Complex & Intelligent Systems (2023) 9:5235–5250
https://doi.org/10.1007/s40747-023-01004-8

ORIGINAL ARTICLE

NEDORT: a novel and efficient approach to the data overlap problem in relational triples

Zhanjun Zhang[1] · Xiaoru Hu[2] · Haoyu Zhang[3] · Jie Liu[4,5]

Received: 11 August 2022 / Accepted: 7 February 2023 / Published online: 16 March 2023
© The Author(s) 2023

Abstract

Relation triple extraction is a combination of named entity recognition and relation prediction. Early works ignore the problem of data overlap when extracting triples, resulting in poor extraction performance. Subsequent works improve the capability of the model to extract overlapping triples through generative and extractive methods. These works achieve considerable performance but still suffer from some defects, such as poor extraction capability for individual triplets and inappropriate spatial distribution of the data. To solve the above problems, we perform sequence-to-matrix transformation and propose the NEDORT model. NEDORT predicts all subjects in the sentence and then completes the extraction of relation–object pairs. There are overlapping parts between relation–object pairs, so we conduct the conversion of sequence to matrix. We design the Differential Amplified Multi-head Attention method to extract subjects. This method highlights the locations of entities and captures sequence features from multiple dimensions. When performing the extraction of relation–object pairs, we fuse subject and sequence information through the Biaffine method and generate relation–sequence matrices. In addition, we design a multi-layer U-Net network to optimize the matrix representation and improve the extraction performance of the model. Experimental results on two public datasets show that our model outperforms other baseline models on triples of all categories

Keywords Relation triple extraction · Overlapping data · Differential amplified Multi-head attention · Biaffine · U-Net network

Introduction

Relation triple extraction[1] is a combination of named entity recognition [1,9,22,27] and relation prediction [13,18,35]. The previous works perform these two tasks separately [24,30], which is inconvenient and ineffective. Subsequently, the methods of joint extraction are proposed, bringing a breakthrough for the extraction of triples.

Early works on relation triple extraction first extract all entities in a sentence, and then predict the relations [14,20,50]. With the development of extraction methods [8,28,36], the performance of such models has been further improved. Although these models can complete the extraction of relational triples, they fail to deal with the problem of data overlap. As shown in Fig. 1, the relational triples are divided into three categories: Normal, SEO,[2] and EPO.[3] The

Zhanjun Zhang and Xiaoru Hu contributed equally to the work.

✉ Jie Liu
liujie@nudt.edu.cn

Zhanjun Zhang
zhangzhanjun@nudt.edu.cn

Xiaoru Hu
188201029@csu.edu.cn

Haoyu Zhang
zhanghaoyu10@nudt.edu.cn

1 College of Computer, National University of Defense Technology, Changsha, China

2 Department of Nephrology, Hunan Key Laboratory of Kidney Disease and Blood Purification, The Second Xiangya Hospital of Central South University, Changsha, China

3 Defense Innovation Institute, Beijing, China

4 Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha, China

5 Laboratory of Software Engineering for Complex Systems, National University of Defense Technology, Changsha, China

[1] The relational triple is expressed in the form of $(subject, relation, object)$.

[2] SEO: Single Entity Overlap.

[3] EPO: Entity Pair Overlap.

Normal category contains triples without overlapping patterns, while triples of the other two categories contain one and two overlapping entities, respectively. In addition, SEO and EPO overlapping patterns can appear simultaneously in a sentence, which brings significant challenges to the extraction of relational triples. The above models do not consider overlapping data and perform poorly on triples of SEO and EPO categories.

[47] points out the problem of data overlap for the first time and classifies the relational triples according to the overlapping patterns. To deal with overlapping data, [47] proposes a generative model, which extracts relational triples through the encoding-decoding structure. Subsequent works optimize on this basis and propose various generative methods[10,25,45]. Generative methods generate the entire triple by designing an end-to-end framework. The performance of the model can be improved by optimizing the encoder and decoder [7,10,46]. Generative models obtain the entire triple without considering the data overlap problem. The implementation of these methods is based on the extraordinary inference capability of the model, which limits the extraction effect of triples. The extractive method can also deal with overlapping data through ingenious framework design. These works divide the triple into several parts and complete the extraction through multiple steps[16,44]. The design of task division can deal with overlapping data more effectively. Extractive methods transform triple extraction into a classification problem, which is easier to implement. Recently, CGT [42] and GraphJoint [40] perform well on the extraction of relational triples. CGT improves the quality of generated triples by constructing negative samples and introducing the dynamic masking mechanism. GraphJoint proposes a relation-based triple extraction model, which predicts all relations in the sentence by graph neural network and then performs entity recognition on this basis. Although the above works achieve good performance in dealing with the problem of data overlap, there are still some defects. The encoder–decoder framework is designed to handle overlapping data, but it does not perform well in extracting individual triplets. The above methods extract triplets in one-dimensional space, which can only improve the inference capability of the model for overlapping data but cannot change the spatial distribution of data to eliminate the problem of data overlap fundamentally.

To solve the above problems, we perform the transformation from sequence to matrix (two-dimensional space) and propose the NEDORT model. NEDORT divides the extraction of relational triples into two steps: the first step extracts all subjects in the sentence, and the second step predicts relation–object pairs corresponding to the subject. The overlapping data are caused by expressing data of multidimensional space in one dimension. The first step does not involve overlapping phenomenon, and the extraction can be performed directly in the sequence. Numerous overlapping cases occur in relation–object pairs, so we convert the sequence to a matrix for extraction. There are no overlapping relation–object pairs in the matrix, so the problem of data overlap can be fundamentally eliminated. Moreover, both extraction steps perform binary classification tasks. There-



Fig. 1 Examples of Normal, SEO and EPO overlapping patterns. Subject, Object and Relation are marked in red, green and orange, respectively. The entities marked in purple are both Subject and Object. Overlapping triples are contained in the SEO and EPO categories

fore, NEDORT performs better than the encoder–decoder framework in extracting individual triplets.

When extracting subjects, this paper proposes the Differential Amplified Multi-head Attention method to highlight the locations of entities. The relation–object pairs to be extracted are subject-related, so we incorporate the relevant information of the subject into the subsequent extraction. After fetching the start and end positions of the subject, we generate its comprehensive representation through self-attention and apply it to the extraction of relation–object pairs. In this paper, the problem of data overlap is solved by matrix representation, so we design the Biaffine method to complete the conversion from sequence to matrix. This method fuses subject and sequence information and generates the relation–sequence matrix without overlapping data. Traditional sequence-based methods (Lstm, Transformer, etc.) are powerless to deal with matrix problems, so we design the U-Net network to optimize matrix features. U-Net realizes the combination of up-sampling and down-sampling blocks and the information interaction between different layers, which has significant advantages in extracting matrix features. Finally, we combine the extraction results of the two steps to construct relational triples.

We conduct experiments on two public datasets WebNLG and NYT. Experimental results show that our proposed NEDORT model outperforms the previous state-of-the-art models in the extraction of triples. Further analysis demonstrates that our model has advantages over other models in dealing with the problem of data overlap.

The main contributions of this paper are as follows:

1. We propose the sequence-to-matrix transformation method to eliminate the problem of data overlap.
2. We propose the Differential Amplified Multi-head Attention method to highlight the position of the subject, which is more conducive to the prediction of the start and end indices.
3. We propose the Biaffine method to generate a relation–sequence matrix and optimize it through the U-Net network. The optimized matrix contains more prosperous entity and relation features.

To show the paper organization more clearly, we make a summary according to the contents of each section [23]. "Introduction" summarizes the motivation and contribution of this paper. "Related work" describes the related works of this task. "Methods" introduces the implementation details of the proposed methods. "Experiments" presents the experimental results and conducts comparative analysis. "Conclusion" summarizes the work of this paper.

Related work

Relation triple extraction is a combination of named entity recognition [15,31] and relation prediction [5,30,43]. Early works extract all entities in a sentence and then predict the relation between them [2,20,32,48]. This is actually performing named entity recognition and relation prediction, but the two parts are jointly trained and share the underlying modules [26,50]. To enhance the link between entity recognition and relation prediction, models based on table filling are proposed [12,21]. These models construct a connection table between tokens, and define a new label to identify the relationship between tokens [19,37,38]. Constructing reasonable labels and improving the inference capability of the model are the keys to improving performance. The span-based method performs well on named entity recognition, so some works conduct the extraction of relational triples based on span [8,17,36]. To obtain better extraction performance, [52] adds new annotations on the basis of span. The new annotations related to entities are only used when predicting relations. Span-based methods have significant advantages in extracting triples. However, these method needs to traverse all the spans in the sequence and make predictions, which is computationally intensive. Despite the excellent performance of the above methods, the data overlap problem is not considered when performing relation triple extraction. Overlapping triples are abundant in this task and become an important factor hindering the extraction performance. These models perform poorly when dealing with triples containing overlapping data.

