



# An Improved Partition Filter Network for Entity-Relation Joint Extraction

Zhenjie Huang<sup>1</sup>, Likeng Liang<sup>1</sup>, Xiaozhi Zhu<sup>1</sup>, Heng Weng<sup>2</sup>, Jun Yan<sup>3</sup>,  
and Tianyong Hao<sup>1</sup>(✉)

<sup>1</sup> School of Computer Science, South China Normal University, Guangzhou, China  
{2020022962, lianglikeng, 2020022975, haoty}@m.scnu.edu.cn

<sup>2</sup> State Key Laboratory of Dampness, Syndrome of Chinese Medicine, The Second Affiliated  
Hospital of Guangzhou University of Chinese Medicine, Guangzhou, China  
wengh@gzucm.edu.cn

<sup>3</sup> AI Lab, Yidu Cloud (Beijing) Technology Co., Ltd., Beijing, China  
Jun.yan@yiducld.com

**Abstract.** The purpose of a joint entity-relation extraction task is to extract entity-relation triples from unstructured text to assist text analysis, knowledge graph construction, etc. The existing sequence-to-sequence or sequence-to-non-sequence models treat the joint extraction task as a triple generation task, sharing the feature space of entity and relation extraction in the same structure. However, fusing the information of both subtasks may cause the problem of feature conflicts and thus decrease model performance. In order to enable each extraction subtask has its own independent feature space to reduce feature conflicts, this paper proposes a dual-decoder to decode entity extraction subtask and relation extraction subtask separately based on an encoder-to-decoder structure. A Dual-Joint-Input-PFN model is proposed by improving the partition filter network as an interaction to capture connection information between two subtasks. The model consists of two Joint-Input-PFNs layers, and each layer accepts two inputs simultaneously and filters the other input according to one of them. The experiments are based on standard datasets WebNLG and NYT, and the effectiveness of the proposed model is verified by comparing with the state-of-the-art baseline methods.

**Keywords:** Joint entity-relation extraction · Dual-decoder · Information extraction

## 1 Introduction

Both entity extraction and relation extraction are fundamental and critical tasks for information extraction in natural language processing. The extracted entity-relation triples can be applied to various downstream tasks, such as automatic knowledge graph construction. The early studies [1, 2] employed the pipeline approach, which extracts entities and relations sequentially. In the pipeline manner, as relation extraction depended on entity extraction, errors such as missing or incorrect entities in entity extraction were propagated to relation extraction, and were amplified [3].

In recent years, more and more studies paid attention to joint extraction approaches, which combine entity and relation extraction by multi-task learning and accomplish the two subtasks within one model. Various joint extraction approaches had been proposed. Table filling-based approach utilized a table structure to achieve joint extraction [3–6]. However, this approach required much computational resources during training. Tagging-based approach [7–11] designed novel tagging methods for extracting entities and relations simultaneously. However, elaborately designing a complex and relatively reasonable tagging method required much expertise. Sequence-to-sequence approach [12] treated joint extraction as a triple extraction task. It extracted triples by a sequence generation model, and was beneficial to solve the relation overlap problem. Nevertheless, the construction of the joint extraction task as a sequence generation task led to increased exposure bias since there was no order information among triples. In order to avoid the bias issue, some researches employed sequence-to-non-sequence approaches. Zhang et al. [13] constructed a seq2tree method, while Sui et al. [14] developed a seq2set method using a non-autoregressive encoder-decoder. All of these models were able to alleviate negative impact caused by the exposure bias.

However, both sequence-to-sequence and sequence-to-non-sequence approaches employed merely one set of encoder-decoders to construct features of the two subtasks. The parameters that were exclusive to each of the subtasks were generally the last parameters utilized for classification. This structure assumed that the features of the subtasks were mutually compatible and conflict-free. However, Zhong et al. [15] mentioned that there was a high possibility of feature conflicts between the two subtasks, which might significantly limit the performance of models.

