

# PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction

Hengyi Zheng<sup>1,2,3</sup>, Rui Wen<sup>3</sup>, Xi Chen<sup>3,4\*</sup>, Yifan Yang<sup>3</sup>, Yunyan Zhang<sup>3</sup>  
Ziheng Zhang<sup>3</sup>, Ningyu Zhang<sup>5</sup>, Bin Qin<sup>2\*</sup>, Ming Xu<sup>2\*</sup>, Yefeng Zheng<sup>3</sup>

<sup>1</sup>College of Electronics and Information Engineering, Shenzhen University

<sup>2</sup>Information Technology Center, Shenzhen University

<sup>3</sup>Tencent Jarvis Lab, Shenzhen, China

<sup>4</sup>Platform and Content Group, Tencent <sup>5</sup>Zhejiang University

zhenghengyi2019@email.szu.edu.cn, {qinbin, xuming}@szu.edu.cn

{ruiwen, jasonxchen, tobyfyang, yunyanzhang, zihengzhang, yefengzheng}@tencent.com

## Abstract

Joint extraction of entities and relations from unstructured texts is a crucial task in information extraction. Recent methods achieve considerable performance but still suffer from some **inherent limitations**, such as **redundancy of relation prediction**, **poor generalization of span-based extraction** and **inefficiency**. In this paper, we decompose this task into **three subtasks**, *Relation Judgement*, *Entity Extraction* and *Subject-object Alignment* from a novel perspective and then propose a joint relational triple extraction framework based on **Potential Relation and Global Correspondence (PRGC)**. Specifically, we design **a component to predict potential relations**, which **constrains the following entity extraction to the predicted relation subset rather than all relations**; then a relation-specific sequence tagging component is applied to handle the overlapping problem between subjects and objects; finally, a global correspondence component is designed to align the subject and object into a triple with low-complexity. Extensive experiments show that PRGC achieves state-of-the-art performance on public benchmarks with higher efficiency and delivers consistent performance gain on complex scenarios of overlapping triples.<sup>1</sup>

## 1 Introduction

Identifying entity mentions and their relations which are in the form of a triple (subject, relation, object) from unstructured texts is an important task in information extraction. Some previous works proposed to address the task with pipelined approaches which include two steps: named entity recognition (Tjong Kim Sang and De Meulder, 2003; Ratnov and Roth, 2009) and relation prediction (Zelenko et al., 2002; Bunescu and Mooney,

\*Corresponding author.

<sup>1</sup>The source code and data are released at <https://github.com/hy-struggle/PRGC>.

Subtask	Model	Component
Relation Judgement	CasRel	None (Take all relations)
	TPLinker	None (Take all relations)
	PRGC	Potential Relation Prediction
Entity Extraction	CasRel	Span-based
	TPLinker	Span-based
	PRGC	Rel-Spec Sequence Tagging
Subject-object Alignment	CasRel	Cascade Scheme
	TPLinker	Token-pair Matrix
	PRGC	Global Correspondence

Table 1: Comparison of the proposed PRGC and previous methods in the respect of our new perspective with three subtasks.

2005; Pawar et al., 2017; Wang et al., 2020b). Recent end-to-end methods, which are based on either multi-task learning (Wei et al., 2020) or single-stage framework (Wang et al., 2020a), achieved promising performance and proved their effectiveness, but lacked in-depth study of the task.

To better comprehend the task and advance the state of the art, we propose a novel perspective to decompose the task into three subtasks: *i) Relation Judgement* which aims to identify relations in a sentence, *ii) Entity Extraction* which aims to extract all subjects and objects in the sentence and *iii) Subject-object Alignment* which aims to align the subject-object pair into a triple. On the basis, we review two end-to-end methods in Table 1. For the multi-task method named CasRel (Wei et al., 2020), the relational triple extraction is performed in two stages which applies object extraction to all relations. Obviously, the way to identify relations is redundant which contains numerous invalid operations, and the span-based extraction scheme which just pays attention to start/end position of an entity leads to poor generalization. Meanwhile, it is restricted to process one subject at a time due to its subject-object alignment mechanism, which is inefficient and difficult to deploy. For the single-stage

framework named TPLinker (Wang et al., 2020a), in order to avoid the exposure bias in subject-object alignment, it exploits a rather complicated decoder which leads to sparse label and low convergence rate while the problems of relation redundancy and poor generalization of span-based extraction are still unsolved.

To address aforementioned issues, we propose an end-to-end framework which consists of three components: *Potential Relation Prediction*, *Relation-Specific Sequence Tagging* and *Global Correspondence*, which fulfill the three subtasks accordingly as shown in Table 1.

For *Relation Judgement*, we predict potential relations by the *Potential Relation Prediction* component rather than preserve all redundant relations, which reduces computational complexity and achieves better performance, especially when there are many relations in the dataset.<sup>2</sup> For *Entity Extraction*, we use a more robust *Relation-Specific Sequence Tagging* component (Rel-Spec Sequence Tagging for short) to extract subjects and objects separately, to naturally handle overlapping between subjects and objects. For *Subject-object Alignment*, unlike TPLinker which uses a relation-based token-pair matrix, we design a relation-independent *Global Correspondence* matrix to determine whether a specific subject-object pair is valid in a triple.

Given a sentence, PRGC first predicts a subset of potential relations and a global matrix which contains the correspondence score between all subjects and objects; then performs sequence tagging to extract subjects and objects for each potential relation in parallel; finally enumerates all predicted entity pairs, which are then pruned by the global correspondence matrix. It is worth to note that the experiment (described in Section 5.2.1) shows that the *Potential Relation Prediction* component of PRGC is overall beneficial, even though it introduces the exposure bias that is usually mentioned in prior single-stage methods to prove their advantages.

