Entity Relative Position Representation based Multi-head Selection for Joint Entity and Relation Extraction

Tianyang Zhao[1]* and **Zhao Yan**[2] and **Yunbo Cao**[2] and **Zhoujun Li**[1]
[1]State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2]Tencent Cloud Xiaowei, Beijing, China
{tyzhao,lizj}@buaa.edu.cn, {zhaoyan,yunbocao}@tencent.com

Abstract

Joint entity and relation extraction has received increasing interests recently, due to the capability of utilizing the interactions between both steps. Among existing studies, the Multi-Head Selection (MHS) framework is efficient in extracting entities and relations simultaneously. However, the method is weak for its limited performance. In this paper, we propose several effective insights to address this problem. First, we propose an entity-specific Relative Position Representation (eRPR) to allow the model to fully leverage the distance information between entities and context tokens. Second, we introduce an auxiliary Global Relation Classification (GRC) to enhance the learning of local contextual features. Moreover, we improve the semantic representation by adopting a pre-trained language model BERT as the feature encoder. Finally, these new keypoints are closely integrated with the multi-head selection framework and optimized jointly. Extensive experiments on two benchmark datasets demonstrate that our approach overwhelmingly outperforms previous works in terms of all evaluation metrics, achieving significant improvements for relation F1 by +2.40% on CoNLL04 and +1.90% on ACE05, respectively.

1 Introduction

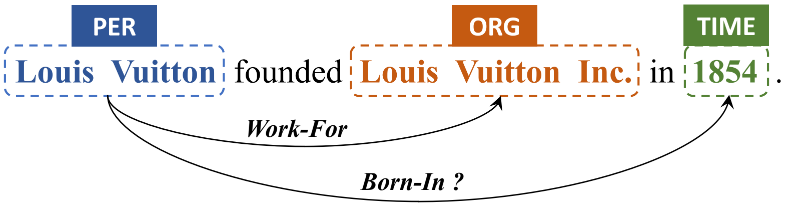
The entity-relation extraction task aims to recognize the entity spans from a sentence and detect the relations holds between two entities. Generally, it can be formed as extracting triplets (e_1, r, e_2), which denotes that the relation r holds between the head entity e_1 and the tail entity e_2, i.e., (*John Smith*, `Live-In`, *Atlanta*). It plays a vital role in the information extraction area and has attracted increasing attention in recent years.

Traditional pipelined methods divide the task into two phases, named entity recognition (NER) and relation extraction (RE) (Miwa et al., 2009; Chan and Roth, 2011; Lin et al., 2016). As such methods neglect the underlying correlations between the two phases and suffer from the error propagation issue, recent works propose to extract entities and relations jointly. These joint models fall into two paradigms. The first paradigm can be denoted as $(e_1, e_2) \rightarrow r$, which first recognizes all entities in the sentence, then classifies the relation depend on each extracted entity pairs. However, these methods require enumerating all possible entity pairs and the relation classification may be affected by the redundant ones. While another paradigm is referred as $e1 \rightarrow (r, e2)$, which detects head entities first and then predicts the corresponding relations and tail entities (Bekoulis et al., 2018; Li et al., 2019; Zhao et al., 2020). Comparing with the first paradigm, the second one can jointly identify entities and all the possible relations between them at once. A typical approach is the Multi-Head Selection (MHS) framework (Bekoulis et al., 2018). It first recognizes head entities using the BiLSTM-CRF structure and then performs tail entity extraction and relation extraction in one pass based on multiclass classification. The advantage of the MHS framework is obvious - it is efficient to work with the scenario, that one entity can involve several relational triplets, making this solution suitable for large scale practical applications. In this paper, we focus on the second paradigm of the joint models, especially on the MHS framework.

Despite the efficiency of the MHS framework, it is weak for the limited performance comparing with other complex models. Intuitively, the distance between entities and other context tokens provide impor-

* Work done during an internship at Tencent.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 962-973,
Hainan, China, October 30 - Novermber 1, 2020. (c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China



Golden Relation:
(Louis Vuitton, ***Work-For***, Louis Vuitton Inc.)

Figure 1: An example to show the impact of entity-specific relative position.

tant evidence for entity and relation extraction. Meanwhile, the distance information of non-entity words is less important. As shown in the sentence of Fig. 1, the "Louis Vuitton" that is far from the word "Inc." is a person entity, while the one adjacent to "Inc." denotes an organization. Such entity-specific relative position can be a useful indicator to differentiate entity tokens and non-entity tokens and enhance interactions between entities. While the existing model pays equal attention to each context tokens and ignores the relative distance information of entities. As a result, the entity-specific features may become less obscure and mislead the relation selection. Second, the existing model predicts the relations and tail entities merely based on the local contextual features of the head entity, and the incomplete local information may confuse the predictor. While the semantic of the whole sentence always has a significant impact on relation prediction. For example, in Fig. 1, the relation between "Louis Vuitton" and "1854" may easily be mislabeled as "`Born-In`" without considering the meaning of the whole sentence. Therefore, the global semantics should also be taken into account.

