Relation Classification via Bidirectional Prompt Learning with Data Augmentation by Large Language Model

Yizhi Jiang[1] , **Jinlong Li**[1*], **Huanhuan Chen**[1]
School of Computer Science and Technology
[1]University of Science and Technology of China, Hefei, China
jyz181035@mail.ustc.edu.cn, jlli@ustc.edu.cn, hchen@ustc.edu.cn

Abstract

The Relation Extraction (RE) task aims to extract the relation between two entities in a sentence. As the performance of methods on RE task depends on datasets' quantity and quality, in this paper, we propose to use the Large Language Model (LLM) to do data augmentation. Moreover, compared to traditional fine-tuning methods, more research focuses on prompt learning. However, all of their prompt templates ignore the relative order of entities, which we believe will affect the prediction error. Due to that, we propose novel bidirectional prompt templates for prompt learning and design a training strategy for utilizing the templates. Then we try to fit the probability distributions of both prompt learning and fine-tuning methods into our model. To this end, we propose Relation Classification via Bidirectional Prompt learning with data augmentation by LLM (RCBP) and conduct experiments on four datasets: TACRED, RETACRED, TACREV and Semeval. The results show that RCBP performs well on these datasets and outperforms the state-of-the-art in the TACREV, RETACRED datasets.

Keywords: relation extraction, data augmentation, prompt learning

1. Introduction

Relation Extraction (RE) is being deeply researched in knowledge graph and other tasks, such as question answering (Xu et al., 2016; Chen et al., 2019) and text summarization (Shang et al., 2011; Lu et al., 2022). The RE task takes a sentence and two entities in the sentence as inputs and outputs a relation that is commonly recognized from a predefined relation set.

Recent studies investigate RE task from two different aspects. Firstly, Data Augmentation (DA) is widely used to improve performance, such as Back Translation (BT), which leverages several existing pre-training models to translate sentences from one language into another and then translate back. BT implies a large number of DA operators, such as synonym substitution, word deletion, word addition and so on (Dai et al., 2023). However, in back translation of RE task, there is no guarantee that two input entities will appear in the sentence after translation. For instance, given the sentence *That man has founded his own company*, words *man* and *company* are two entities, but after we translate the sentence into Chinese and then translate back into English, the sentence becomes *That person has founded his own company*, and the subject entity *man* is replaced by the entity *person*. To solve this issue, entities are recognized by comparing their semantic similarity between the original sentence and the sentence after back translation (Yu et al., 2020). However, there are still lots of errors in matching the corresponding entities. In order to further reduce such errors, we use Large Language Models (LLM) such as ChatGLM2 (Zeng et al., 2022) and Chinese-Alpaca (Cui et al., 2023) to implement BT by requiring the original entities to be kept in the answer of the queries.

Secondly, more attention has been paid to prompt learning in Natural Language Processing (NLP) task. Prompt learning opens up a new paradigm (Liu et al., 2021) of fine-tuning Pre-training Language Models (PLMs) with additional learning prompt templates. For example, in Binary Sentiment Classification (BSC), given the sentence *The movie was nice*, to present positive sentiment, we append the prompt template *It was [MASK]* to this sentence with the label of [MASK] $= great$. In this way, prompt templates designed based on a specific task can be used to fine-tune parameters of PLMs to adapt to the task. In RE task, Han et al. (2022b) proposes Prompt Tuning with Rules (PTR) for many-class text classification and applies logic rules to construct prompts with several sub-prompts. Based on it, more works about prompt learning have been proposed (Cohen et al., 2020; Yang and Song, 2022; Chen et al., 2022a,b,c; Ye et al., 2022). However, in prompt learning, there are still two issues to be addressed.

The first issue is that the order of two entities in the sentence may affect the prediction error. Recent researches (Han et al., 2022b; Yang and Song, 2022; Chen et al., 2022a,b,c; Ye et al., 2022) do not consider the order and use only one order of entities in prompt learning. But we argue that templates composed of different orders may result in different label distributions, and using a prompt tem-

* Corresponding authors

LREC-COLING 2024, pages 13885–13897

plate with only one given order may not reach the maximum probability of correct labels, which leads to the prediction error. So we propose a novel template in prompt learning for RE task to merge two orders of both entities, we name such a method bidirectional prompt learning. For instance, given sentence x and its two entities e_1 and e_2, we design forward directional template e_1 *[MASK]* e_2 and reverse directional template e_2 *[MASK]* e_1, these two templates are jointed after the sentence x. For each relation, we take the maximum probability of two templates. In this way, the issue of prediction error can be alleviated if the correct label has the highest probability in one of these two templates. Besides that, we consider that different label words for these two templates' masks have an impact on the result due to the fact that different label words may expand the scope of target semantics and further alleviate the issue. So the novel template in bidirectional prompt learning is our method to alleviate the first issue of prediction error.

