Position-aware Joint Entity and Relation Extraction with Attention Mechanism

Chenglong Zhang[1] , **Shuyong Gao**[1] , **Haofen Wang**[3*] and **Wenqiang Zhang**[1,2*]
[1]Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China
[2]Academy for Engineering & Technology, Fudan University, Shanghai, China
[3]College of Design and Innovation, Tongji University, Shanghai, China
{clzhang20, sygao18, wqzhang}@fudan.edu.cn, carter.whfcarter@gmail.com

Abstract

Named entity recognition and relation extraction are two important core subtasks of information extraction, which aim to identify named entities and extract relations between them. In recent years, span representation methods have received a lot of attention and are widely used to extract entities and corresponding relations from plain texts. Most recent works focus on how to obtain better span representations from pre-trained encoders, but ignore the negative impact of a large number of span candidates on slowing down the model performance. In our work, we propose a joint entity and relation extraction model with an attention mechanism and position-attentive markers. The attention score of each candidate span is calculated, and most of the candidate spans with low attention scores are pruned before being fed into the span classifier, thus achieving the goal of removing the most irrelevant spans. At the same time, in order to explore whether the position information can improve the performance of the model, we add position-attentive markers to the model. The experimental results show that our model is effective. With the same pre-trained encoder, our model achieves the new state-of-the-art on standard benchmarks (ACE05, CoNLL04 and SciERC), obtaining a 4.7%-17.8% absolute improvement in relation F1.

1 Introduction

Named entity recognition and relation extraction are two core subtasks of information extraction. Early research methods can be divided into two main categories: pipeline methods and joint methods. The pipeline methods usually need to train two models, one is used to identify named entities, and the other is used to extract the relation between entitiess [Zhong and Chen, 2020; Ye *et al.*, 2021]. The joint approach is to model the two tasks of named entity recognition and relation extraction jointly [Luan *et al.*, 2018; Eberts and Ulges, 2019; Ji *et al.*, 2020; Lin *et al.*, 2020; Wang and Lu, 2020], either by projecting them into a structured prediction framework or by performing multi-task learning with a shared representation.

In this work, we propose a joint entity and relation extraction model with an attention mechanism and position-attentive markers, and our encoder for the joint model is built on top of a deeply pre-trained language model [Devlin *et al.*, 2018; Beltagy *et al.*, 2019; Lan *et al.*, 2019]. Unlike previous work based on BIO/BILOU labels [Bekoulis *et al.*, 2018b; Li *et al.*, 2019; Nguyen and Verspoor, 2019], the potential entities of our model are represented by span, which can identify overlapping entities. Although span-based representation models can identify overlapping entities, they also bring a large number of negative samples, and such a large number of negative samples of candidate entities will bring noise to the model and hurt the performance of the model. To address this issue, we consider to filter out some negative samples of candidate entities by designing a specific attention mechanism. On the other hand, the position information of the words in the input sentences helps to improve the performance of the model. We finally desire to know whether combining the above two into a unified model can further improve the overall performance. Our model is called PERA (**P**osition-aware Joint **E**ntity and **R**elation Extraction with **A**ttention Mechanism).

Using the same pre-trained encoders, our model outperforms all previous joint models on three standard benchmarks: ACE05, SciERC and CoNLL04, advancing the previous state-of-the-art 4.7%-17.8% in relation F1. The experimental results show that, (1) introducing the attention mechanism to reduce the number of negative samples of candidate span can improve the performance of the model in named entity recognition; (2) adding position-attentive markers to the model can help improve the overall performance of the model in named entity recognition and relation extraction; (3) attention mechanism and position-attentive markers can complement each other, and combining them can further improve the model's performance.

In summary, we summarize our contributions as follows:

- We propose a new method for joint entity and relation extraction that fuses attention mechanisms and position-attentive markers, where entity recognition and relation extraction share the same encoder. Our model PERA is built on three standard benchmarks and outperforms all previous joint models.

*Corresponding author

- We conduct a series of experiments to understand why our method performs so effectively. We find that a large number of negative samples of candidate entities hurts the performance of the model, and removing some of the negative samples before feeding candidate spans to the span classifier can improve the performance of the model.
- We conduct a careful analysis to examine how different factors affect the final performance of our model. The experimental results show that attention mechanism and position-attentive markers can complement each other, and combining them together can further improve the performance of the model.

2 Related Work

In the past, named entity recognition and relation extraction in the text were generally regarded as two independent tasks. However, in recent years, joint entity and relation extraction have attracted attention and there has been a surge of research interest in joint entity and relation extraction models. At the same time, the span representation method has received extensive attention from the academic community and is widely used to extract entities and relations from plain texts.