To address the aforementioned limitations, we present several new key points to improve the existing multi-head selection framework. First, we propose an entity-specific Relative Position Representation (eRPR) to leverage the distance information between entities and their contextual tokens, which provides important positional information for each entity. Then, in order to better consider the sentence-level semantic during relation prediction, we add up an auxiliary Global Relational Classification (GRC) to guide the optimization of local context features. In addition, different from the original MHS structure, we adopt the pre-trained transformer-based encoder (BERT) to enhance the ability of semantic representations. Notably, the proposed method can address the entity and multiple-relation extraction simultaneously and without relying on any external parsing tools or hand-crafted features. We conduct extensive experiments on two widely-used datasets CoNLL04 and ACE05, and demonstrate the effectiveness of the proposed framework.

To summarize, the contributions of this paper are as follows:

- We propose an entity-specific relative position representation to allow the model aware of the distance information of entities, which provides the model with richer semantics and handles the issue of obscure entity features.
- We introduce a global relation classifier to integrate the essential sentence-level semantics with the token-level ones, which can remedy the problem caused by incompleted local information.
- Experiments on the CoNLL04 and ACE05 datasets demonstrate that the proposed framework significantly outperforms the previous work, achieving $+2.40\%$ and $+1.90\%$ improvements in F1-score on the two datasets.

2 Related Work

In this section, we introduce the related studies for this work, entity and relation extraction as well as the positional representation.

2.1 Entity and relation extracion

As a crucial content of information extraction, the entity-relation extraction task has always been widely concerned. Previous studies (Miwa et al., 2009; Chan and Roth, 2011; Lin et al., 2016) mainly focus on pipelined structure, which divides the task into two independent phases, all entities are extracted first by an entity recognizer, and then relations between every entity pairs are predicted by a relation classifier. The pipelined methods suffer from error propagation issue and they ignore the interactions between the two phrases. To ease these problems, many joint models have been proposed to extract the relational triplets (e_1, r, e_2), simultaneously. According to different extraction order, the joint models can be categorized into two paradigms. The first paradigm first identifies all entities in the sentence, then traverses each pair of entities and determines their potential relation. Various models have achieved promising results by exploiting recurrent neural network (Miwa and Bansal, 2016; Luan et al., 2019), graph convolutional network (Sun et al., 2019; Fu et al., 2019) and transformer-based structure (Eberts and Ulges, 2019; Wang et al., 2019). Though effective, these models need to examine every possible entity pairs, which inevitably contains a lot of redundant pairs. In the second paradigm, the head entities are detected first and the corresponding relations and tail entities are extracted later. Bekoulis et al. (Bekoulis et al., 2018) present the multi-head selection framework to automatically extract multiple entities and relations at once. Huang et al. (Huang et al., 2019) improve the MHS framework by using NER pretraining and soft label embedding features. Recently, Li et al. (Li et al., 2019) cast the task as a question answering problem and identify entities based on a machine reading comprehension model. Different from the first one, the second paradigm is able to extract entities and all the relations between at once without enumerating every entity pair each time, which reduces redundant prediction and improves work efficiency.

Our work is inspired by the multi-head selection framework but enjoys new key points as follows. 1) We propose an entity-specific relative position representation to better encode the distance between entities and context tokens. 2) We incorporate the sentence-level information for relation classification to revise the learning of local features. 3) We enhance the original MHS framework with a pre-trained self-attentive encoder. Together these improvements contribute to the extraction performance remarkably.

2.2 Positional Representation

Generally, non-recurrent models do not contain the sequential order information of input tokens. Therefore, in order to fit for the sequential inputs, they need to design representations to encode positional information explicitly.

The approaches for positional representations can fall into three catagroies. The first one designs the position encodings as a deterministic function of position or learned parameters (Sukhbaatar et al., 2015; Gehring et al., 2017). These encodings are combined with input elements to expose position information to the model. For example, the convolutional neural networks inherently capture the relative positions within each convolutional kernels. The second catagroy is the absolute position representation. The Transformer structure (Vaswani et al., 2017) contains neither recurrence nor convolution, in order to inject the positional information to the model, it defines the sine and consine functions of different frequencies to encode absolute positions. However, such absolute positions cannot model the interaction information between any two input tokens explicitly. Therefore, the third catagroy extends the self-attention mechanism to consider the relative positions or distances between sequential elements. Such as the model by (Shaw et al., 2018) and Transformer-XL (Dai et al., 2019). Different from the relative positions metioned above, we propose the relative positions exspecially for entities. As such information is not necessary for non-entity tokens, and may introduce noise on the contrary.

3 Method

In this section, we briefly present the details of the relative position representation based multi-head selection framework. The concept of multi-head means that any head entity may be relevant to multiple relations and tail entities (Bekoulis et al., 2018).

