# ODEE: A One-Stage Object Detection Framework for Overlapping and Nested Event Extraction

**Jinzhong Ning**, **Zhihao Yang***, **Zhizheng Wang**, **Yuanyuan Sun** and **Hongfei Lin**
Dalian University of Technology, Dalian, China
{Jinzhong_Ning, wzz_dllg}@mail.dlut.edu.cn, {yangzh, syuan, hflin}@dlut.edu.cn

## Abstract

The task of extracting overlapping and nested events has received significant attention in recent times, as prior research has primarily focused on extracting flat events, overlooking the intricacies of overlapping and nested occurrences. In this work, we present a new approach to Event Extraction (EE) by reformulating it as an object detection task on a table of token pairs. Our proposed one-stage event extractor, called ODEE, can handle overlapping and nested events. The model is designed with a vertex-based tagging scheme and two auxiliary tasks of predicting the spans and types of event trigger words and argument entities, leveraging the full span information of event elements. Furthermore, in the training stage, we introduce a negative sampling method for table cells to address the imbalance problem of positive and negative table cell tags, meanwhile improving computational efficiency. Empirical evaluations demonstrate that ODEE achieves the state-of-the-art performance on three benchmarks for overlapping and nested EE (i.e., FewFC, Genia11, and Genia13). Furthermore, ODEE outperforms current state-of-the-art methods in terms of both number of parameters and inference speed, indicating its high computational efficiency. To facilitate future research in this area, the codes are publicly available at https://github.com/NingJinzhong/ODEE.

## 1 Introduction

Event Extraction (EE), a vital and intricate task within the realm of Information Extraction (IE), endeavors to identify event triggers of specific types and their corresponding arguments. As a case in point, the `Gene Expression` event depicted in Figure 1(b) contains a trigger "expression" and the `Theme` argument "ICAM-1".

Traditionally, EE has been approached as a sequence labeling task[Chen *et al.*, 2015; Nguyen *et al.*, 2016; Liu *et al.*, 2018; Yang *et al.*, 2019], with the assumption that event mentions do not overlap. However, these methods fail to account for complex, irregular EE scenarios, such as overlapped [Sheng *et al.*, 2021] and nested [Cao *et al.*, 2022] EE. In reality, events frequently appear in sentences in a complicated manner, with triggers and arguments potentially overlapping within a single sentence. As depicted in Figure 1(a), there are two events, `Investment` and `Share Transfer`, that overlap and share the same trigger word "acquired" and `Subject` argument words "Grand Pharma". Figure 1(b) shows an example of nested events, where the trigger word "reduce" of the `Gene Expression` event also serves as the `Theme` argument in another event, `Positive Regulation`. In this study, we focus on a challenging and realistic problem in event extraction: *Overlapping and Nested Event Extraction*.

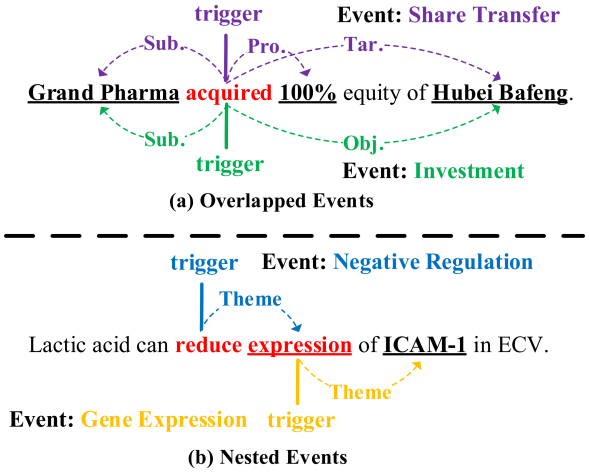


Figure 1: Examples that showcase two types of events, including overlapped events (a) and nested events (b). Different event mentions are highlighted in distinct colors. The triggers are highlighted in red, while the arguments involved in the event are underlined. "Sub.", "Tar.", "Obj." and "Pro." are the abbreviations of "Subject", "Target", "Object" and "Proportion", respectively.

Previously, overlapping and nested event extraction has been approached using pipeline-based methods that extract event triggers and arguments in a series of successive stages [Yang *et al.*, 2019; Li *et al.*, 2020] or that consecutively perform event type detection, trigger extraction, and argument extraction [Sheng *et al.*, 2021]. The main issue with this ap-

* Corresponding Author

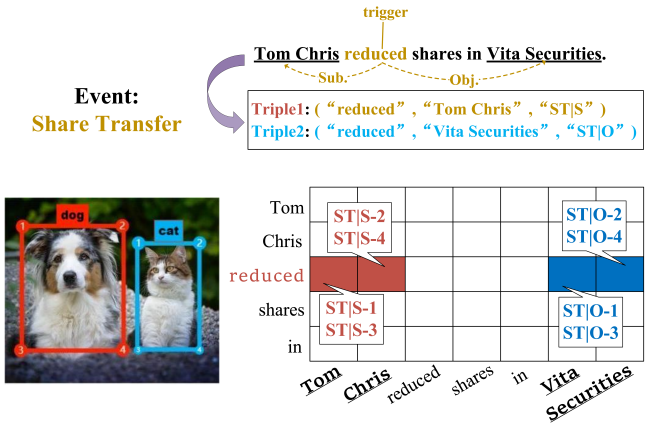


Figure 2: A comparison is made between object detection and event extraction based on the table-filling method. It should be noted that the table cells (representing pairs of tokens), the table itself, and the regions of the table occupied by triples are aligned with pixels, the image, and objects on the visual side, respectively. This alignment allows for a correspondence between the representations in the table and the visual elements in the image. "ST|S" represents the "Share Transfer" event type and the "Subject" argument role, while "ST|O" represents the "Share Transfer" event type and the "Object" argument role.

proach is that errors can propagate from one stage to the next, as the latter stage relies on the results of the former stage. Recently, the table-filling method OneEE [Cao *et al.*, 2022] has introduced a novel tagging scheme that converts the extraction of overlapping and nested events into the recognition of word-word relations.

