Neural Processing Letters (2024) 56:165
https://doi.org/10.1007/s11063-024-11616-x

A Parallel Model for Jointly Extracting Entities and Relations

Zuqin Chen[1] · **Yujie Zheng**[1] · **Jike Ge**[1] · **Wencheng Yu**[1] · **Zining Wang**[1]

Accepted: 6 April 2024 / Published online: 7 May 2024

Abstract

Extracting relational triples from a piece of text is an essential task in knowledge graph construction. However, most existing methods either identify entities before predicting their relations, or detect relations before recognizing associated entities. This order may lead to error accumulation because once there is an error in the initial step, it will accumulate to subsequent steps. To solve this problem, we propose a parallel model for jointly extracting entities and relations, called PRE-Span, which consists of two mutually independent submodules. Specifically, candidate entities and relations are first generated by enumerating token sequences in sentences. Then, two independent submodules (Entity Extraction Module and Relation Detection Module) are designed to predict entities and relations. Finally, the predicted results of the two submodules are analyzed to select entities and relations, which are jointly decoded to obtain relational triples. The advantage of this method is that all triples can be extracted in just one step. Extensive experiments on the WebNLG*, NYT*, NYT and WebNLG datasets show that our model outperforms other baselines at 94.4%, 88.3%, 86.5% and 83.0%, respectively.

Keywords Joint learning · Named entity recognition · Relation extraction · Multi-head attention

1 Introduction

Relational triple extraction aims to recognize all entities and semantic relations between entities from unstructured text, which is widely used in various downstream tasks, such as knowledge graph construction [1] and question answering [2].

✉ Zuqin Chen
chenzuq81@163.com

Yujie Zheng
2424263336@qq.com

Jike Ge
576315110@qq.com

Wencheng Yu
littlefish@88.com

Zining Wang
1003283741@qq.com

[1] Chongqing University of Science and Technology, Chongqing, China

Traditional pipeline approaches [3, 4] first identify all entities in a sentence, and then classify the relations between each entity pair. Although these methods are flexible, they suffer from error propagation. To address this shortcoming, joint feature-based extraction models [5, 6] are proposed. However, these methods often rely heavily on external NLP tools and require sophisticated feature engineering.

Recently, deep neural network models based on joint extraction have attracted a lot of interest from researchers. Sun et al. [7] have developed a graph convolutional network model of an entity-relation bipartite graph, which allows joint inference of entity and relation types. Wei et al. [8] and Ren et al. [9] tackled the triple extraction task in two steps, first identifying the head entities, and then detecting multiple tail entities under specific relations. TDEER [10] employed a translation decoding strategy that treats relations as translation operations from head entities to tail entities. RIFRE [11] represented words and relations as nodes on a graph and fuse them to obtain a more efficient representation of nodes for the relation extraction task. RFBFN [12] transformed triple extraction into first detected relations and then recognized entities.

Although the above methods achieve promising performance, most of them often detect entities and relations sequentially, which may lead to error accumulation. Inspired by the above ideas, we propose a parallel model for jointly extracting entities and relations (PRE-Span) in this paper, which consists of two mutually independent submodules. Specifically, for a given sentence, our method generates candidate entities and relations by enumerating token sequences based on span length. Then, the Entity Extraction Module and the Relation Detection Module are designed to perform entity recognition and relation detection, respectively. Finally, the prediction results from two submodules are filtered to retain only those predicted to be entities and relations, which are then decoded jointly. However, enumerating token sequences in this way generates a large number of negative samples. To overcome this problem, we randomly remove some negative samples with downsampling. Extensive experiments are conducted on public datasets (WebNLG*, NYT*, NYT and WebNLG) and the results demonstrate that our method achieves strong competition.

In summary, the main contributions of our work are as follows:

(1) We propose an end-to-end model that transforms the relational triple extraction task into two mutually independent and parallel-executed submodules, which can effectively solve error accumulation.

(2) Unlike most previous methods, the proposed PRE-Span can simultaneously detect both entities and relations in sentences, and the features between submodules do not interfere with each other. This method extracts all the triples in a sentence in just one step.

(3) Extensive experiments are conducted on several datasets (WebNLG*, NYT*, NYT and WebNLG) and the results show that our method outperforms previous baselines.

2 Related Work

In recent years, various neural network models based on joint learning have been proposed by researchers. According to the relational triple extraction procedure, related works can be broadly divided into the following three categories [13]: sequence labeling, table filling and text generation.

The first class is sequence labeling, which converts entity recognition and relation classification into a sequence labeling problem. Zheng et al. [14] and Luo et al. [15] proposed a sophisticated tagging schema that allows both entities and relations to be tagged simultaneously, without the need to identify them separately. Zheng et al. [16] predicted potential relations and constrains subsequent entity extraction to the predicted relation subset, instead of all relations. Wang et al. [17] improved the accuracy of the triple extraction task by integrating the semantic role attention mechanism with position awareness and the attention mechanism based on semantic feature vectors. To effectively leverage correlations between semantic relations, Wang et al. [18] proposed a tensor learning model based on Tucker decomposition, which used a three-dimensional word relation tensor to depict the relations between words within a sentence. Jiang et al. [19] designed an entity and relation heterogeneous graph attention network, comprising word nodes, subject nodes, and relation nodes. This architecture aims to learn and enhance semantic information between entities and relations.