Span-level representation Since the BIO/BILOU-based model can only give each token a tag, and a token cannot be part of multiple entities at the same time, this does not cover the case of overlapping entities. Our method is based on a span representation approach, which performs an exhaustive search for all spans and effectively solves the problem of entity overlap. This uses span representation in both coreference resolution [Lee *et al.*, 2017; Lee *et al.*, 2018] and semantic role labeling [Ouchi *et al.*, 2018].

Recently, some joint entity and relation extraction models [Luan *et al.*, 2018; Dixit and Al-Onaizan, 2019] based on span representation have been proposed, using the span representation of BiLSTM. [Dixit and Al-Onaizan, 2019] study a relation extraction model based on span-level representation. [Luan *et al.*, 2018] perform a beam search on the hypothesis space to estimate which spans are involved in entity classes, relations and coreferences. DyGIE++ [Wadden *et al.*, 2019] has replaced the BiLSTM encoder with BERT and is a Transformer-based span method for joint entity and relation extraction.

Joint model In recent years, joint entity and relation extraction models have been intensively studied and explored. [Li and Ji, 2014] propose an incremental joint framework to extract mentioned entities and relations simultaneously using structure perceptrons and effective cluster search. [Miwa and Sasaki, 2014] adopt a history-based structured learning approach to jointly extract entities and relations in sentences. [Zheng *et al.*, 2017] use an annotation model containing information about entities and relations to transform the joint extraction into an annotation problem. [Zhang *et al.*, 2017] build a globally optimized neural network model to extract end-to-end relations and propose LSTM features to learn contextual representations. [Fu *et al.*, 2019] present an end-to-end relation extraction model GraphRel, which uses GCN to jointly learn named entities and relations. [Li *et al.*, 2019] propose an approach of multi-turn question answering to solve entity and relation extraction. [Wang and Lu, 2020] treat the joint extraction task as a form-filling problem. These methods have a common feature, and it is necessary to find a good global optimization in the two tasks of named entity recognition and relation extraction.

[Miwa and Bansal, 2016] present a model for entity detection and relation extraction separately, while the two parts form an overall single model by stacking and sharing their respective parameters so that named entity recognition and relation extraction interact with each other. [Bekoulis *et al.*, 2018a] model the relation extraction task as a multi-label head selection problem. [He *et al.*, 2018] adopt an end-to-end method to jointly predict all predicates, parameter ranges and their relation. The model makes independent decisions about the relation between each possible word span pair, and learns the span representation of the context, providing rich shared input features for each decision. [Lin *et al.*, 2020] proposes an information extraction framework called ONEIE, which adds a global feature to make joint decisions between instances and subtasks. These methods are similar to pipeline methods. There is a sequential order between the two tasks, with named entity recognition performed first and then relation extraction performed based on the predicted entities.

The closest to our work is coreference resolution [Lee *et al.*, 2017] and DYGIE [Luan *et al.*, 2019]. In their work, the key idea of DYGIE [Luan *et al.*, 2019] is to use a dynamic graph propagation layer to achieve a shared span representation between two tasks. [Lee *et al.*, 2017] use the attention mechanism combined with pruning to train to obtain the most likely reference span in the common reference cluster. We will show our method in detail in Section 3.

3 Method

In this section, we first formally define the named entity recognition and relation extraction problem in Section 3.1, and then describe our approach in detail in Sections 3.2 and Section 3.3. Finally, we present the details of the training and inference in Section 3.4.

3.1 Problem Definition

The input of the problem is a sentence X consisting of n tokens, $X = \{x_1, x_2, ..., x_n\}$. Let $S = \{s_1, s_2, .., s_m\}$ be all the possible spans in X of up to length L, where s_i consists of tokens $(x_{a(i)}, ..., x_{b(i)})$ and $a(i)$ and $b(i)$ denote start and end indices of s_i. The problem can be decomposed into two sub-tasks: named entity recognition and relation extraction.

Named entity recognition Let $\mathcal{E}$ to be a pre-defined set of entity categories. The named entity recognition task is, for each span $s_i \in S$, to predict an entity type $y_e(s_i) \in \mathcal{E} \cup \{none\}$. $none$ represents spans that do not constitute entities. The output of the task is $Y_e = \{(s_i, e) : s_i \in S, e \in \mathcal{E}\}$.