While CasEE [Sheng *et al.*, 2021] and OneEE have achieved significant success as the current state-of-the-art methods for overlapping and nested event extraction, they still have the following limitations: 1) The span information of trigger words and argument entities in OneEE is not fully exploited due to its table-filling tagging scheme. 2) The incorporation of event type information in CasEE and OneEE enhances their performance, but the diverse representation of text (derived from pretrained language models) and event types (initialized randomly) impairs the interaction between them. 3) OneEE, like other table-filling methods [Li *et al.*, 2022; Shang *et al.*, 2022] in information extraction, also struggles with an imbalance between positive and negative table cell tags.

As shown in Figure 2, we transform the EE task into a triple extraction problem, referred to as the relational triple extraction of trigger and argument (RTE-TA) task in this paper, in the form `(trigger,argument,trigger type|argument role)`. We noticed a strong similarity between the table-filling-based RTE-TA task and the Object Detection (OD) task in computer vision. As depicted in Figure 2, both tasks require the identification of Regions of Interest (ROIs) within a two-dimensional array of pixels or table cells. Further, inspired by the keypoint-based one-stage object detection methods [Duan *et al.*, 2019; Law and Deng, 2018; Zhou *et al.*, 2019], we propose a one-stage **O**bject **D**etection framework for **E**vent **E**xtraction (**ODEE**) to address the overlapping and nested event extraction problem.

The primary contributions of this research, along with the point-by-point solution of our proposed method, address the three issues mentioned above in the current state-of-the-art methods as follows:

- ODEE directly predicts bounding boxes through the identification and grouping of four vertices of each Region of Interest as shown in Figure 2. By using vertex-based bounding box detection and type and span prediction of trigger words and argument entities, our method fully utilizes the span information of trigger words and argument entities, in contrast to the existing table-filling method OneEE.
- We map candidate event types to natural language text based on their semantic definitions, and then combine them with sentences into a continuous sequence. We then use a pretrained transformer language model (PLM) to encode the sequence, and obtain a unified representation of event types and text through the interaction of the two in the transformer blocks of the PLM.
- During the training phase, we incorporate a negative sampling approach for table cells as a means of enhancing the training efficiency and mitigating the imbalanced distribution of positive and negative table cell tags.

## 2 Related Work

### 2.1 Event Extraction

Event extraction (EE) is a crucial and intricate challenge in the realm of information extraction. Traditional approaches to event extraction frequently involve formulating it as a sequence labeling task, in which each token in the text is assigned a single label using a tagging scheme such as BIO. Representative models, such as CNN [Chen *et al.*, 2015], RNN [Nguyen *et al.*, 2016], attention-based GCN [Liu *et al.*, 2018], are utilized to model the dependency information in the text. However, these methods are not able to handle event extraction tasks that involve overlapping and nested events. Recently, overlapping and nested event extraction has received widespread attention due to its challenging and practical nature. Early methods [Yang *et al.*, 2019; Li *et al.*, 2020; Sheng *et al.*, 2021] used a pipeline approach that sequentially cascades multiple modules to extract nested and overlapping entities, resulting in error accumulation. Recently, the OneEE method [Cao *et al.*, 2022] based on table-filling achieved one stage extraction of overlapping and nested events by transforming the EE task into a word-word relation recognition problem.

### 2.2 Table-filling Information Extraction Method

The task of information extraction can be transformed into word-word relation prediction problem using a table-filling method, enabling the extraction of information to be performed in a single stage. Recently, the table-filling method has gained widespread use in a variety of information extraction tasks, including opinion mining [Wu *et al.*, 2020], relation extraction [Shang *et al.*, 2022], and named entity recognition [Li *et al.*, 2022]. This method is characterized by its

ability to represent relations between tokens, which makes it particularly well-suited to these tasks. Recently, the table-filling method has also been applied to tasks involving overlapping and nested event extraction [Cao *et al.*, 2022]. In contrast to this approach, we propose a novel tagging scheme from a unique perspective of object detection. Additionally, for the issue of imbalanced positive and negative table cell tags which exists in most table-filling methods , we introduce a negative sampling strategy for table cells to alleviate this problem.

