KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction

Xiang Chen[1,2*], **Ningyu Zhang**[1,2 *], **Xin Xie**[1,2 *], **Shumin Deng**[1,2], **Yunzhi Yao**[1,2],
Chuanqi Tan[3], **Fei Huang**[3], **Luo Si**[3], **Huajun Chen**[1,2†]

[1] Zhejiang University & AZFT Joint Lab for Knowledge Engine
[2] Hangzhou Innovation Center, Zhejiang University [3] Alibaba Group
{xiang_chen,zhangningyu,xx2020,231sm,yyztodd,huajunsir}@zju.edu.cn
{chuanqi.tcq,f.huang,luo.si}@alibaba-inc.com

Abstract

Recently, prompt-tuning has achieved promising results for certain few-shot classification tasks. The core idea of prompt-tuning is to insert text pieces (i.e., templates) into the input and transform a classification task into a masked language modeling problem. However, for relation extraction, determining an appropriate prompt template requires domain expertise, and it is cumbersome and time-consuming to obtain a suitable label word. Furthermore, there exist abundant semantic knowledge among the entities and relations that cannot be ignored. To this end, we focus on incorporating knowledge into prompt-tuning for relation extraction and propose a **Know**ledge-aware **Prompt**-tuning approach with synergistic optimization (**KnowPrompt**). Specifically, we inject entity and relation knowledge into prompt construction with learnable virtual template words as well as answer words and synergistically optimize their representation with knowledge constraints. Extensive experimental results on five datasets with standard and low-resource settings demonstrate the effectiveness of our approach.

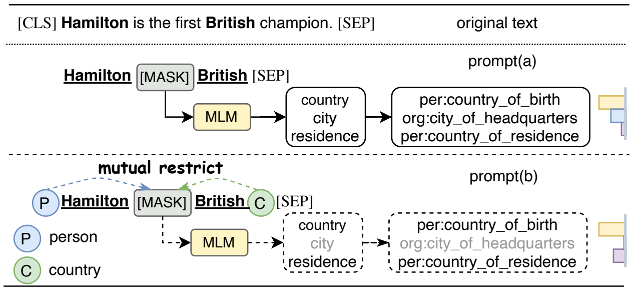

Figure 1: Examples of prompt-tuning for RE.

1 Introduction

Relation Extraction (RE) is an important task in information extraction. RE appeals to many researchers (Zhang et al. 2017, 2018; Baumgartner et al. 2018; Zhang et al. 2019, 2020; Nan et al. 2020; Wu et al. 2021) due to the capability to extract textual information and benefit many NLP applications, e.g., information retrieval, dialog generation, and question answering.

Previous self-supervised pre-trained language models (PLMs) such as BERT (Devlin et al. 2019), which can learn powerfully contextualized representations, have achieved state-of-the-art (SOTA) results in lots of RE benchmarks. However, since fine-tuning requires adding extra classifiers on top of PLMs and further training the models under classification objectives, their performance heavily depends on time-consuming and labor-intensive annotated data, making it hard to generalize well. Recently, a series of studies using prompt-tuning (Schick and Schütze 2021, 2020) has arisen to address this issue: adopting the pre-trained LM directly as a predictor by completing a cloze task to bridge the gap between pre-training and fine-tuning. Prompt-tuning fuses the original input with the prompt template to predict `[MASK]` and then maps the predicted label words to the corresponding class sets, which has induced better performances for PLMs on few-shot tasks. As shown in Figure 1, in prompt-learning for RE, a typical prompt consists of a template (e.g. "<S_1> Hamilton `[MASK]` British") and a set of label words ("country", "city", "residence", etc.) as candidates to predict `[MASK]`. PLMs predict ("country", "city", "residence", etc.) at the masked position to determine the relation between "Hamilton" and "British" given <S_1> to be ("per:country_of_birth", "org:city_of_headquarters", "per:country_of_residence", etc.) . In a nutshell, RE with prompt-tuning involves template engineering and answer engineering, which aims to search for the best template and an answer space (Liu et al. 2021a).

However, there are still several non-trivial challenges for RE with prompt-tuning as follows: On one hand, determining the appropriate prompt template for RE requires domain expertise, and auto-constructing a high-performing prompt with input entities often requires additional computation cost for generation and verification (Schick, Schmid, and Schütze 2020; Schick and Schütze 2021; Shin et al. 2020; Gao, Fisch, and Chen 2021). On the other hand, the computational complexity of the label word search process is very high (e.g., usually exponential depending on the number of categories), and it is non-trivial to obtain a suitable

[*] Equal contribution and shared co-first authorship.

[†] Corresponding author.

target label word in the vocabulary when the length of the relation label varies. For example, the relation labels of $per : country\_of\_birth$ and $org : city\_of\_headquarters$ cannot specify a suitable single label word in the vocabulary. In addition, there exists rich semantic knowledge among entity types, relation labels, and structural knowledge implications among relational triples, which cannot be ignored. For example, as shown in Figure 1, if one pair of entities contains types of "person" and "country", there is a low probability that it will contain the relation "org:city_of_headquarters". Conversely, the relation also restricts the entity types of its subject and object entity. Previous studies (Li et al. 2019; Distiawan et al. 2019; Bastos et al. 2021) indicate that incorporating the relational knowledge will provide evidence for the prediction of relations.