The second class is table filling, which treats relational triple extraction as a table filling problem. Fu et al. [20] proposed a relation-weighted graph convolutional network to improve relation extraction by accounting for the interaction of information between named entities and relations. Wang et al. [21] designed a one-stage joint extraction model, TPLinker, that converts joint extraction into a token pair linking problem and introduces a novel handshaking tagging scheme to align boundary tokens of entity pairs under each relational type. Wang et al. [22] used a unified classifier to predict the label of each cell so that information between entities and relations could be better learned. Shang et al. [23] treat the joint extraction task as a fine-grained triple classification to tackle the challenge posed by the interdependence and indivisibility of the three components within a triple. Ren et al. [9] proposed a global feature-oriented model for relational triple extraction that enhances the global associations between relations and token pairs. Gao et al. [24] proposed a novel lightweight joint extraction model based on a global entity matching strategy, which uses relation attention to fuse candidate relations into the entity recognition module to identify entities in sentences more accurately. Wang et al. [25] devised a W-shaped DNN (WNet) to capture coarse-level high-order connections, aiming to encompass more comprehensive information than first-order word-by-word interactions.

The third class is text generation, which uses the encoder-decoder framework to generate relational triples. Zeng et al. [26] developed an end-to-end model to generate the relation and its corresponding entities through a copy mechanism, but it is limited to predicting only the last word of an entity. To address this limitation, CopyMTL [27] can effectively identify entities with multiple tokens. TransRel [28] was a novel unified translation framework that addresses redundant predictions, overlapping triplets, and relational connections simultaneously. Huang et al. [29] used encoder-to-decoder to decompose relational triple extraction into two subtasks and capture the connection information between them via a partition filter network.

However, most existing methods detect entities and relations sequentially, which may result in error accumulation if the initial step is incorrectly identified. Unlike previous methods, our proposed model consists of two mutually independent submodules and uses the output features of the BERT encoder as their input. Therefore, it would effectively solve the above problem.

3 Proposed Model

In this section, we first define the relational triple extraction task in Sect. 3.1. Next, the generating principle for candidate entities and relations is introduced in Sect. 3.2. Subsequently, the Entity Extraction Module and the Relation Detection Module are described in detail in

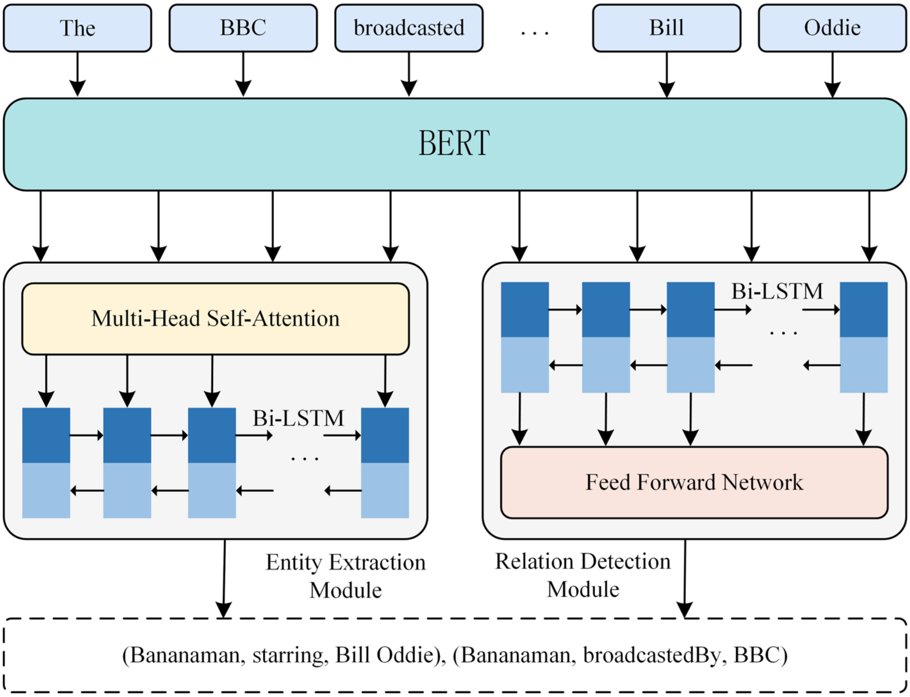

The
BBC
broadcasted
. . .
Bill
Oddie
BERT
Multi-Head Self-Attention
Bi-LSTM
. . .
Entity Extraction Module
Bi-LSTM
. . .
Feed Forward Network
Relation Detection Module
(Bananaman, starring, Bill Oddie), (Bananaman, broadcastedBy, BBC)

Fig. 1 The overall architecture of the end-to-end model

Sects. 3.3 and 3.4, respectively. To effectively train our model, a joint learning method is introduced in Sect. 3.5. Finally, a specific decoding process is described for two submodules in Sect. 3.6. Figure 1 shows an overview architecture of the proposed model.

3.1 Task Definition

The purpose of relational triple extraction is to recognize all entities and their corresponding relations in sentences. Given a sentence with n tokens $X = (x_1, x_2, ..., x_n)$, our model is designed to detect all possible triples $T(X) = \{(s, r, o)\,|s, o \in \mathrm{E}, r \in \mathcal{R}\}$, where E is the head and tail entities of the triples, and $\mathcal{R}$ is the set of predefined relation types.