### 2.3 Object Detection (OD)

Object detection, a task in computer vision which involves locating and identifying objects of interest within natural images, has garnered significant attention in recent years. While two-stage object detection models such as R-CNN [Girshick *et al.*, 2014], Faster-RCNN [Ren *et al.*, 2015], and Mask-RCNN [He *et al.*, 2017] have achieved notable success, one-stage models like YOLO [Redmon *et al.*, 2016], SSD [Liu *et al.*, 2016] and FCOS [Tian *et al.*, 2019] have gained popularity due to their real-time performance capabilities. Our approach is inspired by keypoint-based one-stage object detection methods [Duan *et al.*, 2019; Law and Deng, 2018; Zhou *et al.*, 2019]. Shen *et al.*[2022] also proposed a two-stage detector which considers the nested named entity recognition task as an object detection task. In contrast to this two-stage nested NER detector, our proposed one-stage detector, ODEE, demonstrates improvements in both performance and computational efficiency for the task of overlapping and nested event extraction.

## 3 Reformulation of EE Task

### 3.1 Task Definition

The objective of event extraction is to detect and extract event triggers and their related arguments, which may exhibit overlapping or nested relationships. To address this complexity, we propose a solution of transforming the EE task, characterized by overlapping and nested triggers and arguments, into the task of RTE-TA (as outlined in the Section 1).

The RTE-TA task involves identifying a set of all $N$ potential relational triples of trigger and argument, referred to as TAR triples, from a given sentence. The sentence, represented as a sequence of words $S = \{w_1, w_2, \cdots, w_L\}$, where $L$ is the length of the sentence $S$. Each TAR triple $\Gamma_i = (t_i, a_i, r_i)$ contains a trigger $t_i$, an argument $a_i$, and a relation $r_i$ between $t_i$ and $a_i$ which is a combination of event type $type_i$ and argument role $role_i$ represented as $type_i\,|role_i$ as illustrated in Figure 3. The event type and argument role are chosen from pre-defined sets of candidate event types $\mathcal{E}$ and candidate argument roles $\Upsilon$, respectively.

### 3.2 OD-style TAR Triple Tagging Scheme

The Figure 2 and Figure 3 illustrate that the trigger and argument of a TAR triple can be represented by a rectangular region in a table of token pair representations. Previous research [Cao *et al.*, 2022] has demonstrated that the span and type of elements in an event can be inferred from the relations of bounding tokens and that one-stage object detection can be achieved through the identification and grouping of key points within the bounding box [Duan *et al.*, 2019; Law and Deng, 2018; Zhou *et al.*, 2019]. Inspired by these ideas, the proposed approach utilizes the four vertices of the rectangular region enclosed by the trigger and argument in a TAR triple in a relation-specific table to determine the relevant region of the TAR triple. These vertices include the upper left (**UL**) vertex, which indicates the start position of both the trigger and argument, the upper right (**UR**) vertex, which indicates the start position of the trigger and the end position of the argument, the lower left (**LL**) vertex, which indicates the end position of the trigger and the start position of the argument, and the lower right (**LR**) vertex, which indicates the end position of both the trigger and argument.

## 4 Methodology

In this section, we provide a detailed explanation of the implementation of ODEE, the overall structure of which is illustrated in Figure 3.

### 4.1 Unified Encoder of Event Types and Sentence

The process begins by utilizing a verbalizer to convert each event type within the schema into a sequence of natural language words, which are selected manually. Due to the intricacies of the event types, it may be necessary for a certain event type to be represented by multiple words. For instance, the `Share Transfer` event type depicted in Figure 3 is expressed as a sequence composed of the words "share" and "transfer". The resulting concatenation of the input sentence $S$ and the text sequence of event types, represented as $ET = [et_1, \cdots, et_M]$, is then inputted into a pre-trained BERT [Kenton and Toutanova, 2019] encoder (as illustrated in Figure 3):

$$input = concat\,(ET, S) \tag{1}$$

$$[H_{et}, H_w] = BERT\,(input) \tag{2}$$

The output of BERT encoder includes representations of $ET$ and $S$, which are denoted as $H_{et} = [het_1, \cdots, het_M] \in \mathbb{R}^{M\times 768}$ and $H_w = [hw_1, \cdots, hw_L] \in \mathbb{R}^{L\times 768}$, respectively, where $M$ is the length of $ET$.

The multi-layer Transformer blocks in BERT possess advanced capabilities for modeling global dependencies within sequences. The steps described above facilitate the interaction and representation of event types and the input sentence within BERT in a unified semantic space.

### 4.2 Span-aware Word Representation

In order to fully exploit the span information of event elements, we propose the incorporation of two auxiliary tasks: predicting the span and type of both trigger words and argument entities. These tasks serve as guidance for BERT to obtain span-aware word representations.

For the task of predicting the type and span of trigger words, we formulate it as a classification problem of identifying the relationship between words in $ET$ and words in