Experimental results show that PRGC outperforms the state-of-the-art methods on public benchmarks with higher efficiency and fewer parameters. Detailed experiments on complex scenarios such as various overlapping patterns, which contain the *Single Entity Overlap (SEO)*, *Entity Pair Overlap*

(*EPO*) and *Subject Object Overlap (SOO)* types<sup>3</sup> show that our method owns consistent advantages. The main contributions of this paper are as follows:

1. We tackle the relational triple extraction task from a novel perspective which decomposes the task into three subtasks: *Relation Judgement*, *Entity Extraction* and *Subject-object Alignment*, and previous works are compared on the basis of the proposed paradigm as shown in Table 1.
2. Following our perspective, we propose a novel end-to-end framework and design three components with respect to the subtasks which greatly alleviate the problems of redundant relation judgement, poor generalization of span-based extraction and inefficient subject-object alignment, respectively.
3. We conduct extensive experiments on several public benchmarks, which indicate that our method achieves state-of-the-art performance, especially for complex scenarios of overlapping triples. Further ablation studies and analyses confirm the effectiveness of each component in our model.
4. In addition to higher accuracy, experiments show that our method owns significant advantages in complexity, number of parameters, floating point operations (FLOPs) and inference time compared with previous works.

## 2 Related Work

Traditionally, relational triple extraction has been studied as two separated tasks: entity extraction and relation prediction. Early works (Zelenko et al., 2002; Chan and Roth, 2011) apply the pipelined methods to perform relation classification between entity pairs after extracting all the entities. To establish the correlation between these two tasks, joint models have attracted much attention. Prior feature-based joint models (Yu and Lam, 2010; Li and Ji, 2014; Miwa and Sasaki, 2014; Ren et al., 2017) require a complicated process of feature engineering and rely on various NLP tools with cumbersome manual operations.

Recently, the neural network model which reduces manual involvement occupies the main part of the research. Zheng et al. (2017) proposed a

<sup>2</sup>For example, the WebNLG dataset (Gardent et al., 2017) has hundreds of relations but only seven valid relations for one sentence mostly.

<sup>3</sup>More details about overlapping patterns are shown in Appendix A.

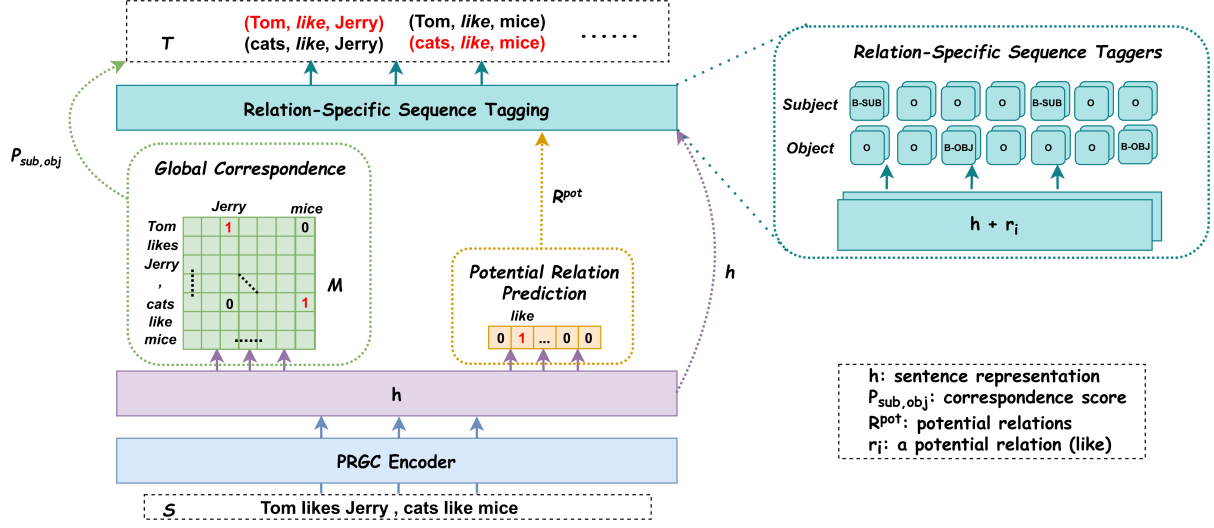


Figure 1: The overall structure of PRGC. Given a sentence  $S$ , PRGC predicts a subset of potential relations  $R^{pot}$  and a global correspondence  $M$  which indicates the alignment between subjects and objects. Then for each potential relation, a relation-specific sentence representation is constructed for sequence tagging. Finally we enumerate all possible subject-object pairs and get four candidate triples for this particular example, but only two triples are left (marked red) after applying the constraint of global correspondence.

novel tagging scheme that unified the role of the entity and the relation between entities in the annotations, thus **the joint extraction task was converted to a sequence labeling task but it failed to solve the overlapping problems**. Bekoulis et al. (2018) proposed to first extract all candidate entities, then predict the relation of every entity pair as a multi-head selection problem, which shared parameters but did not decode jointly. Nayak and Ng (2020) employed an encoder-decoder architecture and a pointer network based decoding approach where an entire triple was generated at each time step.

