# ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning

**Yujia Qin**[♣♠♢], **Yankai Lin**[♢], **Ryuichi Takanobu**[♣♢], **Zhiyuan Liu**[♣*], **Peng Li**[♢], **Heng Ji**[♠*], **Minlie Huang**[♣], **Maosong Sun**[♣], **Jie Zhou**[♢]

[♣]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[♠]University of Illinois at Urbana-Champaign
[♢]Pattern Recognition Center, WeChat AI, Tencent Inc.
yujiaqin16@gmail.com

## Abstract

Pre-trained Language Models (PLMs) have shown superior performance on various downstream Natural Language Processing (NLP) tasks. However, conventional pre-training objectives do not explicitly model relational facts in text, which are crucial for textual understanding. To address this issue, we propose a novel contrastive learning framework ERICA to obtain a deep understanding of the entities and their relations in text. Specifically, we define two novel pre-training tasks to better understand entities and relations: (1) the entity discrimination task to distinguish which tail entity can be inferred by the given head entity and relation; (2) the relation discrimination task to distinguish whether two relations are close or not semantically, which involves complex relational reasoning. Experimental results demonstrate that ERICA can improve typical PLMs (BERT and RoBERTa) on several language understanding tasks, including relation extraction, entity typing and question answering, especially under low-resource settings.[1]

## 1 Introduction

Pre-trained Language Models (PLMs) (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019) have shown superior performance on various Natural Language Processing (NLP) tasks such as text classification (Wang et al., 2018), named entity recognition (Sang and De Meulder, 2003), and question answering (Talmor and Berant, 2019). Benefiting from designing various effective self-supervised learning objectives, such as masked language modeling (Devlin et al., 2018), PLMs can effectively capture the syntax and semantics in text to generate informative language representations for downstream NLP tasks.

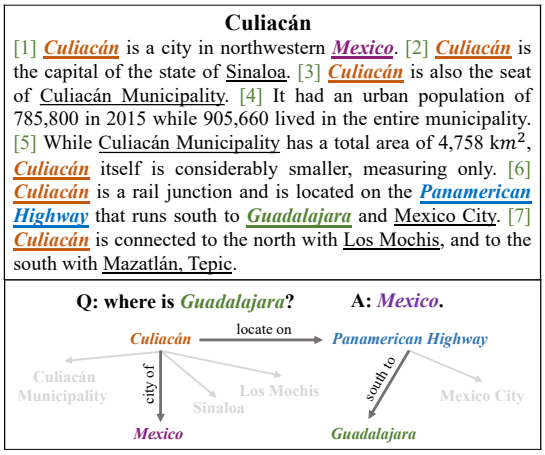


Figure 1: An example for a document "Culiacán", in which all entities are underlined. We show entities and their relations as a relational graph, and highlight the important entities and relations to find out "where is Guadalajara".

However, conventional pre-training objectives do not explicitly model relational facts, which frequently distribute in text and are crucial for understanding the whole text. To address this issue, some recent studies attempt to improve PLMs to better understand relations between entities (Soares et al., 2019; Peng et al., 2020). However, they mainly focus on within-sentence relations in isolation, ignoring the understanding of entities, and the interactions among multiple entities at document level, whose relation understanding involves complex reasoning patterns. According to the statistics on a human-annotated corpus sampled from Wikipedia documents by Yao et al. (2019), at least 40.7% relational facts require to be extracted from multiple sentences. Specifically, we show an example in Figure 1, to understand that "Guadalajara is located in Mexico", we need to consider the following clues jointly: (i) "Mexico" is the country of "Culiacán" from sentence 1; (ii) "Culiacán" is a rail junction lo-

*Corresponding author.

[1]Our code and data are publicly available at https://github.com/thuqinyj16/ERICA.

cated on "Panamerican Highway" from sentence 6; (iii) "Panamerican Highway" connects to "Guadalajara" from sentence 6. From the example, we can see that there are two main challenges to capture the in-text relational facts:

1. To understand an entity, we should consider its relations to other entities comprehensively. In the example, the entity "Culiacán", occurring in sentence 1, 2, 3, 5, 6 and 7, plays an important role in finding out the answer. To understand "Culiacán", we should consider all its connected entities and diverse relations among them.

2. To understand a relation, we should consider the complex reasoning patterns in text. For example, to understand the complex inference chain in the example, we need to perform multi-hop reasoning, i.e., inferring that "Panamerican Highway" is located in "Mexico" through the first two clues.

In this paper, we propose ERICA, a novel framework to improve PLMs' capability of **E**ntity and **R**elat**I**on understanding via **C**ontr**A**stive learning, aiming to better capture in-text relational facts by considering the interactions among entities and relations comprehensively. Specifically, we define two novel pre-training tasks: (1) the entity discrimination task to distinguish which tail entity can be inferred by the given head entity and relation. It improves the understanding of each entity via considering its relations to other entities in text; (2) the relation discrimination task to distinguish whether two relations are close or not semantically. Through constructing entity pairs with document-level distant supervision, it takes complex relational reasoning chains into consideration in an implicit way and thus improves relation understanding.

We conduct experiments on a suite of language understanding tasks, including relation extraction, entity typing and question answering. The experimental results show that ERICA improves the performance of typical PLMs (BERT and RoBERTa) and outperforms baselines, especially under low-resource settings, which demonstrates that ERICA effectively improves PLMs' entity and relation understanding and captures the in-text relational facts.

## 2 Related Work

Dai and Le (2015) and Howard and Ruder (2018) propose to pre-train universal language representations on unlabeled text, and perform task-specific fine-tuning. With the advance of computing power, PLMs such as OpenAI GPT (Radford et al., 2018), BERT (Devlin et al., 2018) and XLNet (Yang et al., 2019) based on deep Transformer (Vaswani et al., 2017) architecture demonstrate their superiority in various downstream NLP tasks. Since then, numerous PLM extensions have been proposed to further explore the impacts of various model architectures (Song et al., 2019; Raffel et al., 2020), larger model size (Raffel et al., 2020; Lan et al., 2020; Fedus et al., 2021), more pre-training corpora (Liu et al., 2019), etc., to obtain better general language understanding ability. Although achieving great success, these PLMs usually regard words as basic units in textual understanding, ignoring the informative entities and their relations, which are crucial for understanding the whole text.

To improve the entity and relation understanding of PLMs, a typical line of work is knowledge-guided PLM, which incorporates external knowledge such as Knowledge Graphs (KGs) into PLMs to enhance the entity and relation understanding. Some enforce PLMs to memorize information about real-world entities and propose novel pre-training objectives (Xiong et al., 2019; Wang et al., 2019; Sun et al., 2020; Yamada et al., 2020). Others modify the internal structures of PLMs to fuse both textual and KG's information (Zhang et al., 2019; Peters et al., 2019; Wang et al., 2020; He et al., 2020). Although knowledge-guided PLMs introduce extra factual knowledge in KGs, these methods ignore the intrinsic relational facts in text, making it hard to understand out-of-KG entities or knowledge in downstream tasks, let alone the errors and incompleteness of KGs. This verifies the necessity of teaching PLMs to understand relational facts from contexts.