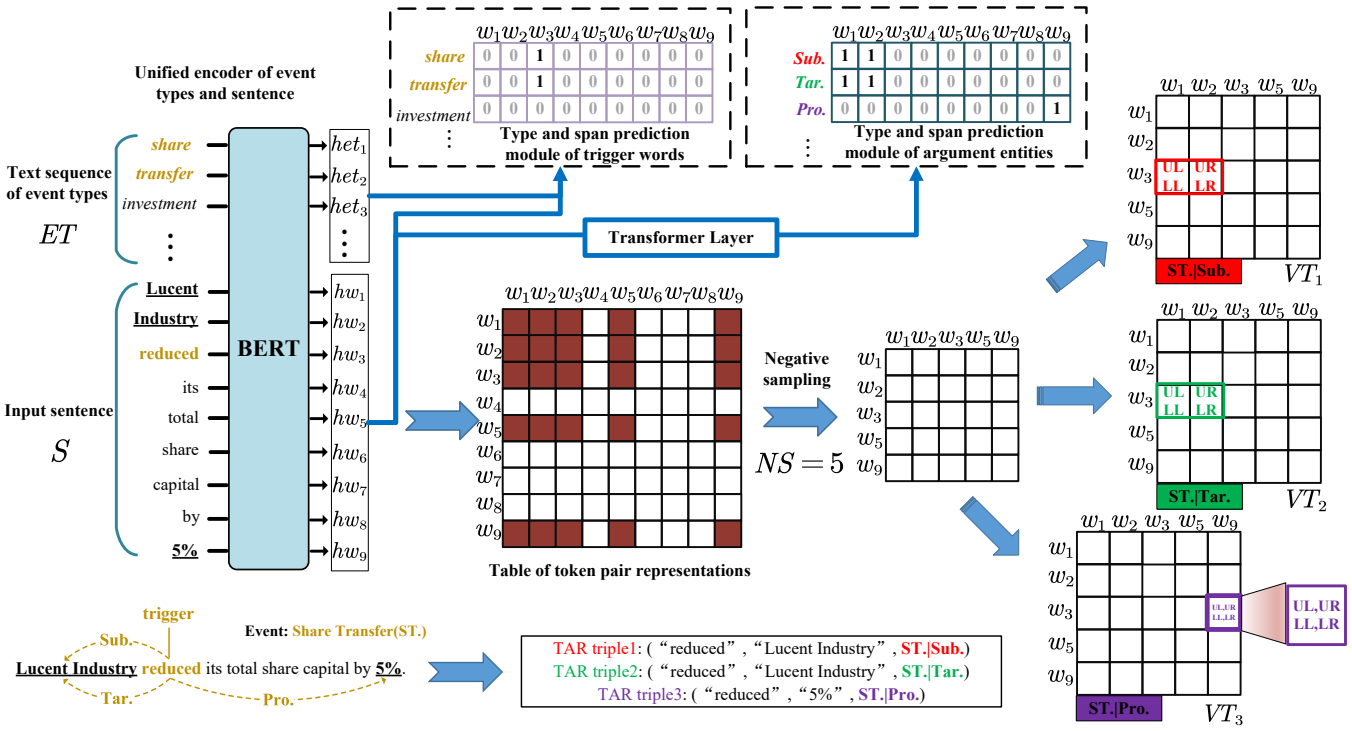


Figure 3: The overall architecture of ODEE. The UL, UR, LL and LR denote the upper left vertex, upper right vertex, lower left vertex and lower right vertex, respectively. In this figure, the total number of tokens after negative sampling, $NS$, is 5.

$S$. The representation of token pair, consisting of $et_i \in ET$ and $w_j \in S$, is calculated as:

$$h_{(et_i,w_j)} = ReLU\left(W_{(et,w)}\left[h_{et_i}; h_{w_j}\right] + b_{(et,w)}\right) \quad (3)$$

where $ReLU(\cdot)$ is the ReLU [Agarap, 2018] activation function, $[\cdot;\cdot]$ is the concatenation operators, $W_{(et,w)} \in \mathbb{R}^{d_e \times 1536}$ and $b_{(et,w)} \in \mathbb{R}^{d_e}$ are learnable parameters.

The prediction score $p_{(et_i,w_j)}$ for token pair $(et_i, w_j)$ is calculated as follows:

$$p_{(et_i,w_j)} = \sigma\left(W_{etw} h_{(et_i,w_j)} + b_{etw}\right) \quad (4)$$

where $\sigma$ denotes sigmoid function, $W_{etw} \in \mathbb{R}^{1 \times d_e}$ and $b_{etw} \in \mathbb{R}^1$ are learnable parameters. If $w_i$ is found to be within the span of a trigger word $trigger_m$, and if $trigger_m$ and $et_j$ are of the same event type, then the ground truth value of $p_{(et_i,w_j)}$ is assigned as 1, as illustrated in Figure 3. Conversely, if these conditions are not met, the ground truth value of $p_{(et_i,w_j)}$ is assigned as 0.

For the task of predicting the span and type of argument entities, we construct it as a role-specific span tagging task of $S$. We first employ a Transformer [Vaswani *et al.*, 2017] layer to obtain the argument-specific hidden representations:

$$H_w^{(T)} = Transformer\left(H_w\right) = \left[h_{w_1}^{(T)}, \cdots, h_{w_L}^{(T)}\right] \quad (5)$$

For each word $w_i$ in the set $S$, the probability score $p_{w_i}^{(role_m)}$ of it being within the span of an entity with a role $role_m$ in the set $\Upsilon$ is computed using the following equation:

$$p_{w_i}^{(role_m)} = \sigma(W_{role_m} hw_i^{(T)} + b_{role_m}) \quad (6)$$

where $\sigma$ represents the sigmoid function, and $W_{role_m} \in \mathbb{R}^{1 \times d_e}$ and $b_{role_m} \in \mathbb{R}^1$ are the learnable parameters.

### 4.3 TAR Triple Region Detector

**Negative Sampling Method for Table Cells**

Existing table-filling methods for information extraction (such as discussed in Section 2.2) generate all possible token pairs from the input text, resulting in a high computational cost, particularly when the input text is long. Additionally, considering all tokens leads to an imbalance in the positive and negative labels for table cells, causing the model to have a bias towards predicting negative labels. We propose a negative sampling strategy for table cells to alleviate these issues. For each word $w_i$ in sentence $S$, if it is within the span of any trigger or argument entity, we define it as a positive token, otherwise, we define it as a negative token.