To handle the problems mentioned above, Wei et al. (2020) presented a cascade framework, which first identified all possible subjects in a sentence, then for each subject, applied span-based taggers to identify the corresponding objects based on each relation. This method leads to redundancy on relation judgement, and is not robust due to the span-based scheme on entity extraction. Meanwhile, the alignment scheme of subjects and objects limits its parallelization. In order to represent the relation of triple explicitly, Yuan et al. (2020) presented a relation-specific attention to assign different weights to the words in context under each relation, but it applied a naive heuristic nearest neighbor principle to combine the entity pairs which means the nearest subject and object entities will be combined into a triple. This is obviously not in accordance with intuition and fact. Meanwhile, it is also redundant

on relation judgement. The state-of-the-art method named TPLinker (Wang et al., 2020a) employs a token pair linking scheme which performs two  $O(n^2)$  matrix operations for extracting entities and aligning subjects with objects under each relation of a sentence, causing extreme redundancy on relation judgement and complexity on subject-object alignment, respectively. And it also suffers from the disadvantage of span-based extraction scheme.

### 3 Method

In this section, we first introduce our perspective of relational triple extraction task with a principled problem definition, then elaborate each component of the PRGC model. An overview illustration of PRGC is shown in Figure 1.

#### 3.1 Problem Definition

The input is a sentence  $S = \{x_1, x_2, \dots, x_n\}$  with  $n$  tokens. The desired outputs are relational triples as  $T(S) = \{(s, r, o) | s, o \in E, r \in R\}$ , where  $E$  and  $R$  are the entity and relation sets, respectively. In this paper, the problem is decomposed into three subtasks:

**Relation Judgement** For the given sentence  $S$ , this subtask predicts potential relations it contains. The output of this task is  $Y_r(S) = \{r_1, r_2, \dots, r_m | r_i \in R\}$ , where  $m$  is the size of potential relation subset.

**Entity Extraction** For the given sentence  $S$  and a predicted potential relation  $r_i$ , this subtask identifies the tag of each token with **BIO (i.e., Begin, Inside and Outside) tag scheme** (Tjong Kim Sang and Veenstra, 1999; Ratinov and Roth, 2009). Let  $t_j$  denote the tag. The output of this task is  $Y_e(S, r_i | r_i \in R) = \{t_1, t_2, \dots, t_n\}$ .

**Subject-object Alignment** For the given sentence  $S$ , this subtask predicts the correspondence score between the start tokens of subjects and objects. **That means only the pair of start tokens of a true triple has a high score**, while the other token pairs have a low score. Let  $M$  denote the global correspondence matrix. **The output of this task is  $Y_s(S) = M \in \mathbb{R}^{n \times n}$** .

### 3.2 PRGC Encoder

The output of PRGC Encoder is  $Y_{enc}(S) = \{h_1, h_2, \dots, h_n | h_i \in \mathbb{R}^{d \times 1}\}$ , where  $d$  is the embedding dimension, and  $n$  is the number of tokens. We use a pre-trained BERT model<sup>4</sup> (Devlin et al., 2019) to encode the input sentence for a fair comparison, but theoretically it can be extended to other encoders, such as Glove (Pennington et al., 2014) and RoBERTa (Liu et al., 2019).

### 3.3 PRGC Decoder

In this section, we describe the instantiation of PRGC decoder that consists of three components.

#### 3.3.1 Potential Relation Prediction

This component is shown as the orange box in Figure 1 where  $R^{pot}$  is the potential relations. Different from previous works (Wei et al., 2020; Yuan et al., 2020; Wang et al., 2020a) which redundantly perform entity extraction to every relation, given a sentence, we first predict a subset of potential relations that possibly exist in the sentence, and then the entity extraction only needs to be applied to these potential relations. Given the embedding  $\mathbf{h} \in \mathbb{R}^{n \times d}$  of a sentence with  $n$  tokens, each element of this component is obtained as:

$$\begin{aligned} \mathbf{h}^{avg} &= Avgpool(\mathbf{h}) \in \mathbb{R}^{d \times 1} \\ P_{rel} &= \sigma(\mathbf{W}_r \mathbf{h}^{avg} + \mathbf{b}_r) \end{aligned} \quad (1)$$

where  $Avgpool$  is the average pooling operation (Lin et al., 2014),  $\mathbf{W}_r \in \mathbb{R}^{d \times 1}$  is a trainable weight and  $\sigma$  denotes the sigmoid function.

We model it as a multi-label binary classification task, and the corresponding relation will be assigned with tag 1 if the probability exceeds a certain threshold  $\lambda_1$  or with tag 0 otherwise (as shown in Figure 1), so next we just need to apply the relation-specific sequence tagging to the predicted relations rather than all relations.

#### 3.3.2 Relation-Specific Sequence Tagging

As shown in Figure 1, we obtain several relation-specific sentence representations of potential relations described in Section 3.3.1. Then, we perform two sequence tagging operations to extract subjects and objects, respectively. The reason why we extract subjects and objects separately is to handle the special overlapping pattern named *Subject Object Overlap (SOO)*. We can also simplify it to one sequence tagging operation with two types of entities if there are no *SOO* patterns in the dataset.<sup>5</sup>

For the sake of simplicity and fairness, we abandon the traditional LSTM-CRF (Panchendrarajan and Amaresan, 2018) network but adopt the simple fully connected neural network. Detailed operations of this component on each token are as follows:

$$\begin{aligned} \mathbf{P}_{i,j}^{sub} &= Softmax(\mathbf{W}_{sub}(\mathbf{h}_i + \mathbf{u}_j) + \mathbf{b}_{sub}) \\ \mathbf{P}_{i,j}^{obj} &= Softmax(\mathbf{W}_{obj}(\mathbf{h}_i + \mathbf{u}_j) + \mathbf{b}_{obj}) \end{aligned} \quad (2)$$

where  $\mathbf{u}_j \in \mathbb{R}^{d \times 1}$  is the  $j$ -th relation representation in a trainable embedding matrix  $\mathbf{U} \in \mathbb{R}^{d \times n_r}$  where  $n_r$  is the size of full relation set,  $\mathbf{h}_i \in \mathbb{R}^{d \times 1}$  is the encoded representation of the  $i$ -th token, and  $\mathbf{W}_{sub}, \mathbf{W}_{obj} \in \mathbb{R}^{d \times 3}$  are trainable weights where the size of tag set  $\{B, I, O\}$  is 3.