3.2 Constructing Candidate Entities and Relations

We generate candidate entities and relations by enumerating all consecutive token sequences with a span length that is less than the sentence length. For example, the sentence "The BBC broadcasted Bananaman which starred Bill Oddie", has two triples: (Bananaman, starring, Bill Oddie) and (Bananaman, broadcastedBy, BBC). All candidate entities {"The", "The BBC", "The BBC broadcasted", …, "Bill Oddie", "Oddie"} are generated based on the span length, as described in previous works [12, 30]. For relations, candidate relations are also generated by enumerating token sequences, except that the span length is not set. It is worth noting that positive and negative samples are separated when enumerating token sequences, and the threshold of candidate entities and relations is set 100. If the total sample size exceeds a certain threshold, N negative samples ($N = 100$ - number of positive samples) are randomly selected and mixed with the positive samples. Otherwise, all negative samples are mixed into positive samples. We denote the subset of candidate entities and relations as ε^e and ε^r.

3.3 Entity Extraction Module

To improve the performance of the entity recognition task, the component consists of Multi-head Self-Attention and Bi-LSTM, which effectively capture contextual representation. Specifically, we use the BERT encoder to obtain the contextual representation of each token. Let $S = [s_1, s_2, \ldots, s_n]$ denote all the feature representations in X, where $S \in \mathbb{R}^{n \times d}$, n is the length of a sentence and d is the embedding dimension. The output S from the BERT encoder is fed into the Multi-Headed Self-Attention layer, which aims to project the hidden representation into different subspaces and learn them individually. The formulas for the Multi-Head Self-Attention layer are presented below:

$$\begin{aligned} head_l &= Attention\left(QW_l^q, KW_l^k, VW_l^v\right) \\ \text{Multihead} &= \text{Concat}\left(\text{head}_1, \text{head}_2, ..., \text{head}_l\right) W^o \end{aligned} \tag{1}$$

where, K, Q and V are derived from matrix S, W_l^q, W_l^k, W_l^v and W_l^o are trainable weights and l is the number of heads in Multi-Head Self-Attention. We represent the outputs as $M = [m_1, m_2, \ldots, m_n]$, with n is the number of tokens in the sentence. Then, the M is input to Bi-directional Long Short-Term Memory (Bi-LSTM) for encoding, which calculates the current forward LSTM $\overrightarrow{h_i^e}$ and backward one $\overleftarrow{h_i^e}$ based on the previous hidden state h_{i-1}, the memory cell c_{i-1} and the current word vector m_i. The detailed formulas for the Bi-LSTM are as follows:

$$\begin{aligned} \overrightarrow{h_i^e} &= \text{LSTM}\left(m_i, \overrightarrow{h_{i-1}}, \overrightarrow{c_{i-1}}\right) \\ \overleftarrow{h_i^e} &= \text{LSTM}\left(m_i, \overleftarrow{h_{i-1}}, \overleftarrow{c_{i-1}}\right) \end{aligned} \tag{2}$$

The $\overrightarrow{h_i^e}$ and $\overleftarrow{h_i^e}$ are concatenated as a sequence-level representation of the m_i. The representation of it can be denoted as:

$$h_i^e = \left[\overrightarrow{h_i^e}; \overleftarrow{h_i^e}\right] \tag{3}$$

The final outputs of the Bi-LSTM are $H^e = \left[h_1^e, h_2^e, \ldots, h_n^e\right]$ after computing the hidden state for each token in a sentence. Finally, based on the candidate entities $e_i \in \varepsilon^e$ mentioned in Sect. 3.2, the corresponding word vectors from Bi-LSTM are selected and subjected to a max pooling operation before passing through the linear layer for classification. The max pooling and linear classification formulas are as follows:

$$e_i = \text{Maxpool}\left(\left[X_{start(i)}^e; X_{end(i)}^e\right]\right) \tag{4}$$

$$\text{Ent}_i = W_e e_i + b_e \tag{5}$$

where, $X_{start(i)}^e$ and $X_{end(i)}^e$ are the contextual presentations of the boundary tokens, $W_e \in \mathbb{R}^{d \times n_e}$ and $b_e \in \mathbb{R}^{1 \times n_e}$ are trainable weights, d is the word vector's dimension and n_e is the size of the tag set.

3.4 Relation Detection Module

Considering the potential mutual influence between the two submodules, along with the design principles of the Relation Detection Module, we combine Bi-LSTM and Feed Forward Network. This combination is intended to capture local contextual representations within sentences to avoid introducing additional noise. To elaborate, we input the features extracted by the BERT encoder into Bi-LSTM to facilitate the acquisition of contextual representations. For vectorized token r_i, the final hidden state h_i^r is obtained by concatenating the features of the forward LSTM $\overrightarrow{h_i^r}$ and the backward one $\overleftarrow{h_i^r}$, as follows:

$$h_i^r = \left[\overrightarrow{h_i^r}; \overleftarrow{h_i^r}\right] \tag{6}$$

Therefore, the final output representation of Bi-LSTM is denoted as $H^r = \left[h_1^r, h_2^r, \ldots, h_n^r\right]$, where h_i^r is the hidden state of the i-th tokens and n is the length of the sentence. Next, the Feed Forward Network (FFN) is connected behind the Bi-LSTM and *Relu* is used as the activation function. The formula for FFN is as follows:

$$\text{Re} = \text{Relu}\left(WH^r + b\right) \tag{7}$$

where, $W \in \mathbb{R}^{d\times n_r}$ and $b \in \mathbb{R}^{1\times n_r}$ are trainable weights. Then, the candidate relations constructed in Sect. 3.2 are used to obtain the FFN feature representation and perform max pooling. Finally, a linear layer is used to predict the type of candidate relations. The detailed formulas are as follows:

$$t_i = \text{Maxpool}\left(\left[X_{start(i)}^r; X_{end(i)}^r\right]\right) \tag{8}$$

$$Relat_i = W_r t_i + b_r \tag{9}$$

where, $X_{start(i)}^r$ and $X_{end(i)}^r$ are the contextual presentations of the boundary tokens, $W_r \in \mathbb{R}^{d\times n_r}$ and $b_r \in \mathbb{R}^{1\times n_r}$ are trainable weights, d is the word vector's dimension and n_r is the number of relation types.