Another line of work is to directly model entities or relations in text in pre-training stage to break the limitations of individual token representations. Some focus on obtaining better span representations, including entity mentions, via span-based pre-training (Sun et al., 2019; Joshi et al., 2020; Kong et al., 2020; Ye et al., 2020). Others learn to extract relation-aware semantics from text by comparing the sentences that share the same entity pair or distantly supervised relation in KGs (Soares et al., 2019; Peng et al., 2020). However, these methods only consider either individual entities or within-sentence relations, which limits the performance in dealing with multiple entities and relations at document level. In contrast, our ERICA considers the interactions among multiple entities

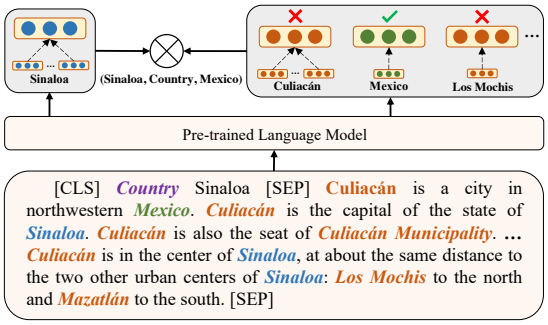


Figure 2: An example of Entity Discrimination task. For an entity pair with its distantly supervised relation in text, the ED task requires the ground-truth tail entity to be closer to the head entity than other entities.

and relations comprehensively, achieving a better understanding of in-text relational facts.

## 3 Methodology

In this section, we introduce the details of ERICA. We first describe the notations and how to represent entities and relations in documents. Then we detail the two novel pre-training tasks: Entity Discrimination (ED) task and Relation Discrimination (RD) task, followed by the overall training objective.

### 3.1 Notations

ERICA is trained on a large-scale unlabeled corpus leveraging the distant supervision from an external KG $\mathcal{K}$. Formally, let $\mathcal{D} = \{d_i\}_{i=1}^{|\mathcal{D}|}$ be a batch of documents and $\mathcal{E}_i = \{e_{ij}\}_{j=1}^{|\mathcal{E}_i|}$ be all named entities in $d_i$, where $e_{ij}$ is the $j$-th entity in $d_i$. For each document $d_i$, we enumerate all entity pairs $(e_{ij}, e_{ik})$ and link them to their corresponding relation $r_{jk}^i$ in $\mathcal{K}$ (if possible) and obtain a tuple set $\mathcal{T}_i = \{t_{jk}^i = (d_i, e_{ij}, r_{jk}^i, e_{ik}) | j \neq k\}$. We assign `no_relation` to those entity pairs without relation annotation in $\mathcal{K}$. Then we obtain the overall tuple set $\mathcal{T} = \mathcal{T}_1 \bigcup \mathcal{T}_2 \bigcup ... \bigcup \mathcal{T}_{|\mathcal{D}|}$ for this batch. The positive tuple set $\mathcal{T}^+$ is constructed by removing all tuples with `no_relation` from $\mathcal{T}$. Benefiting from document-level distant supervision, $\mathcal{T}^+$ includes both intra-sentence (relatively simple cases) and inter-sentence entity pairs (hard cases), whose relation understanding involves cross-sentence, multi-hop, or coreferential reasoning, i.e., $\mathcal{T}^+ = \mathcal{T}^+_{single} \bigcup \mathcal{T}^+_{cross}$.

### 3.2 Entity & Relation Representation

For each document $d_i$, we first use a PLM to encode it and obtain a series of hidden states $\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|d_i|}\}$, then we apply *mean pooling* operation over the consecutive tokens that mention $e_{ij}$ to obtain local entity representations. Note $e_{ij}$ may appear multiple times in $d_i$, the $k$-th occurrence of $e_{ij}$, which contains the tokens from index $n^k_{start}$ to $n^k_{end}$, is represented as:

$$\mathbf{m}^k_{e_{ij}} = \text{MeanPool}(\mathbf{h}_{n^k_{start}}, ..., \mathbf{h}_{n^k_{end}}). \quad (1)$$

To aggregate all information about $e_{ij}$, we average[2] all representations of each occurrence $\mathbf{m}^k_{e_{ij}}$ as the global entity representation $\mathbf{e}_{ij}$. Following Soares et al. (2019), we concatenate the final representations of two entities $e_{ij_1}$ and $e_{ij_2}$ as their relation representation, i.e., $\mathbf{r}^i_{j_1 j_2} = [\mathbf{e}_{ij_1}; \mathbf{e}_{ij_2}]$.

### 3.3 Entity Discrimination Task

Entity Discrimination (ED) task aims at inferring the tail entity in a document given a head entity and a relation. By distinguishing the ground-truth tail entity from other entities in the text, it teaches PLMs to understand an entity via considering its relations with other entities.

As shown in Figure 2, in practice, we first sample a tuple $t^i_{jk} = (d_i, e_{ij}, r^i_{jk}, e_{ik})$ from $\mathcal{T}^+$, PLMs are then asked to distinguish the ground-truth tail entity $e_{ik}$ from other entities in the document $d_i$. To inform PLMs of which head entity and relation to be conditioned on, we concatenate the relation name of $r^i_{jk}$, the mention of head entity $e_{ij}$ and a separation token `[SEP]` in front of $d_i$, i.e., $d^*_i$ ="`relation_name entity_mention [SEP]` $d_i$"[3]. The goal of entity discrimination task is equivalent to maximizing the posterior $\mathcal{P}(e_{ik}|e_{ij}, r^i_{jk}) = \text{softmax}(f(\mathbf{e}_{ik}))$ ($f(\cdot)$ indicates an entity classifier). However, we empirically find directly optimizing the posterior cannot well consider the relations among entities. Hence, we borrow the idea of contrastive learning (Hadsell et al., 2006) and push the representations of positive pair $(e_{ij}, e_{ik})$ closer than negative pairs, the loss function of ED task can be formulated as:

$$\mathcal{L}_{\text{ED}} = -\sum_{t^i_{jk} \in \mathcal{T}^+} \log \frac{\exp(\cos(\mathbf{e}_{ij}, \mathbf{e}_{ik})/\tau)}{\sum_{l=1, l\neq j}^{|\mathcal{E}_i|} \exp(\cos(\mathbf{e}_{ij}, \mathbf{e}_{il})/\tau)}, \quad (2)$$

[2] Although weighted summation by attention mechanism is an alternative, the specific method of entity information aggregation is not our main concern.

