Knowledge-enhanced Prompt Tuning for Dialogue-based Relation Extraction with Trigger and Label Semantic

Hao An, Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Yuexian Zou*
School of ECE, Peking University, China
{anhao, zhihongzhu, chengxx}@stu.pku.edu.cn, {zhiqihuang, zouyx}@pku.edu.cn

Abstract

Dialogue-based relation extraction (DRE) aims to determine the semantic relation of a given pair of arguments from a piece of dialogue, which has received increasing attention. Due to the low information density of dialogue text, it is difficult for the model to focus on key information. To this end, in this paper, we propose a **K**nowledge-**E**nhanced **P**rompt-**T**uning (KEPT) method to effectively enhance DRE model by exploiting trigger and label semantic. Specifically, we propose two beneficial tasks, masked trigger prediction, and verbalizer representation learning, to effectively inject trigger knowledge and label semantic knowledge respectively. Furthermore, we convert the DRE task to a masked language modeling task to unify the format of knowledge injection and utilization, aiming to better promote DRE performance. Experimental results on the DialogRE dataset show that our KEPT achieves state-of-the-art performance in F1 and F1$_c$ scores. Detailed analyses demonstrate the effectiveness and efficiency of our proposed approach. Code is available at https://github.com/blackbookay/KEPT.

Keywords: Knowledge Enhancement, Prompt-tuning, Trigger, Label Semantic

1. Introduction

Relation extraction (RE) has been proposed to identify relation facts between two arguments from unstructured text, which plays a crucial role in a range of knowledge-driven applications, such as knowledge graph construction (Ji et al., 2022) and task-oriented dialogue system (Smirnova and Cudré-Mauroux, 2018). In recent years, a variety of deep models have been proposed to discover relational facts in a single sentence and achieve notable performance (Jiang et al., 2020). However, sentence-level RE is limited in reality by an unavoidable limitation: an enormous proportion of relation triples are presented across multiple sentences (Yao et al., 2019). Therefore, some works explore document-level relation extraction, which aims to identify relational facts within a document (Yao et al., 2019; Zeng et al., 2020a; Zhou et al., 2021).

Recently, dialogue-level relation extraction (Yu et al., 2020) (DRE) has been proposed and attracting increasing attention in the area of information extraction, which aims to identify relation facts within a multi-turn dialogue. Compared to document-level RE, dialogue-based relation extraction is more challenging due to low information density (Wang and Liu, 2011) and majority colloquialism (Bai et al., 2021). Table 1 shows an example of DRE, which includes eleven turns of utterances by two speakers and several arguments, however, only several words in the dialogue are important to relation identification. The aforementioned words are called triggers, which give informative cues to relation extraction. For example, the trigger "mad at" indicates S1 has a negative impression of Estelle.

Dialogue		
S1: Hey Phoebe.		
S2: At least I care about his feelings!		
S1: I'm just mad at my agent.		
S2: Estelle? Why?		
S1: There's a part in a TV movie that I would be perfect for and I didn't even be put up for it! She'd better have a good reason.		
S2: I'm guessing she does.		
S1: Well. I'm wanna hear it, because she keeps doing this.		
S2: Well, no, no, wait, wait, wait. All right, I gotta go. Just listen. Promise me, that you will wait a minute before you call her.		
S1: Ok. Why?		
S2: Because a promise between friends means never having to give a reason.		
S1: I love that saying!		
Arguments	**Trigger**	**Relation**
(Estelle, agent)	-	per:title
(Estelle, S1)	agent	per:client
(S1, Estelle)	mad at	per:negative_impression
(S1, S2)	friends	per:friends
(S2, S1)	friends	per:friends

Table 1: An example of dialogue relation extraction. S1 and S2 are anonymized speakers of each utterance. Triggers are crucial clues of relations annotated in DialogRE.

Therefore, some studies try to enhance DRE models by trigger annotated in the train set, which gives crucial cues for relation recognition. Zhao et al. (2021) proposed a trigger words prediction as a sequence labeling task. Lin et al. (2022) and

[*] Corresponding author.

LREC-COLING 2024, pages 9822–9831
20-25 May, 2024.

An et al. (2023) trained a trigger span extractor to detect the trigger and fuse it with dialogue context feature to enhance final relation extraction. Son et al. (2022) predicted trigger among the prompted dialogue sequence and append it to the input of the answer mapping module.

To conclude, these approaches adopt a pipeline-based manner to introduce trigger knowledge. They first use a trigger extraction module to identify the existence and the location of the trigger in the dialogue text, then fuse the trigger with dialogue context features, and finally feed the fused trigger features to the relation classifier.

However, current trigger-enhanced pipeline approaches (Lin et al., 2022; An et al., 2023; Son et al., 2022) have inevitable limitations in practice. Specifically, There are two limitations to these approaches. Firstly, such approaches do not guarantee complete accuracy in trigger recognition, as erroneous triggers mixed with contextual features can generate noise and lead to error accumulation. This can introduce noise in the final relation extraction task, resulting in limited improvement in the performance. Secondly, these methods require an additional module for extracting triggers to anticipate the trigger span within the dialogue text during the inference phase. This results in lower overall model inference efficiency, making it challenging to employ in real-world relation extraction applications due to resource and time limitations.

