Attention as Relation: Learning Supervised Multi-head Self-Attention for Relation Extraction

Jie Liu[1*] , **Shaowei Chen**[1] , **Bingquan Wang**[1] , **Jiaxin Zhang**[1] , **Na Li**[2] and **Tong Xu**[3]
[1]College of Artificial Intelligence, Nankai University, Tianjin, China
[2]College of Computer Science, Nankai University, Tianjin, China
[3]University of Science and Technology of China, Hefei, China
jliu@nankai.edu.cn, {chenshaowei, wangbq, nkuzjx, 2120180458}@mail.nankai.edu.com,
tongxu@ustc.edu.cn

Abstract

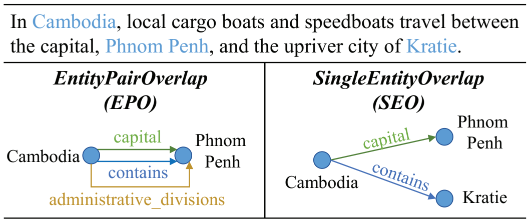
Joint entity and relation extraction is critical for many natural language processing (NLP) tasks, which has attracted increasing research interest. However, it is still faced with the challenges of identifying the overlapping relation triplets along with the entire entity boundary and detecting the multi-type relations. In this paper, we propose an attention-based joint model, which mainly contains an entity extraction module and a relation detection module, to address the challenges. The key of our model is devising a supervised multi-head self-attention mechanism as the relation detection module to learn the token-level correlation for each relation type separately. With the attention mechanism, our model can effectively identify overlapping relations and flexibly predict the relation type with its corresponding intensity. To verify the effectiveness of our model, we conduct comprehensive experiments on two benchmark datasets. The experimental results demonstrate that our model achieves state-of-the-art performances.

1 Introduction

Joint entity and relation extraction is an important and challenging task in information extraction. Given an unstructured text, this task aims to identify relation triplets consisting of two entities and their semantic relation, such as (*Cambodia*, *Capital*, *Phnom Penh*) in Figure 1. As the fundamental task for building knowledge bases, this task has attracted widespread attention.

Traditional methods [Zelenko *et al.*, 2003; Mintz *et al.*, 2009; Chan and Roth, 2011] divide this task into two isolated subtasks, including entity recognition and relation classification, and solve them in a pipeline manner. Although these methods are flexible, they suffer from error propagation and ignore the relevance between the two subtasks.

To avoid error propagation, most recent studies [Miwa and Bansal, 2016; Zheng *et al.*, 2017; Zeng *et al.*, 2018] are dedicated to identifying entities together with their semantic relations in a joint manner and have achieved great progress.



In Cambodia, local cargo boats and speedboats travel between the capital, Phnom Penh, and the upriver city of Kratie.

Figure 1: An example of the *EntityPairOverlap* (EPO) and *SingleEntityOverlap* (SEO) triplets.

Despite the progress, the complicated relation structures still pose great challenges for this task. **First**, it is challenging to identify overlapping relation triplets. As shown in Figure 1, given a sentence, there may exist *EntityPairOverlap* triplets where two entities have multiple relations and *SingleEntityOverlap* triplets where two relation triplets share an overlapped entity. Early joint methods [Miwa and Bansal, 2016; Zheng *et al.*, 2017] fail to extract them due to the relation classifiers, which suppose that a token only belongs to one relation. **Second**, the characteristic of entity semantics under different relation types should be considered. For example, as shown in Figure 1, the location semantics should be captured when the model predicts the *contains* relation for the entity pair (*Cambodia*, *Phnom Penh*), while the semantics of administrative function should be learned when the *capital* relation is predicted for the above entity pair. However, existing studies [Sun *et al.*, 2019; Zeng *et al.*, 2019a; Nayak and Ng, 2019] generally utilize a standard classifier to detect relation types, which neglects the semantic changes and leads to unideal relation triplet extraction. **Third**, it is crucial to recognize multiple possible relation types for an *EntityPairOverlap* triplet accurately. For instance, as shown in Figure 1, the entity pair (*Cambodia*, *Phnom Penh*) has three relations, including *contains*, *capital*, and *administrative_divisions*. Thus, the two entities should have high correlation intensity under all three relation types. But with a multi-class classifier, all the possible relation types share the same probability space, which makes them are essentially mutually exclusive. Although the post-processing can be used to obtain the *EntityPairOverlap* triplets [Fu *et al.*, 2019], the correlation intensity and discrimination for different relation types will decrease. In this work, we argue that

[*]Corresponding author.

this problem should be treated as a multi-label classification task, instead of a simple multi-class classification task.

