,550,000 restricted shares of the Company with Huarong Securities Co. Ltd for stock pledge repurchase transactions, accounting for 3.61% of the total share capital of the Company, with the initial transaction date on *January* 12, 2017 and the repurchase transaction date on *January* 10, 2019.

S21: As of January 19, 2017, the total share capital of the Company was 65,300,7263 shares, and Tacheng International held 286,383,516 shares, accounting for 43.86% of the total share capital of the Company.

S22: After this pledge, Tacheng International pledged a total of 226,160,000 shares of the Company's restricted shares for sale.

Event Table of Equity Pledge

| Event Role | Event-1 | Event-2 |
|---|---|---|
| Pledger | Tacheng International | Tacheng International |
| PledgedShares | 13,000,000 | 23,550,000 |
| Pledgee | Bank of Shanghai Corporation Hongkou Sub-branch | Huarong Securities Co. Ltd |
| TotalHoldingShares | null | 286,383,516 |
| TotalHoldingRatio | null | 43.86% |
| TotalPledgedShares | null | 226,160,000 |
| StartDate | null | January 12, 2017 |
| EndDate | null | January 10, 2019 |
| ReleasedDate | December 29, 2016 | null |

Figure 1: A sample of document-level event extraction.

In summary, our contributions include:

(1) We propose a novel framework (DEERE) based on task transformation, which is suitable for DEE tasks without trigger-word labelling.

(2) We design two key algorithms in the framework. The role selection algorithm can reduce the probability of event arguments overlapping, and the event division algorithm can further deal with the case of arguments overlapping mentioned.

(3) Experimental results show that DEERE significantly outperforms the most recent SOTA method on the widely used DEE dataset, with an average F1-score improvement of 3.7.

## 2. Methodology

As shown in Figure 2, the architecture of DEERE includes three modules: task transformation, relation extraction, and event prediction. Labelled events are transformed into well-designed relational triples, which are used as the training data of a relation extraction model. The relation extraction model adopts a long text supported Transformer to encode the whole document. During event prediction, the relational triples extracted from the input text are reorganized into basic events.

### 2.1. Task Transformation

*2.1.1. Relational Triplet Creation.* The labelled events in the training set are transformed into two kinds of relations: (1) role relation describes the role assignments between arguments, which is designed to resolve the challenge of argument-scattering and multievents without key role overlap; (2) co-event relation describes whether two arguments belong to the same event, which is designed to resolve the challenge of multi-events with key role overlap.

Suppose the event type $E_k$ includes $m$ roles, denoted as $[r_1, r_2, \ldots, r_m]$; an event instance of $E_k$ also includes $m$ arguments correspondingly, denoted as $[a_1, a_2, \ldots, a_m]$. Select one from the roles of $E_k$ as the key role, and the argument that plays the key role in a specific event is called the key argument. The key argument of the event is combined with each nonkey argument to form a role relation triple $(a_{key}, E_k\_r_i, a_i)$, where $a_{key}$ represents the key argument of a specific $E_k$ event, and $a_i$ represents a nonkey argument playing the $i$-th role in the same event $(i \neq \text{key})$.

In a document, a key argument may involve several events of the same type. In order to distinguish these events, a subkey role is selected from the nonkey roles of the event type, and the argument that plays the subkey role in a specific event is called the subkey argument. The subkey argument of the event is combined with every other argument to form a