The second issue is that research shows that both prompt learning and fine-tuning methods have their own advantages in RE tasks (Liu et al., 2021) and finding a method to combine both of them on RE task has been a recent research direction. In RE task, the fine-tuning method (Liu et al., 2019; Peters et al., 2019; Park and Kim, 2021; Wang et al., 2023) adds an extra structure (e.g. classification) after PLMs, both of the extra structure's and PLMs' parameters are fine-tuned to adapt to datasets. While prompt learning only utilizes pre-training task (e.g. masked language modeling) in PLM without adding additional parameters. Due to the limited number of parameters to be adjusted, researches (Han et al., 2022b; Chen et al., 2022b; Liu et al., 2019) show that prompt learning performs better than fine-tuning in few-shot datasets and is not as good as fine tuning in large datasets. On the other hand, research shows that prompt learning utilizes semantic information from labels (Yenicelik et al., 2020), while fine-tuning does not. Because prompt learning preserves the semantics of labels by converting labels into label words of [MASK]s that need to be predicted in templates, while fine-tuning regards labels as numbers and trains connections between entities' embeddings and the numbers. In order to better adapt to large datasets and make use of the semantics of labels, we add an additional MultiLayer Perceptron (MLP) to adaptively fit the final probability distribution of each label in this paper. Compared to the method proposed by Yang and Song (2022) which takes weighted sum of probability distributions of each label as the final probability distribution, our method uses more parameters to control the final probability distributions, and our parameters can be optimized adaptively. Overall, we propose our model named Relation Classification via Bidirectional Prompt learning with data augmentation by LLM (RCBP).

The main contributions of this paper are summarized as follows:

- We implement back translation by using large language models, which is able to alleviate the issue that entities can't be kept in the sentence after back translation.
- We design bidirectional prompt learning for RE task and select maximum similarity of both directions to reduce the prediction errors of a single direction.
- In order to better adapt to large datasets and make use of labels' semantics, we add a MultiLayer Perceptron to combine prompt learning with fine-tuning methods to adaptively fit the final probability distribution.

2. Preliminaries

Before introducing our method RCBP, we first formalize the problem of the RE task.

Formally, given several quintuples (x, s, o, t_s, t_o) and a relation set R, where x is a sentence, s and o are subject and object entities and $s, o \in \mathbb{E}$, where $\mathbb{E}$ is a set of all entities, t_s is type of s while t_o is type of o. The RE task is to predict relation $r \in R$ between s and o. All these quintuples make up the instance set $\mathcal{X}$.

3. Method

Our Relation Classification via Bidirectional Prompt learning (RCBP) method is illustrated in Figure 1, which includes three parts: *Back Translation*, *Bidirectional Prompt Learning* and *MLP-Driven Probability Distribution Fusion*. In *Back Translation*, we firstly translate our initial training data from English to Chinese by Large Language Model (LLM), then we translate them back to English, so we obtain a new set of data, and then we merge them with initial data into our new training data. In *Bidirectional Prompt Learning* part, firstly, we add special tokens and entities' types into sentences. For example, in the sentence shown in Figure 1, we change *Apple* to *@ * organization * Apple @*. And we also add a bidirectional prompt after the sentence. Secondly, we compute the probability distribution over each label through both Masked Language Modeling (MLM) in pre-training model and MultiLayer Perceptron (MLP). In *MLP-Driven Probability Distribution Fusion* part, we feed both probability distributions into another MLP to fit the final probability distributions.