Formally, denote $\mathcal{E}$ and $\mathcal{R}$ as the set of pre-defined entity types and relation categories, respectively. Given an input sentence with N tokens $\boldsymbol{s} = \{s_1, s_2, \ldots, s_N\}$, the entity-relation extraction task aims at extracting a set of named entities $\boldsymbol{e} = \{e_1, e_2, \ldots, e_M\}$ with specific types $\boldsymbol{y} = \{y_1, y_2, \ldots, y_M\}$, and predict the relation r_{ij} for each entity pair (e_i, e_j), where $y_i \in \mathcal{E}$ and $r_{ij} \in \mathcal{R}$. Triplets such as

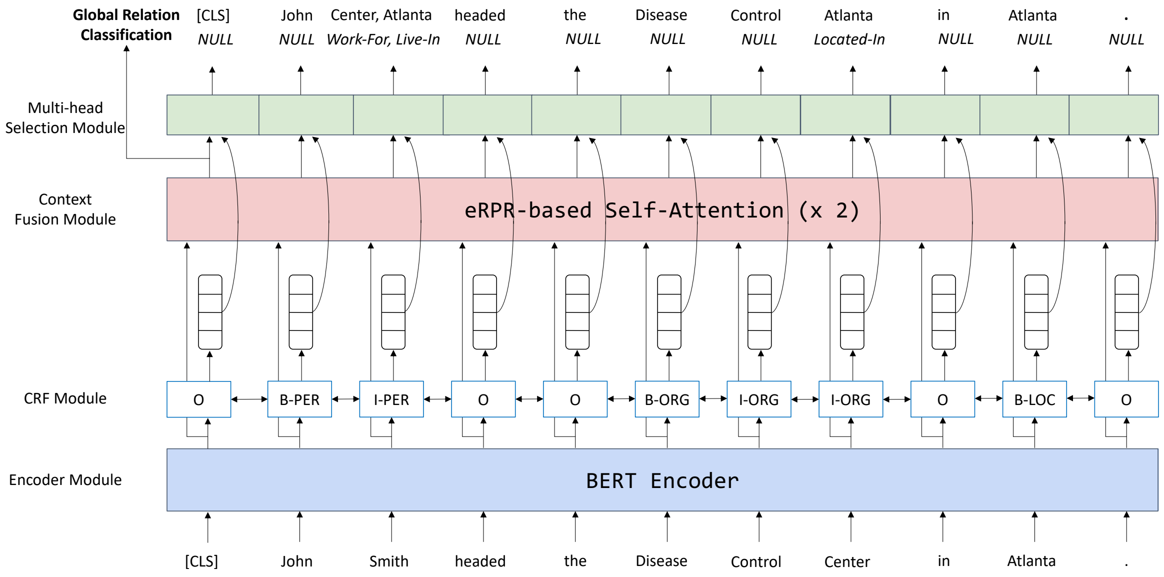


Figure 2: The overview of the relative position representation based multi-head selection framework. We take a sentence from CoNLL04 dataset as an example. In this sentence, the golden relational triplets are: (*John Smith*, `Live-In`, *Atlanta*), (*John Smith*, `Work-For`, *Disease Control Center*) and (*Disease Control Center*, `Located-In`, *Atlanta*). The *NULL* label denotes a case of no relation.

(e_i, r_{ij}, e_j) are formulated as the output, where e_i is the head entity and e_j is the tail entity, e.g., (*John Smith*, `Live-In`, *Atlanta*).

As illustrated in Fig. 2, our framework consists of four modules as follows: the encoder module, the CRF module, the context fusion module and the multi-head selection module. The token sequence is taken as the input of the framework and is fed into the BERT encoder to capture contextual representations. The CRF module is applied afterward to extract potential head entities (i.e., boundaries and types). Then, the hidden states of BERT and the entity information are feed into the context fusion module to encoder the entity position-based features. Finally, a multi-head selection module is employed to simultaneously extract tuples of relation and tail entity for the input token (e.g., (`Work-For`, *Center*) and (`Live-In`, *Atlanta*) for the head entity *Simth*). Additionally, we present the strategy of global relation classification. We will elaborate on each of the modules in the following subsections.

3.1 Encoder Module

The encoder module aims at mapping discrete tokens into distributed semantic representations. Bidirectioal Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a pre-trained language representations built on the bidirectional self-attentive models. It is known as its powerful feature representative ability and recently breaks through the leaderboards of a wide range of natural language processing tasks, such as named entity recognition, word segmentation and question answering. Different from the previous work (Bekoulis et al., 2018) which uses the BiLSTM as the feature encoder, we use the BERT instead to better represent contextual features.

As illustrated in Fig. 2, given a N-token sentence $\boldsymbol{s} = \{s_1, s_2, \ldots, s_N\}$, a special classification token ([CLS]) is introduced as the first token of the input sequence as $\{[\text{CLS}], s_1, s_2, \ldots, s_N\}$. The sequence is encoded by the multi-layer bidirectional attention structure. The output of the BERT layer is the contextual representation of each token as $\boldsymbol{h} = \{h_0, h_1, \ldots, h_N\}$ where $h_i \in \mathbb{R}^{d_h}$, where d_h denotes the dimension of the hidden state of BERT.

3.2 CRF Module

The conditional random field is a probabilistic method that jointly models interactions between entity labels, which is widely used in named Entity recognition task. Similarly, we employ a linear-chain CRF over the BERT layer to obtain the most possible entity label for each token, e.g., B-PER.