Recently, prompt tuning has achieved significant success. It is successful for two main reasons, reducing the gap between pre-trained language model (PLM) pre-training tasks and downstream tasks (Liu et al., 2023b) and making full use of label semantic knowledge (Liu et al., 2023a). Inspired by these points, we propose a Knowledge-enhanced Prompt tuning (KEPT) DRE to avoid the issues mentioned above and exploit triggers effectively. To be specific, we propose a masked trigger prediction task that injects trigger knowledge into the DRE model during the training stage, which naturally avoids the slow inference issue and error accumulation issue. Furthermore, we also propose an auxiliary task to utilize label semantic knowledge. For relation extraction, we redefine it as a masked language modeling task to mitigate the gap between relation classification, pre-training, and the aforementioned knowledge injection tasks. By this way, our KEPT is able to incorporate knowledge and utilize knowledge of the PLM effectively and efficiently for dialogue relation extraction.

In summary, our contributions are as follows:

- We propose a novel method, the Knowledge-Enhanced Prompt-tuning (KEPT) method, which constructs an effective and efficient framework to better tackle the low information density problem with trigger and label semantic knowledge for DRE.
- We introduce a masked trigger prediction task to inject trigger knowledge into the DRE model without any extra module.
- We introduce a verbalizer representation learning task to enhance the DRE model with label semantic knowledge.

We evaluate KEPT on the public DialogRE dataset. It significantly outperforms the previous state-of-the-art model by 6.4 F1 score.

2. Related Works

DRE models are classified into two categories: sequence-based and graph-based models.

Sequence-based models encode dialogue as a long sequence using PLMs and then use complex mechanisms like attention and gate mechanisms to extract crucial information. (Yu et al., 2020) modify the BERT model to include dialogue-specific tokens. (Xue et al., 2022) propose a simple and efficient relation refinement mechanism that achieves good results. (Han et al., 2022) uses multiple [MASK] tokens for each entity and the relation with logical rules by combining the subject and object. KnowPrompt (Chen et al., 2022) inject latent knowledge contained in relation labels into prompt construction with learnable virtual type words and answer words. These models concatenate all utterances in multi-turn dialogues but do not consider dialogue-specific characteristics.

Graph-based models create a graph by linking nodes in various ways. Nodes in a graph represent tokens, utterances, or arguments in the given dialogue context. Nan et al. (2020) create meta-dependence routes between argument pairs and aggregate word representations to improve the model's reasoning capacity. Xue et al. (2021) suggest creating a latent multi-view graph to identify potential links between tokens. Lee and Choi (2021) suggest a heterogeneous conversation graph to model interaction between nodes (e.g., speakers, utterances, arguments) and proposes a GCN method with contextualized representations of turns.

However, DRE suffers from the low-density issue (Wang and Liu, 2011). Recently, a series of trigger-enhanced models have been further explored to solve this, which use triggers as additional information to enhance DRE models. Among them, TREND (Lin et al., 2022) and TLAG (An et al., 2023) use explicit trigger extractors to leverage the trigger information. GRASP (Son et al., 2022) predict triggers in the prompted dialogue sequence and append it to the input of the answer mapping module.