Relation extraction Let $\mathcal{R}$ denote a set of predefined relation classes. The task is, for every pair of spans $s_i \in S$, $s_j \in S$ to predict a relation type $y_r(s_i, s_j) \in \mathcal{R} \cup \{none\}$. $\{none\}$ represents there is no relation between s_i and s_j. The output of the task is $Y_r = \{(s_i, s_j, r) : s_i, s_j \in S, r \in \mathcal{R}\}$.

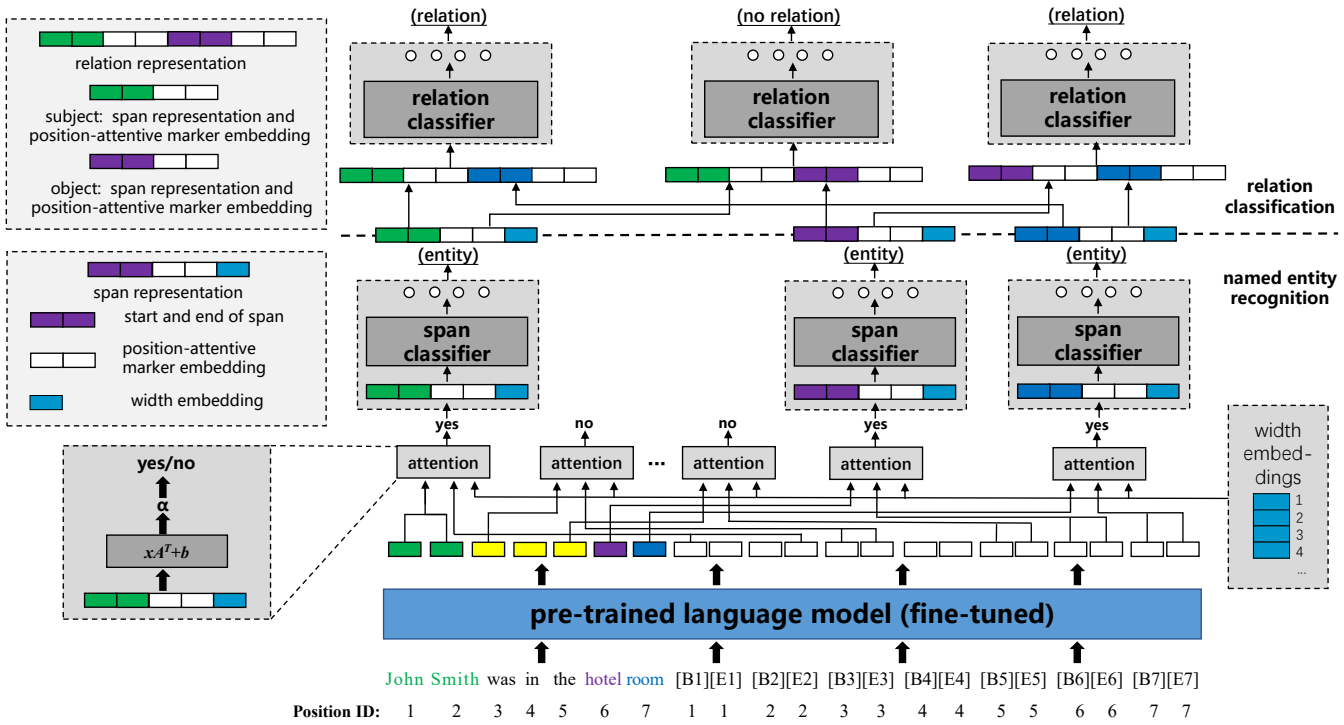

Figure 1: An overview of our models. Given an original input sentence *John Smith was in the hotel room*, it is tokenized and input to the pre-trained language model to obtain its contextual representation embedding. Note that in order to clearly show our approach, Figure 1 omits the step of tokenizing the original input sentences. We append the position-attentive markers to the end of the sentence. In our model, each original word in the original input sentence has a pair of position-attentive markers. For example, the word *John* has a pair of position-attentive markers [B1]/[E1], and [B1]/[E1] shares the position ID of the original word *John*. (1) Before feeding the span representation into the span classifier, we first perform linear calculations on it to get its unary score, and then according to the score, some negative samples are removed first. (2) After pruning, the retained candidate span representations are fed into the span classifier to predict their entity types. (3) We use the span representation of the named entity recognition phase to predict the relation between entities, but not the full span representation of the named entity recognition phase. We first remove the width embedding of two span representations, then concatenate them as our relation representation, and finally feed them into the relation classifier to predict their relation types.