Encoder in deep learning models can be shared by both subtasks in encoding phase. This helps the encoder learn to extract more useful information from input, since multi-task learning integrates the losses of both subtasks during training. However, in decoding phase, dual decoders may help to avoid the feature conflict problem. To this end, a novel model named as Dual-Joint-Input-PFN-Decoder is proposed in this paper. The Dual-Joint-Input-PFN-Decoder is based on the seq2set structure of SPN4RE [14] and integrates a Dual-Joint-Input-PFN strategy into dual-decoder. The Dual-Joint-Input-PFN is implemented by two Joint-Input-PFN layers which are proposed based on Partition Filter Network (PFN) [16]. The original PFN is not applicable to be utilized in dual-decoder directly. One reason is that feature structure constructed by dual-decoder is different from that in the original PFN and the other reason is that dual-decoder needs to construct interactions for both features. Thus, Joint-Input-PFN is proposed by improving the original PFN and it receives two features as input and extract favorable interaction from one feature based on another feature. In order to extract the interaction that is beneficial to both subtasks simultaneously, the Dual-Joint-Input-PFN strategy is constructed based on the pairwise Joint-Input-PFN. The strategy captures interaction features from two features that are beneficial to both subtasks and ensure that there is no feature conflict during the construction of the interactions. Based on the Dual-Joint-Input-PFN strategy, this paper decodes the two subtasks separately with a dual-decoder network incorporating the Dual-Joint-Input-PFN strategy for avoiding feature conflicts.

The main contributions of the paper lie on three-fold:

- 1) A new Dual-Joint-Input-PFN strategy is proposed by incorporating two Joint-Input-PFN strategies improved from Partition Filter Network for construct interactions between entity and relation extractions.
- 2) A new Dual-Joint-Input-PFN-Decoder model integrates the Dual-Joint-Input-PFN strategy into dual-decoder structure is proposed to utilize interactions of entity and relation extraction for reducing feature conflicts.
- 3) The proposed model achieves the best performance on two standard datasets compared with state-of-the-art baseline methods, demonstrating its effectiveness.

## 2 Related Work

The entity-relation extraction task is the fundamental task of many downstream tasks, and the aim of the task is to extract all entity-relation triples from given a sentence. Existing research on joint entity-relation extraction can be divided into categories of pipeline-based models, Table-filling-based models, tagging-based models, seq2seq models, and Multitask learning-based models.

The pipeline-based models [1, 2] were characterized by first extracting entities, and then classifying relations between the entities. However, these models might easily lead to the accumulation of errors. For instance, if a correct entity was missed in entity extraction task, the relations related to this entity could not be extracted correctly in relation extraction. In a backpropagation manner, the information utilized to correct errors can only flow from the relation extraction task to the entity extraction task, not from the entity extraction task to the relation extraction task, resulting in the failure of the models in utilizing connection information between the two subtasks. The Table-filling-based models [3–6] constructed relations between each pair of tokens in given a sentence with the help of table structure, and achieved entity extraction and relation extraction according to relations between tokens. This structure could well solve the problem of triples in overlapping. However, the scale of table structure and the length of sentences were quadratic, so the models often needed to consume a lot of computational resources. The tagging-based models [7–11] elaborated a novel tagging approach to triples extraction. These models could focus on extracting triples with different characteristics by adopting different tagging methods. However, they often required meticulous and complex human involvement consuming a huge amount of time to design an appropriate tagging strategy. Based on the structure of seq2seq models [12, 13, 17], these models adopted a similar encoder-decoder approach to extract entity-relation triples. They could achieve the entity-relation extraction task with excellent performance of existing translation models and could overcome the entity overlapping problem. Although these models had serious exposure bias problem at first, the exposure bias had decreased to be a mainstream problem after continuous improvement. However, there was always a problem of feature conflict with these models. These models tended to employ merely one set of structures to complete the joint extraction task, but the information between the two subtasks was not always beneficial for both, especially the closer to the downstream task, the more the two subtasks have task-specific features. Therefore, the feature conflict brought by mixing two features could limit the performance of the models. Secondly, after separating the two subtasks, an interaction mechanism needed to be constructed to ensure that the association information between the two subtasks were not be lost.

Multitask learning utilized connection information between tasks to integrate multiple tasks into a single model. Joint extraction can be considered as a multi-task learning task. Wang et al. [3] and Sun et al. [18] built interaction mechanisms for the entity extraction and relation extraction through which the model could capture the connection information of the two tasks and thus promote overall model performance. However, these interaction mechanisms did not filter entity and relation features, and direct fusion of two features to construct interaction led to the feature conflict issue.

In the joint entity-relation extraction task, encoding sentences to obtain appropriate features could further improve the performance of models. In early research, there were some other networks utilized as encoders for entity and relation extraction tasks, including CNN, LSTM, GRU, GNN, and GCN. With the emergence of large-scale pre-trained language models and performance breakthroughs achieved by them in various NLP tasks, more and more models have begun to utilize these language models as encoders or embedding layers the extraction tasks to better capture semantic information in sentences. BERT [19] was a pre-trained language model that was obtained by training on a large-scale corpus employing a multilayer Transformer encoder [20].