[3] Here we encode the modified document $d^*_i$ to obtain the entity representations. The newly added `entity_mention` is not considered for head entity representation.

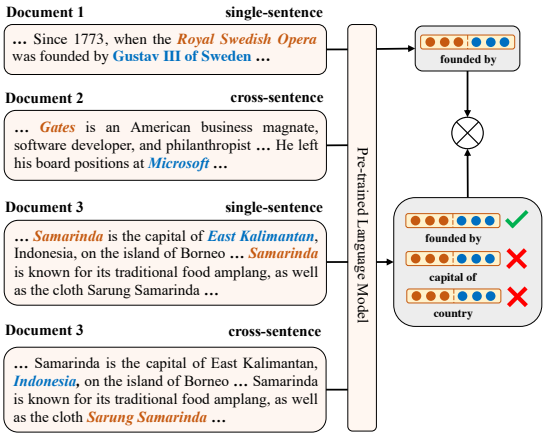


Figure 3: An example of Relation Discrimination task. For entity pairs belonging to the same relations, the RD task requires their relation representations to be closer.

where $\cos(\cdot,\cdot)$ denotes the cosine similarity between two entity representations and $\tau$ (temperature) is a hyper-parameter.

### 3.4 Relation Discrimination Task

Relation Discrimination (RD) task aims at distinguishing whether two relations are close or not semantically. Compared with existing relation-enhanced PLMs, we employ document-level rather than sentence-level distant supervision to further make PLMs comprehend the complex reasoning chains in real-world scenarios and thus improve PLMs' relation understanding.

As depicted in Figure 3, we train the text-based relation representations of the entity pairs that share the same relations to be closer in the semantic space. In practice, we linearly[4] sample a tuple pair $t_A = (d_A, e_{A_1}, r_A, e_{A_2})$ and $t_B = (d_B, e_{B_1}, r_B, e_{B_2})$ from $\mathcal{T}_s^+$ ($\mathcal{T}_{single}^+$) or $\mathcal{T}_c^+$ ($\mathcal{T}_{cross}^+$), where $r_A = r_B$. Using the method mentioned in Sec. 3.2, we obtain the positive relation representations $\mathbf{r}_{t_A}$ and $\mathbf{r}_{t_B}$ for $t_A$ and $t_B$. To discriminate positive examples from negative ones, similarly, we adopt contrastive learning and define the loss function of RD task as follows:

$$\mathcal{L}_{\text{RD}}^{\mathcal{T}_1,\mathcal{T}_2} = -\sum_{t_A\in\mathcal{T}_1, t_B\in\mathcal{T}_2} \log \frac{\exp(\cos(\mathbf{r}_{t_A},\mathbf{r}_{t_B})/\tau)}{\mathcal{Z}},$$

$$\mathcal{Z} = \sum_{t_C\in\mathcal{T}/\{t_A\}}^{N} \exp(\cos(\mathbf{r}_{t_A},\mathbf{r}_{t_C})/\tau),$$

$$\mathcal{L}_{\text{RD}} = \mathcal{L}_{\text{RD}}^{\mathcal{T}_s^+,\mathcal{T}_s^+} + \mathcal{L}_{\text{RD}}^{\mathcal{T}_s^+,\mathcal{T}_c^+} + \mathcal{L}_{\text{RD}}^{\mathcal{T}_c^+,\mathcal{T}_s^+} + \mathcal{L}_{\text{RD}}^{\mathcal{T}_c^+,\mathcal{T}_c^+}, \quad (3)$$

where $N$ is a hyper-parameter. We ensure $t_B$ is sampled in $\mathcal{Z}$ and construct $N-1$ negative examples by sampling $t_C$ ($r_A \neq r_C$) from $\mathcal{T}$, instead of $\mathcal{T}^+$[5]. By additionally considering the last three terms of $\mathcal{L}_{\text{RD}}$ in Eq.3, which require the model to distinguish complex inter-sentence relations with other relations in the text, our model could have better coverage and generality of the reasoning chains. PLMs are trained to perform reasoning in an implicit way to understand those "hard" inter-sentence cases.

### 3.5 Overall Objective

Now we present the overall training objective of ERICA. To avoid catastrophic forgetting (McCloskey and Cohen, 1989) of general language understanding ability, we train masked language modeling task ($\mathcal{L}_{\text{MLM}}$) together with ED and RD tasks. Hence, the overall learning objective is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{ED}} + \mathcal{L}_{\text{RD}} + \mathcal{L}_{\text{MLM}}. \quad (4)$$

It is worth mentioning that we also try to mask entities as suggested by Soares et al. (2019) and Peng et al. (2020), aiming to avoid simply relearning an entity linking system. However, we do not observe performance gain by such a masking strategy. We conjecture that in our document-level setting, it is hard for PLMs to overfit on memorizing entity mentions due to the better coverage and generality of document-level distant supervision. Besides, masking entities creates a gap between pre-training and fine-tuning, which may be a shortcoming of previous relation-enhanced PLMs.

## 4 Experiments

In this section, we first describe how we construct the distantly supervised dataset and pre-training details for ERICA. Then we introduce the experiments we conduct on several language understanding tasks, including relation extraction (RE), entity typing (ET) and question answering (QA). We test ERICA on two typical PLMs, including BERT and RoBERTa (denoted as ERICA$_{\texttt{BERT}}$ and ERICA$_{\texttt{RoBERTa}}$)[6]. We leave the training details

[4]The sampling rate of each relation is proportional to its total number in the current batch.

[5]In experiments, we find introducing `no_relation` entity pairs as negative samples further improves the performance and the reason is that increasing the diversity of training entity pairs is beneficial to PLMs.

[6]Since our main focus is to demonstrate the superiority of ERICA in improving PLMs to capture relational facts and advance further research explorations, we choose base models

for downstream tasks and experiments on GLUE benchmark (Wang et al., 2018) in the appendix.

### 4.1 Distantly Supervised Dataset Construction

Following Yao et al. (2019), we construct our pre-training dataset leveraging distant supervision from the English Wikipedia and Wikidata. First, we use spaCy[7] to perform *Named Entity Recognition*, and then link these entity mentions as well as Wikipedia's mentions with hyper-links to Wikidata items, thus we obtain the Wikidata ID for each entity. The relations between different entities are annotated distantly by querying Wikidata. We keep the documents containing at least 128 words, 4 entities and 4 relational triples. In addition, we ignore those entity pairs appearing in the test sets of RE and QA tasks to avoid test set leakage. In the end, we collect $1,000,000$ documents (about 1G storage) in total with more than $4,000$ relations annotated distantly. On average, each document contains 186.9 tokens, 12.9 entities and 7.2 relational triples, an entity appears 1.3 times per document. Based on the human evaluation on a random sample of the dataset, we find that it achieves an F1 score of 84.7% for named entity recognition, and an F1 score of 25.4% for relation extraction.