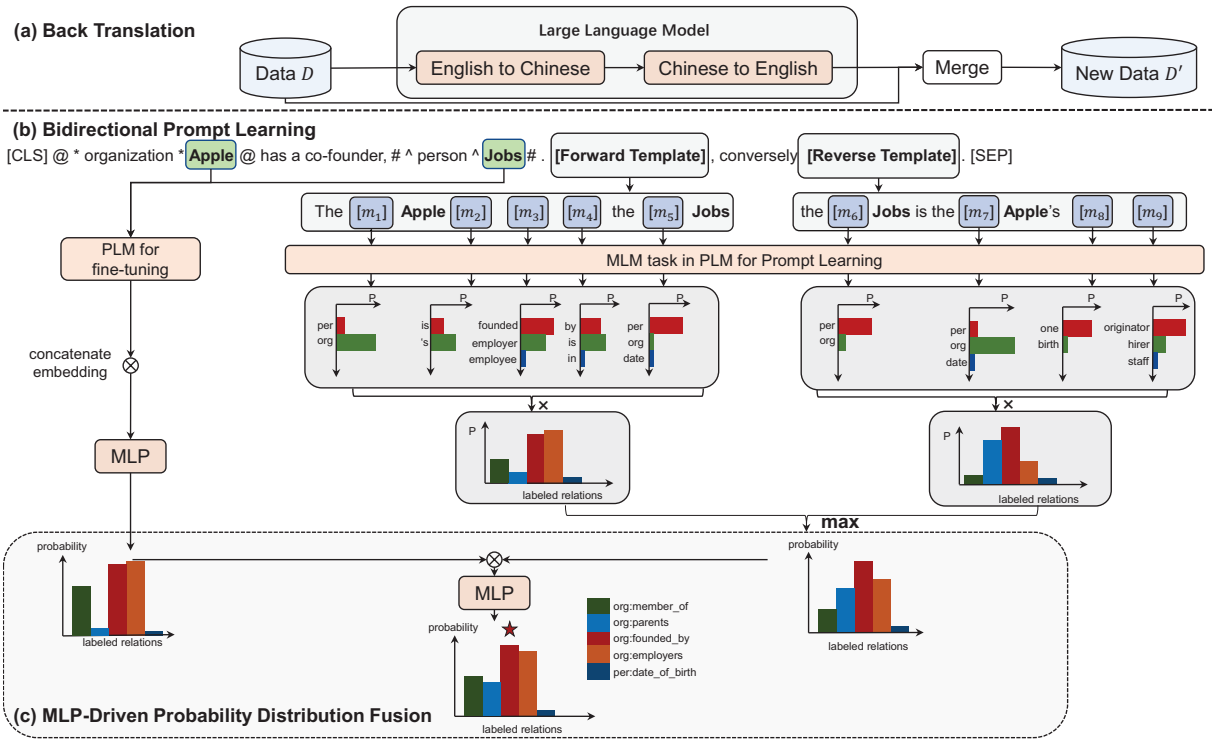

Figure 1: The overview of RCBP. The *[Forward template]* means a forward directional template while the *[Reverse template]* means a reverse directional template. *P* in the charts indicates probability, *PLM* represents a pre-training model, *MLM* represents masked language modeling, and *MLP* represents multilayer perceptron.

3.1. Back Translation

To enhance RE datasets, we use back translation to do data augmentation.

Our *Back Translation* can be concluded into 3 steps, and the input is a dataset D.

Firstly, for each instance d of dataset D, denoting its sentence by x, we construct a query $q_1(\cdot)$ as Equation (1):

$$q_1(x) = \text{“Translate into Chinese: ”} \oplus x, \quad (1)$$

where $\oplus$ is an operation of string concatenation. Then the query string $q_1(x)$ is sent to a Large Language Model (LLM) for completing the task that “Translate the sentence x into Chinese”, as indicated by Equation (2):

$$x' = f(q_1(x)), \quad (2)$$

where $f(\cdot)$ is the translation function of the LLM and x' is the translated Chinese sentence from x.

Secondly, similar to the first step, we construct a query $q_2(\cdot)$ and send it to LLM for translating Chinese sentence x' back into English sentence x'', by Equation (3) and (4):

$$q_2(x') = \text{“Translate the sentence into English, and use phrases ”} \oplus s \oplus \text{“ and ”} \oplus o \oplus \text{“: ”} \oplus x', \quad (3)$$

$$x'' = f(q_2(x')), \quad (4)$$

where s, o are two entities of instance d, x'' is the English sentence after back translation and may be different from the original English sentence x.

Finally, if the two English sentences x and x'' are not the same, we append x'' to dataset D, and update dataset D with Equation (5) :

$$D = D \cup \{x'', s, o, t_s, t_o\}, \quad (5)$$

where t_s and t_o are the types of s and o. After all instances in the dataset D have been augmented, we denote the augmented dataset by D'.

3.2. Bidirectional Prompt Learning

In this section, we introduce *Bidirectional Prompt Learning* part in two steps: building bidirectional prompts and computing bidirectional prompt learning probability.

Building Bidirectional Prompts To consider the relative order of two entities and reduce the similarity error of one single order, with the inputting instance x, we build two types of prompt templates: forward prompt template $T_1(s, o)$ and reverse prompt template $T_2(s, o)$.

Forward prompt template $T_1(s, o)$ is built according to the given relative order of two entities, which can be formalized as Equation (6):

$$T_1(s, o) = \text{The } m_1 \ s \ m_2 \ m_3 \ m_4 \text{ the } m_5 \ o, \quad (6)$$

where $m_i(1 \leq i \leq 5)$ is what we expect the pretraining model to predict. In Equation (6), mask m_1 is predicted as the type of s, mask m_5 is predicted as the type of o while masks $m_i(i = 2, 3, 4)$ are predicted as a predicate phrase that is related to relation class. For instance, in Figure 1, $T_1(\text{Apple}, \text{Jobs})$ = "The m_1 Apple m_2 m_3 m_4 the m_5 Jobs,", the label words for these masks are *organization*, *is*, *founded*, *by*, *person* separately.

Similarly to forward prompt template, reverse prompt template $T_2(s, o)$ is built based on the reversed relative order of two entities, which can be formalized as Equation (7):

$$T_2(s, o) = \text{The } m_6 \ o \text{ is the } m_7 \ s\text{'s } m_8 \ m_9, \quad (7)$$

where m_6 is predicted as the type of o, m_7 is predicted as the type of s while m_8 and m_9 are predicted as noun phrases related to relation class. For example, in Figure 1, $T_2(\text{Apple}, \text{Jobs})$="The m_6 Jobs is the m_7 Apple's m_8 m_9.", the label words for these masks are *person*, *organization*, *one*, *originator* separately.

Computing Bidirectional Prompt Learning Probability Distribution After building the bidirectional prompt templates, we can compute the probability distribution over the relation set R to predict the most possible relation class for each instance. And the way to compute probability distribution of bidirectional prompt learning can be summarized into the following three steps.

The first step is to calculate probability distribution of each mask $m_i(i = 1, 2, ..., 9)$ over its label word set V_i, which is formalized as Equation (8):

$$p(m_i = v|T_j(s, o)) = \frac{exp(h_v \cdot h_{m_i})}{\sum_{\hat{v} \in \mathcal{V}} exp(h_{\hat{v}} \cdot h_{m_i})}, \quad (8)$$

where h_{m_i} is the embedding of m_i, v is a label word in V_i and $T_j(s, o)$ denotes one directional prompt template. Equation (8) denotes calculating the similarity between each label word's embedding h_v and predicted embeddings of corresponding mask h_{m_i}.

