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 specific 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 exists abundant semantic and prior knowledge among the relation labels that cannot be ignored. To this end, we focus on incorporating knowledge among relation labels into prompt-tuning for relation extraction and propose a **Know**ledge-aware **Prompt**-tuning approach with synergistic optimization (**KnowPrompt**). Specifically, we inject latent knowledge contained in relation labels into prompt construction with learnable virtual type words and answer words. Then, we synergistically optimize their representation with structured constraints. Extensive experimental results on five datasets with standard and low-resource settings demonstrate the effectiveness of our approach. Our code and datasets are available in GitHub[1] for reproducibility[2].

1 Introduction

Relation Extraction (RE) aims to extract structured knowledge from unstructured text and plays a critical role in information extraction and knowledge base construction. RE appeals to many researchers (Zhang et al., 2017a, 2018a; Baumgartner et al., 2018; Zhang et al., 2019; Harting et al., 2020; Shen et al., 2021; Wu et al., 2021) due to the capability to extract textual information and benefit many web applications, e.g., information retrieval, web mining, and question answering.

Previous self-supervised pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) 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; Gu et al., 2021; Ding et al., 2021a; Hu et al., 2021a) 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 (a), a typical prompt for text classification consists of a template (e.g. "<S_1> It is `[MASK]` ") and a set of label words ("great", "terrible"etc.) as candidates to predict `[MASK]`. PLMs predict ("great", "terrible", etc.) at the masked position to determine the label of the sentence "<S_1>". In a nutshell, prompt-tuning involves template engineering and verbalizer engineering, which aims to search for the best template and an answer space (Liu et al., 2021a).

Despite the success of prompt-tuning PLMs for text classification tasks, there are still several non-trivial challenges for RE with prompt-tuning as follows: on the 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 et al., 2020; Schick and Schütze, 2021; Shin et al., 2020; Gao et al., 2021); on the other hand, the computational complexity of the label word search process is very high (e.g.,

[*] Corresponding author.
[1] https://github.com/zjunlp/KnowPrompt
[2] This paper has been published at theWebConf2022.

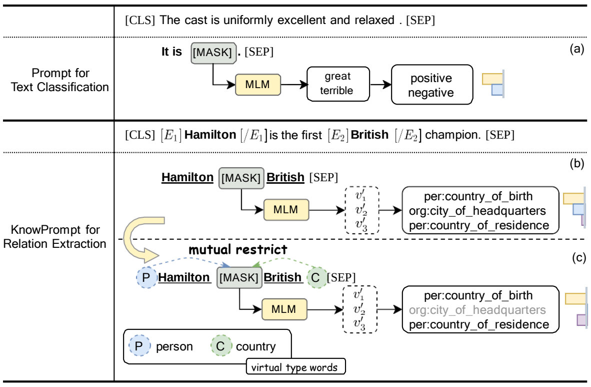

Figure 1: Examples of prompt-tuning to stimulate the knowledge of PLMs by formalizing specific tasks as cloze-style tasks. The **P** and **C** in dashed balls represents the virtual type words with semantics close to person and country.

usually exponentially depending on the number of categories) when the length of the relation label varies, and it is non-trivial to obtain a suitable target label word in the vocabulary to represent the specific relation label. For example, the relation labels of $per : country\_of\_birth$ and $org : city\_of\_headquarters$ cannot specify a single suitable label word in the vocabulary. In addition, there exists rich semantic knowledge among relation labels and structural knowledge implications among relational triples, which cannot be ignored. For example, as shown in Figure 1 (b) and (c), if a pair of entities contains the semantics of "person" and "country", the prediction probability of the [MASK] on the relation "org:city_of_headquarters" will be lower. Conversely, the relation also restricts the 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 RE.

To address those issues, we take the first step to inject knowledge into learnable 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 answer words and virtual type words to alleviate labor-intensive prompt engineering (§4.1). To be specific, instead of a regular verbalizer that mapping from one label word in vocabulary to the particular class, we creatively propose to **leverage learnable virtual answer words by injecting in semantic knowledge to represent relation labels**. Furthermore, we assign learnable virtual type words surrounding entities to hold the role of weakened TYPE MARKER (Zhou and Chen, 2021), which are initialized with prior knowledge maintained in relation labels. Notably, we innovatively **leverage learnable virtual type words to dynamically adjust according to context rather than utilizing annotation of the entity type**, which may not be available in datasets. 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 type 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.

2 Related Work

2.1 Relation Extraction

Relation Extraction (RE) involves extracting the relation between two given entities based on their related context, which plays an essential task in information extraction and knowledge base construction. 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., 2017a) and graph-based methods (Zhang et al., 2018b; Guo et al., 2019, 2020). With the recent advances in pre-trained language models (Devlin et al., 2019), applying PLMs as the backbone of RE systems (Zhou 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). Meanwhile, a series of knowledge-enhanced PLMs have been further explored, which use knowledge bases as additional information to enhance PLMs. Among them, MTB(Baldini Soares et al., 2019) propose matching the blanks based on BERT, which is a RE-oriented pre-trained method to learn relational patterns from text. SPANBERT (Joshi et al., 2020) adopt knowledge to enhance learning objectives, KNOWBERT (Peters et al., 2019) propose to incorporate knowledge into input features, and LUKE (Yamada et al., 2020) leverage knowledge to improve model architectures. We compare with this line of work here for their promotion comes from relational knowledge of external sources. In contrast to them, we focus on learning from the text itself in the paper. Recently, Xue et al. (2021) propose a multi-view graph based on BERT, achieving SOTA performance both on TACRED-Revisit (Alt et al., 2020) and DialogRE (Yu et al., 2020a). Thus, we also choose the latest graph methods based on BERT for RE as our baselines to demonstrate the effectiveness of our KnowPrompt.