To address the aforementioned challenges, we propose a simple but effective attention-based joint model mainly consisting of an entity extraction module and a relation detection module. For entity extraction, we treat it as a sequence labeling task and adopt Conditional Random Field (CRF) [Lafferty *et al.*, 2001] to recognize the entity boundary. For relation detection, we regard it as a multi-label classification task and design a supervised multi-head self-attention mechanism. By mapping each relation type to a subspace of the multiple heads, the distinctive token-level correlation semantics for each relation can be learned, and the correlation intensity under different relation types can be calculated separately. Meanwhile, the interaction between different relation types can be captured by sharing the same input representations of multi-head self-attention. Finally, we infer the triplets based on the two modules with a given threshold. To verify the effectiveness of our model, we make a comprehensive and comparative analysis on two benchmark datasets, and the results demonstrate that our model achieves state-of-the-art performances. In summary, our contributions are three-fold:

- We propose a supervised attention-based joint model[1], which can flexibly identify the overlapping triplets with the entire entity boundary.
- To adequately capture the distinctive correlation semantics and separately learn the correlation intensity under different relation types, we transform the relation detection into a multi-label classification task and design a supervised multi-head self-attention mechanism.
- Extensive experiments are conducted on two benchmark datasets, and the results show that our model achieves state-of-the-art performances with 1.3% and 14.2% improvements, respectively.

2 Related Work

Traditional approaches [Zelenko *et al.*, 2003; Miwa *et al.*, 2009; Mintz *et al.*, 2009; Chan and Roth, 2011; Zeng *et al.*, 2015; Shen and Huang, 2016] generally deal with joint entity and relation extraction task in a pipeline manner, which identifies the entities first and then predicts the relations between them. Because these methods treat entity recognition and relation classification as two isolated steps, they suffer from error propagation.

To consider the interaction between the two steps, Miwa and Bansal [2016], Gupta et al. [2016], Zhang et al. [2017], Zheng et al. [2017], and Sun et al. [2019] jointly extracted entities and relations in a unified framework. But these methods can not precisely identify the overlapping triplets because they assume that a token or an entity pair only belongs to one relation.

Recently, many studies focus on predicting overlapping triplets while considering the interactions between relations. Fu et al. [2019] proposed a two-phase joint model based on graph convolutional network (GCN). Takanobu et al. [2019] introduced a hierarchical reinforcement learning framework, which includes a high-level policy for relation detection and a low-level policy for entity extraction. Dai et al. [2019] designed a novel tagging scheme and proposed a position-attention mechanism to identify overlapping relations. Meanwhile, the seq2seq framework is also utilized to predict the overlapping triplets [Zeng *et al.*, 2018; Zeng *et al.*, 2019b; Zeng *et al.*, 2019a; Nayak and Ng, 2019]. However, these methods fail to treat different relation types as distinctive subspaces and calculate the correlation degree separately under different relation types. Besides, the seq2seq based models generally suffer from inaccurate entity boundary recognition.

To deal with the above issues, we utilize CRF to recognize the entity boundary and design a supervised multi-head self-attention mechanism to flexibly extract overlapping relations via treating each relation type as an isolated subspace.

3 Model

Given a sentence $X = \{x_1, x_2, ..., x_N\}$ with N tokens, joint entity and relation extraction task aims to identify a collection of relation triplets $T = [(f_l, r_l, s_l)]_{l=1}^{L}$ from X, where f_l, s_l, and r_l represent the first entity, the second entity, and their relation, respectively. Note that the entities are extracted from the given sentence, and the relations are selected from a predefined set $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_M\}$ with M types.

To deal with this task, three issues should be considered, including detecting the overlapping relations, learning the characteristic correlation semantics for each relation type, and calculating the correlation intensity for each relation type independently. To this end, a supervised attention-based joint model is proposed in this paper, and the framework of our model is illustrated in Figure 2. Concretely, our model consists of an encoding layer, an entity extraction module, and a relation detection module. We adopt CRF as the entity extraction module to recognize entities and devise a supervised multi-head self-attention mechanism as the relation detection module to learn the fine-grained correlation between tokens for each relation type. Finally, we fuse the predicted results from the two modules and infer triplets via a given threshold.

