EnPaR:Enhancing Entity and Entity Pair Representations for Joint Entity Relation Extraction

Yijun Wang[1, 2], **Changzhi Sun**[4], **Yuanbin Wu**[3], **Hao Zhou**[4], **Lei Li**[4], **and Junchi Yan**[1, 2]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]School of Computer Science and Technology, East China Normal University
[4]ByteDance, AI Lab

yijunwang.cs@gmail.com ybwu@cs.ecnu.edu.cn yanjunchi@sjtu.edu.cn
{sunchangzhi, zhouhao.nlp, lileilab}@bytedance.com

Abstract

Current state-of-the-art systems for joint entity relation extraction (Luan et al., 2019; Wadden et al., 2019) usually adopt the multi-task learning framework. However, annotations for these additional tasks such as coreference resolution and event extraction are always equally hard (or even harder) to obtain. In this work, we propose a pre-training method EnPaR to improve the joint extraction performance. EnPaR requires only the additional entity annotations that are much easier to collect. Unlike most existing works that only consider incorporating entity information into the sentence encoder, we further utilize the entity pair information. Specifically, we devise four novel objectives, *i.e.*, masked entity typing, masked entity prediction, adversarial context discrimination, and permutation prediction, to pre-train an entity encoder and an entity pair encoder. Comprehensive experiments show that the proposed pre-training method achieves significant improvement over BERT on ACE05, SciERC, and NYT, and outperforms current state-of-the-art on ACE05.

1 Introduction

Joint extraction of entities and relations is a fundamental task in information extraction, it aims to extract entities and relations with a unified model. Current approaches (Luan et al., 2019; Wadden et al., 2019) usually adopt the multi-task learning framework that optimizes many objectives simultaneously, including entity recognition, relation extraction, coreference resolution, and event extraction. However, as large-scale manually labeled data required by these methods is unavailable in many domains, their applicability is severely restricted. Therefore, we expect to catch or even surpass the multi-task based joint models with less annotation cost. Compared with the annotations of coreference resolution and event extraction, entity

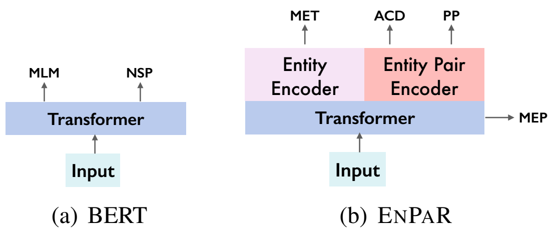


Figure 1: The network architectures and objectives of BERT and EnPaR. Our work introduce an entity encoder and an entity pair encoder. MLM = masked language model, NSP = next sentence prediction, MET = masked entity typing, MEP = masked entity prediction, ACD = adversarial context discrimination, PP = permutation prediction.

annotations can be easily obtained through automatic NER annotation tools (e.g., spaCy [1]). In this paper, we focus on improving the model's performance with just extra entity annotations.

Although pre-trained models, like BERT, have shown impressive performance in many downstream tasks, they have mainly two limitations when applied in the joint entity relation extraction task. One is that currently pre-trained objectives are insufficient for this task. Specifically, these commonly used universal pre-trained model (e.g., BERT) do not consider the entity-related knowledge that is crucial for better extracting entities and relations. The other is that these models only provide pre-trained representations for tokens and sentences, but not entities and entity pairs. To obtain the representations for entities and entity pairs, additional parameters that are not pre-trained are introduced in the fine-tuning stage, which may futher impair the joint extraction performance.

To address the first limitation, recent several works try to incorporate entity-related information

[1]https://spacy.io/

into pre-training objectives. Zhang et al. (2019) fuses heterogeneous information from both texts and knowledge graphs and proposes a denoising entity auto-encoder objective based on BERT. Sun et al. (2019c) presents two knowledge masking strategies in the pre-training stage (entity-level masking and phrase-level masking). Both of them utilize extra entity annotations (*i.e.*, entities in knowledge graphs and automatic entity annotations, respectively). In this paper, we follow this line of works and build a large-scale entity annotated corpus using the spaCy NER tool.

For the second limitation, we propose ENPAR, a pre-training method customized for entity relation extraction. ENPAR consists of an underlying sentence encoder, an entity encoder, and an entity pair encoder. Compared with BERT (Figure 1(a)), the proposed entity encoder and entity pair encoder directly provide representations of entities and entity pairs. To train the three encoders, we devise four novel pre-training tasks: *masked entity typing*, *masked entity prediction*, *adversarial context discrimination* and *permutation prediction* (Figure 1(b)). In the first two tasks, we randomly mask some entity words and then predict the masked tokens and the entity type. These two tasks are natural extensions of the masked language model. To learn a better entity pair encoder, we draw inspirations from the denoising auto-encoder (Zhang et al., 2019) and propose the last two tasks. Specifically, when the entity pair or its context in a sentence are perturbed, we hope that the entity pair encoder is capable of tracking such changes. We employ the parameter-sharing method for these four tasks and train these objectives jointly.