### 4.2 Pre-training Details

We initialize $\text{ERICA}_{\text{BERT}}$ and $\text{ERICA}_{\text{RoBERTa}}$ with *bert-base-uncased* and *roberta-base* checkpoints released by Google[8] and Huggingface[9]. We adopt AdamW (Loshchilov and Hutter, 2017) as the optimizer, warm up the learning rate for the first 20% steps and then linearly decay it. We set the learning rate to $3 \times 10^{-5}$, weight decay to $1 \times 10^{-5}$, batch size to $2,048$ and temperature $\tau$ to $5 \times 10^{-2}$. For $\mathcal{L}_{\text{RD}}$, we randomly select up to 64 negative samples per document. We train both models with 8 NVIDIA Tesla P40 GPUs for $2,500$ steps.

### 4.3 Relation Extraction

Relation extraction aims to extract the relation between two recognized entities from a pre-defined relation set. We conduct experiments on both document-level and sentence-level RE. We test three partitions of the training set (1%, 10% and 100%) and report results on test sets.

for experiments.

[7] https://spacy.io/

[8] https://github.com/google-research/bert

[9] https://github.com/huggingface/transformers

| Size | 1% | | 10% | | 100% | |
|---|---|---|---|---|---|---|
| **Metrics** | F1 | IgF1 | F1 | IgF1 | F1 | IgF1 |
| CNN | | - | | - | 42.3 | 40.3 |
| BILSTM | | - | | - | 51.1 | 50.3 |
| BERT | 30.4 | 28.9 | 47.1 | 44.9 | 56.8 | 54.5 |
| HINBERT | | - | | - | 55.6 | 53.7 |
| CorefBERT | 32.8 | 31.2 | 46.0 | 43.7 | 57.0 | 54.5 |
| SpanBERT | 32.2 | 30.4 | 46.4 | 44.5 | 57.3 | 55.0 |
| ERNIE | 26.7 | 25.5 | 46.7 | 44.2 | 56.6 | 54.2 |
| MTB | 29.0 | 27.6 | 46.1 | 44.1 | 56.9 | 54.3 |
| CP | 30.3 | 28.7 | 44.8 | 42.6 | 55.2 | 52.7 |
| $\text{ERICA}_{\text{BERT}}$ | **37.8** | **36.0** | **50.8** | **48.3** | **58.2** | **55.9** |
| RoBERTa | 35.3 | 33.5 | 48.0 | 45.9 | 58.5 | 56.1 |
| $\text{ERICA}_{\text{RoBERTa}}$ | **40.1** | **38.0** | **50.3** | **48.3** | **59.0** | **56.6** |

Table 1: Results on document-level RE (DocRED). We report micro F1 (F1) and micro ignore F1 (IgF1) on test set. IgF1 metric ignores the relational facts shared by the train and dev/test sets.

| Dataset | TACRED | | | SemEval | | |
|---|---|---|---|---|---|---|
| **Size** | 1% | 10% | 100% | 1% | 10% | 100% |
| BERT | 36.0 | 58.5 | 68.1 | 43.6 | 79.3 | 88.1 |
| MTB | 35.7 | 58.8 | 68.2 | 44.2 | 79.2 | 88.2 |
| CP | **37.1** | **60.6** | 68.1 | 40.3 | 80.0 | **88.5** |
| $\text{ERICA}_{\text{BERT}}$ | 36.5 | 59.7 | **68.5** | **47.9** | **80.1** | 88.0 |
| RoBERTa | 26.3 | 61.2 | 69.7 | 46.0 | 80.3 | 88.8 |
| $\text{ERICA}_{\text{RoBERTa}}$ | **40.0** | **61.9** | **69.8** | **46.3** | **80.4** | **89.2** |

Table 2: Results (test F1) on sentence-level RE (TACRED and SemEval-2010 Task8) on three splits (1%, 10% and 100%).

**Document-level RE** For document-level RE, we choose DocRED (Yao et al., 2019), which requires reading multiple sentences in a document and synthesizing all the information to identify the relation between two entities. We encode all entities in the same way as in pre-training phase. The relation representations are obtained by adding a bilinear layer on top of two entity representations. We choose the following baselines: (1) **CNN** (Zeng et al., 2014), **BILSTM** (Hochreiter and Schmidhuber, 1997), **BERT** (Devlin et al., 2018) and **RoBERTa** (Liu et al., 2019), which are widely used as text encoders for relation extraction tasks; (2) **HINBERT** (Tang et al., 2020) which employs a hierarchical inference network to leverage the abundant information from different sources; (3) **CorefBERT** (Ye et al., 2020) which proposes a pre-training method to help BERT capture the coreferential relations in context; (4) **SpanBERT** (Joshi et al., 2020) which masks

| Metrics | Macro F1 | Micro F1 |
|---|---|---|
| BERT | 75.50 | 72.68 |
| MTB | 76.37 | 72.94 |
| CP | 76.27 | 72.48 |
| ERNIE | 76.51 | 73.39 |
| ERICA$_{\text{BERT}}$ | **77.85** | **74.71** |
| RoBERTa | 79.24 | 76.38 |
| ERICA$_{\text{RoBERTa}}$ | **80.77** | **77.04** |

Table 3: Results on entity typing (FIGER). We report macro F1 and micro F1 on the test set.

and predicts contiguous random spans instead of random tokens; (5) **ERNIE** (Zhang et al., 2019) which incorporates KG information into BERT to enhance entity representations; (6) **MTB** (Soares et al., 2019) and **CP** (Peng et al., 2020) which introduce sentence-level relation contrastive learning for BERT via distant supervision. For fair comparison, we pre-train these baselines on our constructed pre-training data[10] based on the implementation released by Peng et al. (2020)[11]. From the results shown in Table 1, we can see that: (1) ERICA outperforms all baselines significantly on each supervised data size, which demonstrates that ERICA could better understand the relations among entities in the document via implicitly considering their complex reasoning patterns in the pre-training; (2) both **MTB** and **CP** achieve worse results than **BERT**, which means sentence-level pre-training, lacking consideration for complex reasoning patterns, hurts PLM's performance on document-level RE tasks to some extent; (3) ERICA outperforms baselines by a larger margin on smaller training sets, which means ERICA has gained pretty good document-level relation reasoning ability in contrastive learning, and thus obtains improvements more extensively under low-resource settings.

**Sentence-level RE** For sentence-level RE, we choose two widely used datasets: TACRED (Zhang et al., 2017) and SemEval-2010 Task 8 (Hendrickx et al., 2019). We insert extra marker tokens to indicate the head and tail entities in each sentence. For baselines, we compare ERICA with **BERT**, **RoBERTa**, **MTB** and **CP**. From the results shown in Table 2, we observe that ERICA achieves almost comparable results on sentence-level RE tasks with **CP**, which means document-level pre-training in ERICA does not impair PLMs' performance on sentence-level relation understanding.