Some previous studies (Chen et al., 2021) have focused on the few-shot setting since available annotated instances may be limited in practice. 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 et al. (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 et al., 2021; Lester et al., 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. (2021b) propose to incorporate external knowledge into the verbalizer with calibration. Ding et al. (2021b) 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 et al. (2020); Gao et al. (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 et al., 2021; Liu et al., 2021b; He et al., 2021), which focus on utilizing learnable continuous embeddings as prompt templates rather than label words. However, these works can not adapted to RE directly.

For relation extraction, Han et al. (2021) proposes a model called PTR, which creatively applies logic rules to construct prompts with several sub-prompts. Compared with their approach, our approach has three significant differences. **Firstly**, we **propose virtual answer words to represent specific relation labels rather than multiple sub-prompt** in PTR. Essentially, our method is **model-agnostic** that can be applied to generative LMs, while PTR fails due to its sub-prompt mechanism. **Secondly**, we construct prompt with knowledge injection via learnable virtual type words and virtual answer words to alleviate labor-intensive prompt engineering rather than predefined rules; thus, our method is more **flexible** and can generalize to different RE datasets easily. **Thirdly**, we synergistically optimize virtual type words and answer words with knowledge constraints and associate prompt embeddings with each other.

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 PLMs for RE

Given a pre-trained language model (PLM) $\mathcal{L}$ for RE, previous fine-tuning methods first convert the instance $x = \{w_1, w_s \dots w_o, \dots w_n\}$ into an input sequence of PLM, such as $\texttt{[CLS]}x\texttt{[SEP]}$. The PLM $\mathcal{L}$ encodes the input sequence into the corresponding output hidden vectors such as $\mathbf{h} = \{\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_1, \mathbf{h}_s, \dots, \mathbf{h}_o, \dots, \mathbf{h}_{\texttt{[SEP]}}\}$. 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 $\mathcal{X}$.

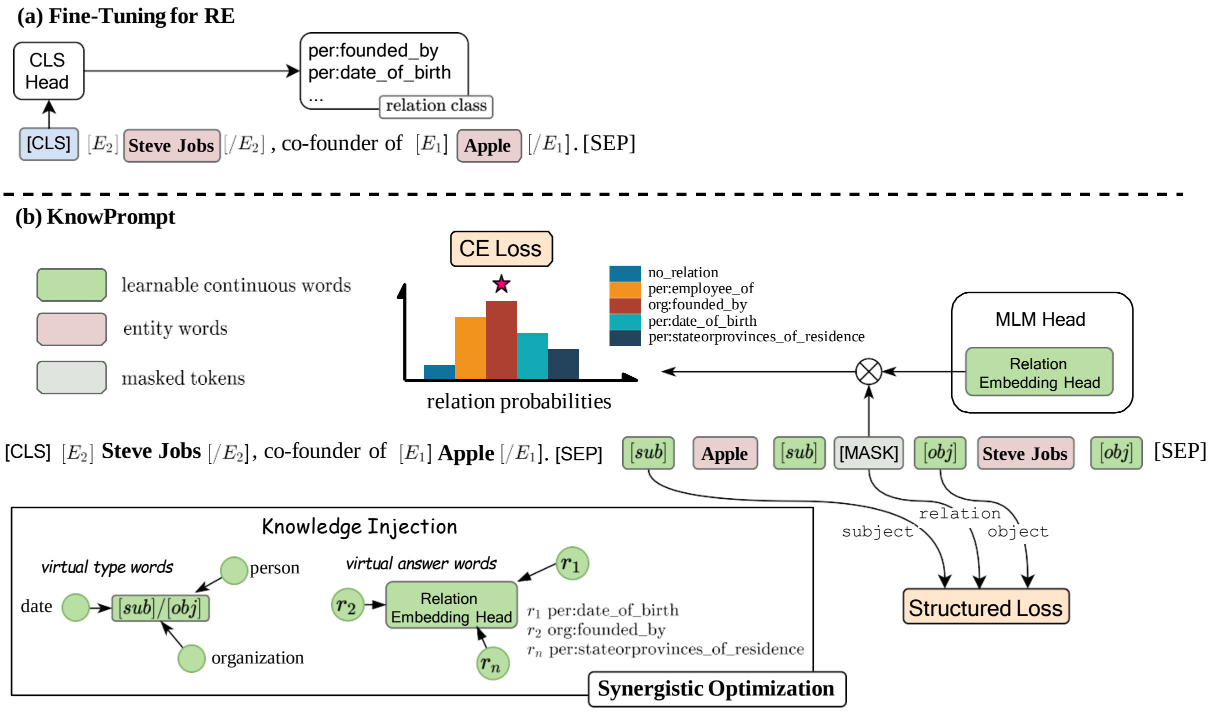

Figure 2: Model architecture of fine-tuning for RE (Figure a), and proposed KnowPrompt (Figure b) approach (Best viewed in color). The answer word described in the paper refers to the virtual answer word we proposed.