To sum up, our main contributions are as follows [2]:

• We introduce an entity encoder and an entity pair encoder to incorporate not only the entity information but also the entity pair information, which were ignored in current universal pre-trained models.

• We propose four novel pre-training tasks that help to learn the proposed encoders. These tasks only require additional entity annotations (with commonly used entity types), which can be automatically generated by public annotation tools, such as spaCy NER.

• We conduct comprehensive experiments and demonstrate that the proposed method achieves significant improvement on ACE05 and NYT dataset and is comparable with the state-of-the-art on the SciERC dataset.

2 Approach

Given an input sentence $s = x_1, \dots, x_{|s|}$ and a set of entities $\mathcal{E}$ (automatically annotated) in s, ENPAR is to encode each entity $e \in \mathcal{E}$ and each entity pair (e_1, e_2) into a contextual representation vector. As shown in Figure 2, ENPAR is composed of a shared Transformer (Vaswani et al., 2017), an entity-level CNN followed by an MLP (multi-layer perceptron), a context-level CNN followed by an MLP, and the last MLP. In the pre-training stage, we optimize ENPAR with four objectives, namely, masked entity typing, masked entity prediction, adversarial context discrimination and permutation prediction. These pre-training objectives can integrate rich entity-related information into the proposed network. After pre-training, we can easily fine-tune the pre-trained network for entity relation extraction task.

2.1 Pre-training Network Architecture

In this section, we will introduce the overall ENPAR architecture in three parts: the sentence encoder, the entity encoder, and the entity pair encoder.

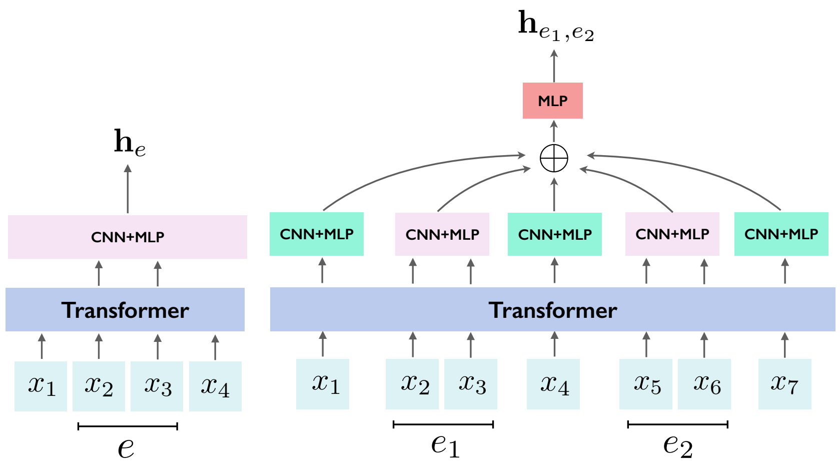
Sentence Encoder As previous pre-training models (UNILM, BERT, and XLM), we also apply the multi-layer Transformer (Vaswani et al., 2017) as the basic sentence encoder for obtaining the contextual representations $\mathbf{h}_i$ for each token in the sentence s. The output of multi-layer Transformer is computed via:

$$\{\mathbf{h}_1, \dots, \mathbf{h}_{|s|}\} = \texttt{Transformer}(\{\mathbf{x}_1, \dots, \mathbf{x}_{|s|}\})$$

The word representation $\mathbf{x}_i$ of x_i follow that of BERT (Devlin et al., 2018), which is a sum of the corresponding token, segment and position embeddings.

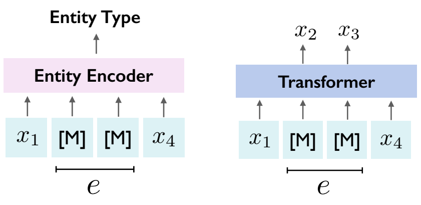
Entity Encoder For each entity $e \in \mathcal{E}$ in the sentence s, the corresponding contextual entity representation $\mathbf{h}_e$ can be obtained by employing a CNN (a single convolution layer with a max-pooling layer) followed by an MLP on vectors $\{\mathbf{h}_i | x_i \in e\}$, as shown in Figure 2(a).

[2]Source code and pre-trained models are available at https://github.com/Receiling/ENPAR.



(a) Entity Encoder (b) Entity Pair Encoder

Figure 2: Entity encoder and entity pair encoder based on a shared sentence encoder. Both share the entity-level CNN with MLP, and entity pair encoder contains a context-level CNN with MLP.



(a) MET (b) MEP

Figure 3: Two objectives to learn enhanced representation of single entities.