To address those issues, we take the first step to inject knowledge into learnable continuous prompts and propose a novel **Know**ledge-aware **Prompt**-tuning with synergistic optimization (**KnowPrompt**) approach for RE. We construct prompt with knowledge injection via learnable virtual template words and virtual answer words to alleviate labor-intensive prompt engineering (§4.1). To be specific, we utilize TYPED MARKER around entities initialized with aggregated entity-type embeddings as learnable virtual template words to inject entity type knowledge. We further leverage the average embeddings of each token in relation labels as virtual answer words to inject relation knowledge. Since there exist implicit structural constraints among entities and relations, and virtual words should be consistent with the surrounding contexts, we introduce synergistic optimization to obtain optimized virtual templates and answer words (§4.2). Concretely, we propose a context-aware prompt calibration method with implicit structural constraints to inject structural knowledge implications among relational triples and associate prompt embeddings with each other. We conclude our contributions as follows:

- We propose a novel knowledge-aware prompt-tuning (KnowPrompt) approach for RE that injects knowledge into prompt template design and answer construction, so as to encode the rich semantic knowledge among entity types and relations.
- We propose jointly optimizing the representation of a virtual prompt template and answer words with knowledge constraints. To the best of our knowledge, it is the first approach to jointly optimize the prompt template and answer words in continuous space.
- Extensive experiments on five RE benchmark datasets (both in general and dialogue domain) illustrate the effectiveness of KnowPrompt in both standard and low-resource settings.

2 Related Work

2.1 Relation Extraction

Relation extraction is a critical task in information extraction. Early approaches involve pattern-based methods (Huffman 1995; Califf and Mooney 1999), CNN/RNN-based (Zeng et al. 2015; Zhou et al. 2016; Zhang et al. 2017) and graph-based methods (Zhang, Qi, and Manning 2018; Guo, Zhang, and Lu 2019; Guo et al. 2020). With the recent advances in pre-trained language models (Devlin et al. 2019), applying PLMs as the backbone of RE systems (Lin et al. 2020; Wang et al. 2020; Li et al. 2020; Zhang et al. 2021b; Zheng et al. 2021; Ye et al. 2021; Zhang et al. 2021a) has become standard procedure. Several studies have shown that BERT-based models significantly outperform both RNN and graph-based models (Wu and He 2019; Joshi et al. 2020; Yu et al. 2020a). Recently, Xue et al. (2021) propose a multi-view graph based on BERT, achieving SOTA performance both on TACRED-Revisit (Alt, Gabryszak, and Hennig 2020) and DialogRE (Yu et al. 2020a).

Since available annotated instances may be limited in practice, some previous studies have focused on the few-shot setting. Han et al. (2018); Gao et al. (2019, 2020); Qu et al. (2020); Yu et al. (2020b); Dong et al. (2020) propose approaches for few-shot RE based on meta-learning or metric learning, with the aim of developing models that can be trained with only a few labeled sentences and nonetheless generalize well. In contrast to previous N-way K-shot approaches, Gao, Fisch, and Chen (2021) utilize a setting that is relatively practical both for acquiring a few annotations (e.g., 16 examples per class) and efficiently training.

2.2 Prompt-tuning

Prompt-tuning methods are fueled by the birth of GPT-3 (Brown et al. 2020) and have achieved outstanding performance in widespread NLP tasks. With appropriate manual prompts, series of studies (Liu et al. 2021a; Ben-David, Oved, and Reichart 2021; Lester, Al-Rfou, and Constant 2021; Scao and Rush 2021; Reynolds and McDonell 2021; Lu et al. 2021) have been proposed, demonstrating the advancement of prompt-tuning. Hu et al. (2021) propose to incorporate external knowledge into the verbalizer with calibration. Ding et al. (2021) apply prompt-tuning to entity typing with prompt-learning by constructing an entity-oriented verbalizer and templates. To avoid labor-intensive prompt design, automatic searches for discrete prompts have been extensively explored. Schick, Schmid, and Schütze (2020); Gao, Fisch, and Chen (2021) first explore the automatic generation of ans words and templates. Shin et al. (2020) further propose gradient-guided search to generate the template and label word in vocabulary automatically. Recently, some continuous prompts have also been proposed (Li and Liang 2021; Hambardzumyan, Khachatrian, and May 2021; Liu et al. 2021b), which directly utilize learnable continuous embeddings as prompt templates.

For relation extraction, Han et al. (2021) proposes a model called PTR, which applies logic rules to construct prompts with several sub-prompts. Compared with their approach, our approach has two major differences. Firstly, we construct prompt with knowledge injection via learnable virtual template words and virtual answer words to alleviate labor-intensive prompt engineering rather than pre-defined rules; thus, our method is flexible and can generalize to different RE datasets. Furthermore, we synergistically optimize virtual template words and answer words with knowledge constraints and associate prompt embeddings with each other.

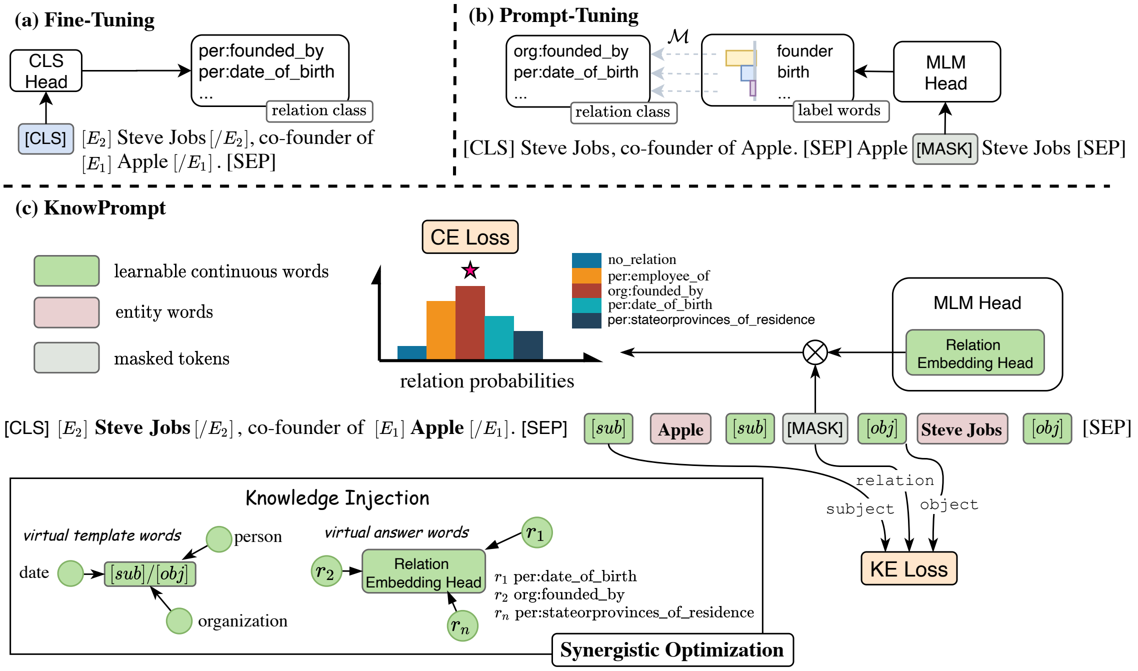

Figure 2: Model architecture of fine-tuning (Figure a), prompt-tuning (Figure b), and proposed KnowPrompt (Figure c) approach (Best viewed in color). The answer word described in the paper refers to the virtual answer word we proposed.