3.2 Prompt-Tuning of PLMs

Prompt-tuning is proposed to bridge the gap between the pre-training tasks 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. $\mathcal{V}$ refers to a set of label words in the vocabulary of a language model $\mathcal{L}$, and $\mathcal{M}\colon \mathcal{Y} \rightarrow \mathcal{V}$ is an injective mapping that connects task labels to label words $\mathcal{V}$. In addition to retaining the original tokens 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 binary sentiment classification task described as an example, we set the template $T(\cdot) =$ "$\cdot$ It is `[MASK]`." and map x to $x_{\text{prompt}} =$ "x It is `[MASK]`.". We can then obtain the hidden vector of `[MASK]` by encoding x_{prompt} by $\mathcal{L}$ and produce a probability distribution $p($`[MASK]`$|x_{\text{prompt}})$, describing which tokens of $\mathcal{V}$ are suitable for replacing the `[MASK]` word. Since previous study for prompt-learning involves searching or generating label words here, we simply set $\mathcal{M}(y =$ "$positive$"$) \rightarrow$ "$great$" and $\mathcal{M}(y =$ "$negative$"$) \rightarrow$ "$terrible$" as examples. According to whether $\mathcal{L}$ predicts "great" or "terrible", we can identify if the label of instance x is either $positive$ or $negative$.

4 Methodology

In this section, we introduce our **Know**ledge-aware **Prompt**-tuning with synergistic optimization (KnowPrompt) approach to be aware of semantic and prior knowledge contained in 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 type words and virtual answer words with knowledge injection for the RE task.

Entity Knowledge Injection. Note that TYPE MARKER (Zhou and Chen, 2021) methods can additionally introduce the type information of entities to improve performance but require additional annotation of type information, which is not always available in datasets. However, we can obtain the **scope of the potential entity types** with prior knowledge contained in a specific relation, rather than annotation. For instance, given the relation "per:country_of_birth", it is evident that the subject entity matching this relation belongs to "person" and the object entity matching this relation belongs to "country". Intuitively, we estimate the **prior distributions** ϕ_{sub} and ϕ_{obj} over the candidate set C_{sub} and C_{obj} of potential entity types, respectively, according to the relation class, where the prior distributions are estimated by frequency statistics. Take C_{sub} and C_{obj} of partial relation labels listed in the Table 1 as an example, the prior distributions for C_{sub} can be counted as: $\phi_{sub} = \{\text{``}organization\text{''} : 3/6, \text{``}person\text{''} : 3/6\}$. Because of this, we assign virtual type words around the entities, which are initialized with aggregated embeddings of the set of potential entity types. Since initialized virtual type words are not precise types for specific entities, those learnable virtual type words can dynamically adjust according to context and play the **weakened role of TYPE MARKER** for RE. The specific initialization method is as follows:

$$\hat{\mathbf{e}}_{[sub]} = \sum \phi_{sub} \cdot \mathbf{e}\left(I\left(\mathbf{C}_{sub}\right)\right), \tag{1}$$

$$\hat{\mathbf{e}}_{[obj]} = \sum \phi_{obj} \cdot \mathbf{e}\left(I\left(\mathbf{C}_{obj}\right)\right), \tag{2}$$

where $\hat{\mathbf{e}}_{[sub]}$ and $\hat{\mathbf{e}}_{[obj]}$ represent the embeddings of virtual type words surrounding the subject and object entities, $I\left(\cdot\right)$ is the deduplication operations on sets, and $\mathbf{e}$ is the word-embedding layer of $\mathcal{L}$. Since the virtual type words designed based on the prior knowledge within relation labels can initially perceive the range of entity types, it can be further optimized according to context to express semantic information close to the actual entity type, holding the role similar to TYPER MARKER.

Relation Knowledge Injection. Previous studies on prompt-tuning usually form a one-one mapping between one label word in the vocabulary and one task label by automatic generation, which maintains large computational complexity of the search process and fails to leverage the abundant semantic knowledge in relation labels for RE. 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 propose to **encodes semantic knowledge about the label and facilitates the process of RE**. Concretely, we set the $\phi_R = [\phi_{r1}, \phi_{r2}, ..., \phi_{rm}]$ and $\mathbf{C}_R = [\mathbf{C}_{r1}, \mathbf{C}_{r2}, ..., \mathbf{C}_{rm}]$, where ϕ_r represent the probability distribution over the candidate set $\mathbf{C}_r$ of the semantic words of relation by **disassembling** the relation label r, m is the number of relation labels. Furthermore, we adopt the weighted average function for ϕ_r to average embeddings of each words among $\mathbf{C}_r$ to initialize these relation embeddings, which can inject the semantic knowledge of relations. The specific decomposition process is shown in Table 1, and the learnable relation embedding of virtual answer word $v' = \mathcal{M}(y)$ is initialized as follows:

$$\hat{\mathbf{e}}_{[rel]}(v') = \phi_r \cdot \mathbf{e}\left(\mathbf{C}_r\right), \tag{3}$$

where $\hat{\mathbf{e}}_{[rel]}(v')$is the embedding of virtual label word v', $\mathbf{e}$ represents the word-embedding layer of $\mathcal{L}$. It is noticed that the knowledgeable initialization process of virtual answer words may be regarded as a great anchor; we can further optimize them based on context to express optimal semantic information, leading to better performance.