Given the BERT outputs $\boldsymbol{h} = \{h_0, h_1, \ldots, h_N\}$, the corresponding entity label sequence is denoted as $\boldsymbol{y} = \{y_0, y_1, \ldots, y_N\}$. Specifically, we uses the BIO (Begin, Inside, Non-Entity) tagging scheme. For example, B-PER denotes the beginning token of a person entity. The probability of using $\boldsymbol{y}$ as the label prediction for the input context is calculated as

$$p(\boldsymbol{y}|\boldsymbol{h}) = \frac{\prod_{i=1}^{N} \phi_i(y_{i-1}, y_i, \boldsymbol{h})}{\sum_{y' \in \mathcal{Y}(\boldsymbol{h})} \prod_{i=1}^{N} \phi_i(y'_{i-1}, y'_i, \boldsymbol{h})}. \tag{1}$$

Here, $\mathcal{Y}(\boldsymbol{h})$ is the set of all possible label predictions. And $\phi_i(y_{i-1}, y_i, \boldsymbol{h}) = \exp(\mathbf{W}_{\text{CRF}}^{y_i} h_i + \mathbf{b}_{\text{CRF}}^{y_{i-1} \to y_i})$, where $\mathbf{W}_{\text{CRF}} \in \mathbb{R}^{d_h \times d_l}$, $\mathbf{b}_{\text{CRF}} \in \mathbb{R}^{d_l \times d_l}$ with d_l denoting the size of the entity label set. $\mathbf{W}_{\text{CRF}}^{y_i}$ is the column corresponding to label y_i, and $\mathbf{b}_{\text{CRF}}^{y_{i-1} \to y_i}$ is the transition probability from label y_{i-1} to y_i.

During training, the NER loss function $\mathcal{L}_{\text{CRF}}$ is defined as the negative log-likelihood:

$$\mathcal{L}_{\text{NER}} = -\sum\nolimits_{\boldsymbol{h}} \log p(\boldsymbol{y}|\boldsymbol{h}). \tag{2}$$

During decoding, the most possible label sequence y^* is the sequence with maximal likelihood of the prediction probability:

$$y^* = \arg\max\nolimits_{\boldsymbol{y} \in \mathcal{Y}(\mathbf{h})} p(\boldsymbol{y}|\boldsymbol{h}). \tag{3}$$

The final labels can be efficiently addressed by the Viterbi algorithm.

3.3 Context Fusion Module

The context fusion module focuses on injecting the entity-specific relative position representation into the semantic feature of entities to capture the distance information between entities and other context tokens. The self-attention structure in BERT introduces sine and cosine functions of varying frequency to represent the absolute position representation (APR) of tokens as:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}), \end{aligned} \tag{4}$$

where d_{model} stands for the hidden dimention of the model. However, such absolute position representation neglects the relative distance information between entities and other tokens, while such distance plays a crucial role in entity-relation prediction. Hence, we introduce an entity-specific relative position representation to efficiently encode the relative distance.

Formally, for the output states of BERT encoder $\boldsymbol{h} = \{h_0, h_1, \ldots, h_N\}$ where $h_i \in \mathbb{R}^{d_h}$, the relative position layer outputs a transformed sequence $\boldsymbol{p} = \{p_0, p_1, \ldots, p_N\}$ where $p_i \in d_p$ with d_p as the hidden dimention of self-attention structure.

Consider two input states h_i and h_j, where h_i denotes an entity and h_j denotes a contextual token, $i, j \in 0, 1, \ldots, N$. In order to inject the relative position information into x_i, we define $a_{ij}^K \in d_p$, $a_{ij}^V \in d_p$ as two different relative distances between h_i and h_j. Suppose that the impacts of tokens beyond a maximum distance on current token are negligible. Therefore, we clip the relative position within a maximum distance δ and only consider the position information of δ tokens on the left and δ tokens on the right. We define $\boldsymbol{\omega^K} = (\omega_{-\delta}^K, \ldots, \omega_{\delta}^K)$ and $\boldsymbol{\omega^V} = (\omega_{-\delta}^V, \ldots, \omega_{\delta}^V)$ as two relative position representations, where $\omega_i^K, \omega_i^V \in \mathbb{R}^{d_p}$ are initialized randomly and will be learned during training. Figure 3 illustrates an example of the relative position representations. Then, a_{ij}^K and a_{ij}^V are assigned as:

$$\begin{aligned} a_{ij}^K &= \omega_{\text{clip}(j-i,\delta)}^K \\ a_{ij}^V &= \omega_{\text{clip}(j-i,\delta)}^V \\ clip(x,\delta) &= \max(-\delta, \min(x,\delta)). \end{aligned} \tag{5}$$

Based on the relative position representations a_{ij}^K, a_{ij}^V, the attention matrix between h_i and h_j is calculated as:

$$\alpha_{ij} = \text{softmax}(\frac{(h_i W^Q)(h_j W^K + a_{ij}^K)^T}{\sqrt{d_p}}), \tag{6}$$