Entity Pair Encoder For each entity pair (e_1, e_2) in the sentence s, to obtain the corresponding contextual entity pair representation $\mathbf{h}_{e_1,e_2}$, we extract two types of features. The first is the features regarding words in e_1, e_2, and the second is the features regarding contexts of the entity pair (e_1, e_2). For features on words in e_1, e_2, we use the output of entity encoder, namely, $\mathbf{h}_{e_1}$ and $\mathbf{h}_{e_2}$. For context features of the entity pair (e_1, e_2), we extract three feature vectors by looking at left context words, middle context words and right context words of the entity pair (e_1, e_2). Similar to entity encoder, we compute three feature vectors by employing another CNN followed by an MLP. Finally, we concatenate the five feature vectors into a single vector. To get the resulting entity pair representation $\mathbf{h}_{e_1,e_2}$, the single vector was fed into another MLP, as shown in Figure 2(b).

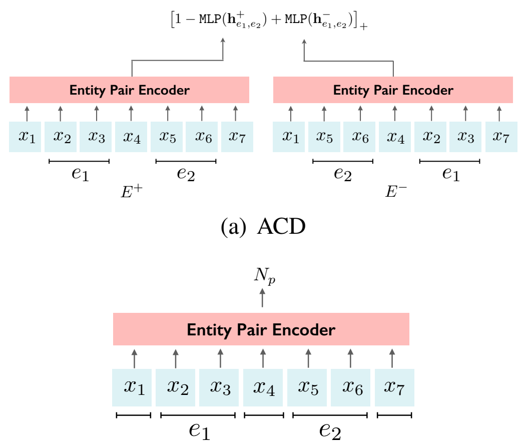
2.2 Pre-training Objectives

We design four pre-training objectives to guide ENPAR to absorb more entity-related knowledge, which is particularly important for entity relation extraction task. The four objectives can be divided into two groups: the first two objectives are to enhance the representations of single entities, and the latter two objectives are to enhance the representations of entity pairs. These objectives are trained jointly (simply sum the objective functions). Our pre-training objectives are based on a dataset with entity annotation, which can be obtained through the public annotation tool. For instance, `PER`("Obama"), `ORG`("Labour Party") were annotated by spaCy NER, and `PER`, `ORG` are entity types (there are 18 entity types).

Masked Entity Typing (MET) In this task, we simply mask some entity words at random, and then predict the corresponding entity type[3]. For instance, given a masked word sequence "x_1, `[M]`, `[M]`, x_4", to predict the masked entity type (e.g., `PER`), we first use the entity encoder to extract the contextual masked entity representation, and then predict the entity type. The objective is to minimize the cross-entropy loss computed using the predicted entity type and the original entity type, as shown in Figure 3(a).

Masked Entity Prediction (MEP) This task is similar to the masked LM in BERT and is identical to the entity-level masking in (Sun et al., 2019c). Specifically, we randomly choose some entity words in the sentence, and replace them with special word `[M]`. Then we feed their corre-

[3]It is worth noting that the entity types annotated by spaCy NER may be different from the entity types of downstream datasets.


$[1 - \texttt{MLP}(\mathbf{h}^+_{e_1,e_2}) + \texttt{MLP}(\mathbf{h}^-_{e_1,e_2})]_+$
Entity Pair Encoder
Entity Pair Encoder
x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_1 x_5 x_6 x_4 x_2 x_3 x_7
e_1 e_2 E^+
e_2 e_1 E^-
N_p
Entity Pair Encoder
x_1 x_2 x_3 x_4 x_5 x_6 x_7
e_1 e_2

(a) ACD

(b) PP

Figure 4: Two objectives to learn enhanced representation of entitie pairs.

sponding output vectors computed by the sentence encoder into a softmax classifier to predict the masked entity word. The model is learned to recover the masked entity words, as shown in Figure 3(b). In short, both mask entity typing and mask entity prediction encourage the model to learn the information of single entities.

Adversarial Context Discrimination (ACD) Given an input sentence s with an entity pair (e_1, e_2), we regard it as a positive sample $E^+ = (s, e_1, e_2)$. According to E^+, we can generate a negative sample $E^- = (s', e_1, e_2)$ that has a main property: the context words and the order of context words w.r.t. the entity pair (e_1, e_2) in E^- are minimally different from those in the original sample E^+. If the entity pair encoder can characterize the context words well, it should be able to recognize small context differences of the entity pair between E^+ and E^-. We refer to this objective as adversarial context discrimination. The hinge loss function was imposed into the positive sample and corresponding negative samples to achieve the goal. Specifically, as shown in Figure 4(a), we can obtain a score using the entity pair encoder with an MLP for each sample, then the hinge loss function is computed via:

$$\left[1 - \texttt{MLP}(\mathbf{h}^+_{e_1,e_2}) + \texttt{MLP}(\mathbf{h}^-_{e_1,e_2})\right]_+ .$$

where $[u]_+ = \max(u, 0)$ is the hinge loss and $\texttt{MLP}$ outputs a scalar value. $\mathbf{h}^+_{e_1,e_2}$ and $\mathbf{h}^-_{e_1,e_2}$ are the output of entity pair encoder for positive sample E^+ and negative sample E^-.