4.2 Synergistic Optimization with Knowledge Constraints

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

Context-aware Prompt Calibration. Although our virtual type and answer words are initialized based on knowledge, they may not be optimal in the latent variable space. They 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 type words as well as answer words by the loss function computed as the cross-entropy between $\mathbf{y}$ and $p(y|x)$ as follows:

Relation Labels	$\mathbf{C}_{sub}$	$\mathbf{C}_{obj}$	$\mathbf{C}_r$ (Disassembled Relation Prepared for Virtual Answer Words)
per:country_of_birth	person	country	{“country”, “of”, “birth” }
per:data_of_death	person	data	{“data”, “of”, “death” }
per:schools_attended	person	organization	{“school”,“attended’}
org:alternate_names	organization	organization	{“alternate”, “names” }
org:city_of_headquarters	organization	city	{“city”, “of”, “headquarters” }
org:number_of_employees/members	organization	number	{“number”, “of”, “employees”, “members” }

Table 1: Examples of some relations of the datasets TACREV, and relation-specific $\mathbf{C}_{sub}$,$\mathbf{C}_{obj}$ and $\mathbf{C}_r$.

$$\mathcal{J}_{\texttt{[MASK]}} = -\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{y} \log p(y|x), \qquad (4)$$

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

Implicit Structured Constraints. To integrate structural knowledge into KnowPrompt, we adopt additional structured constraints to optimize prompts. Specifically, we use a triplet (s, r, o) to describe a relational fact; here, s, o represent the virtual 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[3], we directly leverage the output embedding of virtual type words and virtual answer words through LMs to participate in the calculation. We define the loss $\mathcal{J}_{\texttt{struct}}$ of implicit structured constraints as follows:

$$\begin{aligned}\mathcal{J}_{\texttt{structured}} = &- \log \sigma(\gamma - d_r(\mathbf{s}, \mathbf{o})) \\ &- \sum_{i=1}^{n} \frac{1}{n} \log \sigma(d_r(\mathbf{s'_i}, \mathbf{o'_i}) - \gamma),\end{aligned} \qquad (5)$$

$$d_r(\mathbf{s}, \mathbf{o}) = \|\mathbf{s} + \mathbf{r} - \mathbf{o}\|_2, \qquad (6)$$

where (s'_i, r, o'_i) are negative samples, γ is the margin, σ refers to the sigmoid function 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.

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

Table 2: 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.

4.3 Training Details

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

$$\mathcal{J} = \mathcal{J}_{\texttt{[MASK]}} + \lambda \mathcal{J}_{\texttt{structured}}, \qquad (7)$$

where λ is the hyperparameter, and $\mathcal{J}_{\texttt{structured}}$ and $\mathcal{J}_{\texttt{[MASK]}}$ are the losses for the KE and `[MASK]` prediction, respectively. Second, based on the optimized virtual type 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

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 (Zhang et al., 2017b), TACRED-Revisit (Alt et al., 2020), Re-TACRED (Stoica et al., 2021). Statistical details are provided in Table 2 and Appendix A.1.

5.2 Experimental Settings

For fine-tuning vanilla PLMs and our KnowPrompt, we utilize ROBERT_LARGE for all experiments

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

Standard Supervised Setting						
Methods	Extra Data	SemEval	DialogRE†	TACRED	TACRED-Revisit	Re-TACRED
Fine-tuning pre-trained models						
Fine-tuning-[RoBERTa]	w/o	87.6	57.3	68.7	76.0	84.9
SpanBERT (Joshi et al., 2020)	w/	-	-	70.8	78.0	85.3
KnowBERT (Peters et al., 2019)	w/	89.1	-	71.5	79.3	89.1
LUKE (Yamada et al., 2020)	w/	-	-	**72.7**	80.6	-
MTB (Baldini Soares et al., 2019)	w/	89.5	-	70.1	-	-
GDPNet (Xue et al., 2021)	w/o	-	64.9	71.5	79.3	-
Dual (Bai et al., 2021)	w/o	-	67.3	-	-	-
Prompt-tuning pre-trained models						
PTR-[RoBERTa] (Han et al., 2021)	w/o	89.9	63.2	72.4	81.4	90.9
KnowPrompt-[RoBERTa]	w/o	**90.2** (+0.3)	**68.6** (+5.4)	72.4 (-0.3)	**82.4** (+1.0)	**91.3** (+0.4)

Table 3: Standard RE performance of F_1 scores (%) on different test sets. "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. It is worth noting that "†" indicates we exceptionally rerun the code of KnowPrompt and PTR with RoBERT_base for a fair comparison with current SOTA models on DialogRE. Subscript in red represents advantages of KnowPrompt over the best results of baselines. Best results are bold.

to make a fair comparison (except for DialogRE, we adopt RoBERTa_base to compare with previous methods). 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.All detailed settings for our KnowPrompt, Fine-tuning and PTR can be found in the Appendix A.2, A.3 and A.4.