Subsequent works discover this phenomenon and focus on dealing with overlapping data. [47] points out the problem of data overlap for the first time and divides the triples into three categories. Extracting triples of SEO and EPO categories is the key to solving the problem. The current works are mainly divided into two types: generative and extractive. The generative work generates relational triples through an encoder–decoder framework. Better extraction performance can be achieved by optimizing the encoder and decoder. [10] constructs a two-stage end-to-end model through Bi-GCN, which can capture the implicit features between word pairs. [45] improves the quality of the generated entities through multi-task learning. [7] optimizes the model for information interaction between relation categories. To achieve better training effect, [42] constructs negative samples by random sampling and proposes a dynamic masking mechanism. The generation order also has an impact on model performance. Therefore, [46] determines the generation order of triples through reinforcement learning. The entities generated by the above models are all represented by words. However, [25] adopts the idea of span to generate the start and end indexes corresponding to the entities. Although the generative model can solve the problem of data overlap, the

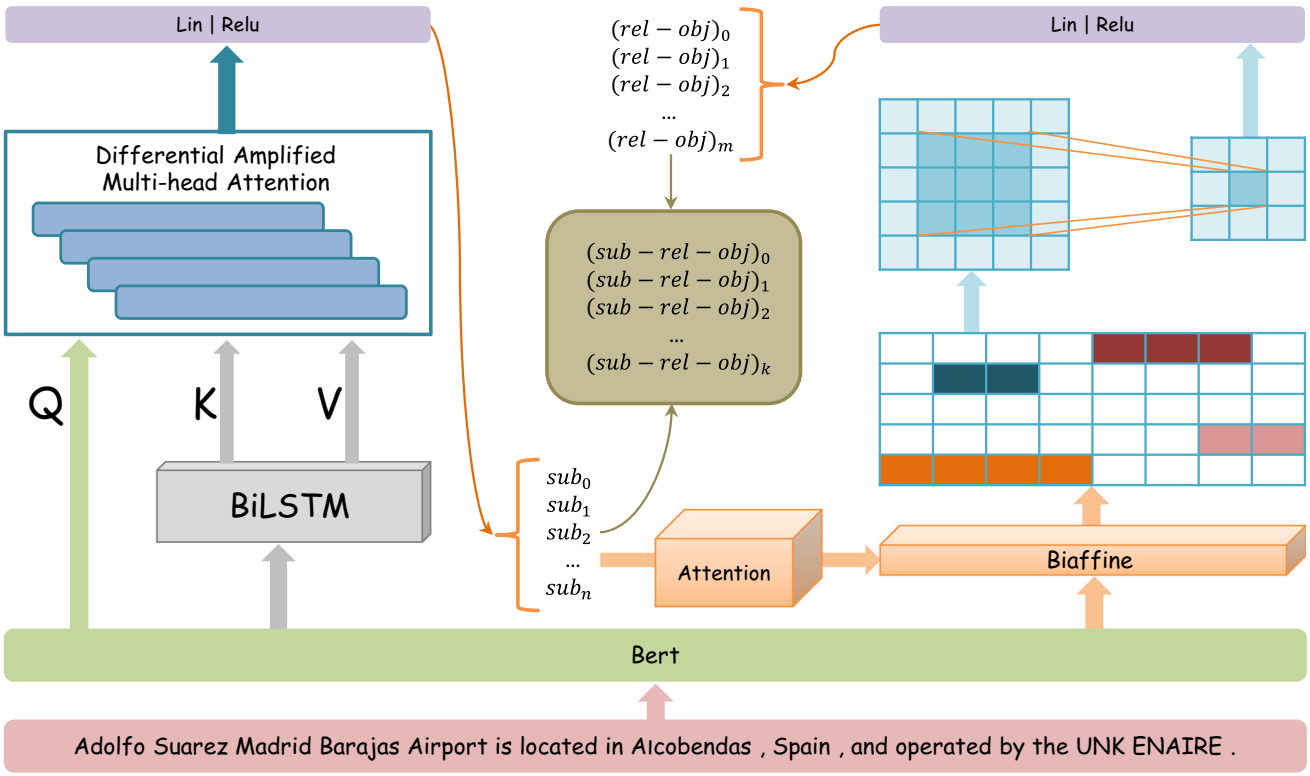


Fig. 2 An overview of the proposed NEDORT framework. The Differential Amplified Multi-head Attention method is employed in the extraction of subjects. The right part presents the generation and optimization of the relation–sequence matrix. Lin and Relu represent linear and activation functions, respectively

characteristics of the model limit the quality of the generated triples. In comparison, the extractive model is easier to implement. [16] learns the connections between each relation category through a multi-head attention mechanism. [39] regards the relation between entities as a function map. [33] extracts entities and relations simultaneously and re-labels multiple relation labels between entities to solve the problem of data overlap. [49] introduces a cascading capsule network to aggregate context representations and proposes a two-way routing mechanism to encourage interactions between relations and entities. [44] first extracts the head entity, and then predicts the relation corresponding to each boundary of the tail entity. Different from [40], [44] predicts all relations in the sentence, and then extracts entity pairs corresponding to a particular relation. The extractive model performs well in dealing with the problem of data overlap. This paper conducts the transformation from sequence to matrix and proposes an extractive model that does not need to deal with overlapping data.

Methods

This section presents the specifics of relation triple extraction. First, we provide a task definition to summarize the entire extraction process. Then we introduce the extraction method employed in each step. An overview illustration of NEDORT is shown in Fig. 1.

Task definition

The overlapping phenomenon is abundant in relational triples. To eliminate the interference of data overlap problem on the model, this paper decomposes the entire extraction task into two subtasks and performs conversion of sequence to matrix on the second subtask. We define the decomposition process of the extraction task by the following formula:

$$
\begin{aligned}
&(C) \rightarrow (Ss, Rs, Os) \\
&= \{(C) \rightarrow (Ss)\} \oplus \{(C, Ss) \rightarrow (Rs, Os)\} \\
&= \{(C) \rightarrow (Ss)\} \oplus \left\{ \underset{S \in Ss}{Iter}(C, S) \rightarrow (Rs, Os) \right\} \\
&= \{(C) \rightarrow (Ss)\} \oplus \left\{ \underset{S \in Ss}{Iter}(C, S) \rightarrow \underset{R \in Rs}{Iter}(R, Os) \right\} \\
&= \{(C) \rightarrow (Ss)\} \oplus \left\{ \underset{S \in Ss}{Iter}(C, S) \rightarrow \underset{R \in Rs}{Iter} \underset{O \in Os}{Iter}(R, O) \right\} \\
&= \{(C) \rightarrow (Ss)\} \oplus \left\{ \underset{S \in Ss}{Iter}(C, S) \rightarrow \underset{R \in Rs}{Iter} \underset{O \in Os}{Iter}(S, R, O) \right\}
\end{aligned} \tag{1}
$$

where C is the input context sequence, Rs is all the relations to be extracted, and R is one of them, Ss and Os are similar, $Iter$ denotes to iterate over all elements in the collection.

The above formula defines the whole extraction process. We first extract all subjects contained in the context, and then predict the relation–object pairs corresponding to one subject. Due to SEO and EPO phenomena, a subject may correspond to multiple relation–object pairs. Moreover, there may be more than one corresponding object for a given subject and relation. Therefore, the extraction of relation–object pairs is essentially a multi-classification based on matrix.

NEDORT encoder

$S = [t_1, t_2, \cdots, t_n]$ is the input sentence sequence and n is the number of tokens. To capture the semantic features of the sequence, we use the pre-trained BERT [6] to encode the input sentence:

$$[h_1, h_2, \cdots, h_n] = BERT([t_1, t_2, \cdots, t_n]) \tag{2}$$

$S_e = [h_1, h_2, \cdots, h_n] \in R^{n \times d}$ is the sentence expression after encoding, n is the length of the sentence, and d is the embedding dimension.

Subject extraction

This section aims to extract all subjects in the sentence. To obtain better extraction performance, we design the Differential Amplified Multi-head Attention method. This method combines data features and optimizes on the basis of Multi-head Self-attention [34]. The Multi-head Self-attention method generates the sequence representation in one dimension by formula $softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$, where Q=K=V. $\sqrt{d}$ is a constant derived from the hyper-parameters of the model and $\sqrt{d} > 1$. According to the features of the softmax function, multiplying the input data can make the large values in the output more prominent. The purpose of this section is to predict the start and end positions for the entity. The prominent output values at these two positions are beneficial to the prediction of subjects [41]. Therefore, we perform extraction without the scaling factor $\sqrt{d}$. Context semantics are crucial for the extraction of subjects, so we use BiLSTM to obtain representations that incorporate sentence context. Different from the Multi-head Self-attention method, we set Q, K, and V to different values to fuse more sentence information. The specific implementation of the Differential Amplified Multi-head Attention method is as follows:

$$\begin{cases} Q = [h_1, h_2, \cdots, h_n] \\ K = BiLSTM([h_1, h_2, \cdots, h_n]) \\ V = BiLSTM([h_1, h_2, \cdots, h_n]) \end{cases} \tag{3}$$

$$head^{(i)} = softmax((QW_q^{(i)})(KW_k^{(i)})^T)(VW_v^{(i)}) \tag{4}$$

$$S_m = \left[head^{(1)}; \ldots; head^{(k)}\right] W_o \tag{5}$$

where $W_q^{(i)}$, $W_k^{(i)}$, $W_v^{(i)}$ and W_o are trainable weights, $[h_1, h_2, \cdots, h_n]$ is the sentence sequence generated in Sect. "NEDORT encoder", k is the number of heads, $[\cdot;\cdot]$ denotes vector connection.