The second step is to merge the result of several masks' probabilities in two directional prompts $T_j(s, o)$ separately and obtain a probability distribution $p_j(\cdot)$ for each prompt template, which can be formalized as Equation (9):

$$p_j(r|x) = \prod_{i, m_i \in T_j(s,o)} p(m_i = v_i|T_j(s, o)), \quad (9)$$

where r is one of the relation classes in the relation set R and its prompt template's label words are $\{v_1, v_2, ..., v_9\}$. Here in Equation (9) we use the multiplication to merge all the masks' probabilities into one single directional probability of relation class r.

The last step is to calculate the maximum probability $p_{\text{BPL}}(\cdot)$ of both directions for each relation class r, which can be calculated as Equation (10):

$$p_{\text{BPL}}(r|x) = \frac{\max\{p_1(r|x), p_2(r|x)\}}{\sum_{r' \in R} \max\{p_1(r'|x), p_2(r'|x)\}}. \quad (10)$$

An additional normalization was added to the equation (10) to guarantee that the sum of all probabilities is equal to 1.

Based on Equation (10) and Cross Entropy Loss, our learning objective for Bidirectional Prompt Learning is to minimize $Loss_{\text{BPL}}$, which is calculated as Equation (11):

$$Loss_{\text{BPL}} = -\frac{1}{D} \sum_{x \in D} \log[p_{\text{BPL}}(r_x|x)], \quad (11)$$

where r_x is the relation class of instance x.

3.3. MLP-Driven Probability Distribution Fusion

After computing the probability distribution $p_{\text{BPL}}(R|x)$ in section 3.2, we add *MLP-Driven Probability Distribution Fusion* part to fit probability distributions by combining bidirectional prompt learning with fine-tuning methods, thus our model can better adapt to large datasets and make use of relation classes' semantics. The *MLP-Driven Probability Distribution Fusion* part can be concluded in the following two steps.

Firstly, we need to compute fine-tuning's probability distribution $p_{\text{FT}}(r|x)$ over the relation set R, which can be formalized as Equation (12):

$$p_{\text{FT}}(r|x) = g(W_{\text{FT}} h_{[CLS]} + b_{\text{FT}}), r \in R, \quad (12)$$

where $g(\cdot)$ represents the softmax function, and W_{FT} and b_{FT} are parameters to be fine-tuned.

Secondly, we add an additional MultiLayer Perceptron (MLP) to adaptively fit the final probability distribution $p(r|x)$, shown by Equation (13):

$$p(r|x) = g(W[p_{\text{BPL}}(r|x), p_{\text{FT}}(r|x)] + b), r \in R, \quad (13)$$

where $p(r|x) \in \mathbb{R}^{|R| \times 1}$, $[\cdot]$ is a function to concatenate two vectors of probability distribution, and W and b are parameters that are fine-tuned to maximize the cross entropy loss function as Equation (14):

$$Loss = -\frac{1}{D} \sum_{x \in D} \log(p(r_x|x)), \quad (14)$$

where r_x is the relation class of instance x.

Statistics	Numbers in TACRED	Numbers in RETACRED	Numbers in TACREV	Numbers in Semeval
Sentences	106264	106264	91467	10717
labeled relations	42	42	40	19
entity pair types	27	27	25	-

Table 1: The Detailed Statistics in TACRED, RETACRED, TACREV and Semeval datasets. RETACRED and TACREV are datasets that correct data with annotation errors in TACRED dataset.

Model	Method	TACRED Test F_1	RETACRED Test F_1	TACREV Test F_1	Semeval Test F_1
TYP Marker‡(Liu et al., 2019)	Fine-tuning	74.6	91.1	83.2	89.9
QA†(Cohen et al., 2020)	Fine-tuning	74.8	-	-	**91.9**
LUKE†(Yamada et al., 2020)	Fine-tuning	72.7	90.3	-	90.3
RECENT‡(Lyu and Chen, 2021)	Fine-tuning	74.6	90.2	83.5	89.7
FPC†(Yang and Song, 2022)	Fine-tuning	76.2	91.6	84.9*	90.4
GenPT†(Han et al., 2022a)	Fine-tuning	75.3	91.1	84.0	-
DeepStruct†(Wang et al., 2023)	Fine-tuning	**76.8**	-	-	-
PTR†(Han et al., 2022b)	Prompt learning	72.4	90.9	83.9	89.9
KnowPrompt†(Chen et al., 2022c)	Prompt learning	72.4	91.3	82.4	90.3
RetrievalRE†(Chen et al., 2022b)	Prompt learning	72.7	91.5	82.7	90.4
OntoPrompt†(Ye et al., 2022)	Prompt learning	-	-	78.2	89.1
RCBP(ours)	both	76.41*(0.13)	**91.95**(0.22)	**85.49**(0.06)	91.43*(0.20)

Table 2: Test F1 on TACRED, RETACRED, TACREV and Semeval datasets. We report mean (and standard deviation) results of RCBP. Method represents whether the model is based on fine-tuning, prompt learning or both. The bold number is the best F1 result for each dataset and the scores with an * indicate the second best results. † indicates results collected from paper and ‡ indicates results collected from code we re-implement.