3 Background

An RE dataset can be denoted as $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ is the set of examples and $\mathcal{Y}$ is the set of relation labels. For each example $x = \{w_1, w_2, w_s \dots w_o, \dots w_n\}$, the goal of RE is to predict the relation $y \in \mathcal{Y}$ between subject entity w_s and object entity w_o (since one entity may have multiple tokens, we simply utilize w_s and w_o to represent all entities briefly.).

3.1 Fine-tuning of PLMs

Given a pre-trained language model (PLM) $\mathcal{L}$ for RE, previous fine-tuning methods first convert the instance $x = \{w_1, w_2, w_s \dots w_o, \dots w_n\}$ into an input sequence of PLM, such as `[CLS]`x`[SEP]`. The PLM $\mathcal{L}$ encodes the input sequence into the corresponding output hidden vectors such as $\mathbf{h} = \{\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_s, \dots, \mathbf{h}_o, \dots, \mathbf{h}_{\texttt{[SEP]}}\}$. Normally, a `[CLS]` head is utilized to compute the probability distribution over the class set $\mathcal{Y}$ with the softmax function $p(\cdot|x) = \texttt{Softmax}(\mathbf{W}\mathbf{h}_{\texttt{[CLS]}})$, where $\mathbf{h}_{\texttt{[CLS]}}$ is the output embedding of `[CLS]` and $\mathbf{W}$ is a randomly initialized matrix that needs to be optimized. The parameters of $\mathcal{L}$ and $\mathbf{W}$ are fine-tuned by minimizing the cross-entropy loss over $p(y|x)$ on the entire $\mathcal{X}$.

3.2 Prompt-Tuning of PLMs

Prompt-tuning is proposed to bridge the gap between the pre-training and downstream tasks. The challenge is to construct an appropriate template $\mathcal{T}(\cdot)$ and label words $\mathcal{V}$, which are collectively referred to as a prompt $\mathcal{P}$. For each instance x, the template is leveraged to map x to prompt the input $x_{\text{prompt}} = T(x)$. Concretely, template $\mathcal{T}(\cdot)$ involves the location and number of added additional words (including actual and learnable continuous words). $\mathcal{V}$ refers to a set of label words in the vocabulary of a language model $\mathcal{L}$, and $\mathcal{M}\colon Y \to \mathcal{V}$ is an injective mapping that connects task labels to label words $\mathcal{V}$.

In addition to retaining the original words in x, one or more `[MASK]` is placed into x_{prompt} for $\mathcal{L}$ to fill the label words. As $\mathcal{L}$ can predict the right token at the masked position, we can formalize $p(y|x)$ with the probability distribution over $\mathcal{V}$ at the masked position, that is, $p(y|x) = p(\texttt{[MASK]} = \mathcal{M}(y)|x_{\text{prompt}})$. Taking the RE task described in Figure 2 (right) as an example, we map x to $x_{\text{prompt}} =$ "x Apple `[MASK]` Steve Jobs[SEP]". We can then obtain the hidden vector of `[MASK]` by encoding x_{prompt} by $\mathcal{L}$ and produce a probability distribution $p(\texttt{[MASK]}|x_{\text{prompt}})$, describing which words of $\mathcal{V}$ are suitable for replacing the `[MASK]` word. Finally, we respectively set $\mathcal{M}(y =$ "$per : data\_of\_birth$"$) \to$ "$birth$", and conduct $\mathcal{M}(y =$ "$org : founded\_by$"$) \to$ "$founder$", etc. According to whether $\mathcal{L}$ predicts "$birth$" or "$founder$", we can identify if the relation label of instance x is either "$per : data\_of\_birth$" or "$org : founded\_by$".

4 Methodology

In this section, we begin to introduce our **Know**ledge-aware **Prompt**-tuning with synergistic optimization (KnowPrompt) approach to be aware of knowledge in entity types and relation labels for relation extraction, as shown in Figure 2. We elucidate the details of how to construct (§4.1), optimize (§4.2) the KnowPrompt.

4.1 Prompt Construction with Knowledge Injection

Because a typical prompt consists of two parts, namely a template and a set of label words, we propose the construction of virtual template words and virtual answer words with knowledge injection for the RE task.

Entity Knowledge Injection. We follow the previous approach (Zhou and Chen 2021) with ENTITY/TYPE MARKER to inject special symbols such as [E] and [/E] around entities in the raw input sequence to index the positions of entities that are widely used in RE tasks. Note that TYPE MARKER methods can additionally introduce the type information of entities to improve performance but require additional annotation of type information. However, we can obtain the scope of the entity types of two marked entities with prior knowledge given a specific relation. For instance, given the relation "per:country_of_birth", it is obvious that the subject entity belongs to "person" and the object entity belongs to "country". Intuitively, we estimate the probability distributions ϕ_{sub} and ϕ_{obj} over the candidate set $C_{sub} = \{$"person","organization", ...$\}$ and $C_{obj} = \{$"organization","data", ...$\}$ of entity types, respectively, according to the relation class, where the distributions are estimated by frequency statistics of possible entity types. As Figure 2 shows, we utilize learnable continuous words to perceive the entity type information. Specifically, the learnable embeddings of virtual template words are initialized with aggregated entity-type embeddings as follows:

$$\begin{aligned} \hat{\mathbf{e}}_{[sub]} &= \phi_{sub} \cdot \mathbf{e}\left(\mathbf{C}_{sub}\right), \\ \hat{\mathbf{e}}_{[obj]} &= \phi_{obj} \cdot \mathbf{e}\left(\mathbf{C}_{obj}\right), \end{aligned} \tag{1}$$

where $\hat{\mathbf{e}}_{[sub]}$ and $\hat{\mathbf{e}}_{[obj]}$ represent the embeddings of virtual template words surrounding the subject and object entities, and $\mathbf{e}$ is the word-embedding layer of $\mathcal{L}$. By designing learnable virtual template words based on the knowledge of entity type, our method has an effect similar to that of the ENTITY MARKER methods without additional annotation of type information. Essentially, the ENTITY MARKER and TYPED MARKER methods can be regarded as a type of template for prompts, which provide rich knowledge of entity position and entity types.

Relation Knowledge Injection. Previous studies on prompt-tuning usually form a one-one mapping between label words and task labels, which may fail to leverage the abundant semantic knowledge in relation labels. To this end, we assume that there exists a virtual answer word $v' \in \mathcal{V}'$ in the vocabulary space of PLMs, which can represent the implicit semantics of the relation. From this perspective, we expand the `MLM Head` layer of $\mathcal{L}$ with extra learnable relation embeddings as the virtual answer word sets $\mathcal{V}'$ to completely represent the corresponding relation labels $\mathcal{Y}$. Thus, we can reformalize $p(y|x)$ with the probability distribution over $\mathcal{V}'$ at the masked position. We also set the probability distribution ϕ_{rel} over the candidate set C_{rel} of the semantic words of relation by disassembling the relation type. Concretely, we adopt the weighted average function for ϕ_{rel} to average embeddings of each token in the tokenizations of relation labels to initialize these relation embeddings, which can inject the semantic knowledge of relations. Taking the relation $y_1 = per : countries\_of\_residence$ as an example, we decompose it as $C_{rel_1} = \{$"person","countries", "residence" $\}$; then, the learnable relation embedding of virtual answer word $v'_1 = \mathcal{M}(y_1)$ is initialized as follows:

$$\hat{\mathbf{e}}(v'_1) = \phi_{rel} \cdot \mathbf{e}\left(C_{rel_1}\right), \tag{2}$$

4.2 Synergistic Optimization with Knowledge Constraints

Since there exist rich interaction and connection between entity types and relation labels, and those virtual template words as well as answer words should be associated with the surrounding context, we further introduce a synergistic optimization method with implicit structural constraints. Specific, we further ***synergistically optimize the parameter set*** $\{\hat{\mathbf{e}}_{[sub]}, \hat{\mathbf{e}}_{[obj]}, \hat{\mathbf{e}}_{[rel]}(\mathcal{V}')\}$ of virtual template words and virtual answer words.

Context-aware Prompt Calibration. Although our virtual template and answer words are initialized based on semantic knowledge, they may not be optimal in the latent variable space and should be associated with the surrounding context. Thus, further optimization is necessary by perceiving the context to calibrate their representation. Given the probability distribution $p(y|x) = p(\texttt{[MASK]} = \mathcal{V}'|x_{\text{prompt}})$ over $\mathcal{V}'$ at the masked position, we optimize the virtual template words as well as answer words by the loss function computed as the cross-entropy between $\mathbf{y}$ and $p(y|x)$ as follows:

$$\mathcal{J}_{\texttt{[MASK]}} = -\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{y} \log softmax(p(y|x)), \tag{3}$$

where $|\mathcal{X}|$ represents the numbers of the train dataset. The learnable continuous words may adaptively obtain optimal representations for prompt-tuning through a synergistic template and answer optimization.

Implicit Structural Constraints. To integrate structural knowledge into KnowPrompt, we adopt the knowledge embedding (KE) objective in KnowPrompt as an additional constraint for optimizing prompts. Specifically, we use a triplet (s, r, o) to describe a relational fact; here, s, o represent the types of subject and object entities, respectively, and r is the relation label within a pre-defined set of answer words $\mathcal{V}'$. In KnowPrompt, instead of using pre-trained knowledge graph embeddings[1], we directly leverage the output embedding of virtual template words and virtual answer words through LMs to participate in the calculation. We use the loss below as our KE objective:

$$\begin{aligned} \mathcal{J}_{\text{KE}} = &-\log sigmoid(\gamma - d_r(\mathbf{h}, \mathbf{t})) \\ &- \sum_{i=1}^{n} \frac{1}{n} \log sigmoid(d_r(\mathbf{h}'_\mathbf{i}, \mathbf{t}'_\mathbf{i}) - \gamma), \end{aligned} \tag{4}$$

$$d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p, \tag{5}$$

[1] Note that pre-trained knowledge graph embeddings are heterogeneous compared with pre-trained language model embeddings.

where (s_i', r, o_i') are negative samples, γ is the margin, and d_r is the scoring function. For negative sampling, we assign the correct virtual answer words at the position of [MASK] and randomly sample the subject entity or object entity and replace it with an irrelevant one to construct corrupt triples, in which the entity has an impossible type for the current relation.

4.3 Training Details

Our approach has a two-stage optimization procedure. First, we synergistically optimize the parameter set $\{\hat{\mathrm{e}}_{[sub]}, \hat{\mathrm{e}}_{[obj]}, \hat{\mathrm{e}}_{[rel]}(\mathcal{V}')\}$ of virtual template words and virtual answer words with a large learning rate lr_1 to obtain the optimal prompt as follows:

$$\mathcal{J} = \mathcal{J}_{\texttt{[MASK]}} + \lambda \mathcal{J}_{KE}, \tag{6}$$

where λ is the hyperparameter, and $\mathcal{J}_{\text{KE}}$ and $\mathcal{J}_{\texttt{[MASK]}}$ are the losses for the KE and [MASK] prediction, respectively.