In this paper, the structure of SPN4RE [14] is used as a backbone structure for improvement. In order to solve the problems of feature conflicts, decoder in the original structure improved to be a dual-decoder. In addition, we propose an improved Partition Filter Network strategy into the Dual-Joint-Input-PFN-Decoder model to generate an interaction mechanism for entity extraction and relation extraction to enhance connections during of forward information propagation.

### 3 Methods

This paper proposes a new Dual-Joint-Input-PFN-Decoder model on by taking the sequence-to-set framework of Sui et al. [14] as a backbone structure for jointly extracting entities and relations in sentences. Our model integrates dual-decoder with Dual-Joint-Input-PFN strategy and Dual-Joint-Input-PFN is implemented by two Joint-Input-PFN strategies for performance improvement. The overall network structure is as shown in Fig. 1. Dual-Joint-Input-PFN-Decoder model needs to generate a fixed size set predictions for each sentence, and the input of the model is initialized by a fixed-size number of learnable embeddings that termed as triple embeddings. After encoding sentences by an encoder, sentence features are extracted and input into the model to transform triple embeddings into output features. Afterwards, the model compares output features with the corresponding labels through bipartite matching and calculates the loss. The improved Dual-Joint-input-PFN-Decoder model receive two inputs at the same time to build interactions between entity extraction and relation extraction and is able to separately decode entity and relation features.

Given a sentence  $s = \{x_1, x_2, \dots, x_n\}$ ,  $x_i$  denotes a token and  $n$  denotes the length of the sentence. The model encodes  $s$  by pre-trained BERT [19]. The encoding output is denoted as  $H^E \in R^{l \times d}$ , where  $l$  is the length of the encoded sentence containing the three specified symbols [CLS], [SEP], and [PAD], while  $d$  is the hidden dimension size of hidden features.

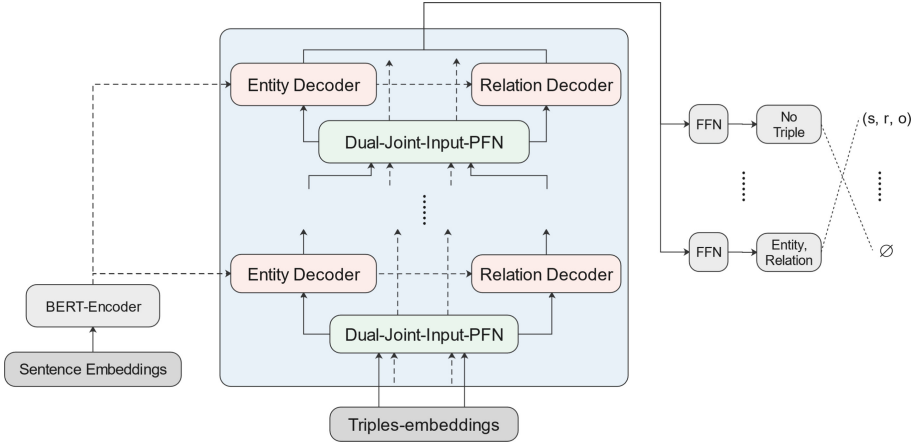
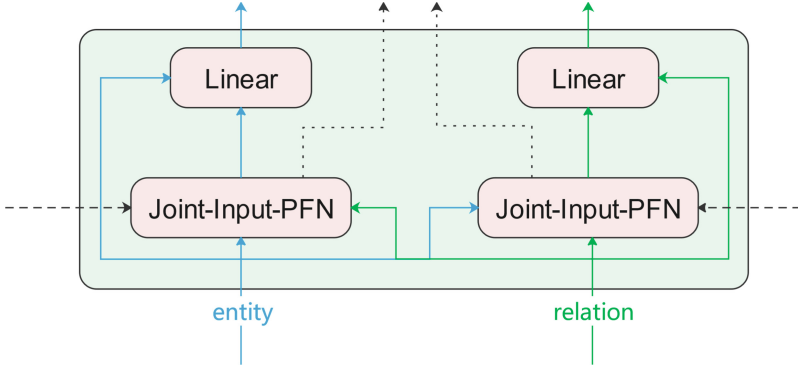


Fig. 1. The architecture of the proposed Dual-Joint-Input-PFN-Decoder model

### 3.1 The Dual-Joint-Input-PFN Strategy

The original PFN is an interaction strategy built on a multi-task learning framework, in which PFN encodes a feature and generates three types of features: features for entity extraction, features for relation extraction, and features for entity-relation extraction. Particularly, PFN considers that features for entity extraction is irrelevant or even harmful for relation extraction and vice versa. Afterwards, the features for entity extraction/relation extraction and the features for entity-relation extraction are combined to build entity/relation features. However, features constructed by dual-decoder are different from the feature accepted by PFN. Interaction strategy employed in dual-decoder need to be able to receive entity features and relation features simultaneously and select beneficial feature from one type of features based on another type of features. PFN cannot meet the requirements above due to structural limitations.