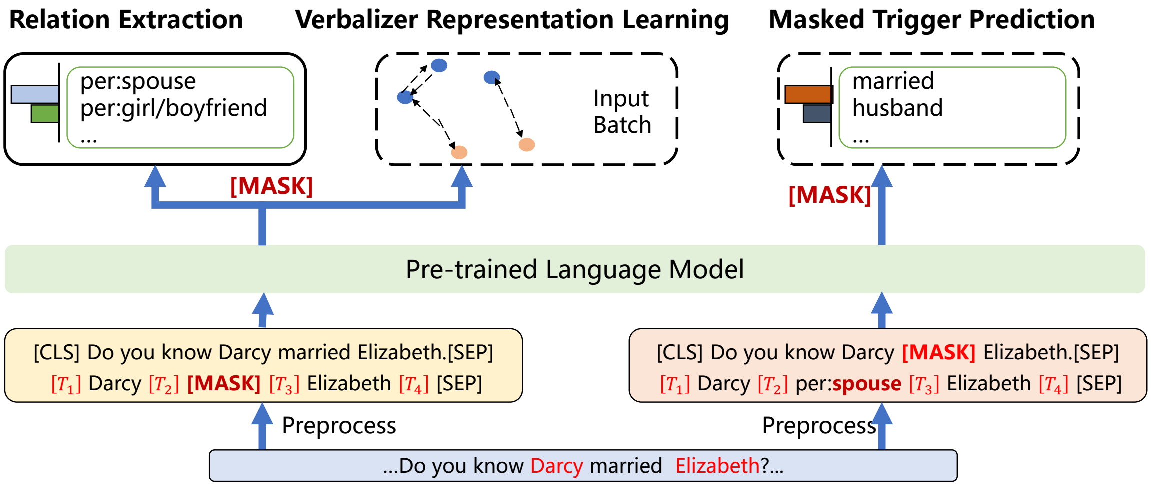

Relation Extraction
Verbalizer Representation Learning
Masked Trigger Prediction
per:spouse
per:girl/boyfriend
...
Input Batch
married
husband
...
[MASK]
[MASK]
Pre-trained Language Model
[CLS] Do you know Darcy married Elizabeth.[SEP]
[T_1] Darcy [T_2] [MASK] [T_3] Elizabeth [T_4] [SEP]
[CLS] Do you know Darcy [MASK] Elizabeth.[SEP]
[T_1] Darcy [T_2] per:spouse [T_3] Elizabeth [T_4] [SEP]
Preprocess
Preprocess
...Do you know Darcy married Elizabeth?...

Figure 1: The overall architecture of proposed KEPT. It consists of three tasks, i.e., relation extraction, masked trigger prediction, and verbalizer representation learning. The latter two are calculated only during training.

3. Method

This section introduces the details of our KEPT which reformulates the DRE task as a cloze task and designs two beneficial auxiliary tasks. The overview of KEPT is shown in Figure 1. First, we describe the relation extraction task based on prompt-tuning in Section 3.2. Then, section 3.3 introduces the masked trigger prediction task that injects trigger knowledge into the model. Section 3.4 presents the verbalizer representation learning task that enhances the model with label semantic knowledge. Finally, section 3.5 introduces the training objective.

3.1. Problem Formulation

Given a dialogue $D = s_1 : u_1, s_2 : u_2, ..., s_n : u_n$ with an argument pair $a = (a_1,\ a_2)$, where s_i and u_i denote the speaker ID and the i^{th} turn of the dialogue, and n is the total number of turns, the goal of dialogue-based relation extraction (DRE) is to identify the relation type r between the given argument pair from a predefined relation set R.

3.2. Prompt-tuning-based Relation Extraction

In order to enable the relation classifier to fully exploit the pre-trained knowledge of the PLM and trigger knowledge obtained by masked trigger prediction, we adopt a prompt-tuning method and redefine the relation extraction task as the masked language modeling (MLM) task, which is the pre-training task format of PLM, thus improve the performance on dialogue relation extraction.

Given the input $X = \{D, a_1, a_2\}$, we first adopt the speaker-aware method of BERTs (Yu et al., 2020) to construct $\hat{D} = \{\hat{s}_1 : u_1, \hat{s}_2 : u_2, ..., \hat{s}_n : u_n\}$, where $\hat{s}_i$ is:

$$\hat{s}_i = \begin{cases} [S_1], & \text{if } s_i = a_1 \\ [S_2], & \text{if } s_i = a_2 \\ s_i, & \text{otherwise} \end{cases} \quad (1)$$

where $[S_1]$ and $[S_2]$ are two newly-defined tokens. $\hat{a}_k (k \in \{1, 2\})$ is defined as $[S_k]$ if $\exists i(s_i = a_k)$, and a_k otherwise. Then we construct a template Z for the input:

$$Z = [T_1]a_1[T_2][MASK][T_3]a_2[T_4], \quad (2)$$

where $[T_1], [T_2], [T_3]$ and $[T_4]$ are trainable prompt token. We concatenate the input Z and the template to obtain the final input X_1:

$$\begin{aligned} X_1 = & [CLS]\hat{D}[SEP][T_1]\hat{a}_1[T_2] \\ & [MASK][T_3]\hat{a}_2[T_4][SEP] \end{aligned} \quad (3)$$

Feed X_1 to the PLM, we can predict the relation by the probability distribution of [MASK]:

$$p(r_i \mid X_1) = softmax(p([MASK] = V_i)) \quad (4)$$

where V_i is the answer token of relation r_i. We compute the cross entropy $\mathcal{L}_R$ between the predicted relation and label. Hence, the DRE task is reformulated as an MLM task that mitigates the gap between knowledge injection and knowledge utilization. The $[MASK]$ is inserted into X_1 to predict the label words of relation y.

3.3. Masked Trigger Prediction

In previous studies, triggers in dialogues are utilized by pipeline manners, which extract triggers and then fuse them with dialogue features (Lin et al., 2022). However, the extraction process does

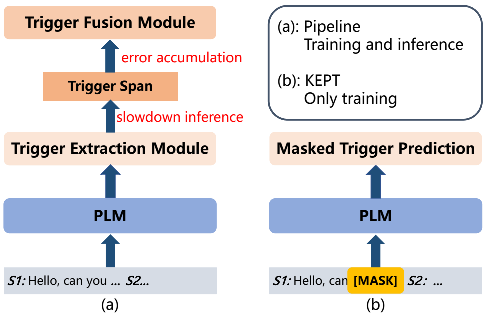


Figure 2: Illustration of the two injection methods.

not guarantee the correctness of the trigger word and takes a long time, which decreases the performance of the model and reduces the inference efficiency. To this end, we propose a masked trigger prediction task that incorporates trigger knowledge into PLM effectively and efficiently during training. As shown in Figure 2, our masked trigger prediction task only involved calculation during training, which avoids the problems of error accumulation and slowdown inference.