3.2 Named Entity Recognition

As shown in Figure 1, our method consists of named entity recognition and relation extraction. First, a raw sentence is input, tokenized to obtain the tokens corresponding to the sentence, and then these tokens are fed into the pre-trained language model to obtain a contextual representation of each token. Note that in order to clearly show our approach, Figure 1 does not show tokenization of the original input sentences. As in previous work [Lee *et al.*, 2017; Luan *et al.*, 2019; Wadden *et al.*, 2019], our approach is a standard span-based model. For each input token x_t, we first use a pre-trained language model (e.g., BERT) to obtain contextualized representations h_t. Given a span $s_i = (x_{a(i)}, ..., x_{b(i)}) \in S$, the span representation $E_e(s_i)$ contains $H_e(s_i)$ and $M_e(s_i)$.

$H_e(s_i)$ is defined as:

$$H_e(s_i) = [h_{a(i)}; h_{b(i)}; w(s_i)] \quad (1)$$

where $h_{a(i)}$ and $h_{b(i)}$ respectively denotes the contextual representation of $x_{a(i)}$ and $x_{b(i)}$, $w(s_i) \in \mathcal{R}^{d_F}$ denotes the learned embeddings of span width features.

$M_e(s_i)$ is defined as:

$$M_e(s_i) = [m_{B(i)}; m_{E(i)}] \quad (2)$$

where $B(i)$ indicates the position ID of the word where span s_i starts, $E(i)$ indicates the position ID of the word where span s_i ends. $m_{B(i)}$ and $m_{E(i)}$ respectively denotes the contextual representation of the position-attentive marker $[B_{B(i)}]$ and $[E_{E(i)}]$.

Therefore, $E_e(s_i)$ can be defined as:

$$E_e(s_i) = [H_e(s_i); M_e(s_i)] = [h_{a(i)}; h_{b(i)}; m_{B(i)}; m_{E(i)}; w(s_i)] \quad (3)$$

where $[A; B; C; D; E]$ denotes the concatenation operation on the vector A, B, C, D, and E.

Inspired by [Lee *et al.*, 2017], we incorporate an attention mechanism in the entity naming entity recognition phase. For all the possible spans $S = \{s_1, s_2, .., s_m\}$, we first compute its attention score $A = \{\alpha_1, \alpha_2, .., \alpha_m\}$. $S = \{s_1, s_2, .., s_m\}$ is sorted according to their scores $A = \{\alpha_1, \alpha_2, .., \alpha_m\}$, and only those span s_i with high scores α_i have the chance to be retained and fed into the span classifier. Before sorting, we set the attention score α_i of the positive samples $s_i \in S_{gold} \subset S$ to the maximum value. For example, if $s_i \in S_{gold} \subset S$, we let the attention score $\alpha_i = MAX$, where $MAX = max(\alpha_1, \alpha_2, .., \alpha_m)$. At the same time, we set the attention score α_j to the minimum for those padded

negative samples s_j due to batch computation. For example, if s_j is a padded negative sample due to batch computation, we let the attention score $\alpha_j = MIN$, where $MIN = min(\alpha_1, \alpha_2, .., \alpha_m)$. Finally, all possible spans $S = \{s_1, s_2, .., s_m\}$ are ranked from highest to lowest according to the attention score $A = \{\alpha_1, \alpha_2, .., \alpha_m\}$.

In our work, the number of reservations $n = \lambda m$, where λ is the *retention factor*. By controlling the size of the *retention factor* λ, not only the number of candidate spans input to the span classifier can be controlled, but also a part of the padded negative samples can be removed.

When some of the negative samples are removed from $S = \{s_1, s_2, .., s_m\}$, we feed span representation $E^{'} = \{E_e(s_1^{'}), E_e(s_2^{'}), ..., E_e(s_n^{'})\}$ of the retained candidate spans $S^{'} = \{s_1^{'}, s_2^{'}, .., s_n^{'}\}$ to a feedforward network to predict the probability distribution of the entity type $e \in \mathcal{E} \cup \{none\}$: $P_e(e \mid s_i^{'})$.

3.3 Relation Classification

A simple way to predict the relation between s_i and s_i is use the span representation $E_e(s_i)$ and $E_e(s_j)$ of the named entity recognition phase [Luan *et al.*, 2019; Wadden *et al.*, 2019]. However, there are some differences with [Luan *et al.*, 2019; Wadden *et al.*, 2019], our span representation $E_e(s_i)$ has the position-attentive information $M_e(s_i) = [m_{B(i)}; m_{E(i)}]$, as shown in the white rectangle wireframe in Figure 1. Here, we do not include the width feature embedding of span, because the width feature embedding of span may capture the semantic information of entities, but may fail to capture the semantic information of the relation between two entities. As shown in Figure 1, our span-pair representation $R_r(s_i, s_j)$ can be defined as:

$$R_r(s_i, s_j) = [h_{a(i)}; h_{b(i)}; m_{B(i)}; m_{E(i)}; h_{a(j)}; h_{b(j)}; m_{B(j)}; m_{E(j)}] \quad (4)$$

where $[A; B; C; D; E; F; G; H]$ denotes the concatenation operation on the vector A, B, C, D, E, F, G, and H.