#### 3.3.3 Global Correspondence

After sequence tagging, we acquire all possible subjects and objects with respect to a relation of the sentence, then we use a global correspondence matrix to determine the correct pairs of the subjects and objects. It should be noted that the global correspondence matrix can be learned simultaneously with potential relation prediction since it is independent of relations. The detailed process is as follows: first we enumerate all the possible subject-object pairs; then we check the corresponding score in the global matrix for each pair, retain it if the value exceeds a certain threshold  $\lambda_2$  or filter it out otherwise.

<sup>4</sup>Please refer to the original paper (Devlin et al., 2019) for detailed descriptions.

<sup>5</sup>For example, the *SOO* pattern is rare in the NYT (Riedel et al., 2010) dataset.



Dataset	#Sentences			Details of test set							
	Train	Valid	Test	Normal	SEO	EPO	SOO	$N = 1$	$N > 1$	#Triples	#Relations
NYT*	56,195	4,999	5,000	3,266	1,297	978	45	3,244	1,756	8,110	24
WebNLG*	5,019	500	703	245	457	26	84	266	437	1,591	171
NYT	56,196	5,000	5,000	3,071	1,273	1,168	117	3,089	1,911	8,616	24
WebNLG	5,019	500	703	239	448	6	85	256	447	1,607	216

Table 2: Statistics of datasets used in our experiments where  $N$  is the number of triples in a sentence. Note that one sentence can have SEO, EPO and SOO overlapping patterns simultaneously, and the relation set of WebNLG is bigger than WebNLG\*.

As shown in the green matrix  $M$  in Figure 1, given a sentence with  $n$  tokens, the shape of global correspondence matrix will be  $\mathbb{R}^{n \times n}$ . Each element of this matrix is about the start position of a paired subject and object, which represents the confidence level of a subject-object pair, the higher the value, the higher the confidence level that the pair belongs to a triple. For example, the value about “Tom” and “Jerry” at row 1, column 3 will be high if they are in a correct triple such as “(Tom, like, Jerry)”. The value of each element in the matrix is obtained as follows:

$$P_{i_{sub}, j_{obj}} = \sigma(\mathbf{W}_g[\mathbf{h}_i^{sub}; \mathbf{h}_j^{obj}] + \mathbf{b}_g) \quad (3)$$

where  $\mathbf{h}_i^{sub}, \mathbf{h}_j^{obj} \in \mathbb{R}^{d \times 1}$  are the encoded representation of the  $i$ -th token and  $j$ -th token in the input sentence forming a potential pair of subject and object,  $\mathbf{W}_g \in \mathbb{R}^{2d \times 1}$  is a trainable weight, and  $\sigma$  is the sigmoid function.

### 3.4 Training Strategy

We train the model jointly, optimize the combined objective function during training time and share the parameters of the PRGC encoder. The total loss can be divided into three parts as follows:

$$\mathcal{L}_{rel} = -\frac{1}{n_r} \sum_{i=1}^{n_r} (y_i \log P_{rel} + (1 - y_i) \log (1 - P_{rel})) \quad (4)$$

$$\mathcal{L}_{seq} = -\frac{1}{2 \times n \times n_r^{pot}} \sum_{t \in \{sub, obj\}} \sum_{j=1}^{n_r^{pot}} \sum_{i=1}^n \mathbf{y}_{i,j}^t \log \mathbf{P}_{i,j}^t \quad (5)$$

$$\mathcal{L}_{global} = -\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (y_{i,j} \log P_{i_{sub}, j_{obj}} + (1 - y_{i,j}) \log (1 - P_{i_{sub}, j_{obj}})) \quad (6)$$

where  $n_r$  is the size of full relation set and  $n_r^{pot}$  is the size of potential relation subset of the sentence. The total loss is the sum of these three parts,

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{rel} + \beta \mathcal{L}_{seq} + \gamma \mathcal{L}_{global}. \quad (7)$$

Performance might be better by carefully tuning the weight of each sub-loss, but we just assign equal weights for simplicity (i.e.,  $\alpha = \beta = \gamma = 1$ ).

## 4 Experiments

### 4.1 Datasets and Experimental Settings

For fair and comprehensive comparison, we follow Yu et al. (2019) and Wang et al. (2020a) to evaluate our model on two public datasets NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017), both of which have two versions, respectively. We denote the different versions as NYT\*, NYT and WebNLG\*, WebNLG. Note that NYT\* and WebNLG\* annotate the last word of entities, while NYT and WebNLG annotate the whole entity span. The statistics of the datasets are described in Table 2. Following Wei et al. (2020), we further characterize the test set w.r.t. the overlapping patterns and the number of triples per sentence.

Following prior works mentioned above, an extracted relational triple is regarded as correct only if it is an exact match with ground truth, which means the last word of entities or the whole entity span (depending on the annotation protocol) of both subject and object and the relation are all correct. Meanwhile, we report the standard micro Precision (Prec.), Recall (Rec.) and F1-score for all the baselines. The implementation details are shown in Appendix B.