To improve the PFN, we propose Dual-Joint-Input-PFN strategy which is implemented by two Joint-Input-PFN strategy. The Joint-Input-PFN strategy receives two inputs at the same time and utilize one type of features to partition and filter the other features to obtain beneficial information. However, it is insufficient to utilize merely one Joint-Input-PFN strategy to construct two interactions, since one Joint-Input-PFN cannot construct interactions beneficial to both extraction task. In terms of entity extraction, Joint-Input-PFN utilize entity features to select features from the relation features for keeping useful ones that beneficial for entity extraction. However, it not able to utilize entity feature for selecting useful features from the relation features for relation extraction. Therefore, we proposed Dual-Joint-Input-PFN strategy by employing symmetric Joint-Input-PFN strategy for generating interactions that are beneficial to both entity extraction and relation extraction, presented in Fig. 2. Moreover, interactions are separated constructed, thus it avoids the problem of feature conflicts. For illustration, a Joint-Input-PFN structure for entity extraction is presented as follows since the Dual-Joint-Input-PFN strategy is implemented by two same Joint-Input-PFN strategy, in which one for entity extraction and one for relation extraction.



**Fig. 2.** The Dual-Joint-Input-PFN strategy

The Joint-Input-PFN receives the entity features  $H_i^{\text{ent-D}}$  and relation feature  $H_i^{\text{rel-D}}$  decoded by the dual-decoder. It also takes the hidden state  $H_{i-1}^{\text{Joint-Input-PFN}}$  and cell state  $c_{t-1}$  from previous Joint-Input-PFN strategy. The Joint-Input-PFN calculates current cell state  $\tilde{c}_i$  utilizing the relation features  $H_i^{\text{rel-D}}$  and the hidden state  $H_{i-1}^{\text{Joint-Input-PFN}}$ , as shown in Eq. (1):

$$\tilde{c}_i = \tanh(\text{Linear}([H_i^{\text{rel-D}}; H_{i-1}^{\text{Joint-Input-PFN}}])) \quad (1)$$

The  $[\cdot]$  denotes a connection operation. Afterwards, the master gate [21] is employed to select beneficial features from the current cell state  $\tilde{c}_i$ . The procedure is as Eq. (2):

$$\begin{aligned} \tilde{p}_{\tilde{c}_i} &= \text{cummax}(\text{Linear}([H_i^{\text{ent-D}}; H_{i-1}^{\text{Joint-Input-PFN}}])) \\ \tilde{q}_{\tilde{c}_i} &= 1 - \text{cummax}(\text{Linear}([H_i^{\text{ent-D}}; H_{i-1}^{\text{Joint-Input-PFN}}])) \end{aligned} \quad (2)$$

The selector contains a  $\tilde{p}_{\tilde{c}_i}$  for selecting relation features of current cell state  $\tilde{c}_i$  that are beneficial or harmful to the entity extraction, and a  $\tilde{q}_{\tilde{c}_i}$  for selecting relation features of  $\tilde{c}_i$  that beneficial or irrelevant to the entity extraction. After the selection using the selector, the relation feature of current cell state  $\tilde{c}_i$  is split into three parts using the  $\rho_{\text{useful}, \tilde{c}_i}$  selector,  $\rho_{\text{harmful}, c_t}$  selector and  $\rho_{\text{unrelated}, c_t}$  selector, as shown in Eq. (3). The acquired beneficial features, harmful features, and irrelevant features to entity extraction are denoted as Eq. (4):

$$\begin{aligned} \rho_{\text{useful}, \tilde{c}_i} &= \tilde{p}_{\tilde{c}_i} \cdot \tilde{q}_{\tilde{c}_i} \\ \rho_{\text{harmful}, c_t} &= \tilde{p}_{\tilde{c}_i} \rho_{\text{useful}} \\ \rho_{\text{unrelated}, c_t} &= \tilde{q}_{\tilde{c}_i} - \rho_{\text{useful}} \end{aligned} \quad (3)$$

$$\begin{aligned} \rho_{\text{ent-useful}} &= \rho_{\text{useful}, \tilde{c}_i} \cdot \tilde{c}_t + \rho_{s, c_{t-1}} \cdot c_{t-1} \\ \rho_{\text{ent-unrelated}} &= \rho_{\text{unrelated}, \tilde{c}_i} \cdot \tilde{c}_t + \rho_{e, c_{t-1}} \cdot c_{t-1} \\ \rho_{\text{ent-harmful}} &= \rho_{\text{harmful}, \tilde{c}_i} \cdot \tilde{c}_t + \rho_{r, c_{t-1}} \cdot c_{t-1} \end{aligned} \quad (4)$$