3.1 Encoding Layer

Given a sentence X, we first utilize bidirectional long short-term memory (BLSTM) network [Hochreiter and Schmidhuber, 1997] to encode the contextualized representation for each token. Formally, the initial embedding $\mathbf{e}_i$ of each token is calculated by concatenating the word embedding $\mathbf{e}_i^w \in \mathbb{R}^{d_w}$ and the character-level morphology feature[2] $\mathbf{e}_i^c \in \mathbb{R}^{d_c}$, where d_w and d_c are the dimensions of word embedding and morphology feature, respectively. And then, the contextualized representation sequence $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_N\}$ is obtained as follows:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \tag{1}$$

$$\overrightarrow{\mathbf{h}}_i = \text{LSTM}^f(\mathbf{e}_i, \overrightarrow{\mathbf{h}}_{i-1}), \overleftarrow{\mathbf{h}}_i = \text{LSTM}^b(\mathbf{e}_i, \overleftarrow{\mathbf{h}}_{i+1}), \tag{2}$$

where LSTM^f and LSTM^b denote the forward and backward LSTM, respectively.

[1] https://github.com/NKU-IIPLab/SMHSA

[2] We also adopt BLSTM to capture the morphology features.

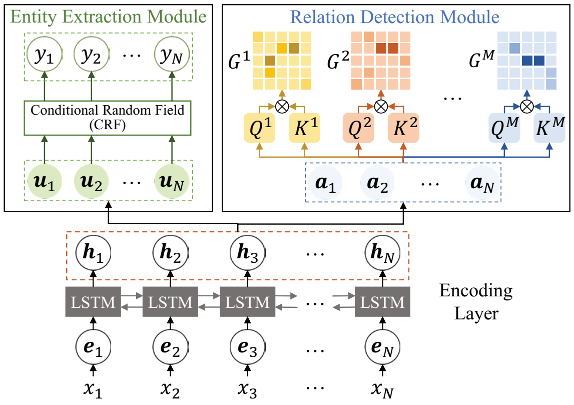


Figure 2: The framework of our model, which consists of an encoding layer, an entity extraction module, and a relation detection module.

3.2 Entity Extraction Module

To recognize the entity boundary accurately, we treat entity extraction as a sequence labeling task and adopt Conditional Random Field (CRF) [Lafferty *et al.*, 2001] as the entity extraction module. Formally, CRF utilizes a state score matrix $P \in \mathbb{R}^{N \times k}$ to model the mappings between tokens and labels. Meanwhile, a transition score matrix $V \in \mathbb{R}^{k \times k}$ is used to learn the dependency between adjacent labels, where k denotes the dimension of the label space[3]. For a sequence of predicted labels $\hat{Y} = \{y_1, y_2, ..., y_N\}$, we define its score as follows:

$$S(X, \hat{Y}) = \sum_{i=1}^{N} V_{y_{i-1}, y_i} + \sum_{i=1}^{N} P_{i, y_i}, \tag{3}$$

$$P = UW_p + b_p, \tag{4}$$

where $U = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ denotes the input hidden representation sequence for the entity extraction module, which is calculated from the context representation sequence H via a fully-connection layer. The matrices $W_p \in \mathbb{R}^{d_u \times k}$ and $b_p \in \mathbb{R}^{N \times k}$ are model parameters, where d_u denotes the dimension of the input hidden representations in U.

Then, the probability of label sequence $\hat{Y}$ can be calculated as follows:

$$p\left(\hat{Y} \mid X\right) = \frac{\exp\left(S\left(X, \hat{Y}\right)\right)}{\sum_{\widetilde{Y} \in Y_X} \exp\left(S\left(X, \widetilde{Y}\right)\right)}, \tag{5}$$

where Y_X denotes all possible label sequences.

During training, we aim to maximize the likelihood probability $p\left(Y \mid X\right)$ of the gold label sequence Y. Thus, we minimize the negative log-likelihood loss function to optimize the parameters as follows:

$$\mathcal{L}_E = \log \sum_{\widetilde{Y} \in Y_X} \exp\left(S\left(X, \widetilde{Y}\right)\right) - S\left(X, Y\right). \tag{6}$$

During decoding, we use the Viterbi algorithm to obtain the predicted label sequence with the maximum score.