The following describes the method of constructing the masked trigger prediction task. Given the training sample $X = \{\hat{D}, a_1, a_2\}$, dialogue text the argument pair are (a_1, a_2) if there exists a trigger of the relation between a_1 and a_2 D, we replace the trigger span [MASK]. Take the training sample in Figure 1 as an example, the trigger is "married", and then "married" is replaced by [MASK]. To further enhance dialogue understanding, we randomly mask 10% of the words in the input dialogue text $\hat{D}$ if there does not exist a trigger for this argument pair. Processing the dialogue text D as described above, we obtain the input X_2 for this task, which is formulated as follows:

$$X_2 = [[CLS], D_{MLM}, [SEP], a_1, Y_{text}, a_2] \qquad (5)$$

where D_{MLM} is the processed dialogue, Y_{text} is the text of the relation label between a_1 and a_2. For example, the Y_{text} corresponding to relation *per:friends* is "person friends". Given the set of masked words, the training objective is as Equal 6:

$$\mathcal{L}_M = -\sum_{i=1}^{M} log\ p(m = m_i|\theta) \qquad (6)$$

where $i \in [1, 2, \cdots, |V|]$, $|V|$ is the size of vocabulary, θ is the parameters of the PLM, M is the number of the masked words.

In this manner, we inject trigger knowledge into the PLM during the training stage without adding any model parameters. Since the task format is the same as prompt-based relation extraction, such trigger knowledge can be effectively utilized for relation extraction.

3.4. Verbalizer Representation Learning

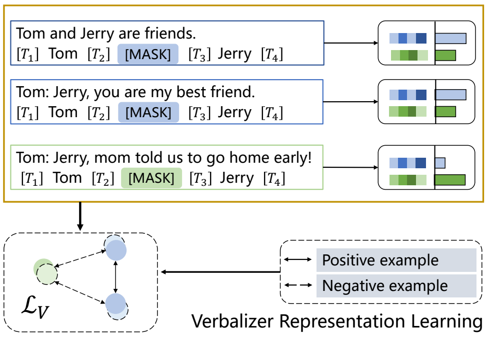

Tom and Jerry are friends.
[T_1] Tom [T_2] [MASK] [T_3] Jerry [T_4]
Tom: Jerry, you are my best friend.
[T_1] Tom [T_2] [MASK] [T_3] Jerry [T_4]
Tom: Jerry, mom told us to go home early!
[T_1] Tom [T_2] [MASK] [T_3] Jerry [T_4]
$\mathcal{L}_V$
Positive example
Negative example
Verbalizer Representation Learning

Figure 3: Illustration of the verbalizer representation learning task. The training objective is supervised contrastive loss.

One reason for the recent popularity of prompt-tuning is its ability to leverage semantic knowledge of labels. Inspired by previous work on differentiable prompt (Zhang et al., 2021), we propose a verbalizer learning task to leverage the label semantic knowledge. To be specific, we treat the verbalizer as a differentiable token, which can be tuned with the label semantic.

On the other hand, we also find that some relations share similar semantics, e.g., "dates" versus "positive_impression", "girl/boyfriend" versus "spouse", and "school_attended" versus "member_of". These relation pairs are semantically difficult to distinguish. To this end, we adopt supervised contrastive learning (Khosla et al., 2020) (SCL) to train the verbalizer. As shown in Figure 3, SCL pulls the verbalizer semantic distances of samples in the same classes closer together, while those in different classes are further apart, making the model more discriminating for similar classes.

To construct this task, we use the label text to initialize the [MASK] token:

$$W_r = \frac{1}{k}\sum_{j=1}^{k} W_{t_j} \qquad (7)$$

For relation *per:employee_or_member_of*, we use an unused token to represent it and initialize this token by averaging the embedding of the words in {"person", "employee", "or", "member", "of"}.

Given the input sample $X = \{x_1, x_2, ..., x_n\}$ and the corresponding labels $y = \{y_1, y_2, ..., y_n\}$, the training objective is as follows:

$$\mathcal{L}_V = -\frac{1}{N}\sum_{i}^{N}\sum_{j\neq i}^{N} 1_{y_i=y_j} \log \frac{\exp(h_i \cdot h_j/\tau)}{\sum_{k\neq i}^{N} \exp(h_i \cdot h_k/\tau)}, \qquad (8)$$

where N is the batch size, $\tau > 0$ is a scalar temperature to stabilize the calculation. $h_i =$

$PLM(MASK)$ denotes the MASK representation encoded by the PLM.

By imposing supervised contrastive learning in training, verbalizer representations belonging to the same class are pulled together in the hidden space, while samples from different classes are pulled away from each other. This task enhances the distinguishability of similar classes and promotes performance.