4. Experiments

To verify the performance of our Relation Classification via Bidirectional Prompt learning with data augmentation by LLM (RCBP), we compare it with the state-of-the-art on four widely used datasets: TACRED, RETACRED, TACREV and Semeval. And then, more studies are conducted to investigate the effectiveness of *Back Translation*, *Bidirectional Prompt Learning* and *MLP-Driven Probability Distribution Fusion*. Finally, we perform analysis on the remaining prediction errors to gain further insights into our RCBP.

4.1. Dataset

We evaluate our RCBP on the following four Relation Extraction tasks: TACRED (Zhang et al., 2017), RETACRED (Stoica et al., 2021), TACREV (Alt et al., 2020) and Semeval (Hendrickx et al., 2019). The detailed dataset statistics are shown in Table 1 and the evaluation index over all these datasets is the micro F_1.

4.2. Experimental Setup

Compared methods To verify the effectiveness of our model, we compare it with the following methods:

- **TYP Marker** (Zhou and Chen, 2021) It creates an entity marking method and produces a new baseline model.
- **QA** (Cohen et al., 2020) It reduces each RE sample to a series of binary spanprediction tasks.
- **LUKE** (Yamada et al., 2020) It proposes new contextualized representations based on a bidirectional transformer.
- **RECENT** (Lyu and Chen, 2021) It partitions dataset into sub-datasets by entity pair types.
- **FPC** (Yang and Song, 2022) It proposes Fine-tuning with Prompt Curriculum for RE.
- **GenPT** (Han et al., 2022a) It generates prompts to reformulate RE as an infilling problem.
- **DeepStruct** (Wang et al., 2023) It pretrains on a large task-agnostic corpora.
- **PTR** (Han et al., 2022b) It applies logic rules to construct prompts with several sub-prompts.

- **KnowPrompt** (Chen et al., 2022c) It proposes knowledge-aware prompt learning.
- **RetrievalRE** (Chen et al., 2022b) It proposes a new retrieval-enhanced prompt learning.
- **OntoPrompt** (Ye et al., 2022) It explores knowledge injection with pre training language models and proposes ontology-enhanced prompt-tuning.

Experimental Configurations Our model is implemented based on the PTR (Han et al., 2022b). The pre-training model we select is Roberta (Liu et al., 2019). Our adam (Kingma and Ba, 2014) rate is 1e-8 with a linear warmup for the first 10% steps. The weight decay is set to 1e-2. For all the datasets, we fine-tune our model for 4 epochs with a batch size of 16 and the learning rate is set to 3e-5. We take the average of F_1 based on three different random seeds as the final result.

We provide the prompt learning templates on TACRED dataset in Appendix A.

4.3. Comparison with State-Of-The-Art

The experimental results of our method with other comparison methods are represented in Table 2.

As shown in Table 2, the first column contains the names of the State-Of-The-Art models and our method RCBP, the second column represents the method used by the models, like fine-tuning, prompt learning and both. While the third to fifth columns contain the results of test F_1 on TACRED, RETACRED, TACREV and Semeval datasets, respectively. Our RCBP is tested on Roberta pre-training language models (PLMs).

From Table 2, we draw three conclusions.

Firstly, our model RCBP achieves significant improvements over all baselines of prompt learning. Compared to the best prompt learning methods, RCBP performs 3.7 higher than REtrievalRE on TACRED, 0.5 higher than REtrievalRE on RETACRED, 1.6 higher than PTR on TACREV and 1.0 higher than REtrievalRE on Semeval. Even comparing with fine-tuning models, RCBP still outperforms these models on TACREV, RETACRED. Although on TACRED, the performance is 0.4 less than the first leader DeepStruct, RCBP's result is 0.2 higher than the second model and 1.1 higher than the third model. While on Semeval, the performance is 0.5 less than the first model QA, RCBP's result is 1.0 higher than the second model.

Secondly, our model RCBP does not perform as effectively as DeepStruct model in TACRED dataset and QA model in Semeval dataset. There are two reasons in TACRED dataset. Firstly, the huge lead of DeepStruct is attributed to the fact that it further adds a large amount of text unrelated to RE task into the PLMs for fine-tuning before simply continuing with fine-tuning in downstream tasks. So another perspective towards certification is that richer knowledge of PLMs from large-scale unlabeled data will perform better in final results. Secondly, excessive noise influences our model. TACRED has lots of manual annotation errors, while RETACRED and TACREV correct some of these error data, so our model reaches high performance on these two datasets. In Semeval dataset, though we both design templates for RE task, QA model aims at answering the span of one entity under each constructed question template, while RCBP predicts the embedding of relations under the bidirectional prompts. We argue that due to the differences between datasets, predicting entities may be more efficient than predicting relations in Semeval dataset, while it's not in TACRED dataset, as RCBP performs better than QA model in TACRED.

The last is that most fine-tuning methods do better than prompt learning methods, especially in TACREV and TACRED datasets. Because recent PLMs have achieved effective results, simply fine-tuning in downstream tasks can capture the rich knowledge of PLMs from large-scale unlabeled data. Also, because fewer parameters can be fine-tuned, prompt learning doesn't have much advantage on large datasets compared to fine-tuning methods.

To understand the insides of our RCBP and find the causes of these improvements, we further investigate our contributed components in the following empirical studies.