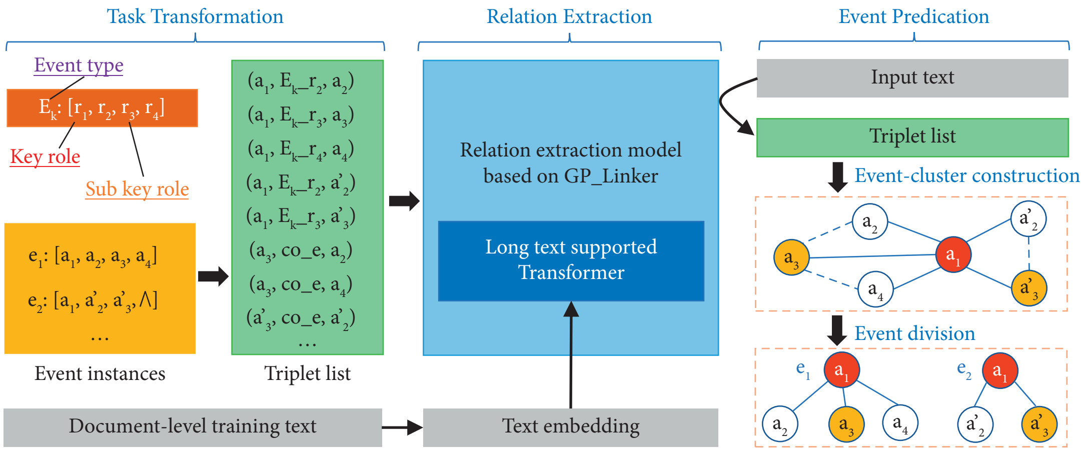


Figure 2: The overall architecture of DEERE.

coevent relation triple $(a_{\text{skey}}, co\_event, a_i)$, which represents that $a_{\text{skey}}$ and $a_i$ belong to a same event ($i \neq$ key and $i \neq$ skey).

According to the definition, there are multiple types of role relation, whose number is related to the total roles of all the event types. There is only one type of coevent relation, which ignores event types and roles. These relational triples will be used as the training data and prediction targets of the relational extraction model.

*2.1.2. Key Role Selection.* Based on the above-given task conversion rules, in order to reduce the situation of a key argument involving multiple events of the same type, we hope to select the role that can best distinguish different events as the key role. For a group of events with the same type, we define role discrimination as the average probability that each argument on a role can accurately identify the event to which it belongs.

Formally, suppose that a document $d$ contains $n$ events of $E_k$, denoted as $[e_1, e_2, \ldots, e_n]^{\mathrm{T}}$; the argument list of $e_i$ is denoted as $[a_i^1, a_i^2, \ldots, a_i^m]$, the argument matrix for all events can be represented as follows:

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} a_1^1 & a_1^2 & \ldots & a_1^m \\ a_2^1 & a_2^2 & \ldots & a_2^m \\ \vdots & \vdots & \ddots & \vdots \\ a_n^1 & a_n^2 & \ldots & a_n^m \end{bmatrix}. \tag{1}$$

The discrimination of role $r_j$ to events of $E_k$ in single document $d$ can be expressed as follows:

$$\text{discrimination}\left(E_k, r_j, d\right) = \sum_{i=1}^{n} p\left(e_i\right) * q\left(a_i^j\right). \tag{2}$$

In equation (2), $p(e_i)$ represents the probability of the occurrence of event $e_i$, and $q(a_i^j)$ represents the probability that the argument $a_i^j$ can accurately identify the event to which it belongs. We assume that each event has an equal probability, that is, $p(e_i) = 1/n$, so the formula can be simplified as follows:

$$\text{discrimination}\left(E_k, r_j, d\right) = \frac{1}{n} \sum_{i=1}^{n} q\left(a_i^j\right). \tag{3}$$

In a group of events of the same type, the higher the repetition rate of arguments on a role, the lower the role discrimination for events. It is not difficult to prove that the summation part in equation (3) is numerically equal to the count of nonrepeated arguments on the role, where arguments with null value are not counted. We can get the following equation:

$$\text{discrimination}\left(E_k, r_j, d\right) = \frac{1}{n} \text{count\_distinct}\left(a_1^j, a_2^j \ldots, a_n^j\right). \tag{4}$$

In equation (4), count_distinct represents the count of nonrepeated arguments, which can be easily obtained by set operation in practice. Suppose that there are $T$ documents in the training set that contain events of $E_k$, we can calculate the discrimination of role $r_j$ in each document $d_t$ separately, and regard the average as the global role discrimination:

$$\text{discrimination\_global}\left(E_k, r_j\right) = \frac{1}{T} \sum_{t=1}^{T} \text{discrimination}\left(E_k, r_j, d_t\right). \tag{5}$$

From the role list of $E_k$, we can select the role with the highest discrimination as the key role and select the one with the second highest discrimination as the subkey role.