We compare PRGC with eight strong baseline models and the state-of-the-art models CasRel (Wei et al., 2020) and TPLinker (Wang et al., 2020a). All the experimental results of the baseline models are directly taken from Wang et al. (2020a) unless specified.

### 4.2 Experimental Results

In this section, we present the overall results and the results of complex scenarios, while the results on different subtasks corresponding to different

Model	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging (Zheng et al., 2017)	-	-	-	-	-	-	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-	-	-	-
MultiHead (Bekoulis et al., 2018)	-	-	-	-	-	-	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel (Fu et al., 2019)	63.9	60.0	61.9	44.7	41.1	42.9	-	-	-	-	-	-
OrderCopyRE (Zeng et al., 2019)	77.9	67.2	72.1	63.3	59.9	61.6	-	-	-	-	-	-
ETL-span (Yu et al., 2019)	84.9	72.3	78.1	84.0	91.5	87.6	85.5	71.7	78.0	84.3	82.0	83.1
WDec (Nayak and Ng, 2020)	<b>94.5</b>	76.2	84.4	-	-	-	-	-	-	-	-	-
RSAN <sup>‡</sup> (Yuan et al., 2020)	-	-	-	-	-	-	85.7	83.6	84.6	80.5	83.8	82.1
CasRel <sub>Random</sub> <sup>‡</sup> (Wei et al., 2020)	81.5	75.7	78.5	84.7	79.5	82.0	-	-	-	-	-	-
CasRel <sub>BERT</sub> <sup>‡</sup> (Wei et al., 2020)	89.7	89.5	89.6	<u>93.4</u>	90.1	91.8	-	-	-	-	-	-
TPLinker <sub>BERT</sub> <sup>‡</sup> (Wang et al., 2020a)	91.3	<b>92.5</b>	<u>91.9</u>	91.8	<u>92.0</u>	<u>91.9</u>	<u>91.4</u>	<b>92.6</b>	<u>92.0</u>	<u>88.9</u>	<u>84.5</u>	<u>86.7</u>
PRGC <sub>Random</sub>	89.6	82.3	85.8	90.6	88.5	89.5	87.8	83.8	85.8	82.5	79.2	80.8
PRGC <sub>BERT</sub>	<u>93.3</u>	<u>91.9</u>	<b>92.6</b>	<b>94.0</b>	<b>92.1</b>	<b>93.0</b>	<b>93.5</b>	<u>91.9</u>	<b>92.7</b>	<b>89.9</b>	<b>87.2</b>	<b>88.5</b>

Table 3: Comparison (%) of the proposed PRGC method with the prior works. **Bold** marks the highest score, underline marks the second best score and <sup>‡</sup> marks the results reported by the original papers.

	Model	Normal	SEO	EPO	SOO
NYT*	OrderCopyRE	71.2	69.4	72.8	-
	ETL-Span	88.5	87.6	60.3	-
	CasRel	87.3	91.4	92.0	77.0§
	TPLinker	90.1	93.4	94.0	<b>90.1</b> §
	PRGC	<b>91.0</b>	<b>94.0</b>	<b>94.5</b>	81.8
WebNLG*	OrderCopyRE	65.4	60.1	67.4	-
	ETL-Span	87.3	91.5	80.5	-
	CasRel	89.4	92.2	94.7	90.4§
	TPLinker	87.9	92.5	95.3	86.0§
	PRGC	<b>90.4</b>	<b>93.6</b>	<b>95.9</b>	<b>94.6</b>

Table 4: F1-score (%) of sentences with different overlapping patterns. **Bold** marks the highest score and § marks results obtained by official implementations.

	Model	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$
NYT*	OrderCopyRE	71.7	72.6	72.5	77.9	45.9
	ETL-Span	88.5	82.1	74.7	75.6	76.9
	CasRel	88.2	90.3	91.9	94.2	83.7
	TPLinker	90.0	92.8	93.1	<b>96.1</b>	90.0
	PRGC	<b>91.1</b>	<b>93.0</b>	<b>93.5</b>	95.5	<b>93.0</b>
WebNLG*	OrderCopyRE	63.4	62.2	64.4	57.2	55.7
	ETL-Span	82.1	86.5	91.4	89.5	91.1
	CasRel	89.3	90.8	94.2	92.4	90.9
	TPLinker	88.0	90.1	94.6	93.3	91.6
	PRGC	<b>89.9</b>	<b>91.6</b>	<b>95.0</b>	<b>94.8</b>	<b>92.8</b>

Table 5: F1-score (%) of sentences with different numbers of triples where  $N$  is the number of triples in a sentence. **Bold** marks the highest score.

components in our model are described in Appendix C.

#### 4.2.1 Overall Results

Table 3 shows the results of our model against other baseline methods on four datasets. Our PRGC method outperforms them in respect of almost all evaluation metrics even if compared with the recent strongest baseline (Wang et al., 2020a) which is quite complicated.

At the same time, we implement PRGC<sub>Random</sub> to validate the utility of our PRGC decoder, where all parameters of the encoder BERT are randomly initialized. The performance of PRGC<sub>Random</sub> demonstrates that our decoder framework (which obtains 7% improvements than CasRel<sub>Random</sub>) is still more competitive and robust than others even

without taking advantage of the pre-trained BERT language model.