3.5. Loss Function

The above three tasks are jointly trained under a multi-task learning framework. We adopt the cross-entropy loss for the relation extraction task. During training, the total loss function is the weighted sum of the three tasks:

$$\mathcal{L}_{total} = \lambda_R \mathcal{L}_R + \lambda_M \mathcal{L}_M + \lambda_V \mathcal{L}_V, \tag{9}$$

where λ_R, λ_M and λ_V are hyperparameters of weight factor. During inference, we only calculate the relation extraction task.

4. Experiments

4.1. Experiments Setup

4.1.1. Dataset and Metrics

We evaluate our method on DialogRE Yu et al. (2020), a human-annotated dataset for DRE from the transcript of the series "*Friends*". DialogRE has 36 relation types, 1788 dialogues, and 8119 relation facts in total. Dialogues in DialogRE contain about 13 utterances on average, and more than 60% relation instances require cross-utterance reasoning. We follow the standard split of the dataset. We calculate both the $F1$ and $F1_c$(Yu et al., 2020). $F1_c$ is computed by taking in the part of the dialogue as input.

4.1.2. Baseline

For a comprehensive performance evaluation, we compared our model with the models using the following baseline and state-of-the-art methods:

Fine-tuning methods: **RoBERTa** (Liu et al., 2019) is a popular pre-trained language model. **GDPNet** (Xue et al., 2021) constructs a latent graph to capture various possible relationships. **TREND** (Lin et al., 2022) and **TLAG** (An et al., 2023) adopt pipeline-based method to inject trigger into DRE. **TUCORE-GCN** (Lee and Choi, 2021), which is designed according to the way people understand dialogues in practice.

Prompt-tuning methods: **PTR** (Han et al., 2022) and **KnowPrompt** (Chen et al., 2022) are the prompt-tuning approaches without trigger assistance. **GRASP** (Son et al., 2022) is a prompt-tuning approach with trigger enhancement.

4.1.3. Experimental Setting

We use the pre-trained model RoBERTa-base as our base model. The training batch size is 8 and the learning rate is 2e-5. All optimizations are performed with the AdamW (Loshchilov and Hutter, 2019) 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. The weights of three loss are set as $\lambda_r = 1.0, \lambda_m = 0.4, \lambda_s = 0.3$. The temperature scale is 0.07. We train the model for 30 epochs. We use grid search to select the best hyperparameters on the validation set. All experiments are conducted on a GeForce RTX 3090 with 24 GB memory. The experimental results for our model are averaged over five runs.

4.2. Main Results

Table 2 shows the results of different methods on DialogRE. We can observe that prompt-tuning methods outperform fine-tuning methods, suggesting that fine-tuning-based approaches are more difficult to exploit knowledge from PLM. Among the prompt-tuning methods, KEPT achieves the best performance, which surpasses the best baseline GRASP by 4.6%/6.4% (F1/F1$_c$) and 2.9%/3.9% (F1/F1$_c$) in English and Chinese, respectively. Compared to other models that incorporate trigger knowledge, KEPT performs the best, suggesting KEPT can successfully inject trigger knowledge into the model and make full use of such knowledge. KEPT demonstrates a significantly higher performance improvement relative to the baseline model at the F1$_c$ metric rather than at the F1 metric, suggesting that our KEPT method can more effectively capture argument and trigger information in practical conversational scenarios.

4.3. Low-Resource Results

We conduct experiments in a low-resource setting and present the results on the DialogRE-EN dataset in Table 3. We draw the following conclusions after analyzing the F1 metrics for four training set sizes.

1) Our KEPT outperforms all baseline methods for three different training sizes in low-resource scenarios, indicating that the KEPT can maintain robustness by utilizing the knowledge in the PLM, especially in the case of extreme lack of samples with K=8, and it outperforms the baseline method by 4.9%.

2) It is suggested that when data are extremely scarce, the fine-tuning-based approaches may have difficulty capturing the relational semantics effectively. In contrast, the prompt-tuning-based approaches can extract the relational semantic knowledge in the PLM by prompting the model.

Method	Trigger	DialogRE-EN		DialogRE-CN	
		F1	F1$_c$	F1	F1$_c$
Fine-tuning					
RoBERTa (Liu et al., 2019)	w/o	62.8	58.8	62.7	58.9
GDPNet (Xue et al., 2021)	w/o	64.9	60.1	62.8	59.8
TREND* (Lin et al., 2022)	w	65.8*	60.4*	65.5*	60.8*
TLAG (An et al., 2023)	w	66.6	60.8	67.0	61.3
TUCORE-GCN (Lee and Choi, 2021)	w	68.7*	61.5*	-	-
Prompt-tuning					
PTR (Han et al., 2022)	w/o	63.2	-	-	-
KnowPrompt (Chen et al., 2022)	w/o	68.6	-	-	-
GRASP (Son et al., 2022)	w	69.0	61.7	69.7*	61.5*
KEPT (Ours)	w	**73.6**	**67.3**	**72.6**	**65.4**

Table 2: Results on DialogRE in English and Chinese. The scores marked by "*" are based on reproduction. The best results are bold.