## 2.2. Relation Extraction

*2.2.1. Relation Extraction Model.* The triples obtained from the task transformation will be input to the relation extraction model together with the original text as training data. In theory, any relation extraction model can be applied here. However, since the results of relation extraction will directly affect the results of EE, we adopt the recently proposed GPLinker [3], an entity-relationship joint extraction model based on GlobalPointer [4]. Preliminary experiments show that the performance of GPLinker is slightly better than the Casrel [5] and comparable to the

TPLinker [6]. In addition, the GPLinker model has fast training speed, high decoding efficiency, and theoretically no exposure bias.

GlobalPointer is essentially a token-pair recognition model, which can be used in nested and nonnested NER. Multilabel categorical cross-entropy is used as the loss function during training. GPLinker converts the extraction of relational triples (*subject*, *predicate*, *object*) into three kinds of token-pair recognition: entity head/tail pair, subject/object heads pair, and subject/object tails pair. Each kind of token-pair is recognized by a specific GlobalPointer, and all GlobalPointer modules share the same text encoder.

*2.2.2. Long Text Encoding.* In recent years, the relationship extraction SOTA models (Casrel, TPLinker, etc.) are mostly based on BERT [7] or other pretrained language models. The original BERT uses absolute position encoding and can handle a maximum text length of 512 tokens. The text length of document-level extraction tasks is usually beyond the above range. If the long text is truncated or segmented, it will inevitably affect the model's perception of the context of the full text, which is also a common problem with previous DEE models. For this, we try to use RoFormer [8], a Transformer that uses relative position encoding, to encode every document as a whole. Of course, other pretrained language models that support long text encoding can also be used here.

*2.3. Event Prediction.* Shown as the event prediction module of Figure 2, we first extract relational triplets from the input text and then integrate these triplets as basic events. Since the triples are predicted by the relation extraction model, there are inevitably some errors or omissions, which requires the event integration algorithm to have a certain fault tolerance. In this regard, we first construct event-clusters by the key argument and role relation. If an event-cluster contains multiple basic events, it will be further divided according to the subkey argument and coevent relation.

*2.3.1. Event-Cluster Construction.* Based on the predicted role relation triplets, a special structure called event-cluster is constructed around each key argument. Every event-cluster includes a key argument and related nonkey arguments, all of which belong to the same event type. An event-cluster can be classified as a single event-cluster or compound event-cluster according to the number of basic events included.

According to the construction rules, there is only one key argument in an event-cluster, but there may be several arguments on a nonkey role, which is called a multivalued role. The higher the proportion of multi-valued roles in an event-cluster, the more likely it contains multiple events. If there are more than one subkey argument and the proportion of multivalued roles exceeds a certain threshold, it will be judged as a compound event-cluster; otherwise, it will be judged as a single event-cluster. The threshold can be adjusted as a hyperparameter.

*2.3.2. Event Division.* A compound event-cluster needs to be further divided into basic events. In detail, we first create one basic event for each subkey argument and then assign arguments to other roles of each event according to the predicted coevent relations. If there are more than one candidate arguments for a role, it can be selected by relation strength.

We have also considered the maximal clique search algorithm for event division, which requires that any two arguments of the same event must be accurately judged as coevent relation. Assuming that an event has $n$ arguments, even if only one-way relations are considered, $n * (n-1)/2$ coevent relations need to be extracted. The missing of just one relation can lead to serious errors in event division. As shown in Figure 3, there is a complete event with 5 arguments [$A$, $B$, $C$, $D$, $E$], if the relation <$B$, $D$> is missing, it will be divided into two incomplete events [$A$, $B$, $C$, $E$] and [$A$, $C$, $D$, $E$].

Although the maximal clique search algorithm is more complete in theory, its application conditions are too demanding. In contrast, our proposed algorithm only needs $n-2$ coevent relations to divide a basic event, which not only reduces the application conditions of the algorithm but also has stronger fault tolerance. This point will be confirmed in the experiments.