[10] In practice, documents are split into sentences and we only keep within-sentence entity pairs.

[11] https://github.com/thunlp/RE-Context-or-Names

| Setting | Standard | | | Masked | | |
|---|---|---|---|---|---|---|
| Size | 1% | 10% | 100% | 1% | 10% | 100% |
| FastQA | | - | 27.2 | | - | 38.0 |
| BiDAF | | - | 49.7 | | - | 59.8 |
| BERT | 35.8 | 53.7 | 69.5 | 37.9 | 53.1 | 73.1 |
| CorefBERT | 38.1 | 54.4 | 68.8 | 39.0 | 53.5 | 70.7 |
| SpanBERT | 33.1 | 56.4 | **70.7** | 34.0 | 55.4 | 73.2 |
| MTB | 36.6 | 51.7 | 68.4 | 36.2 | 50.9 | 71.7 |
| CP | 34.6 | 50.4 | 67.4 | 34.1 | 47.1 | 69.4 |
| ERICA$_{\text{BERT}}$ | **46.5** | **57.8** | 69.7 | **40.2** | **58.1** | **73.9** |
| RoBERTa | 37.3 | 57.4 | 70.9 | 41.2 | 58.7 | 75.5 |
| ERICA$_{\text{RoBERTa}}$ | **47.4** | **58.8** | **71.2** | **46.8** | **63.4** | **76.6** |

Table 4: Results (accuracy) on the dev set of WikiHop. We test both the standard and masked settings on three splits (1%, 10% and 100%).

| Setting | SQuAD | | TriviaQA | | NaturalQA | |
|---|---|---|---|---|---|---|
| Size | 10% | 100% | 10% | 100% | 10% | 100% |
| BERT | 79.7 | **88.9** | 60.8 | 70.7 | 68.4 | 78.4 |
| MTB | 63.5 | 87.1 | 52.0 | 67.8 | 61.2 | 76.7 |
| CP | 69.0 | 87.1 | 52.9 | 68.1 | 63.3 | 77.3 |
| ERICA$_{\text{BERT}}$ | **81.8** | 88.9 | **63.5** | **71.9** | **70.2** | **79.1** |
| RoBERTa | 82.9 | **90.5** | 63.6 | 72.0 | 71.8 | 80.0 |
| ERICA$_{\text{RoBERTa}}$ | **85.0** | 90.4 | **63.6** | **72.1** | **73.7** | **80.5** |

Table 5: Results (F1) on extractive QA (SQuAD, TriviaQA and NaturalQA) on two splits (10% and 100%). Results on 1% split are left in the appendix.

### 4.4 Entity Typing

Entity typing aims at classifying entity mentions into pre-defined entity types. We choose FIGER (Ling et al., 2015), which is a sentence-level entity typing dataset labeled with distant supervision. **BERT**, **RoBERTa**, **MTB**, **CP** and **ERNIE** are chosen as baselines. From the results listed in Table 3, we observe that, ERICA outperforms all baselines, which demonstrates that ERICA could better represent entities and distinguish them in text via both entity-level and relation-level contrastive learning.

### 4.5 Question Answering

Question answering aims to extract a specific answer span in text given a question. We conduct experiments on both multi-choice and extractive QA. We test multiple partitions of the training set.

**Multi-choice QA** For Multi-choice QA, we choose WikiHop (Welbl et al., 2018), which requires models to answer specific properties of an

entity after reading multiple documents and conducting multi-hop reasoning. It has both standard and masked settings, where the latter setting masks all entities with random IDs to avoid information leakage. We first concatenate the question and documents into a long sequence, then we find all the occurrences of an entity in the documents, encode them into hidden representations and obtain the global entity representation by applying mean pooling on these hidden representations. Finally, we use a classifier on top of the entity representation for prediction. We choose the following baselines: (1) **FastQA** (Weissenborn et al., 2017) and **BiDAF** (Seo et al., 2016), which are widely used question answering systems; (2) **BERT**, **RoBERTa**, **CorefBERT**, **SpanBERT**, **MTB** and **CP**, which are introduced in previous sections. From the results listed in Table 4, we observe that ERICA outperforms baselines in both settings, indicating that ERICA can better understand entities and their relations in the documents and extract the true answer according to queries. The significant improvements in the masked setting also indicate that ERICA can better perform multi-hop reasoning to synthesize and analyze information from contexts, instead of relying on entity mention "shortcuts" (Jiang and Bansal, 2019).

**Extractive QA** For extractive QA, we adopt three widely-used datasets: SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017) and NaturalQA (Kwiatkowski et al., 2019) in MRQA (Fisch et al., 2019) to evaluate ERICA in various domains. Since MRQA does not provide the test set for each dataset, we randomly split the original dev set into two halves and obtain the new dev/test set. We follow the QA setting of BERT (Devlin et al., 2018): we concatenate the given question and passage into one long sequence, encode the sequence by PLMs and adopt two classifiers to predict the start and end index of the answer. We choose **BERT**, **RoBERTa**, **MTB** and **CP** as baselines. From the results listed in Table 5, we observe that ERICA outperforms all baselines, indicating that through the enhancement of entity and relation understanding, ERICA is more capable of capturing in-text relational facts and synthesizing information of entities. This ability further improves PLMs for question answering.

## 5 Analysis

In this section, we first conduct a suite of ablation studies to explore how $\mathcal{L}_{\text{ED}}$ and $\mathcal{L}_{\text{RD}}$ contribute to ERICA. Then we give a thorough analysis on how pre-training data's domain / size and methods for entity encoding impact the performance. Lastly, we visualize the entity and relation embeddings learned by ERICA.

| **Dataset** | DocRED | FIGER | WikiHop |
|---|---|---|---|
| BERT | 44.9 | 72.7 | 53.1 |
| -NSP | 45.2 | 72.6 | 53.6 |
| -NSP+$\mathcal{L}_{\text{ED}}$ | 47.6 | 73.8 | **59.8** |
| -NSP+$\mathcal{L}_{\text{RD}}^{\mathcal{T}_c^+,\mathcal{T}_c^+}$ | 46.4 | 72.6 | 52.2 |
| -NSP+$\mathcal{L}_{\text{RD}}^{\mathcal{T}_s^+,\mathcal{T}_s^+}$ | 47.3 | 73.5 | 51.2 |
| -NSP+$\mathcal{L}_{\text{RD}}$ | 48.0 | 74.0 | 52.0 |
| ERICA$_{\text{BERT}}$ | **48.3** | **74.7** | 58.1 |

Table 6: Ablation study. We report test IgF1 on DocRED (10%), test micro F1 on FIGER and dev accuracy on the masked setting of WikiHop (10%).