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. Specifically, 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 strong baselines, which 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 et al., 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.

5.3 Main Results[4]

Standard Result. As shown in Table 3, 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 all baselines and even achieves better performance than those knowledge-enhanced models, which use knowledge as data augmentation or architecture enhancement during fine-tuning. On the other hand, even if the task-specific knowledge is already contained in knowledge-enhanced PLMs such as LUKE, KnowBERT SpanBERT and MTB , it is difficult for fine-tuning to stimulate the knowledge for downstream tasks. Overall, we believe that the development of prompt-tuning is imperative and KnowPrompt is a simple and effective prompt-tuning paradigm for RE.

Low-Resource Result. From Table 4, 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

[4] Evaluations with more datasets can be found in `https://github.com/zjunlp/LREBench`.

		Low-Resource Setting					
Split	Methods	SemEval	DialogRE†	TACRED	TACRED-Revisit	Re-TACRED	Average
K=8	FINE-TUNING	41.3	29.8	12.2	13.5	28.5	25.1
	GDPNET	42.0	28.6	11.8	12.3	29.0	24.7
	PTR	70.5	35.5	28.1	28.7	51.5	42.9
	KNOWPROMPT	**74.3** (+33.0)	**43.8** (+14.0)	**32.0** (+19.8)	**32.1** (+18.6)	**55.3** (+26.8)	**47.5** (+22.4)
K=16	FINE-TUNING	65.2	40.8	21.5	22.3	49.5	39.9
	GDPNET	67.5	42.5	22.5	23.8	50.0	41.3
	PTR	81.3	43.5	30.7	31.4	56.2	48.6
	KNOWPROMPT	**82.9** (+17.7)	**50.8** (+10.0)	**35.4** (+13.9)	**33.1** (+10.8)	**63.3** (+13.8)	**53.1** (+13.2)
K=32	FINE-TUNING	80.1	49.7	28.0	28.2	56.0	48.4
	GDPNET	81.2	50.2	28.8	29.1	56.5	49.2
	PTR	84.2	49.5	32.1	32.4	62.1	52.1
	KNOWPROMPT	**84.8** (+4.7)	**55.3** (+3.6)	**36.5** (+8.5)	**34.7** (+6.5)	**65.0** (+9.0)	**55.3** (+6.9)

Table 4: 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.

Method	**K=8**	**K=16**	**K=32**	**Full**
KNOWPROMPT	**74.3**	**82.9**	**84.8**	**90.2**
-VAW	68.2	72.7	75.9	85.2
-Knowledge Injection for VAW	52.5	78.0	80.2	88.0
-VTW	72.8	80.3	82.9	88.7
-Knowledge Injection for VTW	68.8	79.5	81.6	88.5
-Structured Constrains	73.5	81.2	83.6	89.3

Table 5: Ablation study on SemEval, VAW and VTW refers to virtual answer words and type words.

22.4% and **13.2%** absolute improvement on average compared with 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. 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 in the standard supervised setting may fall off the altar when the data are extremely scarce.

Comparison between KnowPrompt and Prompt Tuning Methods. The typical prompt-tuning methods perform outstandingly on text classification tasks (e.g., sentiment analysis and NLI), such as LM-BFF, but they don't involve RE application. Thus we cannot rerun their code for RE tasks. To our best knowledge, PTR is the only method that uses prompts for RE, which is a wonderful job and works in the same period as our KnowPrompt. Thus, we make a comprehensively comparative analysis between KnowPrompt and PTR, and summarize the comparison in Table 7. The specific analysis is as follows:

Firstly, PTR adopt a fixed number of multi-token answer form and LM-BFF leverage actual label word with single-token answer form, while KnowPrompt propose **virtual answer word with single-token answer form**. Thus, PTR needs to manually formulate rules, which is more labor-intensive. LM-BFF requires expensive label search due to its search process exponentially depending on the number of categories.

Secondly, essentially attributed to to the difference of answer form, our KnowPrompt and LM-BFF is **model-agnostic** and can be plugged into different kinds of PLMs (As show in Figure 3, our method can adopted on GPT-2), while PTR fails to generalize to generative LMs due to it's nultiple discontinuous `[MASK]` prediction.

Thirdly, above experiments, demonstrates that KnowPrompt is comprehensively comparable to the PTR, and can perform better in low-resource scenarios. Especially for DialogRE, a multi-label classification task, our method exceeded PTR by approximately 5.4 points in the standard supervised settings. It may be attributed to the rule method used by PTR that forcing multiple mask predictions will confuse multi-label predictions.

In a nutshell, the above analysis proves that KnowPrompt is more flexible and widely applicable; meanwhile, it can be aware of knowledge and stimulate it to serve downstream tasks better.