## 3. Experiments

### 3.1. Evaluation Dataset

*3.1.1. Dataset.* Our proposed method will be evaluated on the ChFinAnn dataset, which is also applied by DCFEE [9], Doc2EDAG [10], and DE-PPN [2]. ChFinAnn is a DEE dataset without trigger-word labelling, which is automatically annotated by distant supervision. The dataset contains 5 financial event types: Equity Freeze (EF), Equity Repurchase (ER), Equity Underweight (EU), Equity Overweight (EO), and Equity Pledge (EP). It includes a total of 32,040 documents, about 30% of which contain multiple events.

*3.1.2. Evaluation Metric.* We follow the evaluation metrics used in Doc2EDAG and DE-PPN. For each predicted event, the most similar ground truth is selected without replacement to calculate Precision, Recall, and F1-score. Micro-averaged role-level scores are considered as the final metric for each event type. For the global performance, the previous literature only reported the macro-averaged F1-score of all event types, which does not consider the imbalance of sample distribution. Therefore, we will additionally report the microaveraged F1-score that can better reflect the practical performance.

*3.2. Experimental Setup.* The head size of the GlobalPointer in the relation extraction model is set to 64. The text encoder adopts the char-based RoFormer (Chinese_roformer_char_L-12_H-768_A-12), whose parameter scale is comparable to BERT and the vocabulary is reduced to 12000.

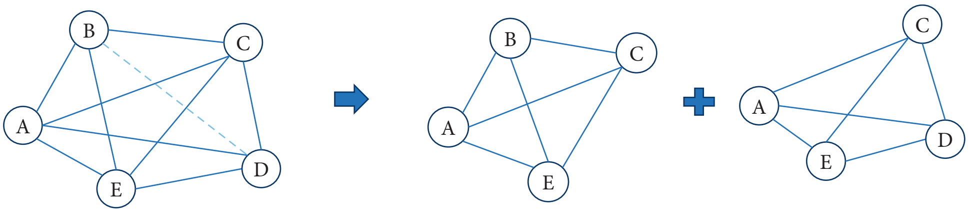


Figure 3: An error case of event division based on the maximal clique search algorithm.

Since most of the documents in the dataset are within 2000 tokens, the maximum encoding length (max_len) of RoFormer is set to 2000. The threshold of multivalued roles proportion for multi-event judgement is set to 0.2. For each event type, we select the role with the highest discrimination as the key role and select the second highest one as the subkey role. The selection results are shown in Table 1.

All experiments run on a workstation with RTX3090. The model is trained for 20 epochs with the optimizer of exponential moving average Adam and the learning rate of $1e-5$. The best performing model on the development set is saved and its performance on the test set is used as the final test result.

### 3.3. Main Results

*3.3.1. Baseline.* Our framework DEERE is compared with the previous SOTA methods as follows: DCFEE [9] proposed a DEE method based on key-event detection and argument completion. DCFEE-O and DCFEE-M are the single-event version and multievent version, respectively. Doc2EDAG [10] is an end-to-end model that transforms event table filling into a path expending of entity-based directed acyclic graph. GreedyDec is a simple version of Doc2EDAG, which only fills one event table entry greedily. DE-PPN [2] is the most recent SOTA model that aggregates the document-level context to predict events in parallel. DE-PPN-1 is the simple version that only generates one event.

*3.3.2. Compare with SOTA.* As shown in Table 2, our framework DEERE achieves the best performance in all 5 event types compared to baseline methods. Specifically, DEERE improves 1.0, 5.4, 6.0, 2.1, and 3.5 F1-score over the DE-PPN on the event type EF, ER, EU, EO, and EP, respectively. Table 3 shows that our model gets the highest score in both single-event and multievent parts for every event type. As to global performance, the macro F1-score of DEERE is 3.6 higher than that of DE-PPN (3.7 higher for the single-event part and 3.4 higher for the multi-event part). In addition, the micro F1-score of DEERE on the test set is 83.7 (90.8 for the single-event part and 75.9 for the multi-event part), which is not reported in the previous literature.