Here we introduce our strategies of generating a negative sample $E^- = (s', e_1, e_2)$ according to a

Permutation	Class
(left context, e_1, middle context, e_2, right context)	001
(left context, e_1, middle context, right context, e_2)	002
(left context, e_1, e_2, middle context, right context)	003
⋮	⋮
(right context, e_2, middle context, e_1, left context)	120

Figure 5: Permutation samples.

positive sample $E^+ = (s, e_1, e_2)$. In fact, there are many negative samples. For the sake of simplicity and clarity, we only adopt following simple rules to generate five negative samples.

- Swap entity e_1 and entity e_2 in the original sample E^+;
- Shift the entity e_1 few positions (n_s) to the left or right.
- Shift the entity e_2 few positions (n_s) to the left or right.

Permutation Prediction (PP) Given an input sentence s with an entity pair (e_1, e_2), the sentence s was split into five parts, namely, left context, e_1, middle context, e_2 and right context. If we shuffle the five parts, does the entity pair encoder have the ability to recognize it? Inspired by this question, we propose an enhanced objective, named permutation prediction, to help to learn a better entity pair encoder. Formally, let $\mathcal{P}$ be the set of all possible permutation of the fives parts, as shown in Figure 5. Obviously, the number of all possible permutations is 5! ($|\mathcal{P}| = 120$). For each permutation $p \in \mathcal{P}$, we first assign it a unique permutation class $N_p (1 \leq N_p \leq 120)$, and then use the entity pair encoder to extract the contextual entity pair representation for predicting the permutation class, as shown in Figure 4(b). The objective is to optimize the cross-entropy loss computed using the predicted permutation class and the gold permutation class. It is costly to consider all permutations. So we sample n_p permutations in practice (we always include the correct permutation).

2.3 Pre-training Setup

In the pre-training stage, we directly optimize the sum of the above four objective functions. Same as UniLM (Dong et al., 2019), we use gelu as activation function. And the sentence encoder is initialized with BERT$_{\text{BASE}}$ weights. We use the

English Wikipedia [4] as pre-training corpus, which has been processed in similarly as (Devlin et al., 2018). The spaCy NER [5] was used to annotate entities. After preprocessing the corpus, there are nearly 820M words and 95M entities in the annotated input. We discard the sentences having less than 3 entities for effectiveness, and only pre-train our model for one epoch. The vocabulary size is 28996, The maximum length of the input sequence is 256. For each entity, we replace the entity words with `[M]` with probability 15%, randomly replace other entity with probability 5% and keep the original entity words for the rest. Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$ is used for optimization. The batch size is 512. The learning rate is 5e-5, with linear warmup rate over the first 10% steps and linear decay. The dropout rate is 0.1. The weight decay is 0.01. It takes about 20 hours for 10, 000 steps using 1 Nvidia Telsa V100 16GB GPU.

2.4 Fine-tuning for Entity Relation Extraction

In the fine-tuning stage, we adopt the same definition of entity relation extraction task as (Sun et al., 2019a). The joint entity relation extraction task can be decomposed into three objectives: entity span detection, entity typing, and relation typing. Firstly, we treat the entity span detection as a sequence labeling task. We regard the first sub-word's output of the sentence encoder as the token-level representation. Then we take the token-level representation as the input to the softmax classifier and compute the cross-entropy loss with respect to gold entity span labels. Secondly, for each detected entity span, the entity classifier uses the corresponding output of the entity encoder to predict entity type. For each detected entity span pair, the relation classifier uses the corresponding output of the entity pair encoder to predict relation type. Both the entity classifier and the relation classifier are randomly initialized softmax layer. Also, we adopt the cross-entropy loss for these two tasks. Besides, all three objectives are optimized simultaneously.

We only tune the hyperparameters on the ACE2005 development set based on the joint performance of entity and relation, then apply the same hyperparameters on the SciERC and NYT datasets. Scheduled sampling strategy (Miwa and Bansal, 2016) and discriminative fine-tuning strategy (Howard and Ruder, 2018) are emploied in fine-tuning. We kepp the same dropout rate as pre-training (*i.e.*, 0.1). The learning rate is 2.5e-5 with weight decay 0.01. We apply a linear warmup scheduler over the first 20% steps and then linear decay. We train our model with a maximum of 200 epochs with early stop strategy in a single Nvidia GeForce GTX 1080 Ti GPU.

3 Experiments

We conduct experiments on three benchmark entity relation extraction datasets: ACE05, SciERC, and NYT. For space limitation, we will mainly discuss the results on ACE05 and report basic results on the remaining two datasets.

ACE05 The ACE05 dataset [6] that is a standard corpus for entity relation extraction task annotates entity and relation labels for a collection of documents. ACE05 contains 7 entity types and 6 relation types. We use the same data split and preprocessing of ACE05 dataset (351 training, 80 validating and 80 testing) as (Miwa and Bansal, 2016) and (Sun et al., 2018).