Finally, the representation $R_r(s_i, s_j)$ will be fed into a feedforward network to predict the probability distribution of the relation type $r \in \mathcal{R} \cup \{none\} : P_r(r \mid s_i, s_j)$.

3.4 Training and Inference

Our training is supervised. Given sentences with entities (including their entity types) and relations (including entity pairs and types of relation), we define a joint loss function for named entity recognition and relation classification:

$$\mathcal{L} = \mathcal{L}^e + \mathcal{L}^r \quad (5)$$

where $\mathcal{L}^e$ is the cross-entropy loss of named entity recognition, $\mathcal{L}^r$ is the cross-entropy loss of relation classification. These two losses are averaged on each batch of samples.

In relation extraction tasks, the number of negative samples of relations is often much larger than the number of positive samples. Therefore, in our model, we reduce the number of relation negative examples before feeding candidate entity pairs into the feedforward neural network to predict the relation type.

Here, we call the entities in the entity pair whose relation type belongs to $\mathcal{R}$ as relational entities, and the set of relational entities is denoted as $S_{rel} = \{s_{rel1}, s_{rel2}, ..., s_{reln}\}$. For example, if $(s_i, s_j) \in \mathcal{R}$, then $s_i, s_j \in S_{rel}$, and $s_i, s_j \in S_{gold}$. Obviously, $S_{rel} \subset S_{gold} \subset S$. For example, suppose there are 7 gold entities $S_{gold} = \{s_{gold1}, s_{gold2}, ..., s_{gold7}\}$ in a sentence, of which only 3 entities $\{s_{gold1}, s_{gold2}, s_{gold3}\}$ in the entity pair whose relation type belongs to $\mathcal{R}$, such as $(s_{gold1}, s_{gold2}) \in \mathcal{R}$ and $(s_{gold2}, s_{gold3}) \in \mathcal{R}$. Then we have $S_{rel} = \{s_{gold1}, s_{gold2}, s_{gold3}\}$. In the training relation classification phase, we only consider the relational entities $S_{rel} = \{s_{rel1}, s_{rel2}, ..., s_{reln}\}$ and use the relational entities labels in the training set. During inference, we first predict the entities type by performing $y_e(s_i) = argmax_{e \in \mathcal{E} \cup \{none\}} P_e(e \mid s_i)$. When all the prediction results $S_{pred} = \{s_i : y_e(s_i) \neq none\}$ are obtained, we enumerate all the spans $s_i, s_j \in S_{pred}$ and use s_i, s_j to construct the input $R_r(s_i, s_j)$ for the relation classification $P_r(r \mid s_i, s_j)$.

4 Experiments

4.1 Setup

The following content shows the setup details of the experiments.

Datasets We use three popular relation extraction datasets: ACE05, CoNLL04 and SciERC. Table 2 shows the statistical information of each dataset. The ACE05 dataset consists of English, Arabic, and Chinese data collected from various domains, such as newswire and online forums. The ACE05 dataset can be used for entity, relation, and event extraction. We use the English part of the ACE05 dataset data. The CoNLL04 dataset contains named entities with annotations and sentences of relations extracted from news articles. SciERC is derived from abstracts of 500 artificial intelligence papers. We follow previous work and use the same preprocessing procedure and splits for the ACE05 and SciERC datasets. For the CoNLL04 dataset, we adopt the training set (1,153 sentences) and test set (288 sentences) split by [Gupta *et al.*, 2016]. To tune the hyperparameters, 20% of the training set is used as the development set.

Evaluation metrics We use the micro F1 measure as the evaluation metric and follow the standard evaluation protocol. (a) For named entity recognition, if a predicted entity span boundaries and its entity type are both correct, we considered this predicted entity as a correct prediction. (b) For relation extraction, considering different situations, we use two evaluation metrics: 1) boundary evaluation (Rel): if the boundary of the two spans is correct and the predicted relation type is correct, then the predicted relation is considered to be a correct prediction. 2) strict evaluation (Rel+): in addition to the requirements in boundary evaluation, the entity's prediction type must also be correct.