*3.4. Ablation Experiments.* To verify the effect of the two key mechanisms in our framework, we conducted a series of ablation experiments. As shown in Table 4, RoleSelection + means to select the key roles and subkey roles according to the discrimination for events, and RoleSelection-means to select them sequentially from the role list. EventDivision + means to perform the event division operation on event-clusters, EventDivision-means to treat every event-cluster as a single event.

Table 1: The result of key and subkey role selection.

| Event type | Key role | Subkey role |
|---|---|---|
| EquityPledge | PledgedShares | Pledgee |
| EquityRepurchase | RepurchasedShares | LowestTradingPrice |
| EquityOverweight | TradedShares | EndDate |
| EquityUnderweight | TradedShares | EndDate |
| EquityFreeze | FrozeShares | LegalInstitution |

Test results show that using either of the two mechanisms can improve the performance, but the improvements overlap significantly when both mechanisms are used. The reason is that after using role selection, the situation that the same key argument involves multiple events is greatly reduced, and the effect of event division will be relatively limited. Although the performance of using only role selection on the ChFinAnn dataset is close to that of using both mechanisms simultaneously, it does not mean that event division is dispensable. If the key roles selected in other EE tasks are not so ideal (i.e., the discrimination for events is not high enough), event division will become more important.

In addition, we also test the event division method based on maximal clique search. The micro F1-score is 81.3 with role selection and 76.8 without role selection. Compared with our event division method, it drops by 2.4 and 5.0, respectively.

*3.5. Effect of Maximum Encoding Length.* We investigate the influence of the max_len of the text encoder on the performance. As shown in Figure 4, with the increase of max_len, the F1-score and Recall increases rapidly, while the Precision is basically flat (with a slight decrease). The model performance no longer improves significantly after the max_len exceeds 2000, which is consistent with our default value. The results affirm the idea of global context awareness and the feasibility of whole document encoding.

*3.6. Effect of Relation Extraction Model.* Our framework does not rely on specific relation extraction models and text encoders, which can be freely chosen. With other settings unchanged, we try to adopt different combinations of relation extraction models and text encoders. The training results are shown in Figure 5, where Casrel [5] is a two-stage entity-relationship joint extraction model, and NEZHA [11]

Table 2: Precision (P), recall (R), and F1-score (F1) for 5 event types.

| Models | EF | | | ER | | | EU | | | EO | | | EP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| DCFEE-O | 66.0 | 41.6 | 51.1 | 84.5 | 81.8 | 83.1 | 62.7 | 35.4 | 45.3 | 51.4 | 42.6 | 46.6 | 64.3 | 63.6 | 63.9 |
| DCFEE-M | 51.8 | 40.7 | 45.6 | 83.7 | 78.0 | 80.8 | 49.5 | 39.9 | 44.2 | 42.5 | 47.5 | 44.9 | 59.8 | 66.4 | 62.9 |
| GreedyDec | 79.5 | 46.8 | 58.9 | 83.3 | 74.9 | 78.9 | 68.7 | 40.8 | 51.2 | 69.7 | 40.6 | 51.3 | **85.7** | 48.7 | 62.1 |
| Doc2EDAG | 77.1 | 64.5 | 70.2 | 91.3 | 83.6 | 87.3 | 80.2 | 65.0 | 71.8 | 82.1 | 69.0 | 75.0 | 80.0 | 74.8 | 77.3 |
| DE-PPN-1 | 77.8 | 55.8 | 64.9 | 75.6 | 76.4 | 76.0 | 76.4 | 63.7 | 69.4 | 77.1 | 54.3 | 63.7 | 85.5 | 43.0 | 57.2 |
| DE-PPN | 78.2 | 69.4 | 73.5 | 89.3 | 85.6 | 87.6 | 69.7 | **79.9** | 74.4 | 81.0 | 71.3 | 75.8 | 83.8 | 73.7 | 78.4 |
| DEERE (ours) | **79.4** | **70.3** | **74.5** | **94.5** | **91.5** | **93.0** | **83.3** | 77.5 | **80.4** | **81.2** | **74.9** | **77.9** | 83.5 | **80.3** | **81.9** |