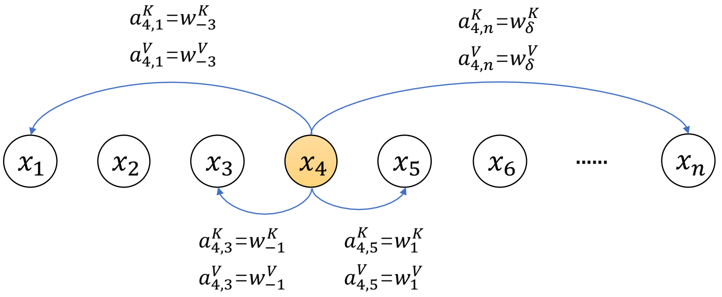


Figure 3: An example to illustrate the entity relative position representation. x_4 is considered as an entity, we show the eRPR between x_4 and the context tokens within the clipped distance δ. Assuming $3 <= \delta <= n - 4$ in this example.

where $W^Q \in \mathbb{R}^{d_h \times d_p}$, $W^K \in \mathbb{R}^{d_h \times d_p}$ are parameter matrices for multi-head projections. The attentional output of h_i is the weighted sum of h_j which also consider the relative position:

$$p_i = \sum_{j=1}^{n} \alpha_{ij}(h_j W^V + a^V_{ij}). \tag{7}$$

Specifically, we only consider the relative position of named entities rather than every tokens in the sentence. So $\boldsymbol{\omega^K}$ and $\boldsymbol{\omega^V}$ are set as 0 for non-entity tokens. the This entity-only RPR approach comes with the following key advantages: 1) it encodes unique features for entities and thus can better differentiate entities from other plain tokens; 2) it provides entity-specific information and helps the relation and tail entity prediction.

3.4 Multi-head Selection Module

The multi-head selection module aims to predict the possible relations and tail entities simultaneously for each head entity (Bekoulis et al., 2018). Given a sequence of entity labels $\boldsymbol{y} = \{y_0, y_1, \ldots, y_N\}$ predicted by the CRF module, we map each label to a distributed label embedding as $\boldsymbol{l} = \{l_0, l_1, \ldots, l_N\}, l_i \in \mathbb{R}^{d_l}$, where d_l is the label embedding size. The mapping dictionary is randomly initialized and be fine-tuned during training. During training, we use the golden entity labels.

As shown in Fig. 2, the input to the multi-head selection layer are the concatentation of label embedding and the outputs of relative position layer as:

$$z_i = [l_i; p_i], i = 0, 1, \ldots, N. \tag{8}$$

For each input state z_i, we compute the score between z_i and z_j given a relation $r_k, r_k \in \mathcal{R}$ as:

$$g(z_i, z_j, r_k) = V^r f(U^r z_j + W^r z_i + b^r), \tag{9}$$

where $V^r \in \mathbb{R}^{d_r}$, $U^r, W^r \in \mathbb{R}^{d_r \times (d_h + d_l)}$, $b^r \in \mathbb{R}^{d_r}$, $f(\cdot)$ is the element-wise RELU function. The most probable tail entity s_j with the relation r_k corresponding to the head entity s_i is predicted as:

$$\Pr(\text{tail} = s_j, \text{relation} = r_k | \text{head} = s_i) = \sigma(g(z_i, z_j, r_k)), \tag{10}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

During training, we optimize the cross-entropy loss $\mathcal{L}_{\text{MHS}}$ for the candidate tail entity s_{ij} and relation r_{ij} given the head entity s_i as:

$$\mathcal{L}_{\text{MHS}} = \sum_{i=0}^{N} \sum_{j=0}^{M} -\log \Pr(tail = s_j, relation = r_j | head = s_i), \tag{11}$$

where M is the number of golden relations for s_i. During testing, we select the tuple of the relation and tail entity $(\hat{r_k}, \hat{s_j})$ with a score exceeding the confidence threshold η. In this way, multiple tail entities and relations for the head entity s_i can be predicted simultaneously.

3.5 Global Relation Classification

Generally, detecting the relation between entites need to consider the theme of the sentence. The previous work only use the local context information for relation and entity prediction, which may lead to the deviation of global semantics. We introduce the global relation classification strategy to guide the training of local semantic features. As illustrated in Fig. 2, the first output of the relative position layer corresponding to the hidden state of [CLS] token p_0, which can be considered as the aggregate representation of the sentence. Therefore, we use the [CLS] token to predict the relations relevant to the whole sentence $\boldsymbol{s}$ as:

$$\Pr(\text{relation} = \boldsymbol{r}|\boldsymbol{s}) = \sigma(W^g p_0 + b^g), \tag{12}$$

where $\boldsymbol{r} \subseteq \mathcal{R}$, $W^g \in \mathbb{R}^{d_h \times |\mathcal{R}|}$, $b^r \in \mathbb{R}^{|R|}$, $\sigma(\cdot)$ is the sigmoid function.

During training, we minimize the binary cross-entropy loss for the global classification as:

$$\mathcal{L}_{\text{GRC}} = \sum_{i=0}^{T} \Pr(\text{relation} = \boldsymbol{r}|\boldsymbol{s}), \tag{13}$$

where T denotes the number of golden relations in the sentence.