We have obtained the sentence representation that incorporates rich semantic features. Then we complete the extraction of subjects by predicting the start and end positions:

$$Sub^s = sigmoid(W_{sub}^s \sigma(S_m) + b_{sub}^s) \tag{6}$$

$$Sub^e = sigmoid(W_{sub}^e \sigma(S_m) + b_{sub}^e) \tag{7}$$

where W_{sub}^s, W_{sub}^e, b_{sub}^s and b_{sub}^e are trainable weights, σ is the activation function, and here is $Relu$. A sentence contains multiple subjects, so we will obtain multiple start positions and multiple end positions.

Relation–object pair extraction

This section completes the extraction of relation–object pairs. To improve the extraction performance, we introduce subject information at this stage. $S = [w_1, w_2, \cdots, w_n]$ is the extracted subject sequence, and w_i is one of the tokens. Each token contained in S contributes differently to global information, so we generate the ensemble representation of the subject by the following formula:

$$S_\omega = \text{softmax}(W_2 \sigma(W_1 S)) \tag{8}$$

$$S_{en} = MeanPool(\odot(S_\omega, S)) \tag{9}$$

where W_{e1} and W_{e2} are trainable parameters, σ is the activation function, and here is $Relu$, $\odot$ denotes element-wise multiplication, $MeanPool$ denotes mean pooling.

The problem of data overlap is abundant in relation–object pairs. To eliminate the interference of overlapping data, we perform the sequence-to-matrix transformation. The relation–object pair to be extracted has a relationship with the subject, so the information related to the subject should be included in the generated matrix. This section employs the Biaffine method to fuse subject features and generate the relation–sequence matrix. As shown in Fig. 3, the labels 1, 2, 3, 4, 5, 6, and 7 correspond to seven matrices M_1, M_2, M_3, M_4, M_5, M_6, and M_7, respectively. $M_1 = S_{en}$ is the subject representation generated earlier. $M_5 = S_e$ is the encoded sentence representation generated in Sect. NEDORT encoder". Then we fuse the information of these two matrices and construct the relation–sequence matrix. To complete the transformation of sequence-to-matrix and introduce subject features, we extend M_1 and generate the relation matrix M_2.

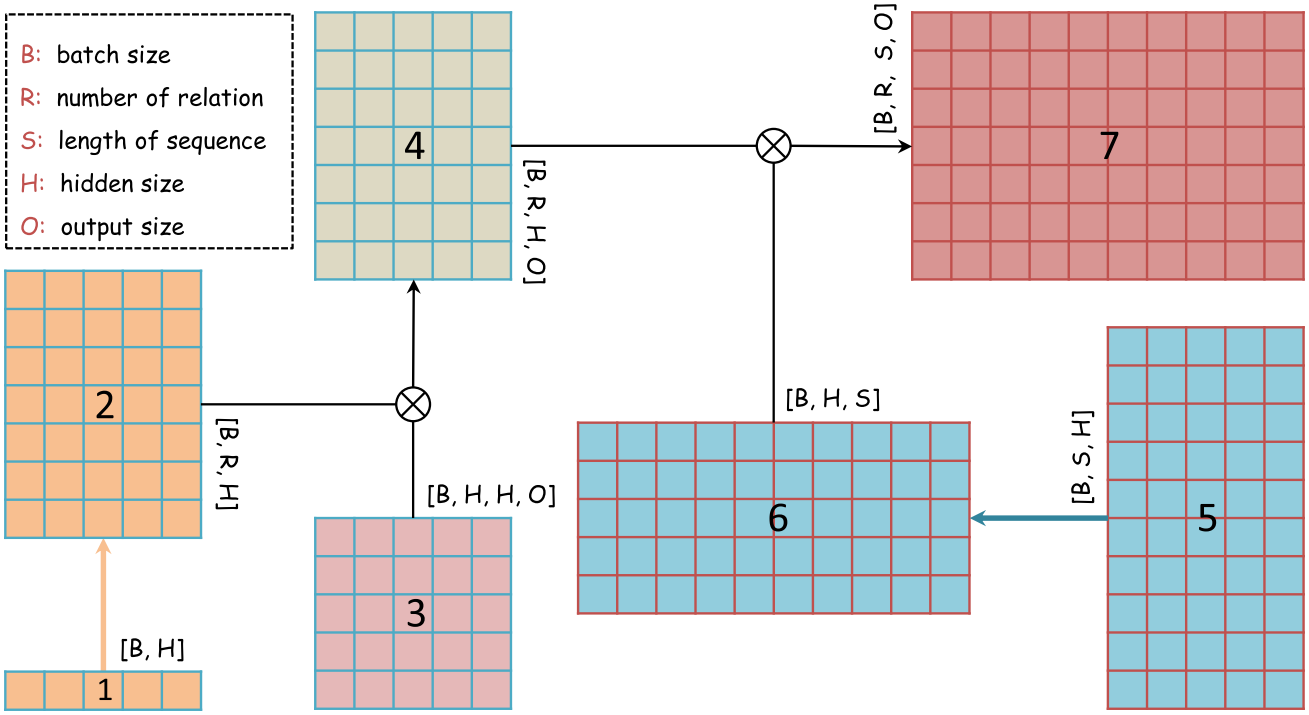


Fig. 3 Details of the Biaffine method. There are seven matrices in total. Matrix 1 is the subject representation obtained. Matrix 5 is the vector representation corresponding to the input sequence. Matrix 7 is the generated relation–sequence matrix

There are numerous repeated expressions in the expanded matrix, so we design the matrix M_3 to optimize it. M_3 is generated by random initialization, and it can be dynamically adjusted during training. The matrix generated by the dynamically adjusted strategy is more conducive to the extraction of relational triples. $M_4 = M_2 M_3$ is the matrix expression after optimization.[4] The matrix to be generated is relation and sequence related. Therefore, we combine the relation-related matrix M_4 and the sequence-related matrix M_5 to generate the final relation–sequence matrix M_7. M_7 can be generated by the formula $M_7 = M_4 M_6$, where $M_6 = (M_5)^T$. The whole process above can be expressed by

$$M_{bia} = Biaffine(S_{en}, S_e) \tag{10}$$

where S_{en} is the subject representation, S_{en} is the sequence representation, M_{bia} is the generated matrix representation. The above formula completes the conversion of sequence to matrix and incorporates the subject features.

The extraction of relation–object pairs is more complicated than that of subjects. Although the design of M_3 improves the inference capability of the model, it is difficult for the generated matrix to accurately extract the relation–object pairs corresponding to the subject. To optimize the matrix representation, we design the U-Net network. U-Net is a multi-layer convolutional neural network that can extract the deep features of the matrix. As shown in Fig. 2, the U-Net network consists of seven parts. Part 1 is the relation–sequence matrix obtained by the Biaffine method. Each of the remaining parts is composed of a two-layer convolutional neural network, and its formula is expressed as follows:

$$M_{out} = \sigma Conv(\sigma Conv(M_{in})) \tag{11}$$

where $Conv$ is the convolution operation and the convolution kernels of all parts are 3×3, σ is the activation function, and here is $Relu$, M_{in} is the input matrix, M_{out} is the output matrix, and the input and output of each part are different.

Figure 2 annotates the matrix dimension of the output for each step. Parts 2, 3, and 4 are down-sampling blocks whose output channels are constantly increasing. More channels can enlarge the receptive field of matrix embedding, thus providing rich global information between relations and sequences. Parts 5, 6, and 7 are up-sampling blocks, and their output channels are trending down. The reduction of the number of extracted features can make the model pay more attention to the token part corresponding to the object rather than the entire sequence. Moreover, there are information interactions between the down-sampling blocks and the up-sampling blocks. The combined matrix representation considers its contextual semantics while focusing on the object. Different from feature extraction in images, the obtained output

[4] Matrix multiplication in this section includes dimension expansion operations.