SciERC The SciERC dataset [7] annotates entity, coreference and relation labels for 500 scientific abstracts from 12 AI conference/workshop proceedings. We only use the annotations of entities and relations. SciERC contains 6 scientific term (entity) types and 7 relation types. We use the same data split and preprocessing of SciERC dataset (350 training, 50 validating and 100 testing) as (Luan et al., 2019).

NYT The NYT dataset[8] is a large-scale corpus which automatically annotates a collection of New York Times news articles. NYT contains 3 types of entities and 12 types of relations. The training set is automatically annotated by distant supervision. While the validation and testing data are manually labeled by (Jia et al., 2019). We choose the latest version of NYT released by (Jia et al., 2019).

Evaluation. As previous works (Miwa and Bansal, 2016; Sun et al., 2019a), we evaluate the

[4]Wikipedia version: enwiki-20190301.

[5]The spaCy model is "en_core_web_md" in version: 2.1.8, which trained on OntoNotes dataset.

[6]https://github.com/tticoin/LSTM-ER

[7]http://nlp.cs.washington.edu/sciIE/

[8]https://github.com/PaddlePaddle/models/tree/develop/PaddleNLP/Research/ACL2019-ARNOR/

Model	**Entity**	**Relation**	**Relation** (exactly)
Sun, 2019a	84.2	–	59.1
Li, 2019$^{\diamond}$	84.8	–	60.2
Luan, 2019$^{\star,\circ}$	88.4	63.2	–
Wadden, 2019 $^{\diamond,\circ}$	**88.6**	63.4	–
ENPAR $^{\diamond}$	86.9	**66.1**	**63.5**

Table 1: Results on the ACE05 test data. $\diamond$ means that the model uses BERT. $\star$ means that the model uses ELMo as token embeddings. $\circ$ stands for training the model with multi-task learning. ENPAR is the proposed model fine-tuned on ACE05 dataset.

performances using F1 score. Specifically, an output entity is correct if its type label and head region match with a gold entity, then an output relation is correct if both its type and its two argument entities are all correct (i.e. exactly match). While some previous works (Luan et al., 2019; Wadden et al., 2019; Sanh et al., 2019) do not consider entity type in relation evaluation. Thus, we also report this result for comparison.

3.1 Results on ACE05

First, we compare the proposed pre-training method with previous works in Table 1. In general, the relation performance of ENPAR significantly exceeds all existing models in two relation evaluation criteria. Specifically, in exactly matching mode, our method achieves 4.4 points improvement compared with the LSTM-based GCN joint model(Sun et al., 2019a) and increases by 3.3 points compared with the BERT-based QA model(Li et al., 2019). Even compared to the multi-task learning models based on BERT (Wadden et al., 2019), our method still achieves 2.7 points improvement on relation performance. Although the entity performance of our method inferior to the multi-task learning models (Luan et al., 2019; Wadden et al., 2019), we believe that those additional supervision signals such as coreference and event information in the fine-tuning step may cause the gap. Besides, they even consider all spans and cross-sentence context, which are empirically beneficial to entity performance. However, even with the slightly inferior entity encoder and lack of additional multi-task training data, our pre-trained entity pair encoder still achieves significantly superior relation performance, which fully demonstrates the powerfulness of the proposed pre-training method.

Next, we evaluate the proposed pre-training method with different settings. We have following four detailed observations regarding the results in Table 2.

Model	**Entity**	**Relation**	**Relation** (exactly)
BERT	87.2	65.1	62.2
ENPAR	86.9	**66.1**	**63.5**
- MET	87.1	65.3	62.5
- MEP	**87.4**	65.6	62.7
- ACD	87.2	65.7	62.9
- PP	87.1	64.2	61.6
- CNN	87.1	66.0	62.9

Table 2: Results on the ACE05 test data in different settings. BERT is our model without pre-training, which is initialized by BERT$_{\text{BASE}}$ and fine-tuned on ACE05 dataset. "- *" is ENPAR without * task, where $* \in$ {MET, MEP, ACD, PP} ; "- CNN" means that only loading parameters of the sentence encoder from ENPAR, and the other parameters (i.e., the entity encoder and the entity pair encoder) are randomly initialized.

Percentage of Data	**Entity**	**Relation**	**Relation** (exactly)
100%	86.9	65.2	61.7
75%	86.8	65.3	62.5
50%	87.1	64.6	61.5
25%	86.9	**66.1**	63.5
15%	87.3	65.9	**63.6**
5%	**87.4**	65.9	63.4

Table 3: Results on the ACE05 test data by varying the size of pre-training data.

• ENPAR (line 2) achieves superior relation performance (1.0 point and 1.3 points improvement) and comparable entity performance compared with BERT (line 1). This result demonstrates that the proposed pre-training objectives inject more entity-related information into the pre-trained model and enhance the relation extraction performance.