$\rho_{s,c_{t-1}}, \rho_{e,c_{t-1}}, \rho_{r,c_{t-1}}$  are selector to select useful features from the previous cell states  $c_{t-1}$ , calculated in the same way as  $\rho_{\text{ent-useful},\tilde{c}_t}$ ,  $\rho_{\text{ent-unrelated},c_t}$ ,  $\rho_{\text{ent-harmful},c_t}$ . Similarly, for the relation decoder, another Joint-Input-PFN strategy can be constructed to extract the features  $\rho_{\text{rel-useful}}$  from the entity feature that are beneficial to the relation feature.

The features  $\rho_{\text{ent-useful},\tilde{c}_t}$  are generated by extracting useful feature from relation feature based on entity feature, as interaction features between entity and relation features. The Skipping connection [22] and Linear layer are employed to generate the interactions of features  $H_i^{\text{ent-DPFN-D}}$  and  $H_i^{\text{rel-DPFN-D}}$ , as shown in Eq. (5). Afterwards, the two features with interactions are sent to next layer dual-decoder for further processing.

$$\begin{aligned} H_i^{\text{ent-DPFN-D}} &= \text{Linear}([H_i^{\text{ent-D}}; \rho_{\text{ent-useful}}]) \\ H_i^{\text{rel-DPFN-D}} &= \text{Linear}([H_i^{\text{rel-D}}; \rho_{\text{rel-useful}}]) \end{aligned} \quad (5)$$

### 3.2 The Dual-Joint-Input-PFN-Decoder Model

In existing studies, sequence-to-sequence or sequence-to-non-sequence models have employed one decoder to decode entity extraction and relation extraction together. However, considering that one decoder may cause the problem of feature conflicts during sharing one structure, as mentioned in [15]. We propose to use a Dual-Joint-Input-PFN-Decoder model incorporated with Dual-Joint-Input-PFN strategy to avoid feature conflicts. Based on the transformer-base non-autoregressive decoder [23], our model utilizes dual-transformers fused with our proposed Dual-Joint-Input-PFN strategy for decoding entity and relation features separately.

The Dual-Joint-Input-PFN-Decoder model consists of two identical decoders, each of which is a non-autoregressive Transformer-decoder structure with  $k$  layers. Before decoding, the Dual-Joint-Input-PFN-Decoder model takes triples embeddings, denoted as  $E \in R^{m \times d}$ , as inputs, where  $m$  is the maximum number of triples in all sentences. The Transformer-decoder contains a self-attention layer, an inter-attention layer, and a feed forward networks (FFN) layer. The forward propagation process for each layer of the dual-decoder can be formalized as Eq. (6), where  $i$  denotes the  $i$ -th layer of decoder:

$$\begin{aligned} H_i^{\text{ent-D}} &= \text{Transformer}_{\text{ent}}(H_{i-1}^{\text{ent-D}}, H^E) \\ H_i^{\text{rel-D}} &= \text{Transformer}_{\text{rel}}(H_{i-1}^{\text{rel-D}}, H^E) \end{aligned} \quad (6)$$

However, the forward propagation of the dual-decoder is completely separated without interactions between entity and relation extraction. The Dual-Joint-Input-PFN formally receives two inputs  $H_{i-1}^{\text{ent-D}}$  and  $H_{i-1}^{\text{rel-D}}$ , and generate  $H_{i-1}^{\text{ent-DPFN}}$  and  $H_{i-1}^{\text{rel-DPFN}}$  with interaction features, as shown in Eq. (7). Thus, the forward propagation process can be revised to Eq. (8) by replacing the inputs with  $H_{i-1}^{\text{ent-DPFN}}$  and  $H_{i-1}^{\text{rel-DPFN}}$ , respectively.

$$(H_{i-1}^{\text{ent-DPFN}}, H_{i-1}^{\text{rel-DPFN}}) = \text{Dual-Joint-Input-PFN}(H_{i-1}^{\text{ent-D}}, H_{i-1}^{\text{rel-D}}) \quad (7)$$

$$\begin{aligned} H_i^{\text{ent-D}} &= \text{Transformer}_{\text{ent}}(H_{i-1}^{\text{ent-DPFN-D}}, H^E) \\ H_i^{\text{rel-D}} &= \text{Transformer}_{\text{rel}}(H_{i-1}^{\text{rel-DPFN-D}}, H^E) \end{aligned} \quad (8)$$