For the input sentence $S$, we obtain a sampled token sequence $\widetilde{S}$ through negative sampling:

$$\widetilde{S} = NegSample\left(S, NS\right) = [\tilde{w}_1, \cdots, \tilde{w}_{NS}] \quad (7)$$

Where the $NegSample(S, NS)$ denotes the operation of preserving all the positive tokens in $S$, randomly sampling the negative tokens, and ensuring that the total number of positive and negative tokens is $NS$. Then, we generate the table of token pair representations by only using the tokens in the sequence $\widetilde{S}$.

**Relation-specific Vertice Regressor**

For a given token pair $(\widetilde{w}_i, \widetilde{w}_j)$, the representation of the token pair $h_{(\widetilde{w}_i,\widetilde{w}_j)}$ is computed as follows:

$$h_{(\widetilde{w}_i,\widetilde{w}_j)} = ReLU\left(W_{ww}\left[h_{\widetilde{w}_i}; h_{\widetilde{w}_j}\right] + b_{ww}\right) \quad (8)$$

Where $W_{ww} \in \mathbb{R}^{d_e \times 1536}$ and $b_{ww} \in \mathbb{R}^{d_e}$ are learnable parameters, and $h_{\widetilde{w}i}$ and $h\widetilde{w}_j$ are the BERT representation of tokens $\widetilde{w}_i$ and $\widetilde{w}_j$ respectively.

Then the probability scores of each token pair $(\widetilde{w}_i, \widetilde{w}_j)$ for a certain vertex tag type $v_k$ (as defined in Section 3.2) under the specific relation $r_m$ (as defined in Section 3.1) between trigger and argument are calculated as follows:

$$S^{(v_k)}_{(i,j,r_m)} = \sigma\left(W^{(v_k)}_{r_m} h_{(\widetilde{w}_i,\widetilde{w}_j)} + b^{(v_k)}_{r_m}\right) \quad (9)$$

where $W^{(v_k)}_{r_m} \in \mathbb{R}^{1\times d_e}$ and $b^{(v_k)}_{r_m} \in \mathbb{R}^1$ are learnable parameters, $S^{(v_k)}_{(i,j,r_m)} \in \mathbb{R}$ is the probability score indicating the probability that the token pair $(\widetilde{w}_i, \widetilde{w}_j)$ is tagged as vertex tag type $v_k$. The candidate types for $v_k$ are UL, UR, LL and LR.

## 4.4 Loss Function

The objective function of ODEE is designed based on the BCE (Binary Cross Entropy) loss, taking into account the task of vertices tagging of token pairs, as well as the auxiliary tasks of predicting the type and span of trigger words and argument entities. The BCE loss function can be formulated as:

$$BCE\left(gt, p\right) = gt\log\left(p\right) + \left(1 - gt\right)\log\left(1 - p\right) \quad (10)$$

The objective function of ODEE is defined as follows:

$$L_{ver} = \frac{\sum_{v_k\in\Psi}\sum_{m=1}^{ReN}\sum_{i=1}^{NS}\sum_{j=1}^{NS} bce^{(v_k)}_{(i,j,r_m)}}{4 \times ReN \times NS \times NS} \quad (11)$$

$$bce^{(v_k)}_{(i,j,r_m)} = BCE\left(gt^{(v_k)}_{(i,j,r_m)}, S^{(v_k)}_{(i,j,r_m)}\right) \quad (12)$$

$$L_{tri} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} BCE\left(gt_{(et_i,w_j)}, p_{(et_i,w_j)}\right)}{M \times N} \quad (13)$$

$$L_{arg} = \frac{\sum_{i=1}^{RoN}\sum_{j=1}^{N} BCE\left(gt^{(role_i)}_{w_j}, p^{(role_i)}_{w_j}\right)}{RoN \times N} \quad (14)$$

$$L_{total} = L_{ver} + \lambda L_{tri} + \gamma L_{arg} \quad (15)$$

where $gt^{(v_k)}_{(i,j,r_m)}$, $gt_{(et_i,w_j)}$ and $gt^{(role_i)}_{w_j}$ are ground truth value of $S^{(v_k)}_{(i,j,r_m)}$, $p_{(et_i,w_j)}$ and $p^{(role_i)}_{w_j}$, respectively. $\Psi$ is a collection of all vertex types, consisting of UL, UR, LL and LR. $ReN$ is the total number of relationship types between trigger words and arguments in the schema. $RoN$ is the number of roles for argument in the schema. $\lambda$ and $\gamma$ are the tuning factors of the loss function.