3.5 Joint Training

To enable the two submodules to learn the features of the BERT encoder, different learning rates are set for them. We adopt cross-entropy loss as the loss function for the two submodules, and the total loss can be divided into two parts, as follows:

$$\mathcal{L}_{ent} = \sum_{i=1}^{k} \log P\left(y_i^* = \hat{l}^*\right) \tag{10}$$

$$\mathcal{L}_{rel} = \sum_{j=1}^{n} \log P\left(y_j^* = \hat{t}^*\right) \tag{11}$$

where, k is the number of entity types, $\hat{l}^*$ is the ground truth of the candidate entity, n is the number of relational types, $\hat{t}^*$ is the true tag of the candidate relation. The total loss is the sum of the Entity Extraction Module and the Relation Detection Module losses, as follows:

$$\mathcal{L} = \mathcal{L}_{ent} + \mathcal{L}_{rel} \tag{12}$$

3.6 Joint Decoder

In the sentence discussed in Sect. 3.2, (“BBC”, “BBC”), (“Bananaman”, “Bananaman”) and (“Bill”, “Oddie”) are predicted to be 1, which means “BBC”, “Bananaman” and “Bill Oddie” are entities. For relation, (“Bananaman”, “BBC”) and (“Bananaman”, “Oddie”) are predicted to be 68 and 52, which means that the relations of “Bananaman broadcasted BBC” and “Bananaman which starred Bill Oddie” are “broadcastedBy” and “starring”, respectively. To form triples, we first construct entity pairs: < “BBC”, “BBC” >, < “BBC”, “Bananaman” >, …, < “Bananaman”, “Bill Oddie” > and < “Bill Oddie”, “Bill Oddie” >. Then, the longest segments are identified as “BBC”, “BBC broadcasted Bananaman”, …, “Bananaman which starred Bill Oddie” and “Bill Oddie”. Finally, if segments and relations are matched, the corresponding entity pairs and relations could form relational triples. Based on the above steps, two triples can be decoded: (Bananaman, broadcastedBy, BBC) and (Bananaman, starring, Bill Oddie).

4 Experiments

4.1 Datasets

For a fair and comprehensive comparison with previous works, we evaluate the performance of our model using NYT [31] and WebNLG [32] datasets, respectively.

- NYT: The dataset is generated by a distantly supervised relation extraction task, and was automatically aligned through Freebase’s relational facts and the New York Times (NYT) corpus. It contains 150 business articles from the New York Times, of which 56k are training sentences and 5k are test sentences.
- WebNLG: It was created for the Natural Language Generation (NLG) task, which contains 5k training sentences and 703 test sentences.

Both of the above datasets have another version, which is annotated only with the last word of the entities. Following the convention of previous works [12, 13], we refer to them as NYT* and WebNLG*. The datasets used in our experiments were provided by Zheng et al. [16] and the detailed statistical results are shown in Table 1.

4.2 Metrics

To be consistent with prior works [33–35], we use standard Precision(Prec), Recall(Rec) and F1-score(F1) as evaluation metrics for our model, as follows:

$$\begin{aligned}\text{Prec} &= \frac{\text{TP}}{\text{TP}+\text{FP}}\\ \text{Rec} &= \frac{\text{TP}}{\text{TP}+\text{FN}}\\ \text{F1} &= \frac{2\times\text{Prec}\times\text{Rec}}{\text{Prec}+\text{Rec}}\end{aligned} \tag{13}$$

where, TP represents the number of correctly predicted triples; FP represents the number of mispredicted triples; FN represents the number of true triples in the corpus that were not predicted.

Table 1 Statistics on WebNLG, NYT, WebNLG* and NYT* datasets

Category	Dataset			Detail of test set							
	Train	Valid	Test	Normal	SEO	EPO	N=1	N=2	N=3	N=4	N≥5
NYT*	56195	4999	5000	3266	1297	978	3244	1045	312	291	108
WebNLG*	5019	500	703	245	457	26	2666	171	131	90	45
NYT	56196	5000	5000	3084	1264	1161	3081	1182	299	316	155
WebNLG	5019	500	703	245	422	4	262	177	131	92	41

Note that there may be different overlapping patterns and a different number of triples in sentences

Table 2 Hyperparameter settings in our model

Hyperparameter	Setting
Batch size	16
Ner length	8
Dropout rate	0.3
Epoch	30 or 100
Clip norm	5.0
Pretrained model learning rate	0.00001
Ner learning rate	0.0001
Re learning rate	0.001

4.3 Experimental Settings

Our model is implemented with PyTorch and uses Adamw to optimize its network weights. We use bert-base-cased [36] as a sentence feature encoder and fine-tune the parameters during training. At the same time, dropout is added to the Relation Detection Module to prevent overfitting of the model. To ensure experimental rigor, all of our experiments are performed on the same device with RTX3060 GPU, AMD R5-5500 CPU and 24 G RAM. The settings of our model hyperparameters are shown in Table 2.