Table 3: F1-score for all event types on single-event (S.) and multievent (M.) parts.

| Models | EF | | ER | | EU | | EO | | EP | | Macro-avg | | | Micro-avg | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S. | M. | S. | M. | S. | M. | S. | M. | S. | M. | S. | M. | S&M | S. | M. | S&M |
| DCFEE-O | 56.0 | 46.5 | 86.7 | 54.1 | 48.5 | 41.2 | 47.7 | 45.2 | 68.4 | 61.1 | 61.5 | 49.6 | 58.0 | | — | |
| DCFEE-M | 48.4 | 43.1 | 83.8 | 53.4 | 48.1 | 39.6 | 47.4 | 42.0 | 67.0 | 60.0 | 58.9 | 47.7 | 55.7 | | — | |
| GreedyDec | 75.9 | 40.8 | 81.7 | 49.8 | 62.2 | 34.6 | 65.7 | 29.4 | 88.5 | 42.3 | 74.8 | 39.4 | 60.5 | | — | |
| Doc2EDAG | 80.0 | 61.3 | 89.4 | 68.4 | 77.4 | 64.6 | 79.4 | 69.5 | 85.5 | 72.5 | 82.3 | 67.3 | 76.3 | | — | |
| DE-PPN-1 | 82.4 | 46.3 | 78.3 | 53.9 | 82.2 | 45.6 | 78.1 | 39.3 | 82.8 | 38.5 | 80.7 | 44.7 | 66.2 | | — | |
| DE-PPN | 82.1 | 63.5 | 89.1 | 70.5 | 79.7 | 66.7 | 80.6 | 69.6 | 88.0 | 73.2 | 83.9 | 68.7 | 77.9 | | — | |
| DEERE (ours) | **84.8** | **65.4** | **95.6** | **71.2** | **84.5** | **75.1** | **83.7** | **71.1** | **89.2** | **77.6** | **87.6** | **72.1** | **81.5** | **90.8** | **75.9** | **83.7** |

Table 4: F1-score of ablation tests on DEERE variants.

| RoleSelection | EventDivision | EF | ER | EU | EO | EP | Macro-avg | Micro-avg |
|---|---|---|---|---|---|---|---|---|
| + | + | **74.5** | **93.0** | 80.4 | 77.9 | **81.9** | **81.5** | **83.7** |
| + | − | 72.7 | 92.9 | **80.7** | **78.4** | 81.6 | 81.3 | 83.5 |
| − | + | 72.6 | 92.2 | 79.8 | 75.0 | 79.6 | 79.8 | 81.8 |
| − | − | 70.0 | 91.9 | 74.3 | 68.0 | 67.6 | 74.4 | 74.6 |

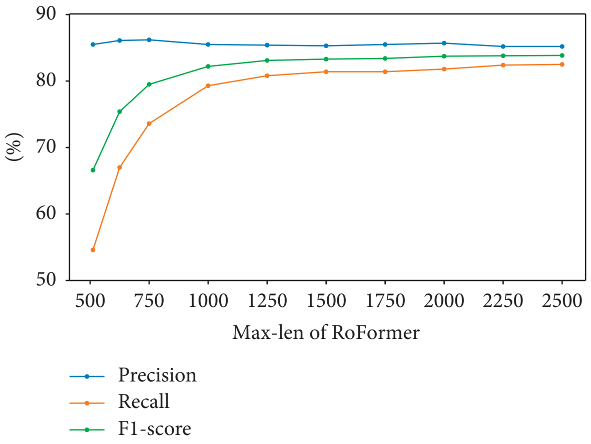


Figure 4: Effect of maximum encoding length.