Method	Shot			
	8	16	32	Full
Fine-tuning				
RoBERTa	29.8	40.8	49.7	62.8
GDPNet	28.6	42.5	50.2	64.9
TUCORE-GCN	24.6	40.0	53.8	68.7
Prompt-tuning				
PTR	35.5	43.5	49.5	63.2
KnowPrompt	43.8	50.8	55.3	68.6
GRASP	45.4	52.0	56.0	69.0
KEPT	**50.3**	**54.1**	**59.1**	**73.6**

Table 3: Few-shot DRE results of F1 scores (%) on different test sets. We use $K = 8, 16, 32$ (# of examples per class). *Full* represents the full training set is used. The best results are bold.

4.4. Ablation Studies

	Method	DialogRE-EN	
		F1	F1$_c$
	KEPT	73.6	67.3
VRL	-w/o SCL	72.5 (-1.1)	67.0 (-0.3)
	-w/o Trainable	71.0 (-2.6)	65.6 (-1.7)
MTP	-w/o Trigger	71.8 (-1.8)	64.9 (-2.4)
	-w/o MLM	70.8 (-2.8)	64.7 (-2.6)

Table 4: Ablation Results. Numbers in parentheses indicate performance degradation relative to the full method.

To further analyze KEPT, we also conduct ablation studies to illustrate the effectiveness of different mechanisms in KEPT. We show the results of the ablation study in Table 4.

Effect of verbalizer representation learning task: In this setting, we first remove the supervised contrastive loss, which is denoted as "-w/o SCL" in Table 4, resulting in a decrease of 1.1% and 0.3% in the F1 and F1$_c$ scores, respectively. This finding demonstrates the efficacy of supervised contrastive learning in effectively differentiating similar samples. Besides, we replace trainable verbalizer tokens with hard verbalizer words (-w/o Trainable) as shown in Table 4. The results show a significant decrease of 2.6% F1 and 1.7% F1$_c$ compared to the full KEPT method, indicating that soft verbalizers possess more enriched relational semantics than non-adjustable hard verbalizers, enabling them to effectively circumvent bias in relation classification.

Effect of masked trigger prediction task: In this setting, the trigger mask is first removed and only randomly masked words are predicted, which is shown as "-w/o Trigger Mask" in Table 4, and the performance decreases by 1.8% and 2.4% in F1 score and F1$_c$ score, respectively. It is proven that trigger knowledge holds significant importance in extracting relation from dialogue text, and our masked trigger prediction task can utilize trigger knowledge effectively. Interestingly, it appears that in the dialogue scenario, the F1$_c$ metric shows a greater decrease of 0.6% relative to the F1 value, which can be attributed to the incomplete information of the input text in the dialogue scenario that only part of the dialogue is available, indicating trigger knowledge plays an important role.

Furthermore, we remove the masked trigger prediction task, as denoted by "w/o MLM" in Table 4. The performance decreases by 1.9% in F1 and 2.6% in F1$_c$. In comparison, when only removing the knowledge of trigger words, the performance decreases by 1.0% in F1 and 0.2% in F1$_c$, suggesting that masking words randomly and predicting them facilitates the understanding of contextual semantics.

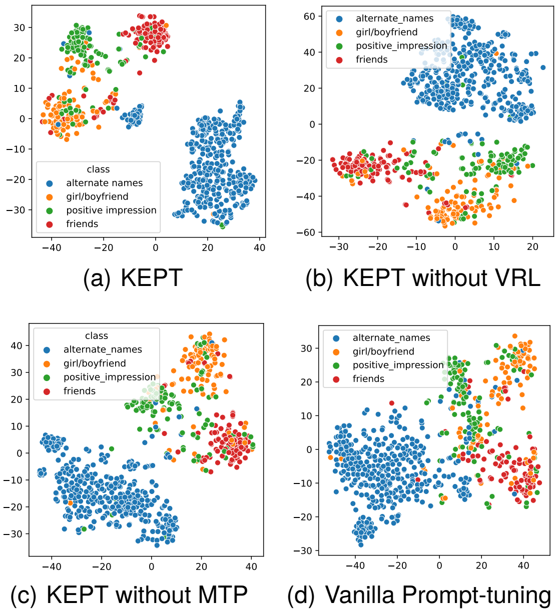
4.5. Inference Efficiency

In practical applications, the inference speed is crucial for DRE, and we conduct experiments on this, and the results are shown in Table 5.