4.4 Compared Method

We compare our method with the following baseline models:

- NovelTagging [14]: The model converted the joint extraction task into a tagging problem, which can extract triples directly without independently identifying entities and their relations.
- CopyRE [26]: The sequence-to-sequence model with copy mechanism that attempted to solve different types of triples.
- MultiHead [37]: The model first identified all entities in a given sentence and then transformed the relation extraction task into a multi-headed selection problem.
- GraphRel [20]: The model facilitated the interaction between entities and relations through a relation-weighted GCN for better relation extraction.
- OrderCopyRE [38]: The method applied reinforcement learning to a sequence-to-sequence model to generate multiple triples in a specific order.
- ETL-Span [39]: The model decomposed the joint extraction task into two associated subtasks and implemented triple extraction by a hierarchical boundary tagger and a multi-span decoding algorithm.
- ImGraph [40]: The GCN model based on a relation-aware attention mechanism was designed to connect entities and relations graphically.
- RSAN [41]: A relation-specific attention network model was proposed to address redundancy in relation prediction.
- CasRel [42]: To tackle the challenge of triples overlap, a novel cascade binary tagging framework (CASREL) has been proposed, treating relations as functions that map subjects to objects within a sentence.

- GCN^2-NAA [43]: A novel joint entity-relation extraction model, GCN^2-NAA, was proposed. The model extracts relation triples in stages by Graph Convolutional Neural Networks and a NodeAware Attention mechanism.
- CBCapsule [44]: A Cascade Bidirectional Capsule Network model was proposed, which first aggregates context representations dynamically and then uses bidirectional routing mechanisms to facilitate information interaction between entities and relations.
- RMAN [45]: A joint extraction model of entities and relations, called RMAN, was proposed that encodes sentence representations and decodes sequence annotations through multiple feature fusion.
- ERHGA [46]: A heterogeneous graph attention network with a gate mechanism was proposed to improve the performance of the model by containing world nodes, subject nodes, and relation nodes.

4.5 Main Results

The comparative results of our model against baseline models across all datasets are shown in Table 3. Experimental results demonstrate that our method achieves better performance in Precision, Recall and F1 scores, outperforming almost all other models. Compared to the best model, ERHGA, PRE-Span achieves a significant performance improvement on the WebNLG* dataset, with an increase of 1.1% in F1 score. In addition, among all baselines, our method also achieves better performance on NYT* and NYT datasets. Comparative experiments validate the rationality of the proposed method and show relatively good performance. We attribute this success to the design concept of two independent submodules, as their feature representations are mutually independent, resulting in experimental results that exceed other methods.

We further observe that the performance in the WebNLG dataset is marginally inferior to GCN^2-NAA. Nested entities are found in sentences by analyzing the dataset. For example, in the sentence “NK University is nicknamed Cornell Big Red”, “Cornell” and “Cornell Big Red” are both entities, and the latter includes the former. The test sentences contain a total of 205 triples across 85 sentences with nested entities. We hypothesize that the performance of our model is affected by nested entities. To test this hypothesis, when sentences with nested entities are removed and reevaluated with the trained model, the result shows a significant increase in F1 score to 86.5%. This indicates that although our method has limitations in extracting Subject Object Overlap, for other types of relational triples, they can be effectively identified and extracted by our model without error accumulation. Given the characteristics of our method, we expect it to be widely used in relevant practical scenarios.

4.6 Detailed Results on Complex Scenarios

To further analyze the performance of our model in different overlapping patterns and different number of triples, we conduct experiments on the WebNLG* and NYT* datasets.

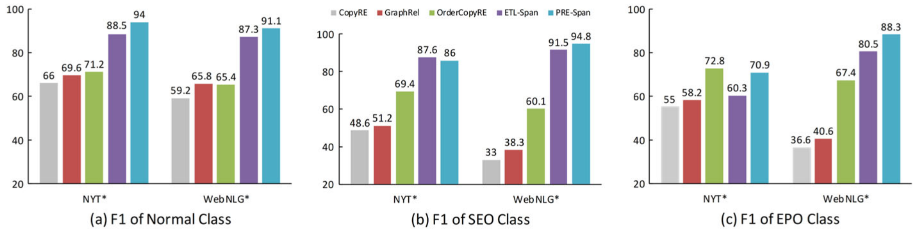
We evaluate sentences containing triples of Normal, SEO and EPO types, and the results are shown in Fig. 2. Our model outperforms all baselines on the WebNLG* dataset, showing significant improvements in different types of triples: Normal, SEO, and EPO by 3.8%, 3.3%, and 7.8%, respectively, compared to ETL-Span. For the NYT* dataset, our method improves by 5.5% in triples of Normal type. However, it performs relatively worse than other types.