5.4 Ablation Study on KnowPrompt

Effect of Virtual Answer Words Modules: To prove the effects of the virtual answer words and its knowledge injection, we conduct the ablation study, and the results are shown in Table 5. For

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 6: Interpreting representation of **virtual type 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 type words.

Method	**# Answer Form**	**# Labor**	**# MA**	**# ML**	**# CC**
LM-BFF	single-token	normal	yes	-	high
PTR	multi-token	normal	no	normal	norm
Ours	single-token	small	yes	better	norm

Table 7: Comparative statistics between KnowPrompt and PTR, including (1)Answer Form of prompt; (2) labor-intensive; (3) MA refers to whether model-agnostic ; (4) ML refers to the ability of multi-label learning; and (4) CC refers to the computational complexity.

-*VAW*, we adopt one specific token in the relation label as the label word without optimization, and for -*Knowledge Injection for VAW*, we randomly initialize the virtual answer words to conduct optimization. Specifically, removing the knowledge injection for virtual answer words has the most significant effect, causing the relation F1 score to drop from 74.3% to 52.5% in the 8-shot setting. It also reveals that the injection of semantic knowledge maintained in relation labels is critical for relation extraction, especially in few-shot scenarios.

Effect of Virtual Type Words Modules: We also conduct an ablation study to validate the effectiveness of the design of virtual type words. As for -*VTW*, we directly remove virtual type words, and for -*Knowledge Injection for VTW*, we randomly initialize the virtual type words to conduct optimization. In the 8-shot setting, the performance of the directly removing virtual type words drops from 74.3 to 72.8, while randomly initialized virtual type words decrease the performance to 68.1, which is much lower than 72.8. This phenomenon may be related to the noise disturbance caused by random initialization, while as the instance increase, the impact of knowledge injection gradually diminishes. Despite this, it still demonstrates that our design of knowledge injection for virtual type words is effective for relation extraction.

Effect of Structured Constrains: Moreover, -*Structured Constraints* refer to the model without implicit structural constraints, which indicates no

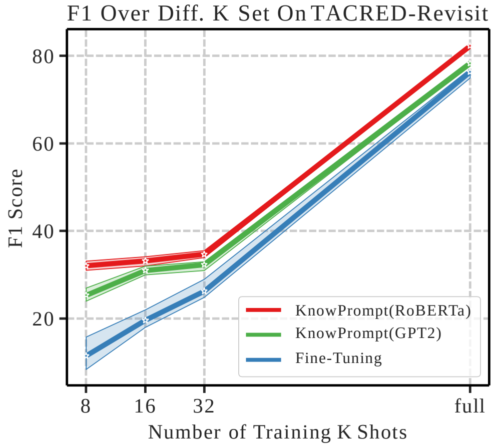

Figure 3: RoBERT-large vs. GPT-2 results on TACRED-Revisit dataset regarding different K (instances per class).

direct correlations between entities and relations. The result demonstrates that structured constraints certainly improve model performance, probably, because they can force the virtual answer words and type words to interact with each other better.

Overall, the result reveals that all modules contribute to the final performance. We further notice that virtual answer words with knowledge injection are more sensitive to performance and highly beneficial for KnowPrompt, especially in low-resource settings.

6 Analysis and Discussion

6.1 Can KnowPrompt Applied to Other LMs?

Since we focus on MLM (e.g., RoBERTa) in the main experiments, we further extend our KnowPrompt to autoregressive LMs like GPT-2. Specifically, we directly append the prompt template with [MASK] at the end of the input sequence for GPT-2. We further apply the relation embedding head by extending the word embedding layer in PLMs; thus, GPT2 can generate virtual answer words. We first notice that fine-tuning leads to poor perfor-

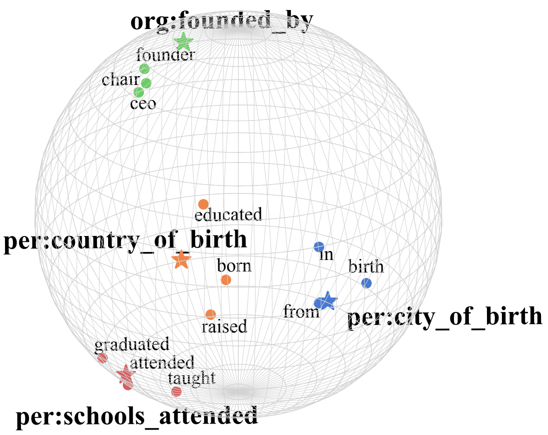

Figure 4: A 3D visualization of several relation representations (**virtual answer words**) optimized in KnowPrompt on TACRED-Revisit dataset using t-SNE and normalization.

mance with high variance in the low-resource setting, while KnowPrompt based on RoBERTa 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 RoBERTa-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.

6.2 Interpreting Representation Space of Virtual Answer Words

Since the embeddings of virtual answer words $\{\hat{\mathrm{e}}_{[rel]}(\mathcal{V}')\}$ are initialized with **semantic knowledge of relation type itself**, and further learned in continuous space, it is intuitive to explore what precisely the optimized virtual answer word is. We use t-SNE and normalization to map the embedding to 3 dimension space and make a 3D visualization of several sampled virtual answer words in the TACRED-Revisit dataset. We also get the top3 tokens nearest the virtual answer word by calculating the L_2 distance between the embedding of the virtual answer word and the actual word in the vocabulary. 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. To some extent, our proposed virtual answer words are similar to prototypical representation for relation labels. This inspired us to further expand knowprompt into the field of prototype representation learning in the future, which can also be applied to other NLP tasks with prompt-tuning.