3.6 Joint Training

To train the model jointly, we optimize the final combined objective function during training:

$$\mathcal{L} = \mathcal{L}_{\text{NER}} + \lambda\mathcal{L}_{\text{GRC}} + \mathcal{L}_{\text{MHS}}, \tag{14}$$

where $\mathcal{L}_{\text{NER}}$, $\mathcal{L}_{\text{GRC}}$, and $\mathcal{L}_{\text{MHS}}$ denote the loss function for head entity recognition, global relation classification and multi-head selection, respectively (Eq. 2, 13, 11), $\lambda \in [0, 1]$ is the weight controling the trade-off of the global relation classification. $\mathcal{L}$ is averaged over samples for each batch.

4 Experiment

In this section, we conduct extensive experiments to verify the effectiveness of our framework, and make detailed analyses to show its advantages.

4.1 Dataset

We evaluate the proposed method on two widely-used benchmarks for entity and relation extaction: CoNLL04 and ACE05.

- **CoNLL04** (Roth and Yih, 2004) defines 4 entity types including Location (`LOC`), Organization (`ORG`), Person (`PER`) and `Other` and 5 relation categories as `Located-In`, `OrgBased-In`, `Live-In`, `Kill` and `Work-For`. It consists of news articles from the Wall Street Journal and Associated Press. We use the data split by Gupta et al. (Gupta et al., 2016) (910 instances for training, 243 for validation and 288 for testing).

- **ACE05** (Doddington et al., 2004) provides 7 entity types: Location (`LOC`), Organization (`ORG`), Person (`PER`), Geopolitical Entity (`GPE`), Vehicle (`VEH`), Facility (`FAC`), Weapon (`WEA`) and 6 relation types: Organization affiliation (`ORG-AFF`), Person-Social (`PER-SOC`), Agent-Artifact (`ART`), `PART-WHOLE`, GPE affiliation (`GEN-AFF`), Physical (`PHYS`). It contains documents from different domains as newswire and online forums. We adopt the same data splits as the previous work (Miwa and Bansal, 2016) (351 documents for training, 80 for validation and 80 for testing).

4.2 Implemental Details

Following previous works, we use the standard precision (P), recall (R), and micro-F1 score (F1) as the evaluation metrics. A relation is correct if the arguments of triplet (e_1, r, e_2) are correct. Other experimental settings are as follows. We initialize the BERT encoder layer using the pre-trained BERT-Base-Cased checkpoint [1] which has 12 layers, a hidden size of 768. We use Adam optimizer with an initial learning rate of 5×10^{-5}. During training, we do warm-up startup first and employ a linearly decrease with 0.05 as the decay rate. For the model structure, we adopt 2-layer eRPR-based self-attention after the BERT encoder layer. The self-attention layer has an identical structure as the layer in BERT. The relative position representations $\boldsymbol{\omega}^{\boldsymbol{K}}, \boldsymbol{\omega}^{\boldsymbol{V}}$ are initiaized randomly with a uniform distribution. The maximum relative distance is set as $\delta = 4$. The GRC loss weight is set as $\lambda = 1$. The size of entity label embedding is set as $d_l = 50$. The threshold for multi-head selection $\eta = 0.5$.

Specifically, we use the both the *relaxed* and *strict* evaluation settings for comparison. In the *relaxed* setting, assuming the entity boundaries are given, a multi-token entity is correct if at least one of its comprising token types is correct; a relation is correct if the two argument entities are correct and the relation type is correct. In the *strict* setting, we consider an entity is correct if the entity type and the boundaries are both correct; a relation is correct if the relation type and the argument entities are both correct.

4.3 Results and Analyses

Table 1: Performance comparision with baseline models on CoNLL04 and ACE05. eRPR denotes models adopt the self-attention with entity-specific relative position representation at the context fusion module. The ✓and ✗marks stand for whether or not the model builds on hand-crafted features or NLP tools. eRPR MHS is the proposed full model.

Model	Pre-calculated Features	Evaluation	Entity P	Entity R	Entity F1	Relation P	Relation R	Relation F1
		CoNLL04						
Gupta et al. (2016)	✓	*relaxed*	92.50	92.10	92.40	78.50	63.00	69.90
Gupta et al. (2016)	✗	*relaxed*	88.50	88.90	88.80	64.60	53.10	58.30
Adel and Schütze (2017)	✗	*relaxed*	-	-	82.10	-	-	62.50
Bekoulis et al. (2018)	✗	*relaxed*	93.41	93.15	93.26	72.99	63.37	67.01
eRPR MHS	✗	*relaxed*	94.32	93.81	**94.06**	73.85	64.41	**68.81**
Miwa and Sasaki (2014)	✓	*strict*	81.20	80.20	80.70	76.00	50.90	61.00
Bekoulis et al. (2018)	✗	*strict*	83.75	84.06	83.90	63.75	60.43	62.04
eRPR MHS	✗	*strict*	86.85	85.62	**86.23**	64.20	64.69	**64.44**
		ACE05						
Miwa and Bansal (2016)	✓	*strict*	80.80	82.90	81.80	48.70	48.10	48.40
Katiyar and Cardie (2017)	✗	*strict*	81.20	78.10	79.60	46.40	45.53	45.70
eRPR MHS	✗	*strict*	86.26	84.66	**85.45**	60.60	60.84	**60.72**