3.3 Relation Detection Module

To flexibly predict the overlapping triplets, we transform the relation detection into a multi-label classification task and devise a *supervised multi-head self-attention* mechanism to deal with it by regarding each relation type as an isolated subspace. Specially, we calculate the attention at token-level, which can help our model learn more fine-grained correlation semantics. Formally, we first couple a fully-connection layer upon the encoding layer to obtain the input hidden representation sequence $A = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N\}$ for the relation detection module:

$$A = HW_a + b_a, \tag{7}$$

where $W_a \in \mathbb{R}^{d_h \times d_a}$ and $b_a \in \mathbb{R}^{N \times d_a}$ are model parameters, while d_h and d_a are the dimensions of the contextualized representations in H and the input hidden representations in A, respectively. By sharing the input hidden representations, the interaction between different relation types can be captured.

Considering that the characteristic correlation semantics should be learned and the correlation degree should be calculated independently for each relation type, we regard each relation type as a subspace and project the input hidden representations to different relation subspaces as follows:

$$Q^m = AW_Q^m, \quad K^m = AW_K^m, \tag{8}$$

where $Q^m \in \mathbb{R}^{N \times d_r}$ and $K^m \in \mathbb{R}^{N \times d_r}$ denote the queries and keys for the m-th relation type. The matrices $W_Q^m \in \mathbb{R}^{d_a \times d_r}$ and $W_K^m \in \mathbb{R}^{d_a \times d_r}$ are model parameters, where d_r is the dimension of each relation subspace.

Then, we calculate the attention matrix $G^m \in \mathbb{R}^{N \times N}$ whose element $G_{i,j}^m$ denotes the correlation intensity between the i-th token and the j-th token under the m-th relation type:

$$G^m = \text{softmax}\left(\frac{Q^m (K^m)^T}{\sqrt{d_r}}\right). \tag{9}$$

To guide this module to detect relation types, we further introduce supervision information by maximizing the likelihood probability as follows:

$$p\left(Z|X\right) = \prod_{m=1}^{M} \prod_{i=1}^{N} \prod_{j=1}^{N} p\left(Z_{i,j}^m | x_i, x_j\right), \tag{10}$$

$$p\left(Z_{i,j}^m | x_i, x_j\right) = \begin{cases} G_{i,j}^m, & if \quad Z_{i,j}^m = 1 \\ 1 - G_{i,j}^m, & if \quad Z_{i,j}^m = 0 \end{cases}, \tag{11}$$

where $Z_{i,j}^m = 1$ denotes the fact that the m-th relation exists between the i-th token and the j-th token, and vice versa. To transform the relation detection into a multi-label classification task, we convert the gold annotation to a one-hot matrix during training and minimize the binary cross-entropy loss between the predicted distribution $\hat{p}\left(Z_{i,j}^m | x_i, x_j\right)$ and the gold distribution $p\left(Z_{i,j}^m | x_i, x_j\right)$ as follows:

$$\mathcal{L}_R = -\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} [p\left(Z_{i,j}^m | x_i, x_j\right) \log \hat{p}\left(Z_{i,j}^m | x_i, x_j\right) + \left(1 - p\left(Z_{i,j}^m | x_i, x_j\right)\right) \log\left(1 - \hat{p}\left(Z_{i,j}^m | x_i, x_j\right)\right)]. \tag{12}$$

With the supervision information, the multi-head self-attention can be guided to detect relations more effectively.

[3] Following the BIO tagging scheme, we define three labels, including B (beginning of entity), I (inside of entity), and O (others).

3.4 Joint Learning

To synchronously learn the proposed two modules and make them mutually improve, we combine the loss functions of the two modules to form the entire loss objective of our model:

$$\mathcal{L}\left(\theta\right) = \mathcal{L}_E + \mathcal{L}_R. \tag{13}$$

The optimization problem in Eq. (13) can be solved by using any gradient descent approach. In this paper, we adopt the RMSprop [Ruder, 2016] approach.