It is important to note that even though TPLinker<sub>BERT</sub> has more parameters than CasRel<sub>BERT</sub>, it only obtains 0.1% improvements on the WebNLG\* dataset, and the authors attributed this to problems with the dataset itself. However, our model achieves a 10× improvements than TPLinker on the WebNLG\* dataset and a significant promotion on the WebNLG dataset. The reason behind this is that the relation judgement component of our model greatly reduces redundant relations particularly in the versions of WebNLG which contain hundreds of relations. In other words, the reduction in negative relations provides an additional boost compared to the models that perform entity extraction under every relation.

Dataset	Model	Complexity	FLOPs (M)	Params <sub>decoder</sub>	Inference Time (1 / 24)	F1-Score
NYT*	CasRel	$O(kn) \rightarrow O(n^2)$	<b>15.05</b>	75,362	24.2 / -	89.6
	TPLinker	$O(kn^2)$	1105.92	110,736	38.8 / 7.7	91.9
	PRGC	$O(n^2)$	32.60	<b>66,085</b>	<b>13.5 / 4.4</b>	<b>92.6</b>
WebNLG*	CasRel	$O(kn^2)$	105.37	527,534	30.5 / -	91.8
	TPLinker	$O(n^3)$	7879.68	788,994	41.7 / 13.2	91.9
	PRGC	$O(n^2)$	<b>33.75</b>	<b>409,534</b>	<b>14.4 / 5.2</b>	<b>93.0</b>

Table 6: Comparison of model efficiency on both NYT\* and WebNLG\* datasets. Results except F1-score (%) of other methods are obtained by the official implementation with default configuration, and **bold** marks the best result. Complexity are the computation complexity, FLOPs and Params<sub>decoder</sub> are both calculated on the decoder, and we measure the inference time (ms) with the batch size of 1 and 24, respectively.

#### 4.2.2 Detailed Results on Complex Scenarios

Following previous works (Wei et al., 2020; Yuan et al., 2020; Wang et al., 2020a), to verify the capability of our model in handling different overlapping patterns and sentences with different numbers of triples, we conduct further experiments on NYT\* and WebNLG\* datasets.

As shown in Table 4, our model exceeds all the baselines in all overlapping patterns in both datasets except the *SOO* pattern in the NYT\* dataset. Actually, the observation on the latter scenario is not reliable due to the very low percentage of *SOO* in NYT\* (i.e., 45 out of 8,110 as shown in Table 2). As shown in Table 5, the performance of our model is better than others almost in every subset regardless of the number of triples. In general, these two further experiments adequately show the advantages of our model in complex scenarios.

## 5 Analysis

### 5.1 Model Efficiency

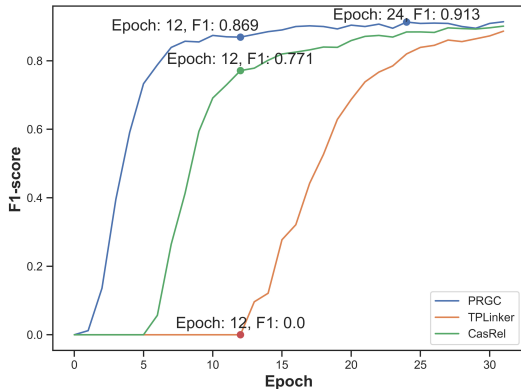


Figure 2: F1-score with respect to the epoch number on the WebNLG\* validation set of different methods. Results of CasRel and TPLinker are obtained by the official implementation with default configuration.

As shown in Table 6, we evaluate the model ef-

iciency with respect to *Complexity*, floating point operations (*FLOPs*) (Molchanov et al., 2017), parameters of the decoder (*Params<sub>decoder</sub>*) and *Inference Time*<sup>6</sup> of CasRel, TPLinker and PRGC in two datasets which have quite different characteristics in the size of relation set, the average number of relations per sentence and the average number of subjects per sentence. All experiments are conducted with the same hardware configuration. Because the number of subjects in a sentence varies, it is difficult for CasRel to predict objects in a heterogeneous batch, and it is restricted to set batch size to 1 in the official implementation (Wang et al., 2020a). For the sake of fair comparison, we set batch size to 1 and 24 to verify the single-thread decoding speed and parallel processing capability, respectively.

The results indicate that the single-thread decoding speed of PRGC is 2× as CasRel and 3× as TPLinker, and our model is significantly better than TPLinker in terms of parallel processing. Note that the model efficiency of CasRel and TPLinker decreases as the size of relation set increases but our model is not affected by the size of relation set, thus PRGC overwhelmingly outperforms both models in terms of all the indicators of efficiency in the WebNLG\* dataset. Compared with the state-of-the-art model TPLinker, PRGC is an order of magnitude lower in *Complexity* and the *FLOPs* is even 200 times lower, thus PRGC has fewer parameters and obtains 3× speedup in the inference phase while the F1-score is improved by 1.1%. Even though CasRel has lower *Complexity* and *FLOPs* in the NYT\* dataset, PRGC still has significant advantages and obtains a 5× speedup in the inference time and 3% improvements in F1-score. Meanwhile, Figure 2 proves our advantage in convergence rate. These all confirm the efficiency of

<sup>6</sup>The *FLOPs* and *Params<sub>decoder</sub>* are calculated via: <https://github.com/sovrasov/flops-counter.pytorch>.