In addition, we explore the performance of the model using sentences containing varying numbers of triples. Based on the number of triples in the sentence, WebNLG* and NYT*

Table 3 Results on NYT*, WebNLG* NYT and WebNLG datasets

Models	NYT*			WebNLG*			NYT			WebNLG		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
NovelTagging† [14]	–	–	–	–	–	–	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE¢ [26]	60.4	62.3	58.7	37.7	36.4	37.1	–	–	–	–	–	–
MultiHead† [37]	–	–	–	–	–	–	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel‡ [20]	63.9	60.0	61.9	44.7	41.1	42.9	–	–	–	–	–	–
OrderCopyRE‡ [38]	77.9	67.2	72.1	63.3	59.9	61.6	–	–	–	–	–	–
ETL-Span¢ [39]	87.2	73.7	79.9	84.0	91.5	87.6	85.5	71.7	78.0	83.6	82.7	83.2
ImGraph‡ [40]	54.0	59.7	56.7	–	–	–	83.2	64.7	72.8	66.4	62.7	64.5
RSAN‡ [41]	–	–	–	–	–	–	85.7	83.6	84.6	80.5	83.8	82.1
CasRel¢ [42]	86.6	**89.9**	88.2	90.6	91.0	90.8	–	–	–	–	–	–
GCN^2-NAA‡ [43]	–	–	–	–	–	-	85.1	73.0	78.6	**85.6**	84.3	**84.9**
CBCapsule‡ [44]	88.4	87.2	87.8	90.9	89.2	90.0	–	–	–	–	–	–
RMAN‡ [45]	55.9	58.6	57.2	–	–	–	**87.1**	83.8	85.4	83.6	**85.3**	84.5
ERHGA‡ [46]	–	–	–	94.3	92.3	93.3	–	–	–	–	–	–
PRE-Span	**90.9**	85.7	**88.3**	**95.6**	**93.3**	**94.4**	86.3	**86.8**	**86.5**	84.4	81.6	83.0

The highest score is highlighted in bold. ‡ denotes results directly from the original papers. † denotes results reproduced by us. ¢ denotes results reported by Wang et al. [19]



(a) F1 of Normal Class

(b) F1 of SEO Class

(c) F1 of EPO Class

Fig. 2 F1 score of different overlapping patterns on sentences

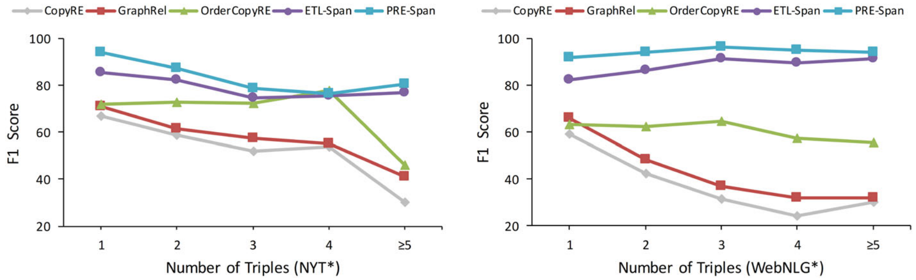


Fig. 3 F1 score of a different number of triples on sentences

datasets can be classified into five groups: 1, 2, 3, 4, and ≥5, as shown in Fig. 3. The results show that with the exception of sentences with four triples in NYT*, the F1 score outperforms all baseline models. It indicates that our method has the ability to extract multiple triples.

4.7 Efficiency Comparison

To show the training efficiency of the methods more clearly, experiments are conducted on the NYT* and WebNLG* datasets, with comparisons made against baselines, as illustrated in Fig. 4.

It is evident from the figure that ETL-Span and CopyRE, which did not use the BERT model, required 165 s and 122 s for training on NYT* and WebNLG* datasets, as well as 11 s and 9 s, respectively. However, CasRel and PRE-Span both use the BERT model and conduct experiments on the same datasets. In comparison, PRE-Span takes only 814 s and 62 s. This indicates that due to the parallelizability of the two submodules in PRE-Span, it takes less time during the training phase.

4.8 Results on Different SubModules

To further investigate the detection capabilities of each component in our model, we conduct more detailed evaluations on the NYT* and WebNLG* datasets. Table 4 shows the results for Precision, Recall and F1. The Entity Extraction Module demonstrates a strong identification capability across all evaluation indices, with a score of over 96%, indicating that the submodule effectively recognizes entities within sentences. The Relation Detection Module performs well on the WebNLG* dataset, but has a lower F1 score of 88.1% on the NYT* dataset.

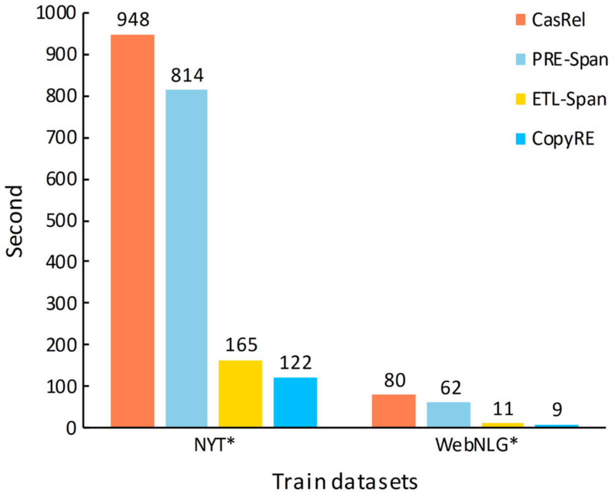


Fig. 4 Time taken for training for each epoch within the NYT* and WebNLG* datasets

Table 4 Results from different submodules of the WebNLG* and NYT* datasets

	Submodule	Prec	Rec	F1
WebNLG*	Entity Extraction Module	97.4	98.5	97.9
	Relation Detection Module	95.1	93.8	94.4
	Combination of the Above All	95.6	92.3	94.4
NYT*	Entity Extraction Module	96.1	96.6	96.3
	Relation Detection Module	89.4	86.9	88.1
	Combination of the Above All	90.9	85.7	88.3

We also observe that the Entity Extraction Module outperformes another submodule across all evaluation indices. It is believed that the module focuses solely on identifying entities in sentences, regardless of their type. Conversely, the Relation Detection Module not only accurately identifies the start and end positions of candidate relations, but also determines their types.