3.5 Inference

Based on the two modules, the triplets can be easily inferred. Concretely, with the label sequence $\hat{Y}$ predicted by the entity extraction module, we can obtain the entity set $\mathcal{E} = \{e_1, e_2, ..., e_{L_\mathcal{E}}\}$ with $L_\mathcal{E}$ entities. Then, given the i-th entity $e_i = \{x_{p_i}, ..., x_{q_i}\}$ and the j-th entity $e_j = \{x_{p_j}, ..., x_{q_j}\}$, the correlated intensity δ between them on the m-th relation can be calculated based on the weight matrix G from the relation detection module:

$$\delta = \frac{1}{|e_i|} \sum_{t=p_i}^{q_i} \sum_{n=p_j}^{q_j} G^m_{t,n}, \tag{14}$$

where $|e_i|$ is the length of entity e_i. The triplet $\langle e_i, \mathcal{R}_m, e_j \rangle$ is extracted if δ is higher than a given threshold $\hat{\delta}$.

4 Experiments

4.1 Datasets

To verify the effectiveness of our model, we conduct extensive experiments on two benchmark datasets, including New York Times (NYT) [Riedel *et al.*, 2010] and WebNLG [Gardent *et al.*, 2017]. The NYT contains 24 types of relations, and the WebNLG has 246 relation types. For NYT, we follow the dataset used in [Zeng *et al.*, 2018]. For WebNLG, we select the sentences containing the most triples in each instance and discard the instances if all triplets are not found in the corresponding sentences. To construct the development set, we randomly select 10% samples from the training set. The statistics of the above datasets are shown in Table 1.

4.2 Experimental Settings

We initialize the word embeddings with pre-trained Glove 840B vectors[4] [Pennington *et al.*, 2014] and randomly initialize the character embeddings with 50 dimensions. The dimensions of hidden states for character LSTM, encoding layer, entity extraction module, and relation extraction module are set to 100, 600, 250, 250, respectively. For the relation extraction module, the head number is the same as the number of relation types, and the dimension of each head is set to 30. During training, we use the RMSprop optimizer [Ruder, 2016]. The learning rate, learning rate decay, and batch size are set to 0.001, 0.95, and 10, respectively. To ensure the balance between entity extraction and relation detection, we adopt an iterative two-step training manner where one training step optimizes the total parameters, and another only optimizes the parameters of the relation detection module and the encoding layer. To avoid overfitting, we apply dropout at a rate of 0.3.

[4] https://nlp.stanford.edu/projects/glove/

Category	NYT		WebNLG	
	Train	Test	Train	Test
#Normal	37013	3266	1712	239
#EPO	9782	978	13	2
#SEO	14735	1297	3639	488
#ALL	56195	5000	5352	727
Average Entity Length	1.4	1.4	2.2	2.2
Max Entity Length	11	8	34	15
#Relation	24		246	

Table 1: Statistics of datasets. #Normal, #EPO, and #SEO represent the numbers of sentences which belong to *Normal*, *EntityPairOverlap*, and *SingleEntityOverlap* types, respectively. #ALL and #Relation are the total number of sentences and relation types, respectively.

4.3 Evaluation

We adopt precision, recall, and standard micro-F1 score to evaluate the performances. Specifically, a predicted triplet (f, r, s) is correct only if the relation type and the two corresponding entities are all the same as the golden standard annotation. We report the corresponding results of the test set when the development set achieves the best result.

4.4 Comparison Methods

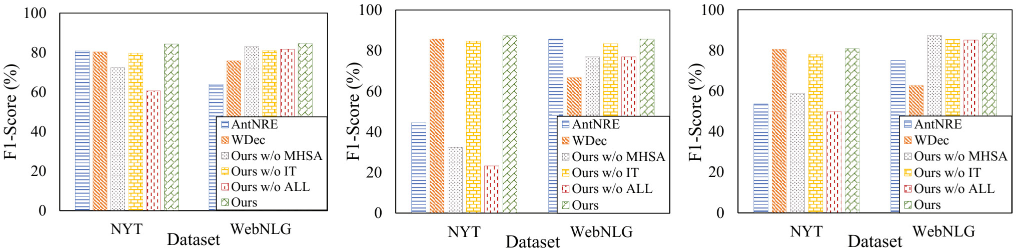
To achieve the comprehensive and comparative analysis of our model, we compare it with a series of advanced models:

- **NovelTagging** [Zheng *et al.*, 2017] introduces a novel tagging scheme which transforms the joint extraction task into a sequence labeling problem.
- **CopyRe** [Zeng *et al.*, 2018] is a seq2seq model with copy mechanism, which can effectively extract overlapping triplets. We report the results of the MultiDecoder.
- **GraphRel** [Fu *et al.*, 2019] is a two phases model based on GCN, where a relation-weighted GCN is utilized to model the interaction between entities and relations.
- **AntNRE** [Sun *et al.*, 2019] decomposes the joint extraction task into entity span detection subtask and entity relation type detection subtask. Specially, the second subtask is tackled based on an entity-relation bipartite.
- **MultiRe** [Zeng *et al.*, 2019b] applies the reinforcement learning into a seq2seq model to automatically learn the extraction order of triplets, where the interactions among triplets can be considered.
- **CopyMTL** [Zeng *et al.*, 2019a] is a multi-task learning framework, where CRF is used to identify entities, and a seq2seq model is adopted to extract relation triplets.
- **WDec** [Nayak and Ng, 2019] designs a new representation scheme and utilizes a seq2seq model to generate triplets with the entire boundaries.

Besides, we also conduct several ablation experiments, including "Ours w/o MHSA" which uses a standard multi-class classifier to detect relations, "Ours w/o IT" which does not utilize the iterative two-step training manner, and "Ours w/o ALL" which does not adopt the multi-head self-attention mechanism and the iterative training manner.

Methods	NYT			WebNLG		
	Precision	Recall	F1	Precision	Recall	F1
NovelTagging [Zheng *et al.*, 2017]	62.4%	31.7%	42.0%	52.5%	19.3%	28.3%
CopyRe [Zeng *et al.*, 2018]	61.0%	56.6%	58.7%	37.7%	36.4%	37.1%
GraphRel [Fu *et al.*, 2019]	63.9%	60.0%	61.9%	44.7%	41.1%	42.9%
AntNRE [Sun *et al.*, 2019]	83.5%*	54.4%*	65.9%*	76.3%*	70.9%*	73.5%*
MultiRe [Zeng *et al.*, 2019b]	77.9%	67.2%	72.1%	63.3%	59.9%	61.6%
CopyMTL [Zeng *et al.*, 2019a]	75.7%	68.7%	72.0%	58.0%	54.9%	56.4%
WDec [Nayak and Ng, 2019]	**88.1%**	76.1%	81.7%	88.6%*	51.3%*	65.0%*
Ours w/o MHSA	68.4%	56.5%	61.9%	85.5%	**88.0%**	86.8%
Ours w/o IT	83.6%	75.8%	79.5%	88.2%	81.7%	84.8%
Ours w/o All	56.1%	48.9%	52.2%	83.7%	85.8%	84.7%
Ours	**88.1%**	**78.5%**	**83.0%**	**89.5%**	86.0%	**87.7%**

Table 2: Results on triplet extraction. The results with '*' are reproduced by us, and all improvements of our model are significant ($p < 0.05$).



(a) Normal (b) EntityPairOverlap (c) SingleEntityOverlap

Figure 3: Results on different sentence types according to the degree of overlapping.

4.5 Results

The results on relation triplet extraction are shown in Table 2. According to the results, our model consistently obtains state-of-the-art performances on two datasets. Compared with the best baseline model, our model outperforms WDec by 1.3% F1-score on NYT and is higher than AntNRE by 14.2% F1-score on WebNLG, respectively. Specially, WDec [Nayak and Ng, 2019] adopts the seq2seq model to generate relation triplets and removes the duplicate triplets and fragmentary triplets via post-processing. Therefore, WDec achieves high precision on two datasets, while its recall is unsatisfactory. It is worth to note that our model achieves the highest precision and recall without any post-processing.

Furthermore, the performances of the seq2seq models are generally better than other baselines because they can identify overlapping relations more flexibly by decoding different triplets one by one. However, compared with other baselines, the seq2seq models fail to precisely recognize the entity boundary. Thus, the longer the length of entity is, the worse these models perform. Specially, the F1-score of the best seq2seq model WDec is inferior to AntNRE on the WebNLG dataset whose max entity length is 15 and the average entity length is 2.2. Note that AntNRE can not deal with the *EntityPairOverlap* triplets, so it performs poorly on the NYT dataset which contains more complex relations. In contrast, our model can effectively capture the overlapping triplets along with the correct entity boundary.