## 4 Experiment

### 4.1 Dataset

Our proposed methods are evaluated based on two standard datasets WebNLG [24] and NYT [25], which are widely applied to the joint entity-relation extraction task, e.g., Zeng et al. [26]. WebNLG contains 5019 sentences in training dataset and 703 sentences in test dataset. There are 171 predefined relation types in WebNLG. NYT contains 56196 sentences in training dataset and 5000 sentences in test dataset, with a total of 24 relation types. The experiments were conducted using the Zeng et al. [26] version of the NYT dataset (Table 1).

**Table 1.** Statistics for WebNLG and NYT

Dataset	#Train	#Valid	#Test	Relation type
WebNLG	5019	500	703	171
NYT	56196	5000	5000	24

### 4.2 Evaluation Metrics

The evaluation metrics are standard precision, recall and F1-score as follows:

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN} \\
 F1 &= \frac{2 \times Precision \times Recall}{Precision + Recall}
 \end{aligned} \tag{9}$$

$TP$  denotes the number of positive classes predicted to be positive,  $FP$  denotes the number of negative classes predicted to be positive, and  $FN$  denotes the number of positive classes predicted to be negative. A triple is regarded as correct if and only if matched the head entity, tail entity, and relation type of the triple are all exactly matched. There are two ways to evaluate whether an entity is correctly matched, namely partial match and exact match. Partial match means that the first word of predicted entity is the same as the first word of the matched label. Exact match means that the whole predicted entity is the same as its matched label. The WebNLG dataset utilizes partial match, and the NYT dataset utilizes exact match for entity matching respectively by following the same strategy in previous work.

### 4.3 Baselines

Our methods are compared with the following baseline models:



- 1) NovelTagging [9]: The model proposed a new tagging strategy featured with extracting a triple from sentences instead of extracting entity and relation separately.
- 2) CopyRE [26]: The model employed an encoder-decoder structure in a sequence-to-sequence manner by utilizing a copy mechanism to identify entities from sentences.
- 3) GraphRel [27]: The model encoded triples using graph neural networks and employed graph structures to construct interactions between entity and relation extraction.
- 4) CasRel [7]: The model treated a relation as a mapping from head entities to tail entities, to enable the extraction of entity-relation triples jointly.
- 5) RIN [18]: Two structures were developed and separately used to implement entity and relation extraction with a Recurrent Interaction Network to construct their connection information.
- 6) TPLinker [11]: The model proposed a Handshaking tagging strategy to extract overlapping triples.
- 7) SPN4RE [14]: The model utilized a non-autoregressive model and bipartite matching loss to implement a sequence-to-set framework, to solve the exposure bias issue.

The experiments were implemented using the base version of BERT [19] as encoder. The initial learning rate of BERT was set to 0.00001, while the learning rate of Dual-Joint-Input-PFN-Decoder model was set to 0.00002. The Dual-Joint-Input-PFN-Decoder model was composed of 3-layer non-autoregressive transformers. In addition, a dropout was applied to prevent overfitting, with a rate of 0.1. The maximum gradient was set to 20 to prevent explosive growth of the gradient. The training was performed utilizing the AdamW method and a Layernorm was employed to accelerate training speed. All experiments were conducted on a server with Intel Xeon CPU E5-2609, 96 GB memory, and RTX 2080Ti.

#### 4.4 The Results

The performance comparison of our proposed model with all baseline models are presented in Table 2. Our model achieved the best performance with a precision of 0.932, a recall of 0.929 and a F1-score of 0.931 on WebNLG, and a precision of 0.930, a recall of 0.918 and a F1-score of 0.924 on NYT. Compared with SPN4RE, our model had an improvement of F1-score by 0.2% and 0.1% on the WebNLG and NYT datasets, respectively. Moreover, our model had a clear improvement on precision over the state-of-the-art SPN4RE by 0.4% and 0.5% on the two datasets, respectively. Comparing the models with interaction mechanism, our model exceeded the RIN by 5.5% and 7.3% in F1-score on the WebNLG and NYT datasets, respectively. Comparing the models with complex tagging strategy, our model exceeded the TPLinker by 1.2% and 0.4% in F1-score on the WebNLG and NYT datasets. The results strongly demonstrate that the Dual-Joint-Input-PFN-Decoder model is able to avoid feature conflicts and build interaction between entity and relation extraction.