Implementation details For fair comparison with previous work, we adopt *bert-base-uncased* [Devlin *et al.*, 2018] and *albert-xxlarge-v2* [Lan *et al.*, 2019] as the base encoders for ACE05 and CoNLL04. Because the pre-trained model *scibert-scivocab-uncased* [Beltagy *et al.*, 2019] is shown to be more effective than BERT [Wadden *et al.*, 2019], we used it as the base encoder for SciERC. We adopt a cross-sentence contextual text. We expand each sentence according to the context and make sure that the original sentence is located

Model	Encoder	ACE05			SciERC			CoNLL04		
		Ent	Rel	Rel+	Ent	Rel	Rel+	Ent	Rel	Rel+
[Li and Ji, 2014]^{j}	-	80.8	52.1	49.5	-	-	-	-	-	-
[Miwa and Sasaki, 2014]^{j}	-	-	-	-	-	-	-	80.7	61.0	-
[Miwa and Bansal, 2016]^{j}	LSTM	83.4	-	55.6	-	-	-	-	-	-
[Katiyar and Cardie, 2017]^{j}	LSTM	82.6	55.9	53.6	-	-	-	-	-	-
[Zhang *et al.*, 2017]	LSTM	83.6	-	57.5	-	-	-	-	-	-
[Bekoulis *et al.*, 2018a]^{j}	LSTM	-	-	-	-	-	-	83.6	62.0	-
[Bekoulis *et al.*, 2018b]^{j}	LSTM	-	-	-	-	-	-	83.9	62.0	-
[Tran and Kavuluru, 2019]	LSTM	-	-	-	-	-	-	84.6	62.7	-
[Luan *et al.*, 2019]$^{\spadesuit j}$	LSTM+ELMo	88.4	63.2	-	65.2	41.6	-	-	-	-
[Dixit and Al-Onaizan, 2019]$^{\spadesuit j}$	LSTM+ELMo	86.0	-	62.8	-	-	-	-	-	-
[Li *et al.*, 2019]^{j}	BERT-base	84.8	-	60.2	-	-	-	87.8	68.9	-
[Lin *et al.*, 2020]^{j}	BERT-large	88.8	67.5	-	-	-	-	-	-	-
[Wadden *et al.*, 2019]$^{\spadesuit j}$	BERT-base	88.6	63.4	-	-	-	-	-	-	-
[Wadden *et al.*, 2019]$^{\spadesuit j}$	SciBERT	-	-	-	67.5	48.4	-	-	-	-
[Wang and Lu, 2020]^{j}	ALBERT-xxlarge-v1	89.5	67.6	64.3	-	-	-	-	-	-
[Zhong and Chen, 2020]$^{\spadesuit p}$	SciBERT	-	-	-	68.9	50.1	**36.8**	-	-	-
[Zhong and Chen, 2020]$^{\spadesuit p}$	ALBERT-xxlarge-v1	90.9	69.4	67.0	-	-	-	-	-	-
PERA (ours)$^{\spadesuit j}$	BERT-base	91.8	71.3	64.4	-	-	-	89.6	86.4	76.7
	SciBERT	-	-	-	**75.5**	**55.3**	35.7	-	-	-
	ALBERT-xxlarge-v2	**92.0**	**74.1**	**68.7**	-	-	-	**92.1**	**86.7**	**77.9**

Table 1: The F1 scores on the test set of ACE05, CoNLL04, and SciERC. $^{\spadesuit}$: indicates that the model makes use of cross-sentence information. ^{j} denotes the joint method. ^{p} denotes the pipeline method. Ent indicates that the entity boundaries and type must be correct. Rel denotes the boundaries evaluation (the entity boundaries must be correct). Rel+ denotes the strict evaluation (both the entity boundaries and types must be correct).

Dataset	$\mid \mathcal{E} \mid$	$\mid \mathcal{R} \mid$	Sentences		
			Train	Dev	Test
ACE05	7	6	10,051	2,424	2,050
SciERC	6	7	1,861	275	551
CoNLL04	4	5	922	231	288

Table 2: The statistics of datasets ACE05, SciERC and CoNLL04. We use these three datasets to evaluate our model.

in the middle of the expanded sentence as much as possible. For datasets ACE05, CoNLL04 and SciERC, we set the maximum length of the extended sentences to 256. We consider spans up to $L = 8$ words. On the data sets ACE05, CoNLL04 and SciERC, we take the retention factor $\lambda = 0.05$.