Comparison Baseline As shown in Table 1, we list the following baselines for comparison. Gupta et al. (2016) propose a table-filling based method that relies on hand-crafted features and external NLP tools. Adel and Schütze (2017) use a global normalized convolutional neural networks to extract entities and relations. Miwa and Bansal (2017) adopt a BiLSTM to extract entities and a Tree-LSTM to model the dependency relations between entities. Bekoulis et al. (2018) propose the multi-head selection structure, which adopts BiLSTM as the feature encoder and uses CRF for entity recognition and can extract the relational triplet simultaneously. The results on CoNLL04 and ACE05 are directly copied from the published paper.

Main Results Table 1 presents the performance comparisions on CoNLL04 and ACE05 datasets. eRPR MHS is the proposed full model, which uses the BERT at encoder module, and follows by two

[1] BERT checkpoints are available at https://github.com/google-research/bert

Table 2: Ablation study on CoNLL04 and ACE05. APR denotes models adopt the general self-attention with absolute position representation at the context fusion module. eRPR denotes models adopt the self-attention with entity-specific relative position representation at the context fusion module. The ✓mark refers to the model include the global relation classification. We use the *strict* evaluation setting here.

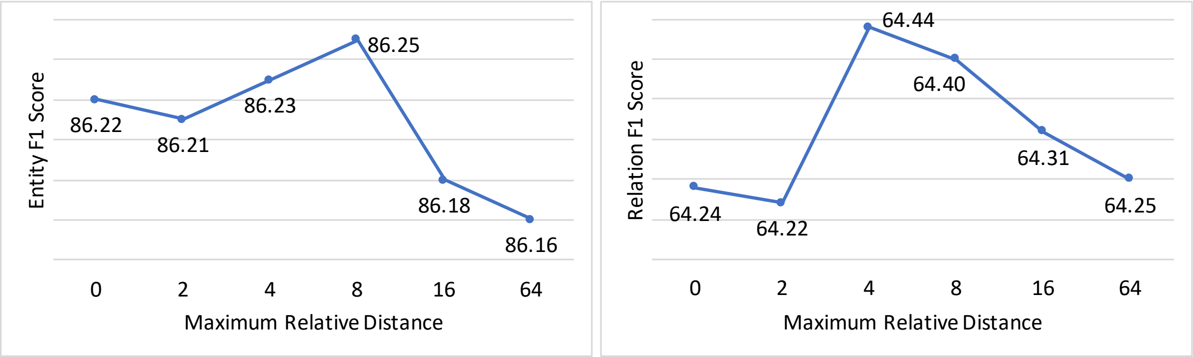
Model	Encoder	Context Fusion	GRC	Entity			Relation		
				P	R	F1	P	R	F1
CoNLL04									
1	BiLSTM	-	-	83.75	84.06	83.90	63.75	60.43	62.04
2	BERT	-	-	85.75	86.28	86.00	65.15	62.56	63.83
3	BERT	APR Layer ×2	-	86.32	85.68	86.00	64.53	63.40	63.96
4	BERT	eRPR Layer ×2	-	86.75	85.56	86.15	63.93	64.50	64.21
5	BERT	APR Layer ×2	✓	86.78	85.66	86.22	64.18	64.30	64.24
6	BERT	eRPR Layer ×2	✓	86.85	85.62	**86.23**	64.20	64.69	**64.44**
ACE05									
1	BiLSTM	-	-	84.88	84.10	84.49	57.40	60.32	58.82
2	BERT	-	-	85.70	84.25	84.96	59.92	60.06	59.99
3	BERT	APR Layer ×2	-	86.18	84.55	85.36	60.23	60.82	60.52
4	BERT	eRPR Layer ×2	-	86.24	84.60	85.41	60.57	60.76	60.66
5	BERT	APR Layer ×2	✓	86.22	84.57	85.39	60.46	60.76	60.61
6	BERT	eRPR Layer ×2	✓	86.26	84.66	**85.45**	60.60	60.84	**60.72**

eRPR self-attention layers and adopts the GRC strategy. As we can see, our eRPR MHS overwhelmingly outperforms all the baseline models in terms of all three evaluation metrics on the two datasets. by a large margin for both entity and relation extraction. Especially, comparing with the model by Bekoulis et al. (2018), our model achieves significant boosts by 2.40% and 1.90% for relation F1 on CoNLL04 and ACE05, respectively. These results show that, with our enhanced components, i.e., the eRPR layers, the global relation classification and the BERT encoder, the model performance can be significantly improved. Such improvements highlight the effectiveness of our proposed framework.