4.9 Ablation Experiment

To insight into the impact of individual submodules on model performance, we conducted an ablation study on the WebNLG* dataset, and the results are shown in Table 5. When the Entity Extraction Module is removed, there is a 0.6% decrease in the F1 score, indicating that the module contributes to improved model performance. Simultaneously removing both submodules results in a 0.9% decrease in the F1 score, suggesting that each submodule contributes to model performance. In addition to the probe components, we study the impact of dropout on the Relation Detection Module. The F1 score drops by 0.4% when the dropout is removed. This suggests that the module is overly complex if certain neurons are not randomly dropped during training. In addition, we also study the impact of freezing the parameters of

Table 5 Performance of different ablations on the WebNLG* dataset. A bold mark indicates the highest score

Model	Prec	Rec	F1
PRE-Span	**95.6**	92.3	**94.4**
Entity Extraction Module	94.7	92.8	93.8
Relation Detection Module	93.7	**93.3**	93.5
Without dropout	94.9	93.1	94.0
Pre-trained parameters(freeze)	92.9	90.7	91.8

the BERT encoder on the downstream task. The result shows that all evaluation indices are down, with Precision being particularly affected by a 2.7% decline.

5 Conclusion and Future Work

In this paper, we propose an end-to-end parallel model comprising two mutually independent modules that can detect both entities and relations in sentences, allowing relational triple extraction in a single step. To verify the validity of our model, extensive experiments are conducted on WebNLG*, NYT*, NYT and WebNLG datasets, and the results show that our method outperforms other baselines. Furthermore, we also explore the impact of submodules on the model, and the ablation study shows that they all contribute to model performance.

However, our model has some limitations. On the one hand, if a sentence contains nested entities, such as “Cornell” and “Cornell Big Red” where “Cornell Big Red” encompasses “Cornell”, this situation may result in the generation of redundant error triples during decoding. On the other hand, the Relation Detection Module encounters challenges in accurately determining the types of candidate relations. We argue that when the number of tokens in a segment is substantial, this submodule could introduce additional information, potentially leading to inaccuracies in detection results.

In future work, we plan to explore other techniques to address the problem of nested entities in relational triples, and implement more advanced approaches to mitigate the challenge of long-distance dependencies in a relation extraction task.

Acknowledgements This research is supported by the National Social Science Foundation Western Project of China under NO. 19XTQ010.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest in this work.

References

1. Dai Quoc Nguyen TDN, Nguyen DQ, Phung D (2018) A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of NAACL-HLT, pp. 327–333 (2018)
2. Hu S, Zou L, Zhang X (2018) A state-transition framework to answer complex questions over knowledge base. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 2098–2108
3. Zelenko D, Aone C, Richardella A (2003) Kernel methods for relation extraction. J Mach Learn Res 3(Feb):1083–1106
4. Chan YS, Roth D (2011) Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp 551–560
5. Yu X, Lam W (2010) Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In: Coling 2010: Posters, pp 1399–1407
6. Miwa M, Sasaki Y (2014) Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1858–1869
7. Sun C, Gong Y, Wu Y, Gong M, Jiang D, Lan M, Sun S, Duan N (2019) Joint type inference on entities and relations via graph convolutional networks. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 1361–1370
8. Nguyen D, Nguyen TD, Nguyen DQ, Phumg Q-D(2018) A novel embedding model for knowledge base completion based on convolutional neutral network
9. Ren F, Zhang L, Yin S, Zhao X, Liu, S, Li B, Liu Y (2021) A novel global feature-oriented relational triple extraction model based on table filling. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2646–2656. https://doi.org/10.18653/v1/2021.emnlp-main.208. https://aclanthology.org/2021.emnlp-main.208
10. Li X, Luo X, Dong C, Yang D, Luan B, He Z (2021) TDEER: an efficient translating decoding schema for joint extraction of entities and relations. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 8055–8064
11. Zhao K, Xu H, Cheng Y, Li X, Gao K (2021) Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. Knowl-Based Syst 219:106888
12. Li Z, Fu L, Wang X, Zhang H, Zhou C (2022) RFBFN: a relation-first blank filling network for joint relational triple extraction. In: Proceedings of the 60th annual meeting of the association for computational linguistics: student research workshop, pp 10–20
13. Shang YM, Huang H, Sun X, Wei W, Mao XL (2022) Relational triple extraction: one step is enough
14. 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: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vol 1, Association for Computational Linguistics, Vancouver, Canada, https://doi.org/10.18653/v1/P17-1113. https://aclanthology.org/P17-1113 1227–1236.pp
15. Luo L, Yang Z, Cao M, Wang L, Zhang Y, Lin H (2020) A neural network-based joint learning approach for biomedical entity and relation extraction from biomedical literature. J Biomed Inform 103:103384
16. Zheng H, Wen R, Chen X, Yang Y, Zhang Y, Zhang Z, Zhang N, Qin B, Ming X, Zheng Y (2021) PRGC: potential relation and global correspondence based joint relational triple extraction. 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), pp 6225–6235
17. Wang Q, Zhang Q, Zuo M, He S, Zhang B (2022) A entity relation extraction model with enhanced position attention in food domain. Neural Process Lett 54(2):1449–1464
18. Wang Z, Nie H, Zheng W, Wang Y, Li X (2023) A novel tensor learning model for joint relational triplet extraction. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2023.3265851
19. Jiang B, Cao J (2023) Joint extraction of entities and relations via entity and relation heterogeneous graph attention networks. Appl Sci 13(2):842
20. Fu T-J, Li P-H, Ma W-Y (2019) GraphRel: modeling text as relational graphs for joint entity and relation extraction. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 1409–1418
21. Wang Y, Yu B, Zhang Y, Liu T, Zhu H, Sun L (2020) TPLinker: single-stage joint extraction of entities and relations through token pair linkingIn: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 1572–1582. https://doi.org/10.18653/v1/2020.coling-main.138. https://aclanthology.org/2020.colingmain.138