4.2 Main Results

In Table 1, we compare our model PERA with all previous joint model results and pipeline model results. Our model uses cross-sentence contextual text, so our F1 scores are cross-sentence F1 scores. As shown in Table 1, our model achieves better performance compared to previous work. Compared to all previous work, our BERT-base model and SciBERT model achieve similar or better results, and our experimental results are further improved by using a larger encoder ALBERT.

For named entity recognition, our method outperforms the previous state-of-the-art, respectively achieving +1.1%, +6.6%, +4.3% absolute F1 improvements on ACE05, SciERC, and CoNLL04. For relation extraction, our method respectively outperforms the previous state-of-the-art with absolute F1 values of +4.7%, +5.2%, and +17.8% on ACE05, SciERC and CoNLL04.

Compared to [Wang and Lu, 2020], our model obtained 2.5% higher F1 for named entity recognition and 6.5% higher F1 for relation extraction using a similar ALBERT pre-trained model on ACE05. Our approach is much more effective than previous best approaches using global features [Lin *et al.*, 2020] or complex neural models such as MT-RNNs [Wang and Lu, 2020], and achieves large improvements on all datasets. This improvement demonstrates the effectiveness of incorporating attention mechanisms and position-attentive markers in a joint entity and relation extraction model. We also note that compared to the previous state-of-the-art model PURE [Zhong and Chen, 2020] based on a similar ALBERT, our model shows a significant improvement not only in entity F1 (90.9 vs 92.0) but also in relation F1 (69.4 vs 74.1). This clearly demonstrates the superiority of our model.

5 Analysis

We effectively combine attention mechanisms and position-attentive markers into a joint entity and relation extraction model, and experimental results show that our approach outperforms all previous joint models. In this section, our goal is to go deeper and understand what factors determine its final performance.

5.1 Effect of Negative Sample Size

The model based on the span representation can exhaust all possible entities and therefore can identify overlapping entities. However, the span representation method generates a

Model	CoNLL04			ACE05		
	Ent	Rel	Rel+	Ent	Rel	Rel+
base	83.4	85.8	64.7	88.6	71.2	62.6
base+PA	83.4 (+0.0)	**86.7 (+0.9)**	66.7 (+2.0)	89.1 (+0.5)	**71.4 (+0.2)**	63.9 (+1.3)
base+attn	88.4 (+5.0)	86.5 (+0.7)	**77.7 (+13.0)**	91.2 (+2.6)	70.7 (-0.5)	64.2 (+1.6)
base+attn+PA	**89.6 (+6.2)**	86.4 (+0.6)	76.7 (+12.0)	**91.8 (+3.2)**	71.3 (+0.1)	**64.4 (+1.8)**

Table 3: Ablation experiment on the test set of CoNLL04 and ACE05. We use the BERT-base model on the CoNLL04 dataset and the ACE05 dataset. If the attention mechanism is used in the model, the size of the retention factor is $\lambda = 0.05$. base: base model without attention mechanisms and position-attentive markers. base+PA: add position-attentive markers to the base model. base+attn: add attention mechanism to the base model. base+attn+PA: our model with attention mechanisms and position-attentive markers.

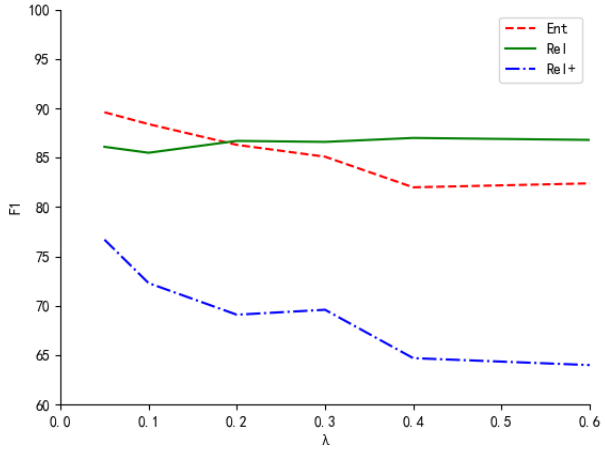

Figure 2: The F1 scores of Ent, Rel and Rel+ for our BERT-base model with different retention factors λ on the CoNLL04 data set.

large number of negative samples of candidate entities, and such a large number of negative samples of candidate entities will bring noise to the model and hurt the performance of the model. For example, if a sentence with n tokens is input, then the number of all its possible candidate spans is $O(n^2)$, but in general the number of gold entities in this sentence is only a few. Our main observation is that it is crucial to reduce the number of input negative samples during the named entity recognition phase. As shown in Figure 2, the Rel F1 basically does not change with the change of retention factor, while the Ent F1 and Rel+ F1 increase drastically with the decrease of retention factor. Therefore, an intuitive conclusion is that a large number of negative samples of candidate entities hurts the performance of the model, and removing some of the negative samples before feeding candidate spans to the span classifier can improve the performance of the model.