Second, based on the optimized virtual template words and answer words, we utilize the object function $\mathcal{J}_{\texttt{[MASK]}}$ to tune the parameters of the PLM with prompt (optimizing overall parameters) with a small learning rate lr_2. For more experimental details, please refer to the appendix.

5 Experiments

In this section, we detail the results of extensive experiments conducted on several RE datasets in both standard and low-resource settings to demonstrate the effectiveness of our proposed approach.

5.1 Datasets

For comprehensive experiments, we carry out our experiments on five RE datasets: SemEval 2010 Task 8 (SemEval) (Hendrickx et al. 2010), DialogRE (Yu et al. 2020a) TACRED-Revisit (Alt, Gabryszak, and Hennig 2020), Re-TACRED (Stoica, Platanios, and Póczos 2021),Wiki80 (Han et al. 2019). More details are provided in Table 1.

5.2 Experimental Settings

For fine-tuning vanilla PLMs and our KnowPrompt, we utilize BERT_LARGE for all experiments to make a fair comparison. For test metrics, we use micro F_1 scores of RE as the primary metric to evaluate models, considering that F_1 scores can assess the overall performance of precision and recall. We use different settings for standard and low-resource experiments.

Standard Setting In the standard setting, we utilize full $\mathcal{D}_{\text{train}}$ to fine-tune. Considering that entity information is essential for models to understand relational semantics, a series of knowledge-enhanced PLMs have been further explored using knowledge graphs as additional information to enhance PLMs. For those knowledge-enhanced PLMs, we select SPANBERT (Joshi et al. 2020), KNOWBERT (Peters et al. 2019), LUKE (Yamada et al. 2020), and MTB (Baldini Soares et al. 2019) as our baselines. Moreover, these four baselines are typical models that use external knowledge to enhance learning objectives, input features, model architectures, or pre-training strategies. We also compare several SOTA models on DialogRE, in which one challenge is that each entity pair has more than one relation.

Low-Resource Setting We conducted 8-, 16-, and 32-shot experiments following LM-BFF (Gao, Fisch, and Chen 2021; Han et al. 2021) to measure the average performance across five different randomly sampled data based on every experiment using a fixed set of seeds $\mathcal{S}_{\text{seed}}$. Specifically, we sample k instances of each class from the initial training and validation sets to form the few-shot training and validation sets. We tune the entire model for 20 epochs and choose the checkpoint with the best validation performance for testing. All detailed hyperparameters for our proposed model can be found in the appendix.

Dataset	# Train.	# Val.	# Test.	# Rel.
SemEval	6,507	1,493	2,717	19
DialogRE	5,963	1,928	1,858	36
TACRED-Revisit	68,124	22,631	15,509	42
Re-TACRED	58,465	19,584	13,418	40
Wiki80	44,800	5,600	5,600	80

Table 1: Statistics for RE datasets used in the paper, including numbers of relations and instances in the different split. For dialogue-level DialogRE, instance refers to the number of documents.

5.3 Main Results

In this subsection, we introduce the specific results and provide possible insights of KnowPrompt.

Standard Result As shown in Table 2, the knowledge-enhanced PLMs yield better performance than the vanilla FINE-TUNING. This result illustrates that it is practical to inject task-specific knowledge to enhance models, indicating that simply fine-tuning PLMs cannot perceive knowledge obtained from pre-training. Note that our KnowPrompt achieves improvements over most baselines and even achieves comparable performance with those knowledge-enhanced models, which use knowledge as data augmentation or architecture enhancement during fine-tuning. We think that KnowPrompt can be aware of knowledge and stimulate it to serve downstream tasks better. Especially for DialogRE, a multi-label classification task with many reversed relations, our method is competitive with the SOTA models GDPNET (Xue et al. 2021) and DUAL (Bai et al. 2021), which are based on PLMs with a complex graph structure. This result indicates that the injected structural knowledge in our approach is particularly beneficial for multi-label RE. Overall, we believe that KnowPrompt is a simple and effective fine-tuning paradigm for RE.

Low-Resource Result From Table 3, KnowPrompt appears to be more beneficial in low-resource settings. We find that KnowPrompt consistently outperforms the baseline method FINE-TUNING, GDPNET, and PTR in all datasets, especially in the 8-shot and 16-shot experiments. Specifically, our model can obtain gains of up to **23.7%** and **14.4%** absolute improvement on average compared with

Stadard Supervised Setting					
Methods	Extra Data	SEMEVAL	DialogRE	TACRED-revisit	Re-TACRED
Fine-tuning pre-trained models					
Fine-tuning	w/o	88.1	57.3	77.0	87.8
SpanBERT (Joshi et al. 2020)	w/	-	-	78.0	85.3
KnowBERT (Peters et al. 2019)	w/	-	-	79.3	89.1
LUKE (Yamada et al. 2020)	w/	-	-	80.6	-
MTB (Baldini Soares et al. 2019)	w/	89.5	-	-	-
GDPNet (Xue et al. 2021)	w/o	-	64.9	79.3	-
Dual (Bai et al. 2021)	w/o	-	67.3	-	-
Prompt-tuning pre-trained models					
PTR (Han et al. 2021)†	w/o	89.1	62.8	80.2	89.0
KnowPrompt	w/o	**90.1** (+0.6)	66.0 (-1.3)	**80.8** (+0.2)	89.8 (+0.7)

Table 2: Standard RE performance of F_1 scores (%) on the on different test sets. For Fine-tuning (Devlin et al. 2019), we report the results of fine-tuning BERT_LARGE (Devlin et al. 2019) with entity markers. In the "Extra Data" column, "w/o" means that no additional data is used for pre-training and fine-tuning, yet "w/" means that the model uses extra data for tasks."†" indicates we rerun their code with BERT_LARGE for a fair comparison. Subscript in red represents advantages of KnowPrompt over the best results of baselines. Best results are bold, and dataset analysis can be seen in Table 1.