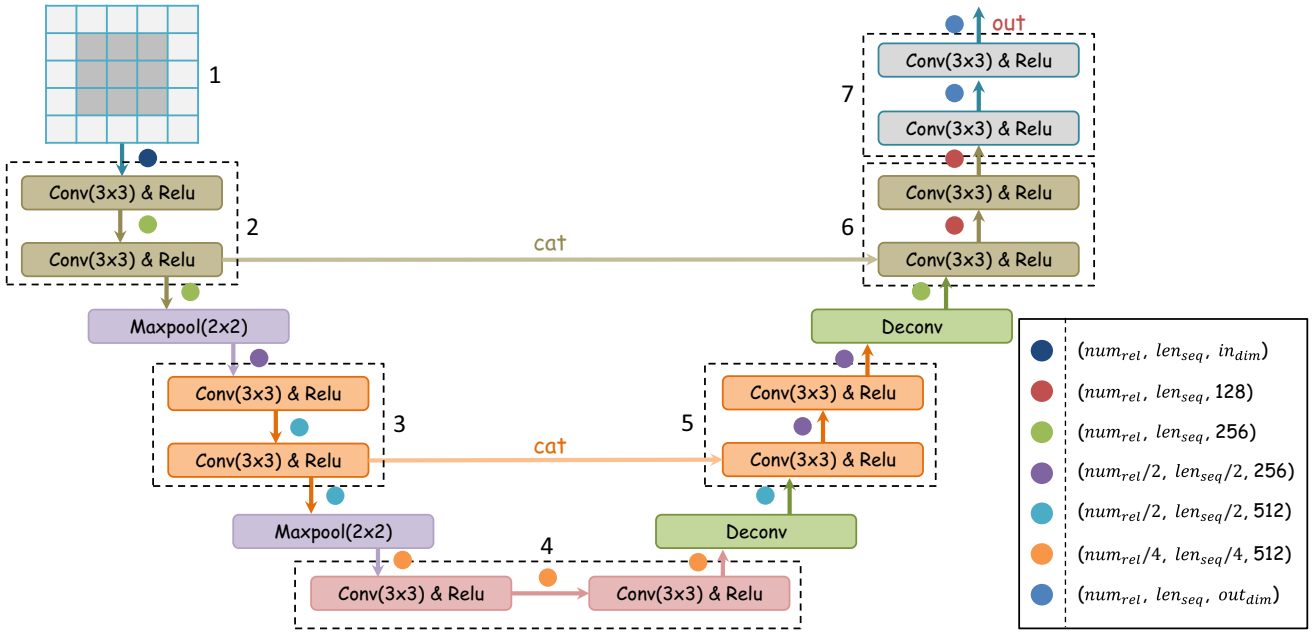


Fig. 4 The structure of the U-Net network. Part 1 is the input matrix. The rest parts are convolution blocks with different number of channels. "cat" represents the information interaction between two blocks. The content in the lower right corner is the matrix dimension of the output

matrix must have the same dimensions as the input matrix. Therefore, we design two Maxpooling layers and two Deconv layers in the down-sampling and up-sampling stages, respectively. The entire process can be expressed by:

$$M_u = UNet(M_{bia}) \tag{12}$$

where M_{bia} is the relation–sequence matrix generated by the Biaffine method, M_u is the matrix representation after optimization.

The optimized relation–sequence matrix has been obtained, and then we extract relation–object pairs on this basis. We complete the extraction of relation–object pairs by predicting the start and end positions of the object corresponding to each relation. The prediction formulas for these two positions are as follows:

$$Ob^s = sigmoid(W_o^{s2}(W_o^{s1}\sigma(M_u) + b_o^{s1}) + b_o^{s2}) \tag{13}$$
$$Ob^e = sigmoid(W_o^{e2}(W_o^{e1}\sigma(M_u) + b_o^{e1}) + b_o^{e2}) \tag{14}$$

where W_o^{s1}, W_o^{s2}, W_o^{e1}, W_o^{e2}, b_o^{s1}, b_o^{s2}, b_o^{e1} and b_o^{e2} are trainable weights, σ is the activation function, and here is $Relu$. The object corresponding to a specific relation may be None, indicating that the relation–object pair containing this relation does not exist.

Relational triple generation

The previous section performs the prediction of the start and end positions. This section combines these predicted values to generate relational triples. We provide a threshold and select the index whose corresponding value is higher than the threshold as the start or end position. For a specific input sentence, our model can obtain multiple start positions and multiple end positions. We match the start position with the first end position after it (the start and end positions can share an index[5]).

In Fig. 5, we set the values above the threshold to 1 and the rest to 0. $S = [T_0, T_1, \cdots, T_7]$ is the input sentence sequence, and two subjects are extracted by the model: $sub_0 = \{T_1\}$, $sub_1 = \{T_3, T_4\}$. Then we predict the relation–object pairs for each subject. Taking sub_1 as an example, the model extracts two objects: $obj_0 = \{T_6, T_7\}$, $obj_1 = \{T_1\}$. Combining the corresponding relations, there are three relation–object pairs obtained: $p_0 = (rel_0, obj_0)$, $p_1 = (rel_1, obj_0)$, $p_2 = (rel_n, obj_1)$. Then the relational triples containing sub1 are: $t_0 = (sub_1, rel_0, obj_0)$, $t_1 = (sub_1, rel_1, obj_0)$, $t_2 = (sub_1, rel_n, obj_1)$. The same entity pairs in t_0 and t_1 correspond to different relations, which belong to the EPO overlapping pattern. obj_1 and sub_0 are the same entity and t_2 contains obj_1, then t_2 and all triples containing sub_0 share an entity, which belongs to the SEO overlapping pattern. As shown in Fig. 5, our model performs perfectly when dealing with the above overlapping data. This paper conducts sequence-to-matrix conversion when extracting relation–object pairs. The transformed expression can avoid the interference of data overlap problem.

[5] If the start value and end value of an index are both 1, then the token corresponding to this index constitutes an entity.

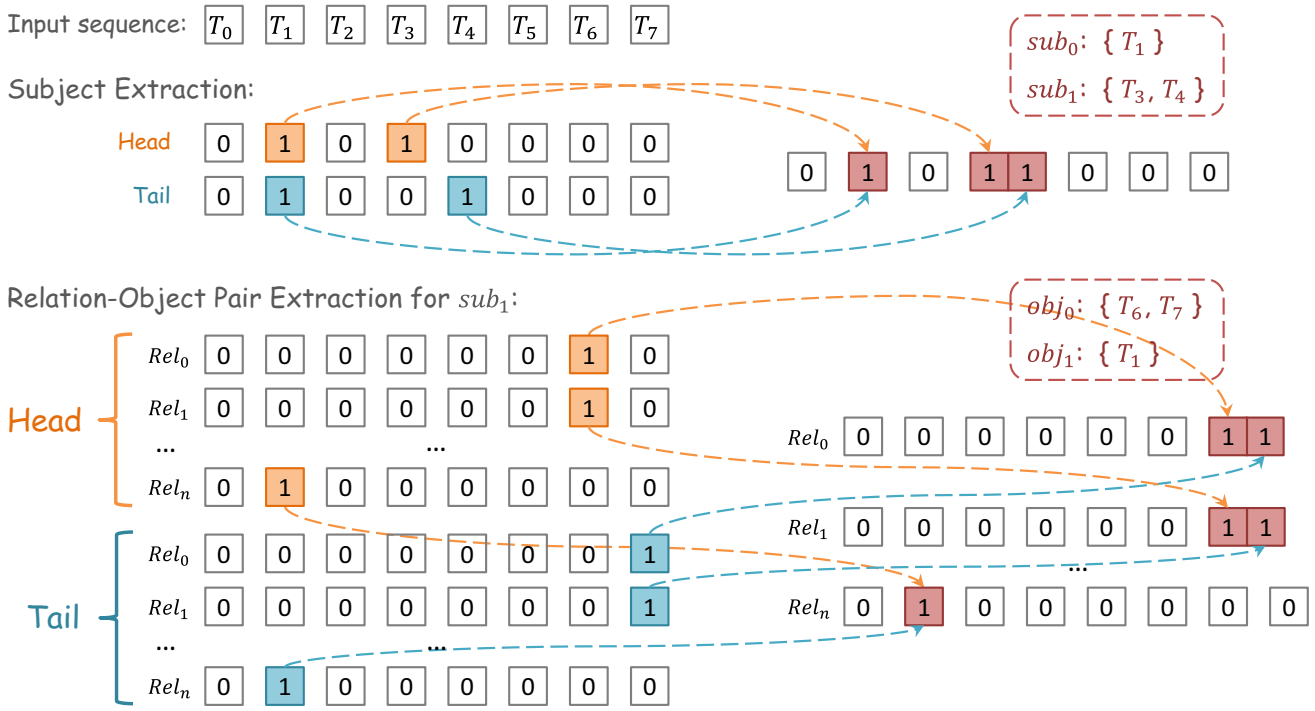


Fig. 5 Implementation details for the extraction of relational triples. "Head" and "Tail" represent predictions for start and end positions, respectively. Consecutive tokens marked in red are entities extracted from the sequence. "*sub*", "*Rel*" and "*obj*" denote subject, relation and object, respectively

Loss function

Our model outputs multiple prediction results, so the training loss consists of multiple parts:

$$\mathcal{L}_S^s = -\frac{1}{n}\sum_{i=1}^{n}(y_i^s \log Sub_i^s + (1-y_i^s)\log(1-Sub_i^s)) \tag{15}$$

$$\mathcal{L}_S^e = -\frac{1}{n}\sum_{i=1}^{n}(y_i^e \log Sub_i^e + (1-y_i^e)\log(1-Sub_i^e)) \tag{16}$$

$$\mathcal{L}_O^s = -\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n} (\bar{y}_{i,j}^s \log Ob_{i,j}^s + (1-\bar{y}_{i,j}^s)\log(1-Ob_{i,j}^s)) \tag{17}$$

$$\mathcal{L}_O^e = -\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n} (\bar{y}_{i,j}^e \log Ob_{i,j}^e + (1-\bar{y}_{i,j}^e)\log(1-Ob_{i,j}^e)) \tag{18}$$

where y^s, y^e, $\bar{y}^s$, and $\bar{y}^e$ are the ground-truth labels corresponding to Sub^s,Sub^e,Ob^s, and Ob^e, respectively. We combine the above four parts to calculate the total loss:

$$\mathcal{L} = \alpha\mathcal{L}_S^s + \beta\mathcal{L}_S^e + \lambda\mathcal{L}_O^s + \mu\mathcal{L}_O^e \tag{19}$$

where $\alpha, \beta, \lambda, \mu \in [0, 1]$ are hyper-parameters used to control the contribution of each part.