5.2 Importance of Position Information

In our model, we append position-attentive markers at the end of the original input sentences with the aim of allowing the model to learn the position information of the original input words to help improve the performance of the model. The experimental results prove that our approach is effective. As shown in Table 3, adding position-attentive markers to the model improved Ent F1 by 0.5% and Rel F1 by 0.2% on dataset ACE05, and Rel F1 improved by 0.9% on dataset CONLL04. Therefore, a reasonable explanation is that adding position-attentive markers to the model enables the model to learn the positional information of the input words. The position information of words is helpful for both named entity recognition and relation extraction.

5.3 Combined Effect

Table 3 shows that although the methods of adding only attention mechanisms or only position-attentive markers to the base model improved the performance of the model compared to the base model, they did not achieve the best performance of our model. The experimental results show that combining attention mechanisms and position-attentive markers in the model can bring out the best performance of the model.

Both adding the attention mechanism to reduce the number of negative samples and adding the position-attentive markers can improve the performance of the model, but they improve the model in different directions. To summarize, (1) introducing the attention mechanism to reduce the number of negative samples of candidate span can improve the performance of the model on named entity recognition; (2) adding position-attentive markers allows the model to learn the position information of the words in the input sentence, which is helpful for both named entity recognition and relation extraction; (3) the results of the ablation experiments show that attention mechanism and position-attentive markers can complement each other, and combining them together can further improve the performance of the model.

6 Conclusion

In this paper, we present a joint entity and relation extraction model that fuses attention mechanism and position-attentive markers. Our model PERA shares the same encoder for named entity recognition and relation extraction, and our experimental results show that it significantly outperforms previous state-of-the-art on three standard benchmarks.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.62072112, 62176185), Scientific and Technological Innovation Action Plan of Shanghai Science and Technology Committee (No.20511103102), Fudan Double First-class Construction Fund (No. XM03211178).

References

[Bekoulis *et al.*, 2018a] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. *arXiv preprint arXiv:1808.06876*, 2018.

[Bekoulis *et al.*, 2018b] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 2018.

[Beltagy *et al.*, 2019] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dixit and Al-Onaizan, 2019] Kalpit Dixit and Yaser Al-Onaizan. Span-level model for relation extraction. In *Proceedings of ACL*, pages 5308–5314, 2019.

[Eberts and Ulges, 2019] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. *arXiv preprint arXiv:1909.07755*, 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 *Proceedings of the 57th Annual Meeting of ACL*, pages 1409–1418, 2019.

[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 *Proceedings of COLING 2016*, pages 2537–2547, 2016.

[He *et al.*, 2018] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*, 2018.

[Ji *et al.*, 2020] Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *Proceedings of COLING*, pages 88–99, 2020.

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

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *ICLR*, 2019.

[Lee *et al.*, 2017] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.

[Lee *et al.*, 2018] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*, 2018.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of ACL*, pages 402–412, 2014.

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

[Lin *et al.*, 2020] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of ACL*, 2020.

[Luan *et al.*, 2018] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *EMNLP*, 2018.

[Luan *et al.*, 2019] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. *NAACL*, 2019.

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

[Miwa and Sasaki, 2014] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of EMNLP*, 2014.

[Nguyen and Verspoor, 2019] Dat Quoc Nguyen and Karin Verspoor. End-to-end neural relation extraction using deep biaffine attention. In *ECIR*. Springer, 2019.

[Ouchi *et al.*, 2018] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. A span selection model for semantic role labeling. *arXiv preprint arXiv:1810.02245*, 2018.

[Tran and Kavuluru, 2019] Tung Tran and Ramakanth Kavuluru. Neural metric learning for fast end-to-end relation extraction. *arXiv preprint arXiv:1905.07458*, 2019.

[Wadden *et al.*, 2019] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*, 2019.

[Wang and Lu, 2020] Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. *EMNLP*, 2020.

[Ye *et al.*, 2021] Deming Ye, Yankai Lin, and Maosong Sun. Pack together: Entity and relation extraction with levitated marker. *arXiv preprint arXiv:2109.06067*, 2021.

[Zhang *et al.*, 2017] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *Proceedings of EMNLP*, 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. *arXiv preprint arXiv:1706.05075*, 2017.

[Zhong and Chen, 2020] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. *arXiv preprint arXiv:2010.12812*, 2020.