### 5.1 Ablation Study

To demonstrate that the superior performance of ERICA is not owing to its longer pretraining (2500 steps) on masked language modeling, we include a baseline by optimizing $\mathcal{L}_{\text{MLM}}$ only (removing the Next Sentence Prediction (-NSP) loss (Devlin et al., 2018)). In addition, to explore how $\mathcal{L}_{\text{ED}}$ and $\mathcal{L}_{\text{RD}}$ impact the performance, we keep only one of these two losses and compare the results. Lastly, to evaluate how intra-sentence and inter-sentence entity pairs contribute to RD task, we compare the performances of only sampling intra-sentence entity pairs ($\mathcal{L}_{\text{RD}}^{\mathcal{T}_s^+,\mathcal{T}_s^+}$) or inter-sentence entity pairs ($\mathcal{L}_{\text{RD}}^{\mathcal{T}_c^+,\mathcal{T}_c^+}$), and sampling both of them ($\mathcal{L}_{\text{RD}}$) during pre-training. We conduct experiments on DocRED, WikiHop (masked version) and FIGER. For DocRED and WikiHop, we show the results on 10% splits and the full results are left in the appendix.

From the results shown in Table 6, we can see that: (1) extra pretraining (-NSP) only contributes a little to the overall improvement. (2) For DocRED and FIGER, either $\mathcal{L}_{\text{ED}}$ or $\mathcal{L}_{\text{RD}}$ is beneficial, and combining them further improves the performance; For WikiHop, $\mathcal{L}_{\text{ED}}$ dominates the improvement while $\mathcal{L}_{\text{RD}}$ hurts the performance slightly, this is possibly because question answering more resembles the tail entity discrimination process, while the relation discrimination process may have conflicts with it. (3) For $\mathcal{L}_{\text{RD}}$, both intra-sentence and inter-sentence entity pairs contribute, which demonstrates that incorporating both of them is necessary for PLMs to understand relations between entities in text comprehensively. We also found empiri-

| Size | 1% | 10% | 100% |
|---|---|---|---|
| BERT | 28.9 | 44.9 | 54.5 |
| $\text{ERICA}_{\text{BERT}}$ | 36.0 | 48.3 | 55.9 |
| $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$ | **36.3** | **48.6** | **55.9** |

Table 7: Effects of pre-training data's entity distribution shifting. We report test IgF1 on DocRED.

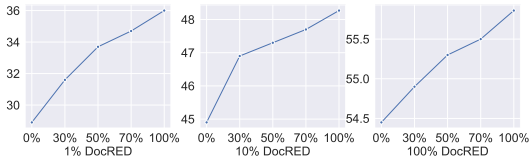


Figure 4: Impacts of relation distribution shifting. X axis denotes different ratios of relations, Y axis denotes test IgF1 on different partitions of DocRED.

cally that when these two auxiliary objectives are only added into the fine-tuning stage, the model does not have performance gain. The reason is that the size and diversity of entities and relations in downstream training data are limited. Instead, pre-training with distant supervision on a large corpus provides a solution for increasing the diversity and quantity of training examples.

### 5.2 Effects of Domain Shifting

We investigate two domain shifting factors: entity distribution and relation distribution, to explore how they impact ERICA's performance.

**Entity Distribution Shifting** The entities in supervised datasets of DocRED are recognized by human annotators while our pre-training data is processed by spaCy. Hence there may exist an entity distribution gap between pre-training and fine-tuning. To study the impacts of entity distribution shifting, we fine-tune a BERT model on training set of DocRED for NER tagging and re-tag entities in our pre-training dataset. Then we pre-train ERICA on the newly-labeled training corpus (denoted as $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$). From the results shown in Table 7, we observe that it performs better than the original ERICA, indicating that pre-training on a dataset that shares similar entity distributions with downstream tasks is beneficial.

**Relation Distribution Shifting** Our pre-training data contains over $4,000$ Wikidata relations. To investigate whether training on a more diverse relation domain benefits ERICA, we train it with the pre-training corpus that randomly keeps only 30%, 50% and 70% the original relations, and compare

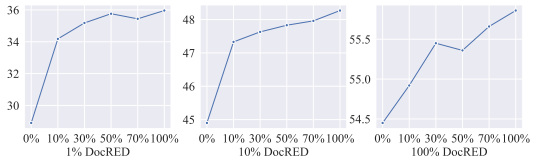


Figure 5: Impacts of pre-training data's size. X axis denotes different ratios of pre-training data, Y axis denotes test IgF1 on different partitions of DocRED.

| Size | 1% | | 10% | | 100% | |
|---|---|---|---|---|---|---|
| **Metrics** | F1 | IgF1 | F1 | IgF1 | F1 | IgF1 |
| **Mean Pool** | | | | | | |
| BERT | 30.4 | 28.9 | 47.1 | 44.9 | 56.8 | 54.5 |
| $\text{ERICA}_{\text{BERT}}$ | 37.8 | 36.0 | 50.8 | 48.3 | 58.2 | 55.9 |
| $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$ | **38.5** | **36.3** | **51.0** | **48.6** | **58.2** | **55.9** |
| **Entity Marker** | | | | | | |
| BERT | 23.0 | 21.8 | 46.5 | 44.3 | 58.0 | 55.6 |
| $\text{ERICA}_{\text{BERT}}$ | 34.9 | 33.0 | 50.2 | 48.0 | 59.9 | 57.6 |
| $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$ | **36.9** | **34.8** | **52.5** | **50.3** | **60.8** | **58.4** |

Table 8: Results (IgF1) on how entity encoding strategy influences ERICA's performance on DocRED. We also show the impacts of entity distribution shifting ($\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$ and $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$) as is mentioned in the main paper.

their performances. From the results in Figure 4, we observe that the performance of ERICA improves constantly as the diversity of relation domain increases, which reveals the importance of using diverse training data on relation-related tasks. Through detailed analysis, we further find that ERICA is less competent at handling unseen relations in the corpus. This may result from the construction of our pre-training dataset: all the relations are annotated distantly through an existing KG with a pre-defined relation set. It would be promising to introduce more diverse relation domains during data preparation in future.

### 5.3 Effects of Pre-training Data's Size

To explore the effects of pre-training data's size, we train ERICA on 10%, 30%, 50% and 70% of the original pre-training dataset, respectively. We report the results in Figure 5, from which we observe that with the scale of pre-training data becoming larger, ERICA is performing better.