Besides, we also investigate the performance of entity extraction, and the results are shown in Table 3. It is clearly shown that our model achieves state-of-the-art results on two datasets, while the results on the seq2seq models are worst. The results also show that the lower F1-score on entity extraction limits the performance of the seq2seq model.

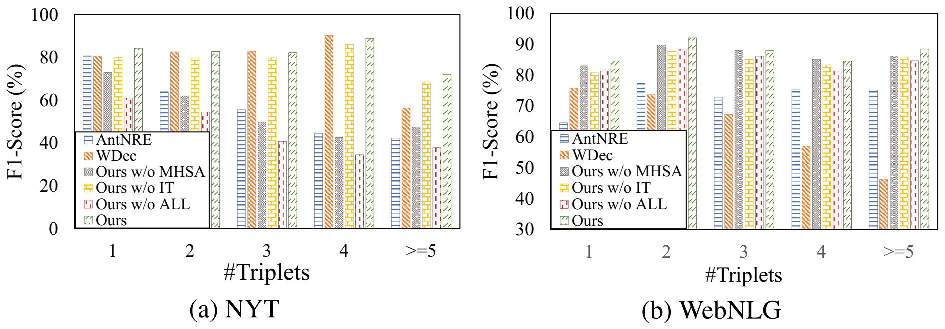
Methods	NYT	WebNLG
GraphRel [Fu *et al.*, 2019]	89.2%	91.9%
AntNRE [Sun *et al.*, 2019]	92.5%*	91.6%*
CopyMTL† [Zeng *et al.*, 2019a]	75.6%	78.2%
WDec† [Nayak and Ng, 2019]	89.1%	88.8%*
Ours	**94.8%**	**96.5%**

Table 3: Results on entity extraction (F_1 score, %). The results with '*' are reproduced by us, and the methods with '†' are seq2seq models.

4.6 Ablation Study

To prove the effectiveness of the proposed relation detection module, we conduct the ablation study. As shown in the second block of Table 2, the performance of "Ours w/o MHSA" decreases significantly on NYT, while it is slightly worse on WebNLG compared with our model. The reason is that the *EntityPairOverlap* and *SingleEntityOverlap* triplets of NYT are significantly more than them of WebNLG, which demonstrates that the proposed supervised multi-head self-attention mechanism can effectively identify overlapping triplets by projecting each relation type to a separate subspace.

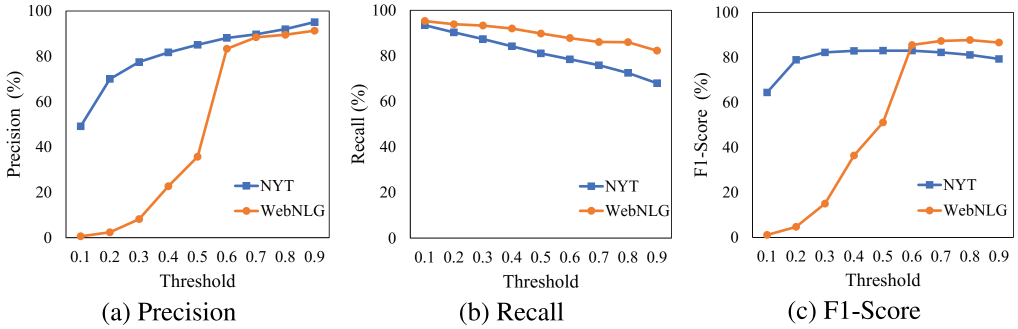
And we also analyze the effectiveness of the iterative two-step training manner. Compared with our model, the F1-score of "Ours w/o IT" decreases 3.5% and 2.9% on two datasets, respectively. This shows that the imbalance between entity



(a) NYT

(b) WebNLG

Figure 4: Results on different sentence types according to the number of triplets.



(a) Precision

(b) Recall

(c) F1-Score

Figure 5: Results on different inference thresholds.

extraction and relation detection can severely impact the performance of triplet extraction. Without the above two components, “Ours w/o ALL” performs worst.