## 4.5 Decoding Method

For each sentence, the tagging results of all sampled token pairs for different vertices under the relation $r_m$ are stored into a matrix called the vertice tagging matrix $VT_m \in \mathbb{R}^{NS\times NS\times 4}$ (as shown in Figure 3). To decode the TAR triples contained in each sentence under the relation $r_m$, we propose a bidirectional decoding method that decodes the triples in two diagonal directions of the object region. The triples are decoded along two decoding directions in parallel: *decoding direction 1* (UL→UR→LR) and *decoding direction 2* (LR→LL→UL). Specifically, for the *decoding direction 1*, we first enumerate all token pairs located at the UL vertices, and then for each UL token pair search for the nearest following token pair located at the UR vertex. Next, for each UR token pair, we search for the nearest following token pair located at the LR vertex. As a result, the tokens between vertices UL and UR form the argument entity, and the tokens between vertices UR and LR form the trigger. Similarly, the meaning of *decoding direction 2* (LR→LL→UL) is similar to that of *decoding direction 1* (UL→UR→LR). Finally, the TAR triples decoded by both *decoding direction 1* and *decoding direction 2* are consolidated in the final decoding results to ensure that both overlapping and nested TAR triples under the relation $r_m$ can be accurately identified.

| | | #Sent. | #Events | #Ovlp. | #Nest. |
|---|---|---|---|---|---|
| FewFC | train | 7,185 | 10,227 | 1,560 | - |
| | dev | 899 | 1,281 | 205 | - |
| | test | 898 | 1,332 | 210 | - |
| Genia11 | train | 8,730 | 6,401 | 954 | 1,628 |
| | dev | 1,091 | 824 | 121 | 199 |
| | test | 1,092 | 775 | 125 | 197 |
| Genia13 | train | 4,000 | 2,743 | 347 | 784 |
| | dev | 500 | 352 | 44 | 100 |
| | test | 500 | 320 | 42 | 88 |

Table 1: Statistics of three datasets. "Ovlp." and "Nest." denote the sentences with overlapping and nested events, respectively.

# 5 Experiments

## 5.1 Experiments Setting

**Datasets**

In this study, we evaluated the performance of our proposed method on three benchmark datasets for overlapping and nested event extraction. Specifically, we utilized the FewFC dataset [Zhou *et al.*, 2021], a Chinese financial event extraction benchmark, which annotates 10 event types and 18 argument role classes, with a significant proportion of sentences containing overlapped events (22%). Additionally, we also conducted experiments on two biomedical event extraction datasets, namely Genia11 [Kim *et al.*, 2011] and Genia13 [Kim *et al.*, 2013], which contain a significant proportion of nested events (18%). Genia11 contains 9 event types and 10 argument role classes, and Genia13 contains 13 event types and 7 argument role classes. The train/dev/test split for these datasets is in accordance with previous work [Sheng *et al.*, 2021; Cao *et al.*, 2022], with an 8:1:1 ratio.

**Implementation Details**

In this study, we employed the Chinese-BERT-Base[1] model for the FewFC dataset and BioBERT[2] [Lee *et al.*, 2020] for the Genia11 and Genia13 datasets. The optimization algorithm used was Adam, with a learning rate of 3e-5. The batch

[1]https://huggingface.co/bert-base-chinese

[2]https://github.com/dmis-lab/biobert

| | | TI(%) | | | TC(%) | | | AI(%) | | | AC(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P(%) | R(%) | F1(%) | P(%) | R(%) | F1(%) | P(%) | R(%) | F1(%) | P(%) | R(%) | F1(%) |
| **Flat EE methods** | 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 |
| **Multi-stage** | 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 |
| **One-stage** | OneEE | 88.7 | 88.7 | 88.7 | 79.1 | 80.3 | 79.7 | 75.4 | 77.0 | 76.2 | 74.0 | 72.9 | 73.4 |
| | ODEE(ours) | 88.9 | **93.6** | **91.2** | 80.7 | **85.0** | **82.8** | **77.5** | **80.7** | **79.0** | **75.7** | **78.8** | **77.2** |

Table 2: Results of event extraction on FewFC dataset. The results are the median values obtained from 5 runs of our model with different random seeds. The best results are highlighted in bold.

| • **Genia11** | TI(%) | TC(%) | AI(%) | AC(%) |
|---|---|---|---|---|
| BERT-softmax | 67.8 | 64.4 | 57.4 | 56.0 |
| BERT-CRF | 68.3 | 64.8 | 58.3 | 56.9 |
| BERT-CRF-joint | 67.0 | 64.1 | 60.2 | 58.1 |
| PLMEE | 67.3 | 65.5 | 60.7 | 59.4 |
| CasEE | 70.0 | 67.0 | 62.0 | 60.4 |
| OneEE | 71.5 | 69.5 | 65.9 | 62.5 |
| ODEE | **76.2** | **73.3** | **71.0** | **69.1** |
| • **Genia13** | TI(%) | TC(%) | AI(%) | AC(%) |
| BERT-softmax | 77.4 | 75.9 | 69.9 | 67.7 |
| BERT-CRF | 78.8 | 77.4 | 70.1 | 68.2 |
| BERT-CRF-joint | 77.6 | 75.7 | 71.9 | 68.2 |
| PLMEE | 79.3 | 78.3 | 72.1 | 70.7 |
| CasEE | 80.5 | 78.5 | 73.7 | 71.9 |
| OneEE | 81.9 | 80.8 | 76.8 | 72.7 |
| ODEE | **83.8** | **81.5** | **79.8** | **79.3** |

Table 3: Results (F1 score) on Genia11 and Genia13 datasets.

size was set to 8 and the hidden size of the model $d_e$ was set to 768. All the hyper-parameters were tuned on the development set. Additionally, the tuning factors of the loss function, $\lambda$ and $\gamma$, were set to 0.1 and 0.01, respectively. And the number of negative samples for token $NS$ is set to 0.4 times the length of the input sentence $L$. The attention heads number of the Transformer layer is set to 8.