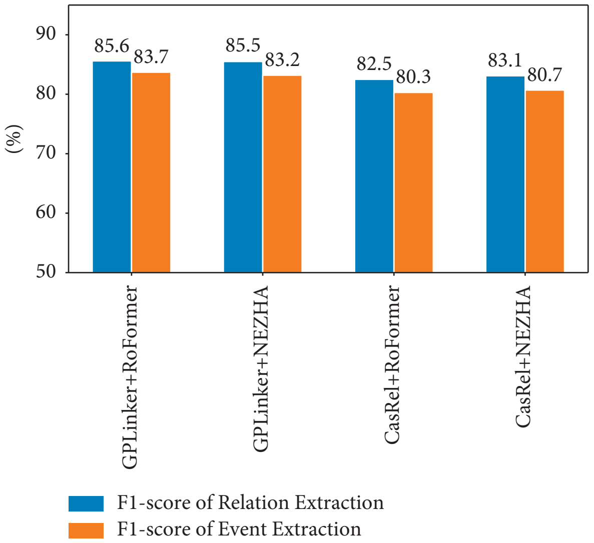


Figure 5: Effect of relation extraction model.

is another pretrained language model using relative position encoding. The performance of EE is basically synchronized with that of relation extraction, and the combination of GPLinker and RoFormer obtains the best score. The results show the single-stage relation extraction model (GPLinker) is significantly better than the multistage model (Casrel), while the effect of text encoder changes is much smaller.

## 4. Related Work

SEE uses only features obtained from intrasentences, and traditional feature engineering-based approaches [12, 13], cannot be adapted to tasks that rely on complex semantic relationships. Recent work on EE is based on deep learning to automatically learn features, mainly using pipeline and joint models. Pipeline methods [14, 15], whether using CNN or RNN, use a pipeline approach to split the extraction process into two separate processes, extracting event trigger words and detecting arguments. This approach inevitably causes error transfer and makes it difficult to capture long-distance dependent information. To reduce error transfer, joint methods [16, 17], consider simultaneous extraction of trigger words and arguments. To address the problem of overlapping roles, pretrained language models [18, 19], are used to model intrasentence and intersentence contextual information, improving the accuracy of the task overall.

Early classification models [20, 21], divide DEE into two subtasks: recognition of event descriptors and detecting arguments, using SVM as classifiers. Neural network-based classification models [22, 23], use word embeddings as the input to the decision tree, and then the structured information of the document is obtained through the integration of information.

To address the argument-scattering, one solution is to transform the extraction task into a sequence annotation task, which dynamically fuses two different levels of information at the sentence level and the document-level [24]. Another solution is to use a sentence as the event-centered sentence for arguments complementation. DCFEE [9] is based on sequence annotation, main event discovery, and arguments complementation strategy to construct the extraction model, which solves the argument-scattering to a certain extent. For the multievents, Doc2EDAG [10] utilizes an end-to-end approach to integrate arguments scattered across different documents and transforms the document-level event table filling task into an entity-based path expansion task for directed acyclic graphs. The multilayer bidirectional network MLBiNet [25] fuses cross-sentence semantic and associative event information to enhance the discrimination of each event mention. In addition, there are methods [26, 27], that transform EE into other tasks such as reading comprehension and intelligent quizzing.

## 5. Conclusion

We propose a novel framework (DEERE) that cleverly transforms the task of DEE into a relation extraction task. The new SOTA performance on the ChFinAnn dataset illustrates the reasonability of the framework design, and ablation experiments verify the effectiveness of the key mechanisms in the framework. The single-stage relation extraction model can mitigate error propagation, and the event integration algorithm with fault tolerance can compensate for some errors of relation extraction. Increasing the maximum length of the text encoder is beneficial to improve document awareness, but it also requires more computational resources. The ChFinAnn dataset used in our experiments has a considerable scale, but labelled data in other application scenarios are often insufficient. In future work, we will focus on few-shot event extraction and data augmentation for it.

## Data Availability

The code and data are available at https://github.com/maomaotfntfn/DEERE.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

Jian Li and Ruijuan Hu are contributed equally to this work.

## References

[1] F. Hogenboom, F. Frasincar, U. Kaymak, F. De Jong, and E. Caron, "A survey of event extraction methods from text for decision support systems," *Decision Support Systems*, vol. 85, pp. 12–22, 2016.