4.4. Ablation Studies

Model	TD Test F_1	RTD Test F_1	TV Test F_1	Semeval Test F_1
RCBP	**76.41**(0.13)	**91.95**(0.22)	**85.49**(0.06)	**91.43**(0.20)
-BT	76.18(0.03)	91.82(0.20)	85.41(0.10)	90.85(0.16)
-BPL	74.89(0.43)	90.91(0.14)	83.90(0.19)	91.38(0.24)
-MPD	76.16(0.26)	91.67(0.10)	85.22(0.12)	90.42(0.38)
-all	74.17(0.42)	90.88(0.06)	82.76(0.25)	89.63(0.29)

Table 3: The mean (and standard deviation) results of our ablation experiments on RCBP. The bold number is the best F1 result. RTD denotes RETACRED, TV denotes TACREV and TD denotes TACRED dataset. BT represents *Back Translation*, BPL represents *Bidirectional Prompt Learning* and MPD represents *MLP-Driven Probability Distribution Fusion*.

Effectiveness of three Parts In order to verify the effectiveness of *Back Translation*, *Bidirectional Prompt Learning* and *MLP-Driven Probability Distribution Fusion* in RCBP, we conduct the ablation

experiments on the four datasets. The experimental results are tested after removing each one of them from our model and they are shown in Table 3. The first column represents the models, the second to fifth columns are test F_1 on each dataset.

Compared with full model RCBP, we draw the following three conclusions.

The first is that *Back Translation* and *MLP-Driven Probability Distribution Fusion* parts have not shown much improvement on TACRED and TACREV datasets. Removing *Back Translation* only decreases the result by 0.16 on TACRED and TACREV datasets on average, and removing *MLP-Driven Probability Distribution Fusion* only decreases by 0.26 on TACRED and TACREV datasets on average, which indicates that both parts have poor handling of noise as TACREV has the same training data as TACRED which has around 25% manual annotation error rate.

Secondly, *Bidirectional Prompt Learning* part almost has the highest improvement among these three parts. *Bidirectional Prompt Learning* increases F_1 result by 0.91 on average, which indicates that the sequential order of two entities in the prompts has a significant impact on the results.

Lastly, we draw the conclusion that *Back Translation*, *Bidirectional Prompt Learning* and *MLP-Driven Probability Distribution Fusion* parts all have positive effects.

Effectiveness of Back Translation by Large Language Model In order to verify the effectiveness of using Large Language Model (LLM) to do back translation, we conduct the following three experiments. We use two LLMs with queries: ChatGLM2 (Zeng et al., 2022), Chinese-Alpaca (Cui et al., 2023) to compare with the common translation method without queries: Google translation.

Dataset	CG test F_1	CA test F_1	Google test F_1	None test F_1
TD	**76.41**(0.13)	76.31(0.11)	75.53(0.07)	76.18(0.03)
RTD	**91.95**(0.22)	91.87(0.28)	91.76(0.30)	91.82(0.20)
TV	**85.49**(0.06)	85.06(0.08)	85.01(0.12)	85.41(0.10)
Semeval	**91.43**(0.20)	90.92(0.26)	90.66(0.27)	90.85(0.16)

Table 4: The mean (and standard deviation) results of comparison with LLMs and common translation. The bold number is the better F1 result. RTD denotes RETACRED, TV denotes TACREV and TD denotes TACRED dataset. CG denotes ChatGLM2, CA denotes Chinese-Alpaca and None denotes without doing back translation.

The first experiment is the RE results of two LLMs and Google translation which are shown in Table 4. From the table, we can observe that the enhanced data improvement of ChatGLM2 is the most significant which can generate higher quality corpora. And compared to the common translation, it further explains that the LLMs perform better in back translation under the guidance of our queries in RE task.

Dataset	ChatGLM2	CA	Google
TACRED	68.33%	**72.00%**	51.19%
RETACRED	**69.29%**	66.50%	51.63%
TACREV	68.33%	**72.00%**	51.19%
Semeval	57.83%	**62.30%**	37.09%

Table 5: The table shows the rates of including both entities after back translation. The bold number is the highest inclusion rate for each dataset. CA denotes Chinese-Alpaca.

The second experiment is about the rates of including both entities after back translation, the results are shown in Table 5. Because TACREV and TACRED have the same training data, they have the same rate of including both entities. From Table 5, we can observe that Chinese-Alpaca has the highest rate of including both entities after back translation. Also, we can observe that the common translation's rates are around 20% less than LLMs' rates which indicates that our queries help guide LLMs to preserve two entities. Because of less data enhanced, the results of common translation are not higher than those of LLMs.

Dataset	ChatGLM2	CA	Google
TACRED	**76.09**	71.64	74.83
RETACRED	75.73	72.05	**77.96**
TACREV	**76.09**	71.64	74.83
Semeval	76.11	73.26	**78.81**

Table 6: The table shows the back translation quality of each model and the evaluation metric is BLEU. The bold number is the highest inclusion rate for each dataset. CA denotes Chinese-Alpaca.

The third experiment shows the translation quality of each model, the results are shown in Table 6. Here we use the BLEU (Papineni et al., 2002) metric to evaluate the quality of back translation, and the n-gram parameter is set to 3. From Table 6, we can observe that ChatGLM2 has higher translation quality than Chinese-Alpaca, even though the latter have slightly higher inclusion rate of entities, ChatGLM2 still performs better in results. Though the common translation has the highest translation quality in RETACRED and Semeval datasets, its inclusion rate of entities is too low, which influences its results in RE task. Overall, we conclude that queries in LLMs can alleviate the issue of keeping entities after back translation.