To verify the effectiveness on extracting element of relational triples, we conducted a comparison of entity pair extraction and relation type extraction with SPN4RE. Although the performance of entity pairs decreased on F1-score by 0.3%, the performance of relation extraction improved on F1-score by 0.3%, as shown in Table 3. The improvement

**Table 2.** The results of performance comparison, where \* denotes results of reproduced models, ‘partial’ denotes entities using partial match, and ‘exact’ denotes entities using exact match.

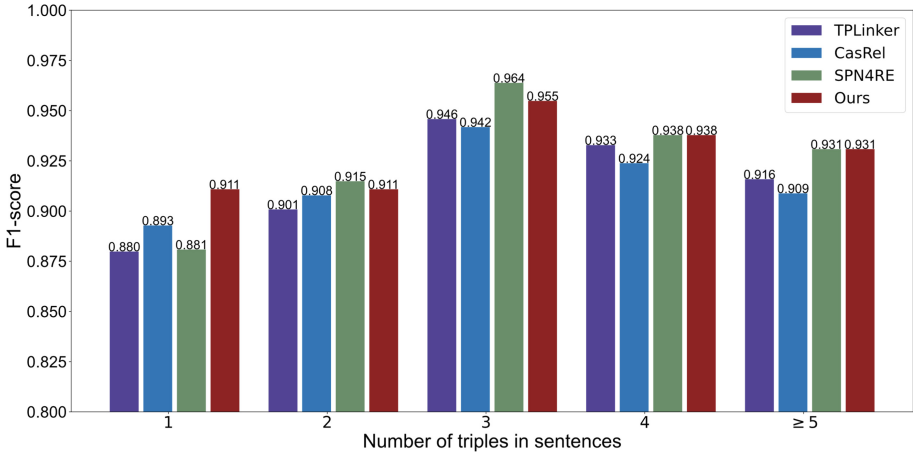
Models	WebNLG(partial)			NYT(exact)		
	Precision	Recall	F1-score	Precision	Recall	F1-score
NovelTagging	0.525	0.193	0.283	–	–	–
CopyRE-One	0.322	0.289	0.305	–	–	–
CopyRE-Mul	0.377	0.364	0.371	–	–	–
GraphRel-1p	0.423	0.392	0.407	–	–	–
GraphRel-2p	0.447	0.411	0.429	–	–	–
CasRel	<b>0.934</b>	0.901	0.918	–	–	–
RIN	0.877	0.879	0.877	0.844	0.860	0.851
TPLinker	0.918	0.920	0.919	0.914	0.926	0.920
SPN4RE	0.928*	<b>0.929*</b>	0.929*	0.925	<b>0.922</b>	0.923
Ours	0.932	<b>0.929</b>	<b>0.931</b>	<b>0.930</b>	0.918	<b>0.924</b>

in relation extraction also driven a 0.2% improvement on F1-score of relational triples extraction, which was an indication that the proposed model correctly matched entity pairs and relation properly.

**Table 3.** The comparison of our model with the SPN4RE on entity pair extraction and relation type extraction on WebNLG

Models	Element	Precision	Recall	F1-score
SPN4RE	(s, o)	<b>0.952</b>	<b>0.952</b>	<b>0.952</b>
	r	0.954	0.955	0.954
	Overall	0.928	0.929	0.929
Ours	(s, o)	0.948	0.945	0.947
	r	<b>0.958</b>	0.955	<b>0.957</b>
	Overall	<b>0.932</b>	0.929	<b>0.931</b>

To verify the effectiveness on extraction various numbers of triples contained in a sentence, we conducted an experiment on a sentence with different numbers of triples. The number of triples in a sentence denoted as  $n$ . The best performance is achieved at  $n = 1$  with a F1-score 0.911. The experimental results are shown in Fig. 3. Although the performance of this model decreases on F1-score by 0.4% and 0.9% compared to SPN4RE at  $n = 2$  and  $n = 3$ , respectively. The performance of the model improves on F1-score by 3% at  $n = 1$ . Many existing models do not perform well for triple extraction at  $n = 1$ . Compared with TPLinker and CasRel, the performance of the proposed model at  $n = 1$  is improved by 3.1% and 1.8% in F1-score, respectively.



**Fig. 3.** The comparison of various number of triples in a sentence on WebNLG

To verify the effectiveness of Dual-Joint-Input-PFN-Decoder model, an ablation experiments was conducted: Ours(w/o interaction) denotes our model without interactions and Ours(PFN) denotes the model with PFN, while Ours(Double-Encoder) denotes the model with separate two encoders. The results are shown in Table 4.