6.3 Interpreting Representation Space of virtual type words

Since we initialize the virtual type words with the average embedding of candidate types of head and tail entities through **prior knowledge maintained in the relation labels**, and synergistically optimize them ($\{\hat{\mathrm{e}}_{[sub]}, \hat{\mathrm{e}}_{[obj]}\}$) with virtual answer words based on context. To this end, we further conduct further analysis to investigate that what semantics do the optimized type words express and whether virtual type words can adaptively reflect the entity types based on context as shown in Table 6. Specifically, we apply the MLM head over the position of the virtual type words to get the output representation and get the top-3 words in vocabulary nearest the virtual type words according to the L_2 distance of embeddings between virtual type words and other words. We observe that thanks to the synergistic optimization with knowledge constraints, those learned virtual type words can dynamically adjust according to context and play a reminder role for RE.

7 Conclusion and Future Work

In this paper, we present KnowPrompt for relation extraction, which mainly includes knowledge-aware prompt construction and synergistic optimization with knowledge constraints. In the future, we plan to explore two 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.

Acknowledgment

We would like to express gratitude to the anonymous reviewers for their kind comments. This work is funded by NSFC91846204/NSFCU19B2027, National Key R&D Program of China (Funding No.SQ2018YFC000004), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Ningbo Natural Science Foundation (2021J190), and Yongjiang Talent Introduction Programme (2021A-156-G).

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of ACL 2020*.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of ACL/IJCNLP 2021*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL/IJCNLP 2019*.

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang, Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. Recon: Relation extraction using knowledge graph context in a graph neural network. In *Proceedings of the Web Conference 2021*, pages 1673–1685.

Matthias Baumgartner, Wen Zhang, Bibek Paudel, Daniele Dell'Aglio, Huajun Chen, and Abraham Bernstein. 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*, pages 21–37. Springer.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. Pada: A prompt-based autoregressive approach for adaptation to unseen domains. *arXiv preprint arXiv:2102.12206*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS 2020*.

Mary Elaine Califf and Raymond J. Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *Proceedings of AAAI*, pages 328–334. AAAI Press / The MIT Press.

Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Jeff Z. Pan, Yuan He, Wen Zhang, Ian Horrocks, and Huajun Chen. 2021. Low-resource learning with knowledge graphs: A comprehensive survey. *CoRR*, abs/2112.10006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021a. Prompt-learning for fine-grained entity typing. *CoRR*, abs/2108.10604.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021b. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of ACL*, pages 229–240.

Bowen Dong, Yuan Yao, Ruobing Xie, Tianyu Gao, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Meta-information guided meta-learning for few-shot relation classification. In *Proceedings of COLING 2020*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of ACL*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of AAAI*.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Neural snowball for few-shot relation learning. In *Proceedings of AAAI 2020*.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. PPT: pre-trained prompt tuning for few-shot learning. *CoRR*, abs/2109.04332.

Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B Cohen. 2020. Learning latent forests for medical relation extraction. In *IJCAI*, pages 3651–3657.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of ACL 2019*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: word-level adversarial reprogramming. In *Proceedings of ACL/IJCNLP 2021*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP, 2018*.

Tom Harting, Sepideh Mesbah, and Christoph Lofi. 2020. LOREM: language-consistent open relation extraction from unstructured text. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1830–1838. ACM / IW3C2.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *CoRR*, abs/2110.04366.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval*, pages 33–38.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021a. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *CoRR*, abs/2108.02035.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021b. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Juan Li, Ruoxu Wang, Ningyu Zhang, Wen Zhang, Fan Yang, and Huajun Chen. 2020. Logic-guided semantic representation learning for zero-shot relation classification. In *Proceedings of COLING*, pages 2967–2978.

Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. 2019. Improving relation extraction with knowledge-attention. In *Proceedings of EMNLP*, pages 229–239.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of ACL/IJCNLP 2021*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP*, pages 43–54.

Meng Qu, Tianyu Gao, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *Proceedings of ICML 2020*.

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

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? *CoRR*, abs/2103.08493.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of COLING*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *CoRR*, abs/2009.07118.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of EACL 2021*.

Yongliang Shen, Xinyin Ma, Yechun Tang, and Weiming Lu. 2021. A trigger-sense memory flow framework for joint entity and relation extraction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1704–1715. ACM / IW3C2.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of EMNLP 2020*.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. *arXiv preprint arXiv:2104.08398*.

Zifeng Wang, Rui Wen, Xi Chen, Shao-Lun Huang, Ningyu Zhang, and Yefeng Zheng. 2020. Finding influential instances for distantly supervised relation extraction. *arXiv preprint arXiv:2009.09841*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of theCIKM 2019*.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Reza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. *CoRR*, abs/2101.01926.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Proceedings of AAAI 2021*.