Low-Resource Setting							
Split	Methods	SEMEVAL	DialogRE	TACRED-Revisit	Re-TACRED	WiKi80	Average
K=8	Fine-tuning	28.8	26.1	10.5	20.1	47.6	26.6
	GDPNet	27.3	23.6	8.3	18.8	45.7	24.7
	PTR	61.9	35.5	25.3	43.6	67.6	46.8
	KnowPrompt	**64.5** (+35.7)	**40.8** (+14.7)	**28.6** (+18.1)	**45.8** (+25.7)	**71.8** (+24.2)	**50.3** (+23.7)
K=16	Fine-tuning	45.7	40.8	19.2	47.4	59.4	42.5
	GDPNet	45.5	38.5	20.8	48.0	61.2	42.8
	PTR	71.8	43.5	27.2	51.8	75.6	53.8
	KnowPrompt	**73.8** (+28.1)	**47.7** (+6.9)	**30.8** (+11.6)	**53.8** (+6.4)	**78.8** (+19.4)	**56.9** (+14.4)
K=32	Fine-tuning	65.4	47.7	26.0	53.6	69.9	52.5
	GDPNet	67.2	47.1	28.1	54.8	72.3	53.9
	PTR	78.3	49.5	33.1	54.8	78.8	59.3
	KnowPrompt	**79.8** (+14.4)	**53.2** (+5.5)	**34.2** (+8.2)	**55.2** (+1.6)	**81.3** (+11.4)	**60.7** (+8.2)

Table 3: Low-resource RE performance of F_1 scores (%) on different test sets. We use $K = 8, 16, 32$ (# examples per class) for few-shot experiments. Subscript in red represents the advantages of KnowPrompt over the results of Fine-tuning.

Fine-tuning. As K increases from 8 to 32, the improvement in our KnowPrompt over the other three methods decreases gradually. For 32-shot, we think that the number of labeled instances is sufficient to optimize the answer words' embeddings. Thus, those rich semantic knowledge injected in our approach may induce fewer gains. We also observe that GDPNet even performs worse than Fine-tuning for 8-shot, which reveals that the complex SOTA model on the standard setting may fall off the altar when the data are extremely scarce. Note that these findings also illustrate that our approach generalizes well to different low-resource scenarios.

5.4 Ablation Study on KnowPrompt

We conduct an ablation study to validate the effectiveness of the components. Specifically, for *-virtual answer words*, we adopt one specific token in the relation label as the answer word without optimization; for *-virtual template words*, we directly remove virtual template words; *-implicit structural constraints* refer to the model without implicit structural constraints, which indicates no explicit correlations between entities and relations. From Table 4, we observe that our KnowPrompt has a performance decay without each module, which demonstrates that all components are necessary. In addition, we find that virtual answer words and virtual template words are incredibly beneficial, especially in low-resource settings. We further notice that virtual answer words is more sensitive to performance and is highly beneficial for KnowPrompt, especially in low-resource settings, which illustrates that a suitable label token is important.

Method	**K=8**	**K=16**	**K=32**	**Full**
KnowPrompt	**64.5**	**73.8**	**79.8**	**89.8**
-virtual answer words	61.3	70.2	78.0	88.5
-virtual template words	62.7	71.5	77.9	89.0
-implicit structural constrains	63.8	72.8	79.0	90.1

Table 4: Ablation study of KnowPrompt on SEMEVAL.

Input Example of our KnowPrompt	Top 3 words around `[sub]`	Top 3 words around `[obj]`
x: `[CLS]` It sold $[E_1]$ **ALICO** $[/E_1]$ to $[E_2]$ **MetLife Inc** $[E_2]$ for $ 162 billion. `[SEP]` `[sub]` **ALICO** `[sub]` `[MASK]` `[obj]` **MetLife Inc** `[obj]`. `[SEP]` **y**: "*org : member_of*"	organization group corporation	company plc organization
x: `[CLS]` $[E_1]$ **Ismael Rukwago** $[/E_1]$, a senior $[E_2]$ **ADF** $[E_2]$ commander, denied any involvement. `[SEP]` `[sub]` **Ismael Rukwago** `[sub]` `[MASK]` `[obj]` **ADF** `[obj]`. `[SEP]` **y**: "*per : employee_of*"	person commander colonel	intelligence organization command

Table 5: Interpreting representation of **virtual template words**. We obtain the hidden state $\mathbf{h}_{\texttt{[sub]}}, \mathbf{h}_{\texttt{[obj]}}$ through the PLM, then adopt `MLM Head` over them to explore which words in the vocabulary is nearest the virtual template words.

6 Analysis and Discussion

Can KnowPrompt Applied to Other LMs? Since we focus on MLM (e.g., BERT) in the main experiments, we further extend our KnowPrompt to autoregressive LMs like GPT-2. Specifically, we place the mask at the end of the sequence input so that the GPT-2 can generate the virtual answer word learned in parameter space. We first notice that fine-tuning leads to poor performance with high variance in the low-resource setting, while KnowPrompt based on BERT or GPT-2 can achieve impressive improvement with low variance compared with FINE-TUNING. As shown in Figure 3, KnowPrompt based on GPT-2 obtains the results on par of the model with BERT-large, which reveals our method can unearth the potential of GPT-2 to make it perform well in natural language understanding tasks such as RE. This finding also indicates that our method is model-agnostic and can be plugged into different kinds of PLMs.

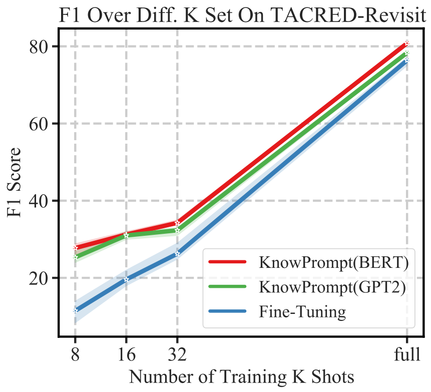

Figure 3: BERT-large vs. GPT-2 results on TACRED-Revisit dataset regarding different K (instances per class).