Texts	Ground Truth	Span-based	Rel-Spec Sequence Tagging
A Fortress of Grey Ice is from the United States. That country has an ethnic group called Asian Americans and they speak English, same as in Great Britain.	(Grey Ice, <i>country</i> , United States) (United States, <i>ethnicGroup</i> , Asian Americans)	(Grey Ice, <i>country</i> , United States) (United States, <i>country</i> , Asian Americans)✗ (Grey Ice, <i>ethnicGroup</i> , United States)✗ (United States, <i>ethnicGroup</i> , Asian Americans)	(Grey Ice, <i>country</i> , United States) (United States, <i>ethnicGroup</i> , Asian Americans)
Above the Veil is an Australian novel and the sequel to Aenir and Castle. It was later followed by Into Battle and The Violet Keystone.	(Above the Veil, <i>precededBy</i> , Aenir)	(Above the Veil, <i>precededBy</i> , Aenir and Castle. It was later followed by Into Battle and The Violet Keystone.)✗	(Above the Veil, <i>precededBy</i> , Aenir)

Figure 3: Case study for the ablation study of Rel-Spec Sequence Tagging. Examples are from WebNLG\*, and we supplement the whole entity span through WebNLG to facilitate viewing. The red cross marks bad cases, the correct entities are in **bold** and the correct relations are colored.

our model.

## 5.2 Ablation Study

In this section, we conduct ablation experiments to demonstrate the effectiveness of each component in PRGC with results reported in Table 7.

	Model	Prec.	Rec.	F1
NYT*	<b>PRGC</b>	<b>93.3</b>	<b>91.9</b>	<b>92.6</b>
	–Potential Relation Prediction	91.5	91.7	91.6
	–Rel-Spec Sequence Tagging	63.8	91.7	75.2
	–Global Correspondence	71.6	91.2	80.2
WebNLG*	<b>PRGC</b>	<b>94.0</b>	<b>92.1</b>	<b>93.0</b>
	–Potential Relation Prediction	80.0	88.2	83.9
	–Rel-Spec Sequence Tagging	33.2	91.3	48.7
	–Global Correspondence	55.9	91.6	69.4

Table 7: Ablation study of PRGC (%).

### 5.2.1 Effect of Potential Relation Prediction

We use each relation in the relation set to perform sequence tagging when we remove the *Potential Relation Prediction* component to avoid the exposure bias. As shown in Table 7, the precision significantly decreases without this component, because the number of predicted triples increases due to relations not presented in the sentences, especially in the WebNLG\* dataset where the size of relation set is much bigger and brings tremendous relation redundancy. Meanwhile, with the increase of relation number in sentences, the training and inference time increases three to four times. Through this experiment, the validity of this component that aims to predict a potential relation subset is proved, which is not only beneficial to model accuracy, but also to efficiency.

### 5.2.2 Effect of Rel-Spec Sequence Tagging

As a comparison for sequence tagging scheme, following Wei et al. (2020) and Wang et al. (2020a), we perform binary classification to detect start

and end positions of an entity with the span-based scheme. As shown in Table 7, span-based scheme brings significant decline of performance.

Through the case study shown in Figure 3, we observe that the span-based scheme tends to extract long entities and identify the correct subject-object pairs but ignore their relation. That is because the model is inclined to remember the position of an entity rather than understand the underlying semantics. However, the sequence tagging scheme used by PRGC performs well in both cases, and experimental results prove that our tagging scheme is more robust and generalizable.

### 5.2.3 Effect of Global Correspondence

For comparison, we exploit the heuristic nearest neighbor principle to combine the subject-object pairs which was used by Zheng et al. (2017) and Yuan et al. (2020). As shown in Table 7, the precision also significantly decreases without *Global Correspondence*, because the number of predicted triples increases with many mismatched pairs when the model loses the constraint imposed by this component. This experiment proves that the *Global Correspondence* component is effective and greatly outperforms the heuristic nearest neighbor principle in the subject-object alignment task.

## 6 Conclusion

In this paper, we presented a brand-new perspective and introduced a novel joint relational extraction framework based on **Potential Relation** and **Global Correspondence**, which greatly alleviates the problems of redundant relation judgement, poor generalization of span-based extraction and inefficient subject-object alignment. Experimental results showed that our model achieved the state-of-the-art performance in the public datasets and successfully handled many complex scenarios with higher efficiency.



## References

- Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. [Joint entity recognition and relation extraction as a multi-head selection problem](#). *Expert Systems with Applications*, 114:34–45.
- Razvan C. Bunescu and Raymond J. Mooney. 2005. [A shortest path dependency kernel for relation extraction](#). In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, BC.
- Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560.
- J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.
- Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 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*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. [Creating training corpora for NLG micro-planners](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, San Diego, CA, USA, Conference Track Proceedings*.
- Qi Li and Heng Ji. 2014. [Incremental joint extraction of entity mentions and relations](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.
- Min Lin, Qiang Chen, and Shuicheng Yan. 2014. [Net-work in network](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2017. [Fixing weight decay regularization in Adam](#). *CoRR*, abs/1711.05101.
- Makoto Miwa and Yutaka Sasaki. 2014. [Modeling joint entity and relation extraction with table representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.
- P Molchanov, S Tyree, T Karras, T Aila, and J Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017- Conference Track Proceedings*.
- Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8528–8535.
- Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. [Bidirectional LSTM-CRF for named entity recognition](#). In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.
- Sachin Pawar, Girish K. Palshikar, and Pushpak Bhat-tacharyya. 2017. [Relation extraction : A survey](#). *CoRR*, abs/1712.05191.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Lev Ratinov and Dan Roth. 2009. [Design challenges and misconceptions in named entity recognition](#). In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.
- Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024.
- S. Riedel, Limin Yao, and A. McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

- Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. [Representing text chunks](#). In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway. Association for Computational Linguistics.
- Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020a. [TPLinker: Single-stage joint extraction of entities and relations through token pair linking](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Zifeng Wang, Rui Wen, Xi Chen, Shao-Lun Huang, Ningyu Zhang, and Yefeng Zheng. 2020b. [Finding influential instances for distantly supervised relation extraction](#). *CoRR*, abs/2009.09841.
- Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488.
- Bowen Yu, Zhenyu Zhang, Jianlin Su, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. 2019. Joint extraction of entities and relations based on a novel decomposition strategy. *24th European Conference on Artificial Intelligence - ECAI 2020*.
- Xiaofeng Yu and Wai Lam. 2010. [Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach](#). In *The 28th International Conference on Computational Linguistics*, pages 1399–1407, Beijing, China.
- Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *International Joint Conference on Artificial Intelligence*, pages 4054–4060. Association for the Advancement of Artificial Intelligence.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. [Kernel methods for relation extraction](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - Volume 10*, page 71–78, USA. Association for Computational Linguistics.
- Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. [Learning the extraction order of multiple relational facts in a sentence with reinforcement learning](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 367–377, Hong Kong, China. Association for Computational Linguistics.
- Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 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)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.
- Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 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 (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

## Appendix

### A Overlapping Patterns

As shown in Figure 4, the *Normal*, *SEO* and *EPO* patterns are usually mentioned in prior works (Nayak and Ng, 2020; Wei et al., 2020; Yuan et al., 2020; Wang et al., 2020a), and *SOO* is a special pattern we identified and addressed.

	Texts	Triples
Normal	The [United States] president [Joe Biden] will visit [Beijing], [China].	(United States, <i>president</i> , Joe Biden) (China, <i>contains</i> , Beijing)
SEO	[LeBron James] and [Anthony Davis] live in [Los Angeles], and [Klay Thompson] was born in there.	(LeBron James, <i>live in</i> , Los Angeles) (Anthony Davis, <i>live in</i> , Los Angeles) (Klay Thompson, <i>born in</i> , Los Angeles)
EPO	[Beijing] is the capital city of [China].	(China, <i>capital city</i> , Beijing) (China, <i>contains</i> , Beijing)
SOO	[[Lebron] James] is a good basketball player.	(Lebron James, <i>first name</i> , Lebron)

Figure 4: Examples of the *Normal*, *Single Entity Overlap (SEO)*, *Entity Pair Overlap (EPO)* and *Subject Object Overlap (SOO)* patterns. The overlapping entities are in **bold**.

### B Implementation Details

We implement our model with PyTorch and optimize the parameters by Adam (Kingma and Ba, 2015) with batch size of 64/6 for NYT/WebNLG. The encoder learning rate for BERT is set as  $5 \times 10^{-5}$ , and the decoder learning rate is set as 0.001 in order to converge rapidly. We also conduct weight decay (Loshchilov and Hutter, 2017) with a rate of 0.01.

For fair comparison, we use the BERT-Base-Cased English model<sup>7</sup> as our encoder, and set the max length of an input sentence to 100, which is the same as previous works (Wei et al., 2020; Wang et al., 2020a). Our experiments are conducted on the workstation with an Intel Xeon E5 2.40 GHz CPU, 128 GB memory, an NVIDIA Tesla V100 GPU, and CentOS 7.2. We train the model for 100 epochs and choose the last model. The performance will be better if the higher the threshold of *Potential Relation Prediction* ( $\lambda_1$ ), but tuning the threshold of *Global Correspondence* ( $\lambda_2$ ) will not help which is consistent with the analysis in Appendix C.

<sup>7</sup>Available at <https://huggingface.co/bert-base-cased>.

### C Results on Different Subtasks

To further verify the results of the three subtasks in our new perspective and the performance of each component in our model, we present more detailed evaluations on NYT\* and WebNLG\* datasets in Table 8.

Subtask	NYT*			WebNLG*		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Relation Judgement	95.3	<b>96.3</b>	95.8	92.8	<b>96.2</b>	94.5
Entity Extraction (Subject)	81.2	<b>95.5</b>	87.8	69.4	<b>96.3</b>	80.7
Entity Extraction (Object)	82.8	<b>95.8</b>	88.8	72.1	<b>95.7</b>	82.2
Subject-object Alignment	94.0	92.3	<b>93.1</b>	96.0	93.4	<b>94.7</b>
Combination of Above All	93.3	91.9	92.6	94.0	92.1	93.0

Table 8: Evaluation (%) of different subtasks on the NYT\* and WebNLG\* datasets. Each subtask corresponds to a component in our model. **Bold** marks the most important metric of each subtask.

**Relation Judgement** We evaluate outputs of the *Potential Relation Prediction* component which are potential relations contained in a sentence. Recall is more important for this task because if a true relation is missed, it will not be recovered in the following steps. We get high recall in this task and the results show that effectiveness of *Potential Relation Prediction* component is not affected by the size of relation set.

**Entity Extraction** This task is related to the *Relation-Specific Sequence Tagging* component, and we evaluate it as a Named Entity Recognition (NER) task with two types of entities: subjects and objects. The predicted entities are from all potential relations of a sentence, and recall is more important for this task because most false negatives can be filtered out by *Subject-object Alignment*. Experimental results show that we extract almost all correct entities, and it further proves that the influence of the exposure bias is negligible.

**Subject-object Alignment** This task is related to the *Global Correspondence* component, and we just evaluate the entity pair in a triple and ignore the relation. Both recall and precision are important for this component, experimental results indicate that our alignment scheme is useful but still can be further improved, especially in the recall.

Overall, the combination of three components in our model accomplishes the relational triple extraction task with a fine-grained perspective, and achieves better and solid results.