• Overall, the entity performances of all models fluctuate quite slightly (0.5 points). Interestingly, "- MEP" (line 4) achieves the best entity performance though its relation performance is not the best. The stable entity performance reflects that the token-level information is likely enough for entity recognition. Besides, the final training objectives may bias entity objectives or entity pair objectives, which leads to more improvement in relation performance than entity performance. Thus, we pay more attention relation perfromance in this paper.

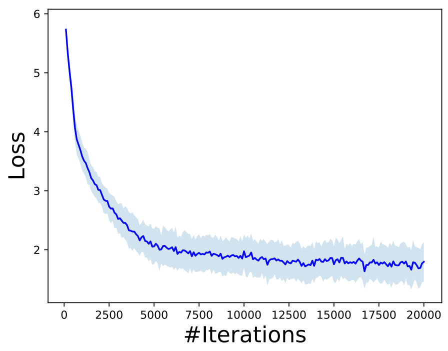


Figure 6: Pre-training loss with respect to the number of pre-training iterations.

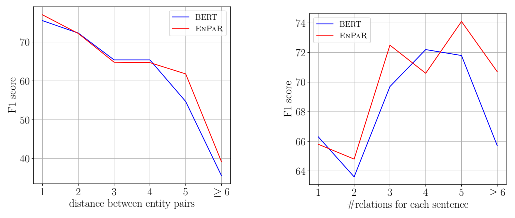
• When any pre-training objective is removed, the relation performance will decrease with varying degrees. Particularly, "- PP" (line 6) drops the most (both 1.9 points decline in two relation evaluation criteria). This phenomenon just indicates the importance of the "PP" objective for the entity pair encoder and relation extraction.

• We also try only to use the pre-trained weights of the Transformer (sentence encoder), while the other parameters are randomly initialized in the fine-tuning step (line 7). The relation performance of the "- CNN" slightly declines compared with EnPar, but it still outperforms BERT. This result reflects that the sentence encoder has absorbed the entity-related knowledge and can achieve more encouraging performance with the entity encoder and the entity pair encoder.

Thirdly, we present the influences of pre-training data size (Table 3). In experiments, we found the loss of the pre-trained model tends to be stable after iterating 5k steps (about 25% of all data), as shown in Figure 6. Table 3 demonstrates that the performance of the pre-trained model was quite competitive using 25% of pre-training data [9], and more pre-training data does not further improve performance. There is a similar conclusion on text classification (Sun et al., 2019b), which performs within-task further pre-training. This observation shows that the proposed pre-training method does not require expensive training costs.

Finally, we examine the relation performance with respect to different distances between entity pairs (Figure 7(a)) and different the number of relations for each sentence (Figure 7(b)), and also give some concrete examples to verify it (Table 4) [10]. Comparing with BERT, EnPar is good at handling long-distance relation dependencies and interactions between multiple relations in a sentence. For S1, BERT does not detect the long distance relation PHYS between "[barbara starr]" and "[pentagon]" while EnPar can handle it. For S2, EnPar identifies a relation PHYS between "[charles]" and "[london]", but BERT fails even the relation PHYS between "[vladimir putin]" and "[london]" was detected. It shows BERT does not fully exploit the multiple relations in a sentence. We attribute these results to the powerful representations learned by the proposed pre-training objectives.

[9]In all experiments, we choose the number of pre-training data, n_p and n_s according to the performance on the development set during the fine-tuning step.



(a) (b)

Figure 7: (a) F1 score with respect to the distance between the entity pairs. (b) F1 score with respect to the number of relation for each sentence.

3.2 Results on SciERC and NYT

SciERC The upper part of Table 5 shows the results of SciERC. Compared with BERT, EnPar achieves 0.5 points improvement on entity performance and 2.2 points (exactly match) improvement on relation performance. This result reflects the effectiveness of the proposed pre-training objectives for entity relation extraction. Compared with the previous state-of-the-art model (Wadden et al., 2019), which is a multi-task learning model based on BERT, EnPar achieves superior entity performance and comparable relation performance without additional multi-task training data. It worth noting that, the entities in the SciERC dataset are different from our pre-training data. Therefore, the entity encoder and the entity pair encoder can encode contexts of entities rather than only encoding information respect to specific entities. This property is desired as it means that we could utilize any entity annotator.

[10]More detailed evaluations are in the Appendix A

S1	…talk about what [barbara starr]$^{\texttt{PER:}\heartsuit\clubsuit\spadesuit}_{\texttt{PHYS-1:}\heartsuit\spadesuit}$ was reporting from the [pentagon]$^{\texttt{FAC:}\heartsuit\clubsuit\spadesuit}_{\texttt{PHYS-2:}\heartsuit\spadesuit}$.
S2	…[vladimir putin]$^{\texttt{PER:}\heartsuit\clubsuit\spadesuit}_{\texttt{PHYS-1:}\heartsuit\clubsuit\spadesuit}$ was greeted by prince [charles]$^{\texttt{PER:}\heartsuit\clubsuit\spadesuit}_{\texttt{PHYS-3:}\heartsuit\spadesuit}$ as he arrived in [london]$^{\texttt{GPE:}\heartsuit\clubsuit\spadesuit}_{\texttt{PHYS-2:}\heartsuit\clubsuit\spadesuit\texttt{\|PHYS-4:}\heartsuit\spadesuit}$ today .