### 5.4 Effects of Methods for Entity Encoding

For all the experiments mentioned above, we encode each occurrence of an entity by mean pooling over all its tokens in both pre-training and downstream tasks. Ideally, ERICA should have consis-

tent improvements on other kinds of methods for entity encoding. To demonstrate this, we try another entity encoding method mentioned by Soares et al. (2019) on three splits of DocRED (1%, 10% and 100%). Specifically, we insert a special start token `[S]` in front of an entity and an end token `[E]` after it. The representation for this entity is calculated by averaging the representations of all its start tokens in the document. To help PLMs discriminate different entities, we randomly assign different marker pairs (`[S1]`, `[E1]`; `[S2]`, `[E2]`, ...) for each entity in a document in both pre-training and downstream tasks[12]. All occurrences of one entity in a document share the same marker pair. We show in Table 8 that ERICA achieves consistent performance improvements for both methods (denoted as **Mean Pool** and **Entity Marker**), indicating that ERICA is applicable to different methods for entity encoding. Specifically, **Entity Marker** achieves better performance when the scale of training data is large while **Mean Pool** is more powerful under low-resource settings. We also notice that training on a dataset that shares similar entity distributions is more helpful for **Mean Pool**, where $\text{ERICA}_{\text{BERT}}^{\text{DocRED}}$ achieves 60.8 (F1) and 58.4 (IgF1) on 100% training data.

### 5.5 Embedding Visualization

In Figure 6, we show the learned entity and relation embeddings of BERT and $\text{ERICA}_{\text{BERT}}$ on DocRED's dev set by t-distributed stochastic neighbor embedding (t-SNE) (Hinton and Roweis, 2002). We label points with different colors to represent its corresponding category of entities or relations[13] in Wikidata and only visualize the most frequent 10 relations. From the figure, we can see that jointly training $\mathcal{L}_{\text{MLM}}$ with $\mathcal{L}_{\text{ED}}$ and $\mathcal{L}_{\text{RD}}$ leads to a more compact clustering of both entities and relations belonging to the same category. In contrast, only training $\mathcal{L}_{\text{MLM}}$ exhibits random distribution. This verifies that ERICA could better understand and represent both entities and relations in the text.

[12]In practice, we randomly initialize 100 entity marker pairs.

[13](Key, value) pairs for relations defined in Wikidata are: (P176, manufacturer); (P150, contains administrative territorial entity); (P17, country); (P131, located in the administrative territorial entity); (P175, performer); (P27, country of citizenship); (P569, date of birth); (P1001, applies to jurisdiction); (P57, director); (P179, part of the series).

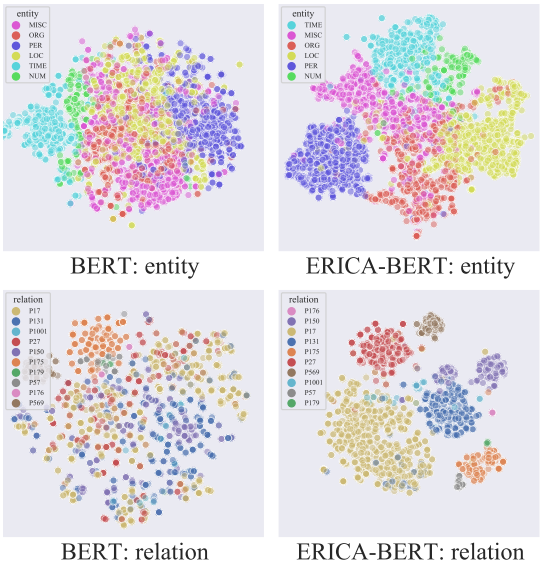


Figure 6: t-SNE plots of learned entity and relation embeddings on DocRED comparing BERT and $\text{ERICA}_{\text{BERT}}$.

## 6 Conclusions

In this paper, we present ERICA, a general framework for PLMs to improve entity and relation understanding via contrastive learning. We demonstrate the effectiveness of our method on several language understanding tasks, including relation extraction, entity typing and question answering. The experimental results show that ERICA outperforms all baselines, especially under low-resource settings, which means ERICA helps PLMs better capture the in-text relational facts and synthesize information about entities and their relations.

## Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106501) and Beijing Academy of Artificial Intelligence (BAAI). This work is also supported by the Pattern Recognition Center, WeChat AI, Tencent Inc.

## References

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

*for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. *CoRR*, abs/1909.07755.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 94–99.

Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. In *Advances in neural information processing systems 15: 16th Annual Conference on Neural Information Processing Systems 2002. Proceedings of a meeting held September 12, 2002, Vancouver, British Columbia, Canada*, volume 15, pages 857–864.

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

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019, July 28, 2019, Florence, Italy*, pages 2726–2736. Association for Computational Linguistics.

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.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7, 2015, Conference Track Proceedings*.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *Proceedings of 8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, April 26, 2020, Conference Track Proceedings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of 8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, April 26, 2020, Conference Track Proceedings*.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *Proceedings of 7th International Conference on Learning Representations, ICLR 2019*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: the sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *Proceedings of 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24, 2017, Con- ference Track Proceedings*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *Proceedings of International Conference on Machine Learning*, pages 5926–5936. PMLR.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921. Association for Computational Linguistics.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. Hin: Hierarchical inference network for document-level relation extraction. In *Advances in Knowledge Discovery and Data Mining-24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11, 2020, Proceedings, Part I, volume 12084 of Lecture Notes in Computer Science*, pages 197–209. Springer.

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 30: Annual Conference on Neural Information Processing Systems 2017, 4 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Net-*

*works for NLP1*. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *Proceedings of 8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, April 26, 2020, Conference Track Proceedings.*

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 764–777.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451. Association for Computational Linguistics.

# Appendices

## A Training Details for Downstream Tasks

In this section, we introduce the training details for downstream tasks (relation extraction, entity typing and question answering). We implement all models based on Huggingface transformers[14].

### A.1 Relation Extraction

**Document-level Relation Extraction** For document-level relation extraction, we did experiments on DocRED (Yao et al., 2019). We modify the official code[15] for implementation. For experiments on three partitions of the original training set (1%, 10% and 100%), we adopt batch size of $10, 32, 32$ and training epochs of $400, 400, 200$, respectively. We choose Adam optimizer (Kingma and Ba, 2014) as the optimizer and the learning rate is set to $4 \times 10^{-5}$. We evaluate on dev set every $20/20/5$ epochs and then test the best checkpoint on test set on the official evaluation server[16].

**Sentence-level Relation Extraction** For sentence-level relation extraction, we did experiments on TACRED (Zhang et al., 2017) and SemEval-2010 Task 8 (Hendrickx et al., 2019) based on the implementation of Peng et al. (2020)[17]. We did experiments on three partitions (1%, 10% and 100%) of the original training set. The relation representation for each entity pair is obtained in the same way as in pre-training phase. Other settings are kept the same as Peng et al. (2020) for fair comparison.

### A.2 Entity Tying

For entity typing, we choose FIGER (Ling et al., 2015), whose training set is labeled with distant supervision. We modify the implementation of ERNIE (Zhang et al., 2019)[18]. In fine-tuning phrase, we encode the entities in the same way as in pre-training phase. We set the learning rate to $3 \times 10^{-5}$ and batch size to 256, and fine-tune the models for three epochs, other hyper-parameters are kept the same as ERNIE.