22. Wang Y, Sun C, Wu Y, Zhou H, Li L, Yan J. UniRE: a unified label space for entity relation extraction In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Vol 1, Association for Computational Linguistics, Online, 220–231. https://doi.org/10.18653/v1/2021.acl-long.19. https://aclanthology.org/2021.acl-long.19
23. Shang Y-M, Huang H, Mao X (2022) OneRel: joint entity and relation extraction with one module in one step. Proceedings of the AAAI conference on artificial intelligence 36:11285–11293
24. Gao C, Zhang X, Li L, Li J, Zhu R, Du K, Ma Q (2023) ERGM: a multi-stage joint entity and relation extraction with global entity match. Knowl-Based Syst 271:110550
25. Wang Y, Sun C, Wu Y, Li L, Yan J, Zhou H (2023) HIORE: leveraging high-order interactions for unified entity relation extraction. arXiv preprint arXiv:2305.04297
26. 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: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 506–514
27. Zeng D, Zhang H, Liu Q (2020) CopyMTL: copy mechanism for joint extraction of entities and relations with multi-task learning. Proceedings of the AAAI conference on artificial intelligence 34:9507–9514
28. Huang H, Shang Y-M, Sun X, Wei W, Mao X (2022) Three birds, one stone: a novel translation based framework for joint entity and relation extraction. Knowl-Based Syst 236:107677
29. Huang Z, Liang L, Zhu X, Weng H, Yan J, Hao T (2022) An improved partition filter network for entity-relation joint extraction. In: Neural computing for advanced applications: third international conference, NCAA 2022, Jinan, China, July 8–10, 2022, Proceedings, Part I. Springer, pp 129–141
30. Zhong Z, Chen D (2021) A frustratingly easy approach for entity and relation extraction. In: 2021 conference of the north American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2021. Association for Computational Linguistics (ACL), pp 50–61
31. Riedel S, Yao L, McCallum A (2010) Modeling relations and their mentions without labeled text. In: Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21. Springer, pp 148–163
32. 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)
33. Sui D, Zeng X, Chen Y, Liu K, Zhao J (2023) Joint entity and relation extraction with set prediction networks. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2023.3264735
34. Yan Z, Zhang C, Fu J, Zhang Q, Wei Z (2021) A partition filter network for joint entity and relation extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 185–197. https://doi.org/10.18653/v1/2021.emnlpmain.17. https://aclanthology.org/2021.emnlp-main.17
35. Ren F, Zhang L, Zhao X, Yin S, Liu S, Li B (2022) A simple but effective bidirectional framework for relational triple extraction. In: Proceedings of the fifteenth ACM international conference on web search and data mining, pp 824–832
36. Bekoulis G, Deleu J, Demeester T, Develder C (2018) Bert: pre-training of deep bidirectional transformers for language understanding. Expert Syst Appl 114:34–45
37. 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
38. 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: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 367–377
39. 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
40. Hong Y, Liu Y, Yang S, Zhang K, Wen A, Hu J (2020) Improving graph convolutional networks based on relation-aware attention for end-to-end relation extraction. IEEE Access 8:51315–51323
41. Yuan Y, Zhou X, Pan S, Zhu Q, Song Z, Guo L (2020) A relation-specific attention network for joint entity and relation extraction. IJCAI 2020:4054–4060
42. Wei Z, Su J, Wang Y, Tian Y, Chang Y (2020) A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 1476–1488. https://doi.org/10.18653/v1/2020.acl-main.136. https://aclanthology.org/2020.acl-main.136
43. Niu W, Chen Q, Zhang W, Ma J, Hu Z (2021) GCN2-NAA: two-stage graph convolutional networks with node-aware attention for joint entity and relation extraction. In: 2021 13th international conference on machine learning and computing, pp 542–549

44. Zhang N, Deng S, Ye H, Zhang W, Chen H (2022) Robust triple extraction with cascade bidirectional capsule network. Expert Syst Appl 187:115806
45. Lai T, Cheng L, Wang D, Ye H, Zhang W (2022) RMAN: relational multi-head attention neural network for joint extraction of entities and relations. Appl Intell 52(3):3132–3142
46. Jiang B, Cao J (2023) Joint extraction of entities and relations via entity and relation heterogeneous graph attention networks. Appl Sci 13(2):842