**Evaluation Metrics & Baselines**

In this study, we evaluated the performance of our proposed method using traditional criteria established in previous work [Chen *et al.*, 2015; Sheng *et al.*, 2021; Cao *et al.*, 2022]. We used the following evaluation metrics: 1) Trigger Identification (**TI**), where a trigger is considered correctly identified if the predicted trigger span matches the golden label; 2) Trigger Classification (**TC**), where a trigger is considered correctly classified if it is correctly identified and assigned the correct type; 3) Argument Identification (**AI**), where an argument is considered correctly identified if its event type is correctly recognized and the predicted argument span matches the golden label; 4) Argument Classification (**AC**), where an argument is considered correctly classified if it is correctly identified and the predicted role matches any of the golden labels. We report Precision (P), Recall (R), and F-measure (F1) for each of these four metrics.

In line with previous work [Sheng *et al.*, 2021; Cao *et al.*, 2022], we selected three types of baseline models for comparison in this study: 1) **Sequence Labeling Methods for Flat Event Extraction** (BERT-softmax, BERT-CRF, and BERT-CRF-joint), 2) **Multi-stage Methods for Overlapping and Nested Event Extraction** (PLMEE [Yang *et al.*, 2019], MQAEE [Li *et al.*, 2020] and CasEE [Sheng *et al.*, 2021]) and 3) **One-stage Methods Based on Table-filling for Overlapping and Nested Event Extraction** (OneEE [Cao *et al.*, 2022]).

| | TI(%) | TC(%) | AI(%) | AC(%) |
|---|---|---|---|---|
| OneEE | 88.7 | 79.7 | 76.2 | 73.4 |
| ODEE | 91.2 | 82.8 | 79.0 | 77.2 |
| w/o TTSP | 90.0 | 81.3 | 78.4 | 76.5 |
| w/o ATSP | 90.6 | 81.9 | 77.8 | 76.3 |
| w/o NST | 90.7 | 82.0 | 78.3 | 76.3 |

Table 4: An ablation study of the proposed model. F1 scores were evaluated on the testset of FewFC. TTSP and ATSP refer to the trigger word type and span prediction task and the argument type and span prediction task, respectively. NST refers to the negative sampling method for tokens.

## 5.2 Experiments Results

**Main Results**

In our study, we compared our proposed ODEE model with several strong baseline models and the results are reported in Table 2 and Table 3. It is evident that ODEE outperforms all baselines and achieves the state-of-the-art performance in terms of F1 scores on all datasets. Through these experimental results, we can conclude that:

1) Comparing with the sequence labeling methods that can only extract flat events, our proposed ODEE model achieved significant improvements in recall and F1 scores, indicating the effectiveness of our model in extracting overlapping and nested events.

2) Comparing with multi-stage methods, our proposed one-stage method still achieved notable improvements in recall and F1 scores, demonstrating the advantage of reducing error propagation in the extraction of nested and overlapping events.

3) When compared with the current state-of-the-art one-stage table-filling method, OneEE, ODEE still achieved a notable improvement in F1 scores. Specifically, the average growth rate on three datasets for TC is 3.41% and for AC is 8.27%, respectively. This highlights the effectiveness of ODEE in utilizing span information of trigger words and argument entities. Additionally, the high average AC growth rate also indicates that our method effectively models the re-

| Model | Stage | #Param. | Inference(sent/s) | GPU Memery(GB) | TC(F1%) | AC(F1%) |
|---|---|---|---|---|---|---|
| CasEE | three | 120.7M(18.4M$^{\emptyset}$) | 59.3$^{\jmath}$(62.3$^{\ell}$) | **12.5** | 78.2 | 71.4 |
| OneEE | one | 114.2M(11.9M$^{\emptyset}$) | **(186.5$^{\ell}$) | ** | 79.7 | 73.4 |
| ODEE | one | **110.4M(8.1M$^{\emptyset}$)** | **204.3** | 14.7 | **82.8** | **77.2** |
| ODEE$^{\dagger}$ | one | **110.4M(8.1M$^{\emptyset}$)** | **204.3** | 23.8 | 82.0 | 76.3 |

Table 5: Comparison of the efficiency with state-of-the-art methods, CasEE and OneEE, on the FewFC dataset. The superscript † indicates an ODEE without negative sampling of tokens. The subscript $\emptyset$ denotes the number of parameters in the model that do not include the BERT component. The subscript $\jmath$ indicates the results obtained from the publicly available implementation of the model. The subscript $\ell$ indicates the results obtained from original paper. The symbol ** denotes that the results for this model are currently unavailable due to the unavailability of its publicly available implementation, but will be included once it becomes accessible.

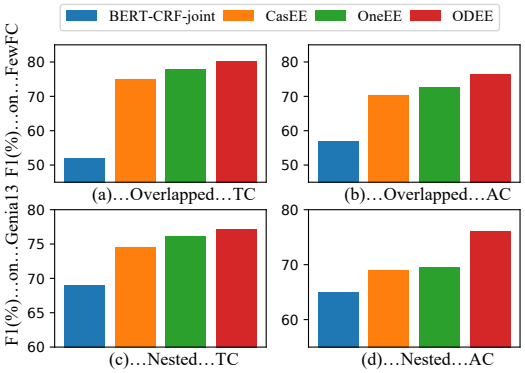


Figure 4: The results of our model's performance in extracting both overlapped triggers (a) and arguments (b) from the FewFC dataset, as well as nested triggers (c) and arguments (d) from the Genia13 dataset are presented. It is important to note that only sentences containing at least one such event were used in these evaluations. Additionally, the results of the comparison experiment are taken from previous research.

lationship between triggers and arguments, leading to superior performance in argument extraction.