Effectiveness of Bidirectional Prompt Learning In order to verify that Bidirectional Prompt Learning (BPL) reduces the prediction errors of one single direction and the design of bidirectional prompts has an impact on the result, we conduct the following two experiments. The *Back Translation* and *MLP-Driven Probability Distribution Fusion* parts are not applied to these experiments.

Dataset	Bidirectional	F	R
TACRED	**76.00**(0.26)	74.17(0.42)	74.05(0.73)
RETACRED	91.06(0.48)	90.88(0.06)	**91.56**(0.21)
TACREV	**84.82**(0.21)	82.76(0.25)	84.09(0.18)
Semeval	**90.28**(0.24)	89.63(0.29)	89.97(0.43)

Table 7: The table shows the mean (and standard deviation) results of bidirectional prompt learning and each single direction. The bold number is the highest result for each dataset. F denotes forward direction while R denotes reverse direction.

The first experiment is to verify that BPL can reduce the prediction errors of one single direction, which are shown in Table 7. The second column indicates using bidirectional prompt templates, the third and fourth columns indicate only using the forward templates and reverse templates respectively.

We draw the following two conclusions from the table.

Firstly, bidirectional prompts perform better than single directional prompts in three datasets except RETACRED. But in RETACRED, the effect of reverse is better than bidirectional , with 0.5 point difference, and both of them do better than the forward templates. So overall, BPL can really reduce the prediction errors in both directions.

Secondly, it's not possible to just choose one effective single direction simply, because forward templates do better than reverse templates on TACRED datasets, but it's opposite on RETACRED, TACREV and Semeval datasets. So both of them have varying degrees of prediction errors on different datasets.

The second experiment is to verify the effectiveness of designing the bidirectional prompts. We test it from the following three points:

- Setting of label words for masks in two directional prompts, like different label words or same label words.
- Different ways of calculating probability distribution of two directions, like maximum or multiplication.
- Different conjunctions of two directional prompts, like *conversely*, *and* or nothing.

We verify the above three points on TACRED and the experimental results are shown in Table 8. From the table, we draw the following three conclusions.

Points	Model	TACRED test F_1
Label words	*Different*	76.00(0.26)
	Same	75.05(0.28)
Calculating ways	*Maximum*	76.00(0.26)
	Multiplication	75.38(0.33)
Conjunction	*conversely*	76.00(0.26)
	and	75.56(0.41)
	,	75.73(0.54)
	(nothing)	75.25(0.61)

Table 8: The mean (and standard deviation) results of three points in designing bidirectional prompts.

Firstly, different label words for two prompts help improve result by 0.95. We consider different label words for MASKs can expand the semantic range which will make it easier for PLMs to predict label words.

Secondly, the maximum calculation is better than the multiplication. This is because the maximum probability distribution chooses the most possible direction which ignore another direction's error, while multiplication will take that part of error into account.

Last is that conjunctions expressing turning point is needed to connect two directional prompts, such as *conversely*, which is straightforward for PLMs to learn that two directional prompts do not represent exactly the same semantics, others like *and* or else do not display the differences between two directional prompts.

Comparison of Probability Distribution Fusion In order to verify the effectiveness of our probability distribution fusion method, we conduct an experiment to compare our MLP-Driven method with the weighted sum method (Yang and Song, 2022). Here, we remove the *Back Translation* part and the experimental results are shown in Table 9. The weighted sum method is formalized as Equation (15):

$$p(r|x) = p_{\text{BPL}}(r|x) + \alpha p_{\text{FT}}(r|x), r \in R, \quad (15)$$

where $p_{\text{BPL}}(r|x)$ and $p_{\text{FT}}(r|x)$ are probability distributions output by bidirectional prompt learning, $p(r|x)$ is the final probability distribution and α is the only hyper-parameter that needs to be set manually.

We can summarize the following two points.

The first is that on RETACRED, TACREV and Semeval datasets, MLP-Driven method does better than weighted sum method. This is due to the fact that compared to the weighted sum, MLP-Driven has more parameters that can be fine-tuned while the weighted sum has only one parameter. MLP-

Dataset	MLP-Driven Test F_1	weighted sum Test F_1
TACRED	76.18(0.03)	**76.45**(0.11)
RETACRED	**91.82**(0.20)	91.53(0.13)
TACREV	**85.41**(0.10)	84.83(0.29)
Semeval	**90.85**(0.16)	90.25(0.44)

Table 9: The mean (and standard deviation) results of a comparison between MLP-Driven and weighted sum methods. The bold number is the better F1 result.

Driven learns to fit the final probability distribution more deeply than the weighted sum.

Secondly, however, in TACRED dataset, the weighted sum does better than MLP-Driven. Because there are too many manual annotation errors in TACRED, MLP-Driven learns too much noise during training, so it performs worse in predicting. This also indicates that our MLP-Driven method will be affected by datasets that have lots of noise.

4.5. Analysis on the error

We argue that the errors can be classified into the annotation errors in TACRED and the multi-relation errors in all datasets.