### A.3 Question Answering

**Multi-choice QA** For multi-choice question answering, we choose WikiHop (Welbl et al., 2018). Since the standard setting of WikiHop does not provide the index for each candidate, we then find them by exactly matching them in the documents. We did experiments on three partitions of the original training data (1%, 10% and 100%). We set the batch size to 8 and learning rate to $5 \times 10^{-5}$, and train for two epochs.

**Extractive QA** For extractive question answering, we adopt MRQA (Fisch et al., 2019) as the testbed and choose three datasets: SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017) and NaturalQA (Kwiatkowski et al., 2019). We adopt Adam as the optimizer, set the learning rate to $3 \times 10^{-5}$ and train for two epochs. In the main paper, we report results on two splits (10% and 100%) and results on 1% are listed in Table 11.

## B Generalized Language Understanding (GLUE)

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) provides several natural language understanding tasks, which is often used to evaluate PLMs. To test whether $\mathcal{L}_{\text{ED}}$ and $\mathcal{L}_{\text{RD}}$ impair the PLMs' performance on these tasks, we compare BERT, $\text{ERICA}_{\texttt{BERT}}$, RoBERTa and $\text{ERICA}_{\texttt{RoBERTa}}$. We follow the widely used setting and use the `[CLS]` token as representation for the whole sentence or sentence pair for classification or regression. Table 9 shows the results on dev sets of GLUE Benchmark. It can be observed that both $\text{ERICA}_{\texttt{BERT}}$ and $\text{ERICA}_{\texttt{RoBERTa}}$ achieve comparable performance than the original model, which suggests that jointly training $\mathcal{L}_{\text{ED}}$ and $\mathcal{L}_{\text{RD}}$ with $\mathcal{L}_{\text{MLM}}$ does not hurt PLMs' general ability of language understanding.

## C Full results of ablation study

Full results of ablation study (DocRED, WikiHop and FIGER) are listed in Table 10.

## D Joint Named Entity Recognition and Relation Extraction

Joint Named Entity Recognition (NER) and Relation Extraction (RE) aims at identifying entities in text and the relations between them. We

[14] https://github.com/huggingface/transformers

[15] https://github.com/thunlp/DocRED

[16] https://competitions.codalab.org/competitions/20717

[17] https://github.com/thunlp/RE-Context-or-Names

[18] https://github.com/thunlp/ERNIE

| Dataset | MNLI(m/mm) | QQP | QNLI | SST-2 | CoLA | STS-B | MRPC | RTE |
|---|---|---|---|---|---|---|---|---|
| BERT | 84.0/84.4 | 88.9 | 90.6 | 92.4 | 57.2 | 89.7 | 89.4 | 70.1 |
| ERICA$_{\text{BERT}}$ | 84.5/84.7 | 88.3 | 90.7 | 92.8 | 57.9 | 89.5 | 89.5 | 69.6 |
| RoBERTa | 87.5/87.3 | 91.9 | 92.8 | 94.8 | 63.6 | 91.2 | 90.2 | 78.7 |
| ERICA$_{\text{RoBERTa}}$ | 87.5/87.5 | 91.6 | 92.6 | 95.0 | 63.5 | 90.7 | 91.5 | 78.5 |

Table 9: Results on dev sets of GLUE Benchmark. We report matched/mismatched (m/mm) accuracy for MNLI, F1 score for QQP and MRPC, spearman correlation for STS-B and accuracy for other tasks.

| Dataset | DocRED | | | WikiHop (m) | | | FIGER |
|---|---|---|---|---|---|---|---|
| Size | 1% | 10% | 100% | 1% | 10% | 100% | 100% |
| BERT | 28.9 | 44.9 | 54.5 | 37.9 | 53.1 | 73.1 | 72.7 |
| -NSP | 30.1 | 45.2 | 54.6 | 38.2 | 53.6 | 73.3 | 72.6 |
| -NSP+$\mathcal{L}_{\text{ED}}$ | 34.4 | 47.6 | 55.8 | **41.1** | **59.8** | **74.8** | 73.8 |
| -NSP+$\mathcal{L}_{\text{RD}}^{\mathcal{T}_c^+,\mathcal{T}_c^+}$ | 34.8 | 46.4 | 54.7 | 37.4 | 52.2 | 72.8 | 72.6 |
| -NSP+$\mathcal{L}_{\text{RD}}^{\mathcal{T}_s^+,\mathcal{T}_s^+}$ | 33.9 | 47.3 | 55.5 | 38.0 | 51.2 | 72.5 | 73.5 |
| -NSP+$\mathcal{L}_{\text{RD}}$ | 35.9 | 48.0 | 55.6 | 37.2 | 52.0 | 72.7 | 74.0 |
| ERICA$_{\text{BERT}}$ | **36.0** | **48.3** | **55.9** | 40.2 | 58.1 | 73.9 | **74.7** |

Table 10: Full results of ablation study. We report test IgF1 on DocRED, dev accuracy on the masked (m) setting of WikiHop and test micro F1 on FIGER.

| Setting | SQuAD | TriviaQA | NaturalQA |
|---|---|---|---|
| BERT | 15.8 | 28.7 | 31.5 |
| MTB | 11.2 | 22.0 | 28.4 |
| CP | 12.5 | 25.6 | 29.4 |
| ERICA$_{\text{BERT}}$ | **51.3** | **51.4** | **42.9** |
| RoBERTa | 22.1 | 40.6 | 34.0 |
| ERICA$_{\text{RoBERTa}}$ | **57.6** | **51.3** | **57.6** |

Table 11: Results (F1) on extractive QA (SQuAD, TriviaQA and NaturalQA) on 1% split.

| Model | CoNLL04 | | ADE | |
|---|---|---|---|---|
| | NER | RE | NER | RE |
| BERT | 88.5 | 70.3 | 89.2 | 79.2 |
| ERICA$_{\text{BERT}}$ | **89.3** | **71.5** | **89.5** | **80.2** |
| RoBERTa | 89.8 | 72.0 | 89.7 | 81.6 |
| ERICA$_{\text{RoBERTa}}$ | **90.0** | **72.8** | **90.2** | **82.4** |

Table 12: Results (F1) on joint NER&RE.

adopt SpERT (Eberts and Ulges, 2019) as the base model and conduct experiments on two datasets: CoNLL04 (Roth and Yih, 2004) and ADE (Gurulingappa et al., 2012) by replacing the base encoders (BERT and RoBERTa) with ERICA$_{\text{BERT}}$ and ERICA$_{\text{RoBERTa}}$, respectively. We modify the implementation of SpERT[19] and keep all the settings the same. From the results listed in Table 12, we can see that ERICA outperforms all baselines, which again demonstrates the superiority of ERICA in helping PLMs better understand and represent both entities and relations in text.

[19] https://github.com/markus-eberts/spert