### Ablation Study

In order to evaluate the effectiveness of each component in our model, we conducted ablation experiments on the FewFC dataset, and the results are reported in Table 4. Firstly, for the two auxiliary tasks of predicting the types and spans of trigger words and argument entities, removing either one of them will result in a decrease in model performance, indicating that both of them contribute to improving the model's ability to perceive the span information of event elements. Specifically, our experimental results reveal that the auxiliary task of TTSP has a more significant impact on the performance of the model in terms of TC, while the auxiliary task of ATSP has a more significant impact on the performance of the model in terms of AC. Secondly, removing the negative sampling method for tokens also leads to a significant decrease in model performance, indicating that this method effectively alleviates the imbalance of positive and negative labels for token pairs.

### Results of Overlapped Events and Nested Events

In order to assess the performance of our proposed model in identifying overlapping and nested event mentions, we present results on sentences containing at least one overlapping event in the FewFC dataset, and sentences containing at least one nested event in the Genia13 dataset. Figure 4 shows the results for trigger classification (TC) and argument classification (AC) on overlapping and nested sentences in the test set. The results demonstrate that our method outperforms other methods in detecting overlapping and nested events. The superior performance of ODEE in recognizing overlapping and nested event mentions is primarily attributed to two factors. Firstly, the single-step event extraction process effectively reduces the accumulation of errors. Secondly, our model is able to more fully perceive and utilize the span information of event triggers and arguments.

### Analysis on Model Efficiency

In order to evaluate the efficiency of our proposed OD-RTE model in comparison to the state-of-the-art methods, CasEE and OneEE, we conduct experiments and analyze the results from three perspectives: number of parameters, inference speed and GPU memory usage in training stage. The results of the experiments are presented in Table 5. To ensure fairness, the results for the above mentioned efficiency metrics are obtained using the same model configurations for all methods. Our model, ODEE, achieves the state-of-the-art performance while utilizing the minimal number of parameters due to the optimizations in its design. Additionally, our model demonstrates the fastest inference speed, indicating high parallelism on GPU. Furthermore, the introduction of negative sampling for tokens not only alleviates the imbalance of positive and negative labels, but also significantly improves performance, particularly in terms of GPU memory usage and inference speed.

## 6 Conclusion

In this paper, we present a novel one-stage object detection framework (ODEE) to address the problem of overlapping and nested event extraction. Our approach utilizes a vertex-based tagging scheme in combination with the prediction of trigger and argument types and spans to better exploit the event element's contextual information. Furthermore, we leverage BERT to achieve unified interaction and representation of sentence and event types. Additionally, the introduction of token negative sampling not only alleviates the issue of imbalanced labels for token pairs but also significantly improves the model's performance. Experimental results demonstrate that our proposed model not only achieves state-of-the-art performance on three datasets but also achieves competitive computational efficiency.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62276043) and the Fundamental Research Funds for the Central Universities (No. DUT22ZD205).

## References

[Agarap, 2018] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[Cao *et al.*, 2022] Hu Cao, Jingye Li, Fangfang Su, Fei Li, Hao Fei, Shengqiong Wu, Bobo Li, Liang Zhao, and Donghong Ji. Oneee: A one-stage framework for fast overlapping and nested event extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1953–1964, 2022.

[Chen *et al.*, 2015] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 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 (Volume 1: Long Papers)*, pages 167–176, 2015.

[Duan *et al.*, 2019] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577. IEEE Computer Society, 2019.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[Kim *et al.*, 2011] Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. Overview of genia event task in bionlp shared task 2011. In *Proceedings of BioNLP shared task 2011 workshop*, pages 7–15, 2011.

[Kim *et al.*, 2013] Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. The genia event extraction shared task, 2013 edition-overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15, 2013.

[Law and Deng, 2018] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[Lee *et al.*, 2020] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234–1240, 2020.

[Li *et al.*, 2020] Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, 2020.

[Li *et al.*, 2022] Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973, 2022.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[Liu *et al.*, 2018] Xiao Liu, Zhunchen Luo, and He-Yan Huang. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, 2018.

[Nguyen *et al.*, 2016] Thien Huu Nguyen, Kyunghyun Cho, and Ralph 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: Human Language Technologies*, pages 300–309, 2016.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[Shang *et al.*, 2022] Yu-Ming Shang, Heyan Huang, and Xianling Mao. Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11285–11293, 2022.

[Shen *et al.*, 2021] Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, 2021.

[Sheng *et al.*, 2021] Jiawei Sheng, Shu Guo, Bowen Yu, Qian Li, Yiming Hei, Lihong Wang, Tingwen Liu, and

Hongbo Xu. Casee: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 164–174, 2021.

[Tian *et al.*, 2019] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wu *et al.*, 2020] Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585, 2020.

[Yang *et al.*, 2019] Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, 2019.

[Zhou *et al.*, 2019] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[Zhou *et al.*, 2021] Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14638–14646, 2021.