Ablation Study As shown in Table 2, we list variant models (Model 1-5) to each component in our framework. Model 1 stands for the original MHS framework proposed by Bekoulis et al. (2018). By comparison, we come to the following conclusions. 1) Replacing the BiLSTM with pre-trained BERT can improve the performance obviously (Model 2 v.s. Model 1). 2) Adding the context fusion module after the encoder module can enhance the semantic representation, leading to higher results (Model 3 v.s. Model 2). 3) Comparing Model 4 with the above variations, incorporating eRPR into the self-attention structure can significantly increase the precision of models and thus contribute to better overall F1 scores. For example, it increases the relation F1 from 63.96% to 64.21% on CoNLL04. We attribute it to that the eRPR injects distance information into entity features, which can provide useful information to the multi-head selection. 4) Comparing Model 5 and Model 4, the GRC strategy can further improve model performance. Therefore, global information is instructive for learning local features. Finally, combining all these components, we achieve significant improvements over the original MHS.

4.4 Effect of the Maximum Relative Distance

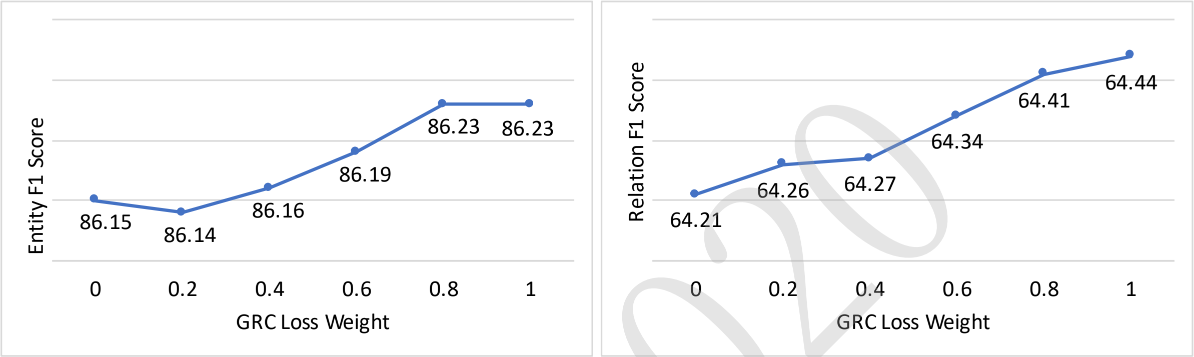
In this subsection, we evaluate the effect of varying the maximum relative distance δ. Following previous studies (Shaw et al., 2018), we conduct experiments on CoNLL04 with different maximum relative distance δ, increases exponentially from 0 to 64. Fig. 4 shows the experimental results. We observe that when $\delta = 8$, the entity F1 has the best result, and when $\delta = 4$, the relation F1 has the best result. Meanwhile, the larger value of δ (i.e., $\delta = 64$) is meaningless for both entity and relation extraction, which verifies that the impacts of tokens beyond a maximum distance can be negligible. Therefore, to ensure a better performance for relation extraction, we set $\delta = 4$ for all the experiements.



(a) Entity F1 Score

(b) Relation F1 Score

Figure 4: Experimental results for varying the maximum relative distance δ.



(a) Entity F1 Score

(b) Relation F1 Score

Figure 5: Experimental results for varying the GRC loss weight λ.

4.5 Effect of the GRC Loss Weight

In this subsection, we evaluate the effect of different GRC loss weight λ to the model performance. We keep the maximum relative distance δ as 4 and conduct the experiments on the CoNLL04 dataset with λ from 0 to 1 at the interval of 0.2. As shown in Fig. 5, the setting with $\lambda = 0$ denotes the GRC is not used in the framework and its performance is much lower than settings with larger λ In addition, with the growth of λ, both entity and relation F1 scores are increased continuously. As such, we keep $\lambda = 1$ for all the above experiments. These comparison results further demonstrate the effectiveness of GRC. Therefore, the sentence-level information can be utilized fruitfully for multi-head selection and helps improve the overall performance.

5 Conclusion

In this paper, we propose a relative position representation based multi-head selection framework for joint entity and relation extraction. Different with the existing multi-head selection method, we introduce the relative position representation to capture the distance information of entities. We then propose a global relation classification to guide the learning of local features. Additionaly, BERT is incorperated in the framework for sematic representation. Experimental results on CoNLL04 and ACE05 datasets show that our framework siginificantly outperforms all the baseline models for both entity and relation extraction.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081,61370126),the Beijing Advanced Innovation Center for Imaging Technology(Grant No.BAICIT-2016001), and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE-2019ZX-17).

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *EMNLP*, pages 1723–1729.

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.

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

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. *arXiv preprint arXiv:1909.07755*.

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

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252.

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

Weipeng Huang, Xingyi Cheng, Taifeng Wang, and Wei Chu. 2019. Bert-based multi-head selection for joint entity-relation extraction. In *NLPCC*, pages 713–723. Springer.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *ACL*, pages 917–928.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, pages 2124–2133.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *NAACL*, pages 3036–3046.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *EMNLP*, pages 1858–1869.

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

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. Technical report, ILLINOIS UNIV AT URBANA-CHAMPAIGN DEPT OF COMPUTER SCIENCE.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL*, pages 464–468.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. *arXiv preprint arXiv:1902.01030*.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2020. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *IJCAI*.