[2] H. Yang, D. Sui, Y. Chen, K. Liu, J. Zhao, and T. Wang, "Document-level event extraction via parallel prediction networks," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 6298–6308, Bangkok, Thailand, 2021.

[3] J. Su, "GPlinker: entity-relation joint extraction based on GlobalPointer," 2022, https://kexue.fm/archives/8373.

[4] J. Su, "GlobalPointer: Handle Nested and Non-nested NER in a Unified Way," 2021, https://kexue.fm/archives/8888.

[5] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476–1488, Seattle, July 2020.

[6] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "TPLinker: single stage joint extraction of entities and relations through token pair linking," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1572–1582, Barcelona, Spain, December 2020.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, Minneapolis, Minnesota, June 2019.

[8] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," vol. abs/2104, Article ID 09864, 2021, https://arxiv.org/abs/2104.09864.

[9] H. Yang, Y. Chen, K. Liu, Y. Xiao, and J. Zhao, "DCFEE: a document level Chinese financial event extraction system based on automatically labeled training data," in *Proceedings of the ACL 2018*, pp. 50–55, System Demonstrations, Melbourne, Australia, February 2018.

[10] S. Zheng, W. Cao, W. Xu, and J. Bian, "An end-to-end document-level framework for Chinese financial event extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 337–346, Hong Kong, China, November 2019.

[11] J. Wei, X. Ren, X. Li et al., "Nezha: Neural contextualized representation for Chinese language understanding," vol. abs/1909, Article ID 00204, 2019, https://arxiv.org/abs/1909.00204.

[12] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 73–82, Sofia, Bulgaria, August 2013.

[13] T. H. Nguyen and R. Grishman, “Event detection and domain adaptation with convolutional neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pp. 365–371, Beijing, China, July 2015.

[14] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “Event extraction via dynamic multi-pooling convolutional neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pp. 167–176, Beijing, China, August 2015.

[15] X. Feng, L. Huang, D. Tang, H. Ji, B. Qin, and T. Liu, “A language-independent neural network for event detection,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 66–71, Berlin, Germany, August 2016.

[16] T. H. Nguyen, K. Cho, and R. Grishman, “Joint event extraction via recurrent neural networks,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 300–309, San Diego, California, June 2016.

[17] Y. S. Chan, J. Fasching, H. Qiu, and B. Min, “Rapid customization for event extraction,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 31–36, Florence, Italy, July 2019.

[18] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, “Exploring pre-trained language models for event extraction and generation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5284–5294, Florence, Italy, July 2019.

[19] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, “Entity, relation, and event extraction with contextualized span representations,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 5784–5789, Hong Kong, China, November 2019.

[20] S. Patwardhan and E. Riloff, “A unified model of phrasal and sentential evidence for information extraction,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 151–160, Singapore, August 2009.

[21] R. Huang and E. Riloff, “Peeling back the layers: detecting event role fillers in secondary contexts,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, June 2011.

[22] E. Boros, R. Besanon, O. Ferret, and B. Grau, “Event role extraction using domain-relevant word representations,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, October 2014.

[23] Y. Lu, H. Lin, J. Xu et al., “Text2Event: controllable sequence-to-structure generation for end-to-end event extraction,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 2795–2806, Bangkok, Thailand, August 2021.

[24] X. Du and C. Cardie, “Document-level event role filler extraction using multi-granularity contextualized encoding,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8010–8020, Seattle, July 2020.

[25] D. Lou, Z. Liao, S. Deng, N. Zhang, and H. Chen, “MLBiNet: a cross-sentence collective event detection network,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4829–4839, Bangkok, Thailand, August 2021.

[26] Y. Chen, T. Chen, S. Ebner, A. S. White, and B. Van Durme, “Reading the manual: event extraction as definition comprehension,” in *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pp. 74–83, November 2020.

[27] X. Du, A. Rush, and C. Cardie, “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*, pp. 634–644, Kyiv, April 2021.