Except for RoBERTa, all the other models use trigger information. When compared with the RoBERTa, our KEPT is slightly slower. Compared to fine-tuning-based methods like TREND and TLAG, our KEPT exhibits 6.8 times faster inference speed. This is due to the fine-tuning-based

Method	T	EP	DialoogRE-EN	
			Latency (s)	Speedup
RoBERTa	No	No	15	0.75x
TREND	Yes	Yes	136	6.8x
TLAG	Yes	Yes	136	6.8x
GRASP	Yes	Yes	99	4.95x
KEPT	Yes	No	20	1.0x

Table 5: Inference speed comparison. **T** denotes trigger. **EP** denotes extra parameters. **Latency** indicates the total time to infer the entire test set. **Speedup** ratio is based on the latency of KEPT.



(a) KEPT (b) KEPT without VRL

(c) KEPT without MTP (d) Vanilla Prompt-tuning

Figure 4: Visualization of the [MASK] feature.

methods introducing extra modules for trigger extraction during inference while KEPT does not. Our KEPT also achieves a 4.95-fold improvement in inference efficiency compared to the GRASP model based on prompt-tuning.

In summary, our KEPT introduces trigger knowledge and label semantic knowledge without adding extra parameters and model structure. Compared to vanilla prompt-tuning, KEPT trades some inference efficiency but dramatically improves relation extraction performance.

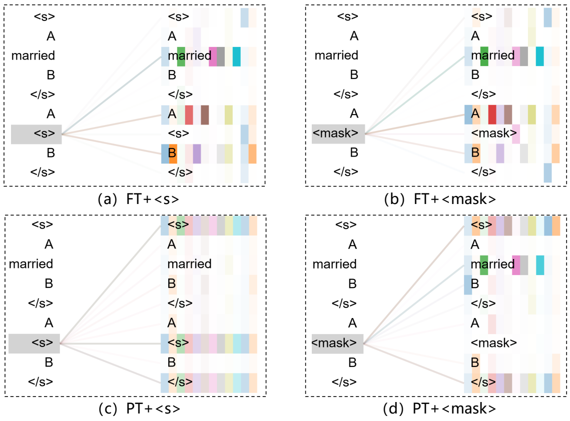
5. Analysis

5.1. Masked Feature Visualization

To verify the effectiveness of the knowledge injection tasks visually, we use t-SNE to map the embeddings of the [MASK] token to two-dimensional space. For good visualization, we choose the four classes *alternate_names*, *girlfriend_boyfriend*,*positive_impressions*, and *friends*, which have the largest number of samples in the DialogRE-EN test set, to conduct experiments. The results are shown in Figure 4.

As illustrated in Figure 4(a), four classes with a high degree of intra-class aggregation and significant inter-class by KEPT. The "alternate_names", i.e. alias, which is easily distinguishable from the other three classes, appears in blue in the figure and displays the highest level of aggregation. The "positive_ impressions" and the "girlfriend/boyfriend" share similarities in natural language semantics, represented in orange and green, respectively, with significant overlap. However, KEPT demonstrates that these two classes are notably more distinguishable in the semantic space compared to the other three cases, i.e. KEPT without one auxiliary task and vanilla prompt-tuning. These results demonstrate the proposed two tasks effectively inject knowledge into the DRE model.

5.2. Effectiveness of Prompt-tuning



(a) FT+<s> (b) FT+<mask>

(c) PT+<s> (d) PT+<mask>

Figure 5: The attention visualization of prompt-tuning and fine-tuning.

To validate the effectiveness of prompt-tuning, we conduct experiments on attention score visualization for prompt-tuning and fine-tuning. Consider the sentence "A married B": the arguments are A and B, the relation is "per:spouse", and the trigger is "married." The results are shown in Figure 5.

Comparing Figure 5(a) and Figure 5(b), in the fine-tuning method, <mask> is not involved during training, it still pays much attention to the trigger "married", since the masked language modeling uses <mask> for training in the pre-training process, hence in the fine-tuning stage <mask> can still retrieve the knowledge learned during pre-training. On the contrary, in Figure 5(c) and Figure 5(d), <s> is unable to focus on the trigger "married" in prompt-tuning.

Compared to fine-tuning, prompt-tuning focuses on more global semantic information and thus can capture relational semantic information more efficiently, while fine-tuning focuses more on local contextual information.

Furthermore, based on the analysis of Figure 5(b) and Figure 5(d), the attention scores for "married" are higher than other words, indicating that <mask> reflects the semantic information of the label. This finding proves that prompt-tuning is a suitable way to utilize trigger knowledge.

5.3. Prompt Analysis

Table 6 shows the experimental results of our KEPT utilizing different prompts. We observe that the hard prompt resulted in the lowest performance. Comparing the first and second rows in the table reveals that performance is better with added prompts than with hard prompts alone. This is because different hard prompts have varying effects. (Liu et al., 2023b) found that there is a large performance variance between different hard prompts. It is possible that the drop in experimental results could be due to insufficiently effective prompts, suggesting hard prompts rely heavily on manual design and are less practical than soft learnable prompts.

When comparing the first and third rows, it can be found that the F1 value of KEPT without adding a prompt is 0.6% higher than that of the method that only adds prompts in front of the arguments, the reason may be that the location of the prompts insertion has a greater impact on the performance, (Webson and Pavlick, 2022) argued that the performance of prompt-tuning depends heavily on the memory patterns during the pre-training.