4.7 Analysis of Different Sentence Types

To further verify the ability of our model to handle complex triplets, we analyze the results on different sentence types. Figure 3 shows the results on *Normal*, *EntityPairOverlap*, and *SingleEntityOverlap* types. Our model achieves the best results on all three types, while the performances of AntNRE, “Ours w/o MHSA” and “Ours w/o ALL” are the worst on *EntityPairOverlap* and *SingleEntityOverlap* for NYT dataset. This shows that it is difficult to deal with the overlapping triplets if the relation detection is treated as a multi-class problem. For WebNLG, WDec performs poorly on all types due to the longer entity length.

Besides, we also evaluate the performance on different triplet numbers, and the results are shown in Figure 4. Following Zeng et al. [2018], we divide the sentence into 5 classes, and each class consists of sentences with 1, 2, 3, 4, or ≥ 5 triplets, respectively. According to the results, our model achieves more stable performance as the number of triples increases, which shows that our model is more robust when it is faced with the complicated relation situation.

4.8 Analysis of Inference Threshold

We also report the effect of different thresholds in Figure 5. As the threshold increases, the recall of our model gradually decreases, and the precision gradually increases. Thus, we can flexibly select a suitable threshold according to the actual demand. Our model has a low precision on WebNLG dataset when the threshold is lower than 0.6, because the prediction process is more difficult when entities are longer. The F1-score first increases and then decreases when the threshold increases, and the results show that our model achieves the best performances on NYT and WebNLG datasets with 0.6 and 0.8 thresholds. Due to the longer sentence length of NYT, the best threshold of NYT is less than WebNLG. In general, our model identifies triplets with high confidence.

5 Conclusion

In this paper, we focused on joint entity and relation extraction task and proposed a simple but very effective attention-based joint model. With the supervised multi-head self-attention mechanism, the relation detection module can flexibly detect *EntityPairOverlap* and *SingleEntityOverlap* relations by regarding each relation type as a subspace and maintaining the independence between them. Meanwhile, the entity extraction module can accurately recognize the entity boundary with CRF. Finally, we inferred the triplets based on the results of the two modules. Extensive experiments showed that our model achieves state-of-the-art performances.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under grant No. 61976119, the Natural Science Foundation of Tianjin under grant No. 18JCYBJC15800, and the Major Program of Science and Technology of Tianjin under grant No. 18ZXZNGX00310.

References

[Chan and Roth, 2011] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *ACL 2011*, pages 551–560, 2011.

[Dai *et al.*, 2019] Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *AAAI 2019*, pages 6300–6308, 2019.

[Fu *et al.*, 2019] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *ACL 2019*, pages 1409–1418, 2019.

[Gardent *et al.*, 2017] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for NLG micro-planners. In *ACL 2017*, pages 179–188, 2017.

[Gupta *et al.*, 2016] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *COLING 2016*, pages 2537–2547, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Lafferty *et al.*, 2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289, 2001.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009*, pages 1003–1011, 2009.

[Miwa and Bansal, 2016] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL 2016*, 2016.

[Miwa *et al.*, 2009] Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *EMNLP 2009*, pages 121–130, 2009.

[Nayak and Ng, 2019] Tapas Nayak and Hwee Tou Ng. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *CoRR*, abs/1911.09886, 2019.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP 2014*, pages 1532–1543, 2014.

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. 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*, pages 148–163, 2010.

[Ruder, 2016] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[Shen and Huang, 2016] Yatian Shen and Xuanjing Huang. Attention-based convolutional neural network for semantic relation extraction. In *COLING 2016*, pages 2526–2536, 2016.

[Sun *et al.*, 2019] Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. Joint type inference on entities and relations via graph convolutional networks. In *ACL 2019*, pages 1361–1370, 2019.

[Takanobu *et al.*, 2019] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI 2019*, pages 7072–7079, 2019.

[Zelenko *et al.*, 2003] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, 2003.

[Zeng *et al.*, 2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP 2015*, pages 1753–1762, 2015.

[Zeng *et al.*, 2018] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. In *ACL 2018*, pages 506–514, 2018.

[Zeng *et al.*, 2019a] Daojian Zeng, Haoran Zhang, and Qianying Liu. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *CoRR*, abs/1911.10438, 2019.

[Zeng *et al.*, 2019b] Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *EMNLP 2019*, pages 367–377, 2019.

[Zhang *et al.*, 2017] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *EMNLP 2017*, pages 1730–1740, 2017.

[Zheng *et al.*, 2017] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *ACL 2017*, pages 1227–1236, 2017.