Interpreting Representation Space Since the embeddings of virtual template words and virtual answer words $\{\hat{\mathrm{e}}_{[sub]}, \hat{\mathrm{e}}_{[obj]}, \hat{\mathrm{e}}_{[rel]}(\mathcal{V}')\}$ are learned in continuous space, it is intuitive to explore what exactly the optimized prompt is. To this end, we sample the top-3 words in vocabulary nearest the virtual answer words according to the L_2 distance of embeddings between prompts and other words. We use t-SNE and normalization to mapping the embedding to 3 dimension space and make a 3D visualization of several sampled virtual answer words in the TACRED-Revisit dataset. For example, "$org : founded\_by$" referred to as green ★ in Figure 4 represents the relation type, which is learned by optimizing virtual answer words in vocabulary space, and the "$founder$","$chair$" and "ceo" referred to as green ● are the words closest to the it. It reveals that virtual answer words learned in vocabulary space are semantic and intuitive. We further investigate whether virtual template words can adaptively reflect the entity type based on context as shown in Table 5. We observe that those learned virtual template words can dynamically adjust according to context and play a reminder role for RE. Note that with synergistic optimization of the virtual template and answer words, our model can automatically obtain an optimized prompt for relation extraction, which can also be applied to other NLP tasks with prompt-tuning.

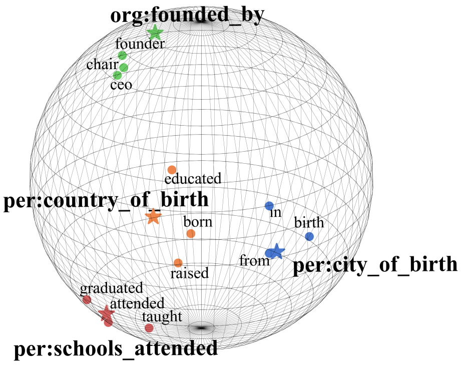

Figure 4: A 3D visualization of several relation representations (**virtual answer words**) optimized in KnowPrompt on TACRED-Revisit dataset using t-SNE and normalization.

7 Conclusion and Future Work

In this paper, we present KnowPrompt, which mainly includes knowledge-injected prompt construction and synergistic optimization with structure constraints. Experimental results on five datasets show that our approach achieves improvement in both standard and low-resource scenarios compared with various baselines. In the future, we plan to explore three directions, including: (i) extending to semi-supervised setting to further leverage unlabelled data; (ii) extending to lifelong learning, whereas prompt should be optimized with adaptive tasks. (iii) applying our approaches to more prompt-tuning tasks.

References

Alt, C.; Gabryszak, A.; and Hennig, L. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of ACL 2020*.

Bai, X.; Chen, Y.; Song, L.; and Zhang, Y. 2021. Semantic Representation for Dialogue Modeling. In *Proceedings of ACL/IJCNLP 2021*.

Baldini Soares, L.; FitzGerald, N.; Ling, J.; and Kwiatkowski, T. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of ACL/IJCNLP 2019*.

Bastos, A.; Nadgeri, A.; Singh, K.; Mulang, I. O.; Shekarpour, S.; Hoffart, J.; and Kaul, M. 2021. RECON: Relation Extraction using Knowledge Graph Context in a Graph Neural Network. In *Proceedings of the Web Conference 2021*, 1673–1685.

Baumgartner, M.; Zhang, W.; Paudel, B.; Dell'Aglio, D.; Chen, H.; and Bernstein, A. 2018. Aligning Knowledge Base and Document Embedding Models Using Regularized Multi-Task Learning. In *International Semantic Web Conference (1)*, volume 11136 of *Lecture Notes in Computer Science*, 21–37. Springer.

Ben-David, E.; Oved, N.; and Reichart, R. 2021. PADA: A Prompt-based Autoregressive Approach for Adaptation to Unseen Domains. *arXiv preprint arXiv:2102.12206*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Proceedings of NeurIPS 2020*.

Califf, M. E.; and Mooney, R. J. 1999. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of AAAI*, 328–334. AAAI Press / The MIT Press.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*.

Ding, N.; Chen, Y.; Han, X.; Xu, G.; Xie, P.; Zheng, H.-T.; Liu, Z.; Li, J.; and Kim, H.-G. 2021. Prompt-Learning for Fine-Grained Entity Typing. *arXiv preprint arXiv:2108.10604*.

Distiawan, B.; Weikum, G.; Qi, J.; and Zhang, R. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of ACL*, 229–240.

Dong, B.; Yao, Y.; Xie, R.; Gao, T.; Han, X.; Liu, Z.; Lin, F.; Lin, L.; and Sun, M. 2020. Meta-Information Guided Meta-Learning for Few-Shot Relation Classification. In *Proceedings of COLING 2020*.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of ACL*.

Gao, T.; Han, X.; Liu, Z.; and Sun, M. 2019. Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification. In *Proceedings of AAAI*.

Gao, T.; Han, X.; Xie, R.; Liu, Z.; Lin, F.; Lin, L.; and Sun, M. 2020. Neural Snowball for Few-Shot Relation Learning. In *Proceedings of AAAI 2020*.

Guo, Z.; Nan, G.; Lu, W.; and Cohen, S. B. 2020. Learning Latent Forests for Medical Relation Extraction. In *IJCAI*, 3651–3657.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of ACL 2019*.

Hambardzumyan, K.; Khachatrian, H.; and May, J. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of ACL/IJCNLP 2021*.

Han, X.; Gao, T.; Yao, Y.; Ye, D.; Liu, Z.; and Sun, M. 2019. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. In *Proceedings EMNLP-IJCNLP 2019*.

Han, X.; Zhao, W.; Ding, N.; Liu, Z.; and Sun, M. 2021. PTR: Prompt Tuning with Rules for Text Classification. *CoRR*, abs/2105.11259.

Han, X.; Zhu, H.; Yu, P.; Wang, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2018. FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of EMNLP, 2018*.

Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Ó Séaghdha, D.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of SemEval*, 33–38.

Hu, S.; Ding, N.; Wang, H.; Liu, Z.; Li, J.; and Sun, M. 2021. Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. *arXiv preprint arXiv:2108.02035*.