5.4. Comprehensive Comparative analysis

Method	Answer Form	Labor	CC	EP
PTR	multi-token	high	normal	✓
KnowPrompt	single-token	normal	normal	✓
GRASP	single-token	normal	high	✓
KEPT	single-token	small	low	✗

Table 7: Comprehensive comparative statistics between KEPT and existing prompt-based methods, including 1) Answer Form: answer form of prompt 2) Labor: labor-intensive 3) CC: computational complexity 4) EP: extra parameters

To comprehensively compare the advantages of each DRE method based on prompt-tuning, we conduct qualitative comparisons in terms of answer forms, label-intensive, computational complexity, and extra parameters. Table 7 shows that the answers of the PTR method are multiple hard-tagged words, whose answer mapping process is more complex and error-prone, while the answers of the other three methods are in the form of trainable tokens, which are easy to train. KEPT has the lowest cost of labor and does not need to design complex templates compared to the other methods. PTR and KnowPrompt have moderate computational complexity, while GRASP has high complexity due to the addition of a trigger prediction process during inference. KEPT employs an auxiliary task to introduce trigger knowledge in the training phase, with no extra computational effort required during inference. The baseline approaches all introduce new parameters, except KEPT does not.

6. Conclusion

In this paper, we proposed an effective framework KEPT for dialogue-based relation extraction. Our main contribution is to inject trigger knowledge and label semantic knowledge into the PLM and improve the DRE model's knowledge utilization ability. In KEPT, we convert the DRE task as the masked language modeling task to mitigate the gap between knowledge injection and knowledge utilization and suggest two beneficial knowledge injection tasks 1) a masked trigger prediction task injecting trigger knowledge into PLM and 2) a verbalizer representation learning task injecting label semantic knowledge into PLM. Experiment results on the DialogRE dataset proved that KEPT has a fast inference speed and outperforms SoTA performance in F1 and F1$_c$ scores without any extra parameters.

7. Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments and suggestions.

8. Bibliographical References

Hao An, Dongsheng Chen, Weiyuan Xu, Zhihong Zhu, and Yuexian Zou. 2023. Tlag: An informative trigger and label-aware knowledge guided model for dialogue-based relation extraction. In *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 59–64. IEEE.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4430–4445.

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. 2020. Dialogue

Prompt	Input Example	F1
None	[CLS] Dialogue [SEP] a1 [MASK] a2 [SEP]	73.2
Hard	[CLS] Dialogue [SEP] The relation between a1 and a2 is [MASK] [SEP]	71.9
Soft	[CLS] Dialogue [SEP] [T1] a1 [MASK] [T2] a2 [SEP]	72.6
Soft	[CLS] Dialogue [SEP] [T1] a1 [T2] [MASK] [T3] a2 [T4] [SEP]	73.6

Table 6: The performance based on different prompt. [MASK] is the token used for relation classification. [T1], [T2], [T3] and [T4] are prompts. Dialogue is the input dialogue text, and a1 and a2 are argument pair.

relation extraction with document-level heterogeneous graph attention networks. *arXiv preprint arXiv:2009.05092*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022*, pages 2778–2788.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *ACL*, pages 745–758.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *TNNLS*, 33(2):494–514.

Haiyun Jiang, Qiaoben Bao, Qiao Cheng, Deqing Yang, Li Wang, and Yanghua Xiao. 2020. Complex relation extraction: Challenges and opportunities. *arXiv preprint arXiv:2012.04821*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Bongseok Lee and Yong Suk Choi. 2021. Graph based network with contextualized representations of turns in dialogue. In *EMNLP*, pages 443–455.

Po-Wei Lin, Shang-Yu Su, and Yun-Nung Chen. 2022. TREND: trigger-enhanced relation-extraction network for dialogues. In *SIGDIAL 2022*, pages 623–629.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023b. Gpt understands, too. *AI Open*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557.

Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation extraction using distant supervision: A survey. *ACM Computing Surveys (CSUR)*, 51(5):1–35.

Junyoung Son, Jinsung Kim, Jungwoo Lim, and Heui-Seok Lim. 2022. Grasp: Guiding model with relational semantics using prompt for dialogue relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 412–423.

Dong Wang and Yang Liu. 2011. A pilot study of opinion summarization in conversations. In *AACL*, pages 331–339.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14194–14202.

Fuzhao Xue, Aixin Sun, Hao Zhang, Jinjie Ni, and Eng Siong Chng. 2022. An embarrassingly simple model for dialogue relation extraction. In *ICASSP*, pages 6707–6711.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *ACL*, pages 764–777.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *ACL*, pages 4927–4940.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020a. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020b. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. Enhancing dialogue-based relation extraction by speaker and trigger words prediction. In *ACL-IJCNLP*, pages 4580–4585.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14612–14620.

9. Language Resource References

Dian Yu and Kai Sun and Claire Cardie and Dong Yu. 2020. *Dialogue-Based Relation Extraction*.