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 EMNLP 2020*.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Proceedings of AAAI, 2021*.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020a. Dialogue-based relation extraction. In *Proceedings of ACL 2020*.

Haiyang Yu, Ningyu Zhang, Shumin Deng, Hongbin Ye, Wei Zhang, and Huajun Chen. 2020b. Bridging text and knowledge with multi-prototype embedding for few-shot relational triple extraction. In *Proceedings of COLING*, pages 6399–6410. International Committee on Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP 2015*.

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021a. Document-level relation extraction as semantic segmentation. In *Proceedings of IJCAI*, pages 3999–4006. ijcai.org.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of NAACL-HLT*.

Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018a. Attention-based capsule network with dynamic routing for relation extraction. In *Proceedings of EMNLP 2018*.

Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang Huang, Zhao Wang, Nengwei Hua, and Huajun Chen. 2021b. Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. In *Proceedings of KDD*, pages 3895–3905. ACM.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP, 2018*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017a. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP 2017*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. PRGC: potential relation and global correspondence based joint relational triple extraction. In *Proceedings of ACL/IJCNLP 2021*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of ACL 2016*.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *CoRR*, abs/2102.01373.

Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. 2020. NERO: A neural rule grounding framework for label-efficient relation extraction. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2166–2176. ACM / IW3C2.

A Appendix

A.1 Detailed Statistics of Dataset

For comprehensive experiments, we carry out our experiments on five relation extraction datasets: TACRED (Zhang et al., 2017b), TACREV (Alt et al., 2020), Re-TACRED (Stoica et al., 2021), SemEval 2010 Task 8 (SemEval) (Hendrickx et al., 2010), and DialogRE (Yu et al., 2020a). A brief introduction to these data is as follows:

TACRED: one large-scale sentence-level relation extraction dataset drawn from the yearly TACKBP4 challenge, which contains more than 106K sentences. It involves 42 different relations (41 common relation types and a special "no relation" type). The subject mentions in TACRED are person and organization, while object mentions are in 16 fine-grained types, including date, number, etc.

TACRED-Revisit: one dataset built based on the original TACRED dataset. They find out and correct the errors in the original development set and test set of TACRED, while the training set was left intact.

Re-TACRED: another version of TACRED dataset. They address some shortcomings of the original TACRED dataset, refactor its training set, development set and test set. Re-TACRED also modifies a few relation types, finally resulting in a dataset with 40 relation types.

SemEval: a traditional dataset in relation classification containing $10,717$ annotated examples covering 9 relations with two directions and one special relation "no_relation".

DialogRE: DialogRE is the first human-annotated dialogue-level RE dataset. It contains 1,788 dialogues originating from the complete transcripts of a famous American television situation comedy. It is multi-label classification, as each entity pair may posses more than one relation.

A.2 Implementation Details for KnowPrompt

This section details the training procedures and hyperparameters for each of the datasets. We utilize Pytorch to conduct experiments with 8 Nvidia 3090 GPUs. All optimizations are performed with the AdamW optimizer with a linear warmup of learning rate over the first 10% of gradient updates to a maximum value, then linear decay over the remainder of the training. Gradients are clipped if their norm exceeded 1.0, margin γ, λ and weight decay on all non-bias parameters are set to 1, 0.001 and 0.01. A grid search is used for hyperparameter tuning (maximum values bolded below).

A.2.1 Standard Supervised Setting

The hyper-parameter search space is shown as follows:

- learning rate lr_1 of synergistic optimization for virtual template and anchor words. [5e-5,**1e-4**, 2e-4]
- learning rate lr_2 of optimization for overall parameters. [1e-5, 2e-5, **3e-5**, 5e-5]
- number epochs 5 (for dialogre as 20)
- batch size: 16 (for tacrev, retacred and dialogre as 8)
- max seq length: 256 (for tacrev, retacred and dialogre as 512)
- gradient accumulation steps: 1 (for dialogre as 4)

A.2.2 Low-Resource Setting

The hyper-parameter search space is shown as follows:

- learning rate lr_1 of synergistic optimization for virtual template and anchor words: [5e-5,**1e-4**, 2e-4]
- learning rate lr_2 of optimization for overall parameters: [1e-5, **2e-5**, 3e-5, 5e-5]
- number of epochs: 30
- batch size: 16 (for tacrev, retacred and dialogre as 8)
- max seq length: 256 (for tacrev, retacred and dialogre as 512)
- gradient accumulation steps: 1 (for dialogre as 4)

A.3 Implementation Details for Fine-tuning

The fine-tuning method is conducted as shown in Figure 2, which is both equipped with the same entity marker in the raw text for a fair comparison. The hyper-parameters such as batch size, epoch, and learning rate are the same as KnowPrompt.

A.4 Implementation Details for PTR

Since PTR does not conduct experiments on DialogRE in standard supervised setting and SemEval and DialogRE in few-shot settings, we rerun its public code to supplement the experiments we described above with these data and scenarios. As for SemEval, the experiment process completely follows the original setting in his code, while for DialogRE, we modify his code to more adapt to the setting of this data set. The specific hyper-parameters such as batch size, epoch, and learning rate are the same as KnowPrompt.