Huffman, S. B. 1995. Learning information extraction patterns from examples. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*.

Joshi, M.; Chen, D.; Liu, Y.; Weld, D. S.; Zettlemoyer, L.; and Levy, O. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguistics*, 8: 64–77.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. *arXiv preprint arXiv:2104.08691*.

Li, J.; Wang, R.; Zhang, N.; Zhang, W.; Yang, F.; and Chen, H. 2020. Logic-guided Semantic Representation Learning for Zero-Shot Relation Classification. In *Proceedings of COLING*, 2967–2978.

Li, P.; Mao, K.; Yang, X.; and Li, Q. 2019. Improving Relation Extraction with Knowledge-attention. In *Proceedings of EMNLP*, 229–239.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of ACL/IJCNLP 2021*.

Lin, Y.; Ji, H.; Huang, F.; and Wu, L. 2020. A Joint Neural Model for Information Extraction with Global Features. In *Proceedings ACL 2020*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021b. GPT Understands, Too. *CoRR*, abs/2103.10385.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2021. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. *arXiv preprint arXiv:2104.08786*.

Nan, G.; Guo, Z.; Sekulic, I.; and Lu, W. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *Proceedings of ACL*.

Peters, M. E.; Neumann, M.; Logan, R.; Schwartz, R.; Joshi, V.; Singh, S.; and Smith, N. A. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of EMNLP-IJCNLP*, 43–54.

Qu, M.; Gao, T.; Xhonneux, L. A. C.; and Tang, J. 2020. Few-shot Relation Extraction via Bayesian Meta-learning on Relation Graphs. In *Proceedings of ICML 2020*.

Reynolds, L.; and McDonell, K. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proceeding of CHI*, 1–7.

Scao, T. L.; and Rush, A. M. 2021. How Many Data Points is a Prompt Worth? *CoRR*, abs/2103.08493.

Schick, T.; Schmid, H.; and Schütze, H. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In *Proceedings of COLING*.

Schick, T.; and Schütze, H. 2020. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. *CoRR*, abs/2009.07118.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of EACL 2021*.

Shin, T.; Razeghi, Y.; IV, R. L. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of EMNLP 2020*.

Stoica, G.; Platanios, E. A.; and Póczos, B. 2021. Re-TACRED: Addressing Shortcomings of the TACRED Dataset. *arXiv preprint arXiv:2104.08398*.

Wang, Z.; Wen, R.; Chen, X.; Huang, S.-L.; Zhang, N.; and Zheng, Y. 2020. Finding influential instances for distantly supervised relation extraction. *arXiv preprint arXiv:2009.09841*.

Wu, S.; and He, Y. 2019. Enriching Pre-trained Language Model with Entity Information for Relation Classification. In *Proceedings of theCIKM 2019*.

Wu, T.; Li, X.; Li, Y.; Haffari, R.; Qi, G.; Zhu, Y.; and Xu, G. 2021. Curriculum-Meta Learning for Order-Robust Continual Relation Extraction. *CoRR*, abs/2101.01926.

Xue, F.; Sun, A.; Zhang, H.; and Chng, E. S. 2021. GDPNet: Refining Latent Multi-View Graph for Relation Extraction. In *Proceedings of AAAI 2021*.

Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; and Matsumoto, Y. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of EMNLP 2020*.

Ye, H.; Zhang, N.; Deng, S.; Chen, M.; Tan, C.; Huang, F.; and Chen, H. 2021. Contrastive Triple Extraction with Generative Transformer. In *Proceedings of AAAI, 2021*.

Yu, D.; Sun, K.; Cardie, C.; and Yu, D. 2020a. Dialogue-Based Relation Extraction. In *Proceedings of ACL 2020*.

Yu, H.; Zhang, N.; Deng, S.; Ye, H.; Zhang, W.; and Chen, H. 2020b. Bridging Text and Knowledge with Multi-Prototype Embedding for Few-Shot Relational Triple Extraction. In *Proceedings of COLING*, 6399–6410. International Committee on Computational Linguistics.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of EMNLP 2015*.

Zhang, N.; Chen, X.; Xie, X.; Deng, S.; Tan, C.; Chen, M.; Huang, F.; Si, L.; and Chen, H. 2021a. Document-level Relation Extraction as Semantic Segmentation. In Zhou, Z., ed., *Proceedings of IJCAI*, 3999–4006. ijcai.org.

Zhang, N.; Deng, S.; Bi, Z.; Yu, H.; Yang, J.; Chen, M.; Huang, F.; Zhang, W.; and Chen, H. 2020. OpenUE: An Open Toolkit of Universal Extraction from Text. In *Proceedings of EMNLP*.

Zhang, N.; Deng, S.; Sun, Z.; Chen, X.; Zhang, W.; and Chen, H. 2018. Attention-Based Capsule Network with Dynamic Routing for Relation Extraction. In *Proceedings of EMNLP 2018*.

Zhang, N.; Deng, S.; Sun, Z.; Wang, G.; Chen, X.; Zhang, W.; and Chen, H. 2019. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks. In *Proceedings of NAACL-HLT*.

Zhang, N.; Jia, Q.; Deng, S.; Chen, X.; Ye, H.; Chen, H.; Tou, H.; Huang, G.; Wang, Z.; Hua, N.; and Chen, H. 2021b. AliCG: Fine-grained and Evolvable Conceptual Graph Construction for Semantic Search at Alibaba. In *Proceedings of KDD*, 3895–3905. ACM.

Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of EMNLP, 2018*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of EMNLP 2017*.

Zheng, H.; Wen, R.; Chen, X.; Yang, Y.; Zhang, Y.; Zhang, Z.; Zhang, N.; Qin, B.; Ming, X.; and Zheng, Y. 2021. PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction. In *Proceedings of ACL/IJCNLP 2021*.

Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; and Xu, B. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of ACL 2016*.

Zhou, W.; and Chen, M. 2021. An Improved Baseline for Sentence-level Relation Extraction. *CoRR*, abs/2102.01373.