Table 4: Examples from the ACE05 dataset with label annotations from BERT and ENPAR for comparison. The ♡ is the gold standard, and the ♣, ♠ are the output of the BERT ,ENPAR respectively.

Model	**Entity**	**Relation**	**Relation** (exactly)
Luan, 2019$^{\star,\circ}$	65.2	41.6	–
Wadden, 2019 $^{\diamond,\circ}$	67.5	**48.4**	–
BERT $^{\diamond}$	67.4	46.0	34.3
ENPAR $^{\diamond}$	**67.9**	48.0	**36.5**
BERT $^{\diamond}$	92.7	50.8	49.3
ENPAR $^{\diamond}$	**94.8**	**54.4**	**52.6**

Table 5: Results on the SciERC test data (upper part) and the NYT test data (bottom part).

NYT For entity relation extraction task, there are no previous works on this dataset. We list the BERT and ENPAR results in the same way as the previous two datasets. For the bottom part of the Table 5, we observe that ENPAR significantly outperforms BERT on entity performance and relation performance. This again verifies the effectiveness of our proposed pre-training method.

4 Related Work

Joint entity relation extraction is an important task that has been extensively studied. One simple method to achieve joint learning is through parameters sharing, which usually share some input embeddings or sentence encoders (Miwa and Bansal, 2016; Katiyar and Cardie, 2017). To further explore the interactions between the outputs of the entity model and the relation model, many joint decoding algorithms were introduced into this joint task (Yang and Cardie, 2013; Li and Ji, 2014; Katiyar and Cardie, 2016; Zheng et al., 2017; Ren et al., 2017; Wang et al., 2018; Sun et al., 2018; Fu et al., 2019). Besides, (Li et al., 2019) tackle this task under the framework of multi-turn QA. And (Sun et al., 2019a) conduct joint type inference via GCN on a bipartite graph composed of entities and relations. Recently, transfer learning (Sun and Wu, 2019), multi-task learning (Sanh et al., 2019; Wadden et al., 2019; Luan et al., 2019) were also applied in this task. In this work, we investigate the pre-trained model for entity relation extraction. For simplicity, we restrict the joint model of parameters sharing, which can be easily extended to jont decoding methods.

Pre-trained models (Yang et al., 2019; Dong et al., 2019; Joshi et al., 2020) have made many amazing breakthroughs on various NLP downstream tasks. Pre-training paradigm first pre-train networks with some pre-training objectives on large-scale unlabeled text corpora, and then fine-tune the pre-trained networks on downstream tasks. These pre-training objectives determine the knowledge absorbed by the pre-trained models. For example, BERT (Devlin et al., 2018) adopts masked language model and next sentence prediction to learn deep contextual representations and the relation between two sentences respectively. XLNet (Yang et al., 2019) uses permutation language model to learn bidirectional representations with autoregressive model. UNILM (Dong et al., 2019) fuses three types of language model: unidirectional, bidirectional and seq2seq prediction. However, these models and objectives all ignore the entity-related information, which is crucial for entity relation extraction task. Recently, there are several public works that explore how to properly integrate entity inforamtion into pre-trained models. Specifically, (Zhang et al., 2019) achieve enhanced language representation by injecting informative entities in KGs into pre-training models. And (Sun et al., 2019c) propose two higher-level masking strategies: entity-level masking and phrase-level masking. In this work, we not only integrate entity information, but also extend to entity pair information and learn more powerful representations for entities and entity pairs.

5 Conclusion

We propose ENPAR, a pre-training method customized for entity relation extraction only with additional entity annotations. Instead of only pre-training sentence encoder in universal pre-trained models, we also pre-train an entity encoder and an entity pair encoder. Then the proposed four objectives can incorporate entity and entity-pair knowledge into the pre-trained encoders to enhance the encoders' representations. Experiments on three datasets demonstrate that EnPaR achieves comparable or even superior performances compared with multi-task based joint models.

Acknowledgement

The authors wish to thank the reviewers for their helpful comments and suggestions. This research is (partially) supported by NSFC (62076097, 61972250, U19B2035), National Key Research and Development Program of China (2018AAA0100704), STCSM (18ZR1411500), the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK(COGOS-20190003). The corresponding authors are Junchi Yan, Changzhi Sun, and Yuanbin Wu.

References

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: Attention regularization based noise reduction for distant supervision relation classification. In *Proc. of ACL*, pages 1399–1408.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proc. of ACL*, pages 919–929.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

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*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. *arXiv preprint arXiv:1904.03296*.

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

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 *Proc. of WWW*, pages 1015–1024.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proc. of AAAI*, pages 6949–6956.