Experiments

This section first elaborates the details of the experiment and then presents and analyzes the experimental results.

Experimental setting

Datasets

We evaluate our model on two public datasets WebNLG [11], and NYT [29]. The WebNLG dataset was originally created for the Natural Language Generation task, and the NYT dataset was initially used for the distant supervision method. [47] applies the above two datasets to relation triple extraction. The sentences of the two datasets contain multiple overlapping relational triples, which is quite suitable for this task. The WebNLG dataset contains 5019 training sentences, 500 validation sentences, and 703 test sentences. The NYT dataset holds more sentences, including 56195 training sentences, 5000 validation sentences, and 5000 test sentences. According to the overlapping pattern, we divide the test set into three categories: Normal, SEO (SingleEntityOverlap), and EPO (EntityPairOverlap). In addition, we also divide the

Table 1 Statistics of datasets used in our experiments

Category	WebNLG	NYT
Train	5019	56195
Valid	500	5000
Test	703	5000

test set into five test subsets according to the number of triples in a sentence to verify the capability of the model in dealing with multiple triples. The statistics of the two datasets are described in Table 1.

Implementation details

We implement our model with the Pytorch library and the hyper-parameters are set as follows. We set the batch size on WebNLG and NYT to be 6 and 10, respectively. Except for the batch size, other hyper-parameters are the same on both datasets. We encode the input sentence through Bert and use its default parameters. The BiLSTM we employed has only one layer, and the designed Differential Amplified Multi-head Attention method contains 8 heads. We set the output dimension of Biaffine to 256 and the hidden size to 768. The input and output channels of U-Net are both 256, where the convolution kernel is 3×3 and the pooling kernel is 2×2. The thresholds for predicting the start and end positions of entities are both set to 0.5. The activation functions used in this paper are all Relu. We optimize the model by SGD with an initial learning rate of 0.1.

Baselines

We consider the following strong baselines for comparison. NovelTagging [51] constructs a new annotation method to complete the extraction of relational triples. CopyR [47] designs an encoder–decoder model with three decoders. GraphRel [10] proposes an end-to-end extraction model based on graph convolutional networks. CopyR$_{RL}$ [46] determines the extraction order of triples by reinforcement learning. Att-as-Rel [16] designs a supervised multi-head self-attention module to learn the correlation of each relation category. RIN [32] proposes a multi-task learning model, which effectively extracts task-specific features through dynamic interaction. CasRel [39] regards relations as functional mappings between entities. ETL-Span [44] first extracts the head entity, and then predicts the relation corresponding to each boundary of the tail entity. WDec [25] also proposes an encoder–decoder model. Different from CopyR, WDec [25] decodes the boundary position of the entity instead of the token span. Similar to the previous encoder–decoder models, CopyMTL, MA-DCGCN, and CGT$_{UniLM}$ all extract relational triples through generative methods. CopyMTL [45] proposes a generative model based on multi-task learning. MA-DCGCN [7] optimizes the model for information interaction between relation categories. CGT$_{UniLM}$ [42] proposes a novel triplet contrastive training object and designs a dynamic masking mechanism to improve the quality of generated triples. GraphJoint [40] proposes a relation-based two-step extractive model. The first step predicts the relations in the sentence through a graph neural network. The second step introduces the obtained relations to perform the extraction of entities.

Table 2 Results (%) of different models on WebNLG

Model	Prec	Rec	F1
NovelTagging[§] (Zheng et al., 2017) [51]	52.5	19.3	28.3
OneDecoder[§] (Zeng et al., 2018) [47]	32.2	28.9	30.5
MultiDecoder[§] (Zeng et al., 2018) [47]	37.7	36.4	37.1
GraphRel$_{1p}$[‡] (Fu et al., 2019) [10]	42.3	39.2	40.7
GraphRel$_{2p}$[‡] (Fu et al., 2019) [10]	44.7	41.1	42.9
CopyR$_{RL}$[‡] (Zeng et al., 2019) [46]	63.3	59.9	61.6
Att-as-Rel[‡] (Liu et al., 2020) [16]	89.5	86.0	87.7
RIN[‡] (Sun et al., 2020) [32]	87.6	87.0	87.3
CasRel[‡] (Wei et al., 2020) [39]	**93.4**	90.1	91.8
ETL-Span[⋆] (Yu et al., 2020) [44]	84.0	91.5	87.6
CopyMTL$_{One}$[‡] (Zeng et al., 2020) [45]	57.8	60.1	58.9
CopyMTL$_{Mul}$[‡] (Zeng et al., 2020) [45]	58.0	54.9	56.4
MA-DCGCN[‡] (Duan et al., 2021) [7]	67.4	65.1	66.3
CGT$_{UniLM}$[‡] (Ye et al., 2021) [42]	92.9	75.6	83.4
GraphJoint[‡] (Xu et al., 2022) [40]	88.3	87.7	87.9
NEDORT	92.2	**91.5**	**91.9**

Bold text indicates the best results
[§]Marks results reported by [47].
[⋆]Marks results produced with official implementation.
[‡]Marks results quoted directly from the original papers

Results and discussions

We conduct experiments on two public datasets WebNLG and NYT. For a fair comparison, we present the scores of all models on Precision (Prec.), Recall (Rec.) and F1. F1 is the synthesis of Precision and Recall, which can better reflect the overall performance of the model [3,4]. Tables 2 and 3 show the comparison results of our proposed NEDORT model and other baseline models. The NEDORT model achieves the best results on both datasets, and the F1 scores all exceed 90 (%). CasRel and CGT$_{UniLM}$ obtain the highest scores on Precision, while their poor performance on Recall results in relatively low F1 scores. In contrast, the performance of our model on the three indicators is relatively balanced, and the comprehensive performance is more prominent.

Table 3 Results (%) of different models on NYT

Model	Prec	Rec	F1
NovelTagging§ (Zheng et al., 2017) [51]	62.4	31.7	42.0
OneDecoder§ (Zeng et al., 2018) [47]	59.4	53.1	56.0
MultiDecoder§ (Zeng et al., 2018) [47]	61.0	56.6	58.7
GraphRel$_{1p}$‡ (Fu et al., 2019) [10]	62.9	57.3	60.0
GraphRel$_{2p}$‡ (Fu et al., 2019) [10]	63.9	60.0	61.9
CopyR$_{RL}$‡ (Zeng et al., 2019) [46]	77.9	67.2	72.1
WDec‡ (Nayak and Ng, 2020) [25]	94.5	76.2	84.4
Att-as-Rel‡ (Liu et al., 2020) [16]	88.1	78.5	83.0
RIN‡ (Sun et al., 2020) [32]	87.2	87.3	87.3
CasRel‡ (Wei et al., 2020) [39]	89.7	89.5	89.6
ETL-Span⋆ (Yu et al., 2020) [44]	84.9	72.3	78.1
CopyMTL$_{One}$‡ (Zeng et al., 2020) [45]	72.7	69.2	70.9
CopyMTL$_{Mul}$‡ (Zeng et al., 2020) [45]	75.7	68.7	72.0
MA-DCGCN‡ (Duan et al., 2021) [7]	81.3	76.7	79.4
CGT$_{UniLM}$‡ (Ye et al., 2021) [42]	**94.7**	84.2	89.1
GraphJoint‡ (Xu et al., 2022) [40]	88.7	83.8	86.2
NEDORT	91.8	**89.7**	**90.7**

Bold text indicates the best results
§Marks results reported by [47].
⋆Marks results produced with official implementation.
‡Marks results quoted directly from the original papers

CGT$_{UniLM}$ is a generative model, and GraphJoint is an extractive model. The extraction performance of CGT$_{UniLM}$ is better than that of GraphJoint on the NYT dataset. Generative models require a large amount of training data to achieve good results. The NYT dataset contains 56195 training sentences and satisfies this condition. Furthermore, there are few relation categories (24) in NYT, which is beneficial for generative models. For the WebNLG dataset with less training data (5000) and more relation categories (246), CGT$_{UniLM}$ performs much worse than GraphJoint. In general, the extraction performance of GraphJoint is slightly better than that of CGT$_{UniLM}$.

The above two models show certain advantages on different datasets, but they are inferior to our model in terms of triple extraction. Compared with CGT$_{UniLM}$, the F1 score improvement of our model on the two datasets is 8.5 (%) and 1.6 (%), respectively. The output of CGT$_{UniLM}$ is not included in any input tokens, and the extraction result depends on the powerful inference capability of the model. In this paper, we only need to perform binary classification for each input token. Compared with the generative task, the binary classification task is simpler, so the model proposed in this paper can achieve better extraction results. Compared with GraphJoint, the F1 score improvement of our model on the two datasets is 4.0 (%) and 4.5 (%), respectively. GraphJoint extracts triplets in a one-dimensional space, where multiple overlapping cases occur between triplets, and overlapping data seriously interferes with the extraction results of the model. Different from GraphJoint, NEDORT expresses the input data in two-dimensional spaces. After dimension expansion, there is no overlap between triplets, which brings great convenience to triple extraction. The transformation from sequence to matrix is the key for our model to obtain good extraction results.