**Table 4.** Results of ablation experiments on WebNLG

Ablation	Precision	Recall	F1-score
Ours(PFN)	0.915	0.908	0.912
Ours(Double-Encoder)	0.918	0.916	0.917
Ours(w/o interaction)	0.927	0.918	0.923
Ours	<b>0.932</b>	<b>0.929</b>	<b>0.931</b>

The performance of the Ours(PFN) decreased on F1-score by 1.9% verifying that the PFN was not suitable in our proposed model. The performance of the Ours(Double-Encoder) decreased by 1.4%, indicating that the utilization of two encoders still lowered the performance. In addition, the performance of the Ours(w/o interaction) decreased by 0.8%, indicating the importance of building interactions between in entity and relation extractions.

## 5 Conclusion

This paper proposed a Dual-Joint-Input-PFN-Decoder model and Dual-Joint-Input-PFN strategy to solve the problem of feature conflicts in the original model and to better capture the connection information between subtasks. Dual-Joint-Input-PFN-Decoder

model is employed to separate the construction process of two subtask features. Dual-Joint-Input-PFN strategy can simultaneously receive two inputs and obtain favorable information from the one input based on another input, enabling Dual-Joint-Input-PFN strategy to be applied to the dual-decoder. Experiment shown that our model achieved the best performance compared with the state-of-the-art baseline models, demonstrating the effectiveness of the method.

**Acknowledgements.** The work is supported by grants from National Natural Science Foundation of China (No. 61871141), Natural Science Foundation of Guangdong Province (2021A1515011339), and Collaborative Innovation Team of Guangzhou University of Traditional Chinese Medicine (No. 2021XK08).

## References

1. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. *J. Mach. Learn. Res.* **3**, 1083–1106 (2003)
2. Chan, Y.S., Roth, D.: 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 (2011)
3. Wang, J., Lu, W.: Two are better than one: joint entity and relation extraction with table-sequence encoders. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1706–1721 (2020)
4. Miwa, M., Sasaki, Y.: 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 (2014)
5. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2537–2547 (2016)
6. Ma, Y., Hiraoka, T., Okazaki, N.: Named entity recognition and relation extraction using enhanced table filling by contextualized representations. *arXiv preprint [arXiv:2010.07522](https://arxiv.org/abs/2010.07522)* (2020)
7. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1476–1488 (2020)
8. Yu, B., et al.: Joint extraction of entities and relations based on a novel decomposition strategy. In: *ECAI 2020*, pp. 2282–2289 (2020)
9. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: 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 (Volume 1: Long Papers)*, pp. 1227–1236 (2017)
10. Luo, X., Liu, W., Ma, M., Wang, P.: A bidirectional tree tagging scheme for jointly extracting overlapping entities and relations. *arXiv e-prints, arXiv:2008* (2020)
11. Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., Sun, L.: TPLinker: single-stage joint extraction of entities and relations through token pair linking. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1572–1582 (2020)
12. Nayak, T., Ng, H.T.: Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8528–8535 (2020)

13. Zhang, R.H., et al.: Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 236–246 (2020)
14. Sui, D., Chen, Y., Liu, K., Zhao, J., Zeng, X., Liu, S.: Joint entity and relation extraction with set prediction networks. arXiv preprint [arXiv:2011.01675](https://arxiv.org/abs/2011.01675) (2020)
15. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 50–61 (2021)
16. Yan, Z., Zhang, C., Fu, J., Zhang, Q., Wei, Z.: A partition filter network for joint entity and relation extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 185–197 (2021)
17. Zeng, X., He, S., Zeng, D., Liu, K., Liu, S., Zhao, J.: 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 (2019)
18. Sun, K., Zhang, R., Mensah, S., Mao, Y., Liu, X.: Recurrent interaction network for jointly extracting entities and classifying relations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3722–3732 (2020)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
20. Fu, T.J., Li, P.H., Ma, W.Y.: 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 (2019)
21. Shen, Y., Tan, S., Sordani, A., Courville, A.: Ordered neurons: integrating tree structures into recurrent neural networks. arXiv preprint [arXiv:1810.09536](https://arxiv.org/abs/1810.09536) (2018)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
23. Gu, J., Bradbury, J., Xiong, C., Li, V.O., Socher, R.: Non-autoregressive neural machine translation. arXiv preprint [arXiv:1711.02281](https://arxiv.org/abs/1711.02281) (2017)
24. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 179–188 (2017)
25. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6323, pp. 148–163. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-15939-8\\_10](https://doi.org/10.1007/978-3-642-15939-8_10)
26. Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J.: 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 (2018)
27. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)