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

Changzhi Sun and Yuanbin Wu. 2019. Distantly supervised entity relation extraction with adapted manual annotations. In *Proc. of AAAI*, volume 33, pages 7039–7046.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proc. of EMNLP*, pages 2256–2265.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune bert for text classification? *arXiv preprint arXiv:1905.05583*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019c. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

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.

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

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*, pages 4461–4467.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proc. of ACL*, pages 1640–1649.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

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. *arXiv preprint arXiv:1706.05075*.

Appendices

A More Evaluations

Table 6 and Table 7 show the performances of "BERT" and our "ENPAR" on each entity type and relation type, respectively. For entity performance, "ENPAR" is almost the same as "BERT". However, for relation performance, we can find "ENPAR" is superior to "BERT" on all relation types except "PER-SOC" relation. The major reason is "ENPAR" significantly improves recall while keeping or sacrificing little precision.

Figure 8 and Figure 9 demonstrate the relation performances (exactly match) of "BERT" and "ENPAR" in different situations. Specifically, Figure 8 shows the results with respect to the number of entities for each sentence, while Figure 9 shows the results with respect to the sentence length for each sentence. In Figure 8, our "ENPAR" significantly outperforms "BERT" when the number of entities in a sentence is more than 2. Similarly, when the sentence length is more than 10, our "ENPAR" is also superior to "BERT" as shown in Figure 9. Both results show the powerful ability of "ENPAR" to identify relations in complicate sentences and long sentences. Moreover, it also proves that the proposed pre-training objectives indeed prompt the model to learn entity related information, which contributes to improving the relation extraction performance.

Entity Type	Model	P	R	F
WEA (109)	BERT	75.9	**78.0**	**76.9**
	ENPAR	**79.0**	72.5	75.6
FAC (286)	BERT	77.9	75.2	76.5
	ENPAR	**78.1**	**76.2**	**77.2**
VEH (116)	BERT	**80.9**	**80.2**	**80.5**
	ENPAR	**80.9**	**80.2**	**80.5**
LOC (136)	BERT	**72.4**	77.2	**74.7**
	ENPAR	70.6	**79.4**	**74.7**
PER (2928)	BERT	**91.7**	**92.3**	**92.0**
	ENPAR	91.0	92.1	91.5
GPE (1013)	BERT	**86.8**	89.4	88.1
	ENPAR	**86.8**	**89.9**	**88.3**
ORG (817)	BERT	**77.6**	**77.0**	**77.3**
	ENPAR	77.3	75.8	76.5

Table 6: The entity performance of "BERT" and "ENPAR" for different entity types on ACE05 test data. The numbers in the first column are counts of entities.

Relation Type	Model	P	R	F
ART (146)	BERT	60.4	39.7	47.9
	ENPAR	**64.1**	**45.2**	**53.0**
PART-WHOLE (175)	BERT	**57.1**	**57.7**	**57.4**
	ENPAR	55.6	56.6	56.1
PER-SOC (73)	BERT	**75.0**	74.0	74.5
	ENPAR	73.1	**78.1**	**75.5**
PHYS (278)	BERT	**61.4**	48.6	54.2
	ENPAR	60.1	**54.7**	**57.3**
GEN-AFF (99)	BERT	**66.2**	45.5	53.9
	ENPAR	63.6	**49.5**	**55.7**
ORG-AFF (354)	BERT	**78.9**	71.8	75.2
	ENPAR	78.4	**72.6**	**75.4**

Table 7: The relation performance (exactly match) of "BERT" and "ENPAR" for different relation types on ACE05 test data. The numbers in the first column are counts of relations.

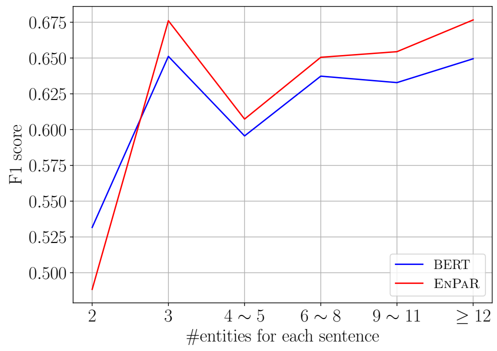


Figure 8: The relaiton F1 score (exactly match) with respect to the number of entities on ACE05 test data.

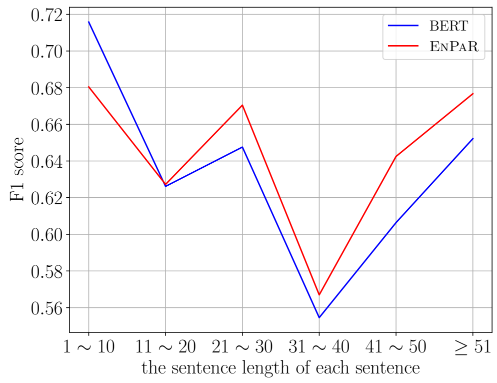


Figure 9: The relaiton F1 score (exactly match) with respect to the the sentence length for each sentence on ACE05 test data.