Results on triples of different categories

The problem of data overlap is a significant obstacle for relation triple extraction. To demonstrate the performance of the model in dealing with overlapping data, we divide triples into three categories: Normal, SEO, and EPO. The Normal category has no data overlap problem, while the triples of the other two categories contain overlapping parts. Therefore, performance in the SEO and EPO categories is the key to evaluate the capability of the model to address the data overlap problem. We conduct experiments on these three categories and present the comparison results in Figs. 6, 7, and 8. The data in the figure is the F1 score of each model on different categories of triples.

The comparison shows that our model achieves the best results on triples of three categories for both datasets. Figure 6 presents the extraction results on triples of the Normal category. Compared with the GraphJoint model, NEDORT only achieves an F1 score improvement of 0.1 (%) on the NYT dataset. The extraction performance on the WebNLG dataset is better, and the F1 score is improved by 0.8 (%). Our model performs better but the advantage is not obvious. Figures 7 and 8 present the extraction results on triples of overlapping patterns. Our model significantly outperforms other baseline models in extracting triples of these two categories. For the SEO category, our model obtains F1 score improvements of 0.8 (%) and 4.2 (%) on both datasets, respectively. The advantage is more obvious for the triple of EPO category. Compared with other baseline models, NEDORT improves the F1 score by 12.9 (%) and 19.8 (%), respectively. There is only one overlapping entity in the triple of SEO category, and it is relatively easy to extract such triples. The triple of the EPO category contains two overlapping entities, so it is difficult for the baseline model to make a correct judgment on the overlapping part. In this paper, the interference of overlapping data on the model is eliminated by matrix presentation, and the categories of overlapping triples have no influence on our model. Therefore, the F1 score improvement of NEDORT in the triple of EPO category is more pronounced. In conclusion, our model is superior to other baseline models in dealing with triples of all categories, and the advantage is more prominent in overlapping data.

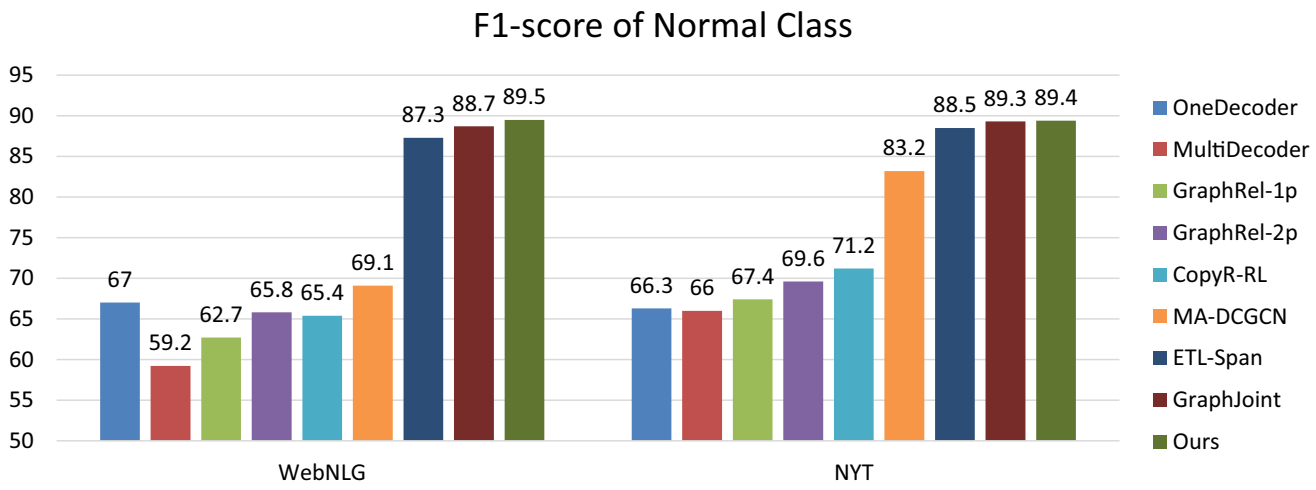


Fig. 6 Extraction results (%) for different models on triples of Normal category. The data in the figure is the F1 score on the specified dataset

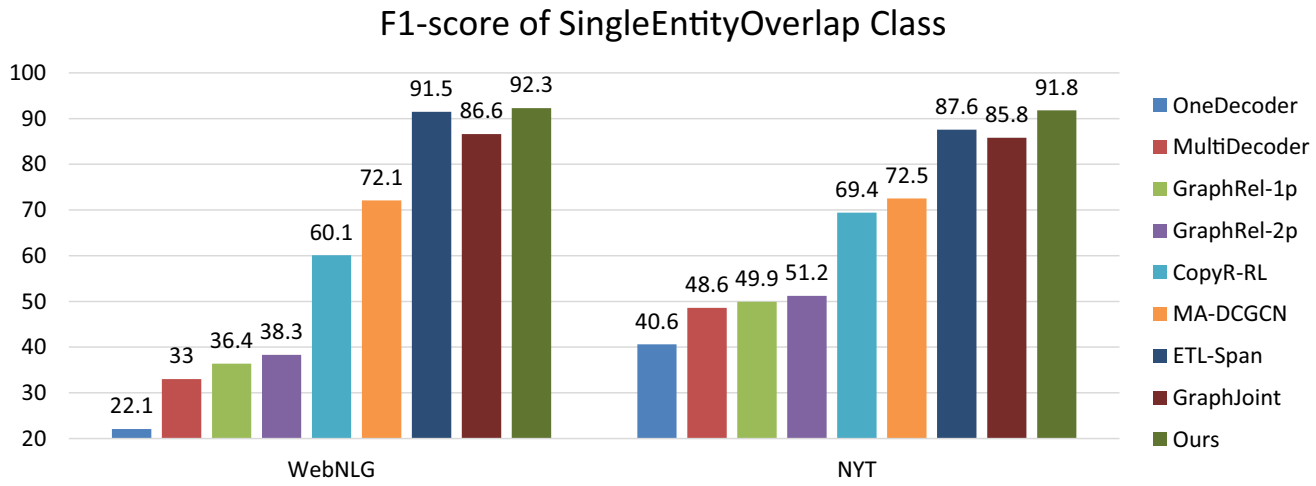


Fig. 7 Extraction results (%) for different models on triples of SEO category. The data in the figure is the F1 score on the specified dataset

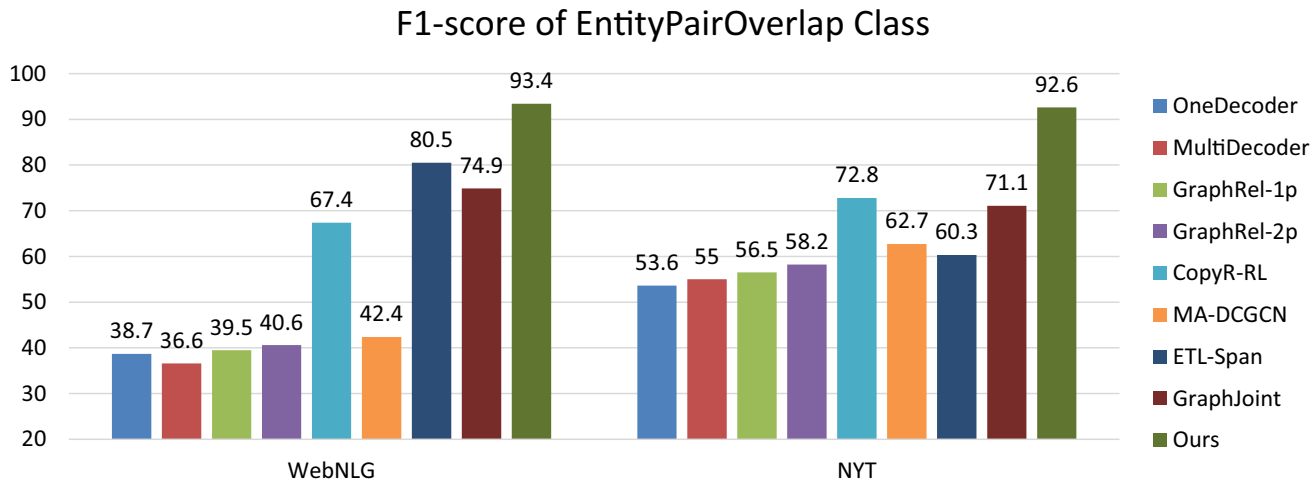


Fig. 8 Extraction results (%) for different models on triples of EPO category. The data in the figure is the F1 score on the specified dataset

Table 4 F1-score (%) of extracting relational triples from sentences with different number (denoted as N) of triples on WebNLG

Model	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$
OneDecoder	65.2	33.0	22.2	14.2	13.2
MultiDecoder	59.2	42.5	31.7	24.2	30.0
GraphRel$_{1p}$	63.8	46.3	34.7	30.8	29.4
GraphRel$_{2p}$	66.0	48.3	37.0	32.1	32.1
CopyR$_{RL}$	63.4	62.2	64.4	57.2	55.7
MA-DCGCN[$]	67.2	50.3	39.4	37.8	35.7
ETL-Span	82.1	86.5	91.4	89.5	91.1
GraphJoint	87.8	86.7	87.1	85.3	87.0
NEDORT	**89.0**	**90.5**	**94.9**	**92.2**	**91.7**

Bold text indicates the best results
[$]marks results inferred from the graph in the original paper, and these values float up and down by 1.0 (%)

Table 5 F1-score (%) of extracting relational triples from sentences with different number (denoted as N) of triples on NYT

Model	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$
OneDecoder	66.6	52.6	49.7	48.7	20.3
MultiDecoder	67.1	58.6	52.0	53.6	30.0
GraphRel$_{1p}$	69.1	59.5	54.4	53.9	37.5
GraphRel$_{2p}$	71.0	61.5	57.4	55.1	41.1
CopyR$_{RL}$	71.7	72.6	72.5	77.9	45.9
MA-DCGCN[$]	72.1	61.8	59.3	57.5	42.3
ETL-Span	85.5	82.1	74.7	75.6	76.9
GraphJoint	86.8	85.3	87.0	86.1	85.7
NEDORT	**89.2**	**91.6**	**91.2**	**94.4**	**88.3**

Bold text indicates the best results
[$]Marks results inferred from the graph in the original paper, and these values float up and down by 1.0 (%)

Results on triples of different numbers

The previous sections focus on the optimization of the data overlap problem. However, too many triples corresponding to one sentence can also make the extraction more difficult. This section verifies the capability of the model to handle sentences containing multiple triples. We divide the test set into 5 subsets according to the number of triples in the sentence, and the last subset contains all triples whose number is greater than 4. It can be seen from the data division that a sentence may contain 5 or more triples, which brings great challenges to the extraction of relational triples. We conduct experiments on 10 subsets of both datasets and compare the F1 score of each model. Table 4 shows the extraction results of all models on the WebNLG dataset. Our model achieves the best results on all subsets compared to other baseline models. The comparison results on the NYT dataset are shown in Table 5. Our model also performs the best on all subsets, and the advantage is more prominent. These experiments demonstrate the superiority of our model on sentences containing multiple triples.

Effectiveness analysis of single step

In this paper, the extraction of relational triples is divided into two steps. The extraction results of previous experiments are based on the entire triple. This section verifies the performance of our model on a single extraction step. We conduct experiments on two datasets and present the results of the two steps in Table 6. For the WebNLG dataset, the F1 score of our model in the first step is 96.7 (%). The extraction of relation–object pairs is much more difficult than that of subjects, and our model can still obtain an F1 score of 93.3 (%). In addition, we also present the respective extraction results of relation and object. The F1 scores for relation and object are 94.4 (%) and 95.6 (%), respectively. It is very difficult to obtain an F1 score of 94.4 (%) on a dataset with 246 relations. Compared with WebNLG, our model performs slightly worse on the NYT dataset. NYT contains more test sentences than WebNLG, and obtaining a high score on this dataset is challenging. Nonetheless, our model achieves F1 scores of over 92 (%) on both extraction steps. For relations and objects, our model also performs well and obtains F1 scores of 94.5 (%) and 93.9 (%), respectively. Such good performance benefits from the specific method designed for each extraction step in this paper.

Table 6 Results (%) of single step on WebNLG and NYT

WebNLG			
Category	Prec	Rec	F1
(Subject)	98.3	95.1	96.7
(Relation, Object)	94.0	92.7	93.3
Relation	95.6	93.2	94.4
Object	96.5	94.7	95.6
(Subject, Relation, Object)	92.2	91.5	91.9
NYT			
Category	Prec	Rec	F1
(Subject)	94.8	92.6	93.7
(Relation, Object)	93.6	91.0	92.3
Relation	96.1	92.9	94.5
Object	94.8	93.0	93.9
(Subject, Relation, Object)	91.8	89.7	90.7

Complexity analysis

This section analyzes the computational complexity of the proposed methods and presents the results in Table 7. As

Table 7 Complexity analysis of the proposed method

Method	Complexity
Differential Amplified Attention	$O(dn^2 + d^2n)$
Biaffine	$O(drn)$
U-Net	$O(drn)$
Subject Prediction	$O(n)$
Relation–Object Pair Prediction	$O(rn)$

Table 8 Ablation study (%) of NEDORT on WebNLG

Model	Prec	Rec	F1
NEDORT	92.2	91.5	91.9
w/o differential amplified attention	92.6	90.5	91.5
w/o subject	60.5	70.3	65.0
w/o Biaffine	91.5	90.8	91.1
w/o U-Net	00.0	00.0	00.0

shown in the table, n is the length of the input sequence, d is the dimension of the word embedding, and r is the number of relations contained in the dataset. The Differential Amplified Attention method captures the subject features contained in the sequence, and the computational complexity is $O(dn^2+d^2n)$. This method performs feature extraction in multiple dimensions, so the computation is relatively complex. The Biaffine method performs the transformation from sequence to matrix, and the U-Net method is used to optimize the representation of the matrix. The computational complexity of these two methods is $O(drn)$, indicating that the computation is not complicated. The last two methods are used to predict the start and end positions of the entity. Both methods perform binary classification, so they perform well in terms of computational complexity. NEDORT combines all the above methods to extract triples, so the comprehensive computational complexity of the model is $O(dn^2 + d^2n + drn + rn + n)$. For a given dataset, both d and r are constants, so the computational complexity of NEDORT can be expressed as $O(n^2)$. The above analysis shows that NEDORT performs well in terms of computational complexity. Combined with the excellent performance of the model in overlapping data, NEDORT is well suited to perform triple extraction.

Ablation study

This section verifies the effectiveness of the proposed methods through ablation studies. We conduct experiments on the WebNLG dataset and show the results in Table 8.

The Differential Amplified Multi-head Attention method is used to extract the subject. We first remove it to perform ablation. The result shows that the extraction performance of the entire triple decreases significantly after ablation. This method extracts entity features from multiple dimensions and highlights the start and end indices of entities, which is essential for the extraction of subjects.

The relation–object pair to be extracted has a relationship with the subject, so we incorporate subject information in the second extraction step. To verify the necessity of information introduction, we perform ablation of the subject. The model after ablation does not input subject information when extracting relation–object pairs. Table 8 shows that the F1 score of the model dropped by 26.5 (%) when performing subject removal. Such obvious performance degradation indicates that the subject information is crucial for the extraction of relation–object pairs. Without subject information, our model cannot associate the extracted relation–object pairs with a specific subject, which is the reason for the poor performance.

The Biaffine method is employed to generate relation–sequence matrices. We replace it with matrix expansion to complete the ablation. Compared with NEDORT, the extraction capability of the model after ablation decreased significantly, indicating that this method is irreplaceable. The Biaffine method realizes the complete fusion of subject and sentence sequence, which lays the foundation for the extraction of relation–object pairs.

Finally, we remove the U-Net network to perform ablation. Surprisingly, the model without U-Net fails to converge. U-Net is used to optimize matrix representation and extract features between entities and relations. The extraction of relation–object pairs is more complicated than that of subjects, and a matrix without optimization cannot capture the complex connections between relations and objects. Therefore, the U-Net network is the key to the excellent performance of our model and cannot be removed.

Conclusion

The problem of data overlap is a significant obstacle to the extraction of relational triples. This paper performs the sequence-to-matrix transformation to fundamentally eliminate the interference of overlapping data. We design the Biaffine method to fuse subject information and generate relation–sequence matrices. Furthermore, we also design the U-Net network to optimize the matrix representation. When performing the extraction of subjects, we employ the Differential Amplified Multi-head Attention method to highlight the start and end positions of entities and extract sequence features from multiple dimensions. The experimental results show that the proposed NEDORT model outperforms other baseline models on triples of all categories. Our model completes the extraction of relational triples in two steps, which are linked only by subject information. If there are more con-

nections between the two steps, the extraction performance of the model will be further improved. Future work will explore new models to increase the information interaction between these two steps. The above experiments have proved that there is no data overlap problem in multidimensional space. In the future, we will explore more effective multidimensional space expression methods to further improve the extraction results of triples. In addition, more efficient word embedding generation methods can also be combined with our model, which can provide better initialization representation for the input data.

Acknowledgements The authors wish to thank the reviewers for their helpful comment.

Funding This work is supported by the National Key Research and Development Program of China (2018YFB0204301).

Data Statement The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declared that they have no conflicts of interest to this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Aras G, Makaroglu D, Demir S, Cakir A (2021) An evaluation of recent neural sequence tagging models in turkish named entity recognition. Expert Syst Appl 182:115049
2. Bekoulis G, Deleu J, Demeester T, Develder C (2018) Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst Appl 114:34–45
3. Chakraborty C, Kishor A (2022) Real-time cloud-based patient-centric monitoring using computational health systems. IEEE Trans Comput Soc Syst 9(6):1613–1623
4. Chakraborty C, Kishor A, Rodrigues JJPC (2022) Novel enhanced-grey wolf optimization hybrid machine learning technique for biomedical data computation. Comput Electr Eng 99:107778
5. Cohen AD, Rosenman S, Goldberg Y (2020) Relation classification as two-way span-prediction. arXiv preprint arXiv:2010.04829
6. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
7. Duan G, Miao J, Huang T, Luo W, Hu D (2021) A relational adaptive neural model for joint entity and relation extraction. Front Neurorobotics 15:635492
8. Eberts M, Ulges A (2020) Span-based joint entity and relation extraction with transformer pre-training. In: G.D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (eds.) ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2006–2013. IOS Press
9. Fang Z, Zhang Q, Kok S, Li L, Wang A, Yang S (2021) Referent graph embedding model for name entity recognition of chinese car reviews. Knowl Based Syst 233:107558
10. Fu T, Li P, Ma W (2019) Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In: A. Korhonen, D.R. Traum, L. Màrquez (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 1409–1418. Association for Computational Linguistics
11. Gardent C, Shimorina A, Narayan S, Perez-Beltrachini L (2017) Creating training corpora for nlg micro-planning. In: 55th annual meeting of the Association for Computational Linguistics (ACL)
12. Gupta P, Schütze H, Andrassy B (2016) Table filling multi-task recurrent neural network for joint entity and relation extraction. In: N. Calzolari, Y. Matsumoto, R. Prasad (eds.) COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 2537–2547. ACL
13. Li P, Mao K (2019) Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts. Expert Syst Appl 115:512–523
14. Li Q, Ji H (2014) Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pp. 402–412. The Association for Computer Linguistics
15. Li X, Yan H, Qiu X, Huang X (2020) FLAT: chinese NER using flat-lattice transformer. In: D. Jurafsky, J. Chai, N. Schluter, J.R. Tetreault (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pp. 6836–6842. Association for Computational Linguistics
16. Liu J, Chen S, Wang B, Zhang J, Li N, Xu T (2020) Attention as relation: Learning supervised multi-head self-attention for relation extraction. In: C. Bessiere (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 3787–3793. ijcai.org
17. Luan Y, Wadden D, He L, Shah A, Ostendorf M, Hajishirzi H (2019) A general framework for information extraction using dynamic span graphs. In: J. Burstein, C. Doran, T. Solorio (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 3036–3046. Association for Computational Linguistics
18. Lyu S, Chen H (2021) Relation classification with entity type restriction. In: C. Zong, F. Xia, W. Li, R. Navigli (eds.) Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, Findings of ACL, vol. ACL/IJCNLP 2021, pp. 390–395. Association for Computational Linguistics
19. Ma Y, Hiraoka T, Okazaki N (2020) Named entity recognition and relation extraction using enhanced table filling by contextualized representations. CoRR **abs/2010.07522**

20. Miwa M, Bansal M (2016) End-to-end relation extraction using lstms on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics
21. Miwa M, Sasaki Y (2014) Modeling joint entity and relation extraction with table representation. In: A. Moschitti, B. Pang, W. Daelemans (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1858–1869. ACL
22. Molina-Villegas A, Muñiz-Sanchez V, Arreola-Trapala J, Alcántara F (2021) Geographic named entity recognition and disambiguation in mexican news using word embeddings. Expert Syst Appl 176:114855
23. Musheer RA (2022) Application of nature inspired soft computing techniques for gene selection: a novel frame work for classification of cancer. Soft Comput 26(22):12179–12196
24. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1):3–26
25. Nayak T, Ng HT (2020) Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 8528–8535. AAAI Press
26. Nguyen DQ, Verspoor K (2019) End-to-end neural relation extraction using deep biaffine attention. In: L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, D. Hiemstra (eds.) Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I, Lecture Notes in Computer Science, vol. 11437, pp. 729–738. Springer
27. Nozza D, Manchanda P, Fersini E, Palmonari M, Messina E (2021) Learningtoadapt with word embeddings: domain adaptation of named entity recognition systems. Inf Process Manag 58(3):102537
28. Ren L, Sun C, Ji H, Hockenmaier J (2021) Hyspa: Hybrid span generation for scalable text-to-graph extraction. In: C. Zong, F. Xia, W. Li, R. Navigli (eds.) Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, Findings of ACL, vol. ACL/IJCNLP 2021, pp. 4066–4078. Association for Computational Linguistics
29. Riedel S, Yao L, McCallum A (2010) Modeling relations and their mentions without labeled text. In: Balcázar JL, Bonchi F, Gionis A, Sebag M (eds) Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20–24, 2010, Proceedings, Part III, vol 6323. Lecture Notes in Computer Science. Springer, pp 148–163
30. Rink B, Harabagiu SM (2010) UTD: classifying semantic relations by combining lexical and semantic resources. In: K. Erk, C. Strapparava (eds.) Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010, pp. 256–259. The Association for Computer Linguistics
31. Sapci AOB, Tastan Ö, Yeniterzi R (2021 Focusing on possible named entities in active named entity label acquisition. CoRR **abs/2111.03837**)
32. Sun K, Zhang R, Mensah S, Mao Y, Liu X (2020) Recurrent interaction network for jointly extracting entities and classifying relations. In: B. Webber, T. Cohn, Y. He, Y. Liu (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pp. 3722–3732. Association for Computational Linguistics
33. Sun Q, Zhang K, Lv L, Li X, Huang K, Zhang T (2022) Joint extraction of entities and overlapping relations by improved graph convolutional networks. Appl Intell 52(5):5212–5224
34. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5998–6008
35. Vo D, Bagheri E (2019) Feature-enriched matrix factorization for relation extraction. Inf Process Manag 56(3):424–444
36. Wadden D, Wennberg U, Luan Y, Hajishirzi H (2019) Entity, relation, and event extraction with contextualized span representations. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 5783–5788. Association for Computational Linguistics
37. Wang J, Lu W (2020) Two are better than one: Joint entity and relation extraction with table-sequence encoders. In: B. Webber, T. Cohn, Y. He, Y. Liu (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pp. 1706–1721. Association for Computational Linguistics
38. Wang Y, Sun C, Wu Y, Zhou H, Li L, Yan J (2020) Unire: A unified label space for entity relation extraction. In: C. Zong, F. Xia, W. Li, R. Navigli (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pp. 220–231. Association for Computational Linguistics (2021)
39. Wei Z, Su J, Wang Y, Tian Y, Chang Y (2020) A novel cascade binary tagging framework for relational triple extraction. In: D. Jurafsky, J. Chai, N. Schluter, J.R. Tetreault (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pp. 1476–1488. Association for Computational Linguistics
40. Xu M, Pi D, Cao J, Yuan S (2022) A novel entity joint annotation relation extraction model. Applied Intelligence pp. 1–17
41. Yan H, Deng B, Li X, Qiu X (2019) TENER: adapting transformer encoder for named entity recognition. CoRR **abs/1911.04474**
42. Ye H, Zhang N, Deng S, Chen M, Tan C, Huang F, Chen H (2021) Contrastive triple extraction with generative transformer. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pp. 14257–14265. AAAI Press
43. Yu B, Zhang Z, Liu T, Wang B, Li S, Li Q (2019) Beyond word attention: Using segment attention in neural relation extraction. In: S. Kraus (ed.) Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 5401–5407. ijcai.org
44. Yu B, Zhang Z, Shu X, Liu T, Wang Y, Wang B, Li S (2020) Joint extraction of entities and relations based on a novel decomposition strategy. In: Giacomo GD, Catalá A, Dilkina B, Milano M, Barro S, Bugarín A, Lang J (eds) ECAI 2020–24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), Frontiers in Artificial Intelligence and Applications, vol 325. IOS Press, pp 2282–2289

45. Zeng D, Zhang H, Liu Q (2020) Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 9507–9514. AAAI Press
46. Zeng X, He S, Zeng D, Liu K, Liu S, Zhao J (2019) Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In: K. Inui, J. Jiang, V. Ng, X. Wan (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 367–377. Association for Computational Linguistics
47. Zeng X, Zeng D, He S, Liu K, Zhao J (2018) Extracting relational facts by an end-to-end neural model with copy mechanism. In: I. Gurevych, Y. Miyao (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pp. 506–514. Association for Computational Linguistics
48. Zhang M, Zhang Y, Fu G (2017) End-to-end neural relation extraction with global optimization. In: M. Palmer, R. Hwa, S. Riedel (eds.) Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pp. 1730–1740. Association for Computational Linguistics
49. Zhang N, Deng S, Ye H, Zhang W, Chen H (2022) Robust triple extraction with cascade bidirectional capsule network. Expert Syst. Appl. 187:115806
50. Zhao S, Hu M, Cai Z, Liu F (2020) Modeling dense cross-modal interactions for joint entity-relation extraction. In: C. Bessiere (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 4032–4038. ijcai.org
51. Zheng S, Wang F, Bao H, Hao Y, Zhou P, Xu B (2017) Joint extraction of entities and relations based on a novel tagging scheme. In: R. Barzilay, M. Kan (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pp. 1227–1236. Association for Computational Linguistics
52. Zhong Z, Chen D (2021) A frustratingly easy approach for entity and relation extraction. In: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pp. 50–61. Association for Computational Linguistics

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.