Because of the annotation errors in TACRED, the final F_1 is limited to around 76.4. Although RETACRED modifies 23.9% of TACRED data, we still find some errors and deficiencies in RETACRED. For example, to those data with two entities' type *(person, country)*, some relations are labeled as *per:city_of_death*, the relation and entity pair type are not relevant, which is one of the reasons leading to the error. The deficiency is that for those data with two entities' type *(person, location)*, *location* is too generalized which results in that the labeled relations may be related to *city*, *country* or *stateorprovince*. We consider if *location* is specifically refined into *city*, *country* or *stateorprovince*, the result of such data will be improved.

We observe that there may exist multiple labeled relations between the given two entities. For example, in sentence *John was born and passed away in New York*, there are relations *per:city_of_birth* and *per:city_of_death* between two entities *John* and *New York*. We call such errors as multi-relation errors. In our model, we deal with this error by choosing one relation with the maximum probability computed through our model, as the RE task requires model to output only one single relation. But when encountering multiple relations with similar high probabilities, our model is difficult to predict the same relation as the labeled relation.

5. Conclusion and Future Work

In this paper, we propose a new method for RE tasks called Relation Classification via Bidirectional Prompt learning with data augmentation by large language models (RCBP). We use large language models to do back translation to enhance data, which alleviate issue of keeping two entities in answers to queries. Based on this, we propose bidirectional prompt learning and design a strategy for utilizing its templates. Otherwise, we add an additional MultiLayer Perceptron to adaptively fit final probability distribution to combine prompt learning with fine-tuning. The experimental results on RE datasets demonstrate that RCBP outperforms existing prompt learning methods and reaches high performance.

In the future, we'll strengthen our model's robustness to reduce the impact of noise and apply more large language models (such as *ChatGPT*, *Xinghuo*, etc.) to more languages to do back translation. Moreover, in order to better solve multi-relation errors, we define another different field in RE task, which is called multi-relation RE task. In this task, we need to output a subset of the given relation set rather than just outputting one relation, and the completeness and correctness of the output set become the evaluation metrics. We will also consider this task in our future work.

6. Acknowledgements

We would like to thank Zhi Cheng, Bin Yang and Zhiyuan Yu for helpful discussions, support, and feedback on earlier versions of this work. We would also like to thank the anonymous reviewers for their insightful comments and suggestions.

7. Bibliographical References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Decoupling knowledge from memorization: Retrieval-augmented prompt learning. *arXiv preprint arXiv:2205.14704*.

Xiang Chen, Lei Li, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Relation extraction as open-book examination: Retrieval-enhanced prompt tuning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2443–2448.

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

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering. *arXiv preprint arXiv:1904.01246*.

Amir D. N. Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. Relation extraction as two-way span-prediction. *CoRR*, abs/2010.04829.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022a. Generative prompt tuning for relation classification.

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *CoRR*, abs/1911.10422.

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

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

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

Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022. Summarization as indirect supervision for relation extraction. *arXiv preprint arXiv:2205.09837*.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. *CoRR*, abs/2105.08393.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Seongsik Park and Harksoo Kim. 2021. Improving sentence-level relation extraction through curriculum learning. *CoRR*, abs/2107.09332.

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 the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Yue Shang, Yanpeng Li, Hongfei Lin, and Zhihao Yang. 2011. Enhancing biomedical text summarization using semantic relation extraction. *PLoS one*, 6(8):e23862.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the TACRED dataset. *CoRR*, abs/2104.08398.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2023. Deepstruct: Pretraining of language models for structure prediction.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. *arXiv preprint arXiv:1603.00957*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. *CoRR*, abs/2010.01057.

Sicheng Yang and Dandan Song. 2022. Fpc: Fine-tuning with prompt curriculum for relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 1065–1077.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced prompt-tuning for few-shot learning. In *Proceedings of the ACM Web Conference 2022*, pages 778–787.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does bert capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162.

Junjie Yu, Tong Zhu, Wenliang Chen, Wei Zhang, and Min Zhang. 2020. Improving relation extraction with relational paraphrase sentences. In *Proceedings of the 28th international conference on computational linguistics*, pages 1687–1698.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

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

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

A. Prompt learning templates

In this section, we list all the detailed prompt learning templates on TACRED dataset in Table 10, the first column contains all the relation labels, and the following columns are label words of masks.

Relation Label	m_1	$m_2 m_3 m_4$	m_5	m_6	m_7	$m_8 m_9$
per:date_of_birth	person	was born in	date	date	person	birth time
per:city_of_birth	person	was born in	city	city	person	birth city
per:cause_of_death	person	was died of	event	event	person	dead causation
per:city_of_death	person	was died in	city	city	person	dead city
per:spouse	person	's spouse was	person	person	person	own partner
per:charges	person	was charged with	event	event	person	accusatory crime
per:date_of_death	person	was died on	date	date	person	dead time
per:country_of_death	person	was died in	country	country	person	dead country
per:state_of_death	person	was died in	state	state	person	dead state
per:state_of_birth	person	was born in	state	state	person	birth state
per:other_family	person	's relative is	person	person	person	one relative
per:country_of_birth	person	was born in	country	country	person	birth country
per:title	person	's title is	title	title	person	one title
per:countries_of_residence	person	was living in	country	country	person	residence country
per:state_of_residence	person	was living in	state	state	person	residence state
per:cities_of_residence	person	was living in	city	city	person	residence city
per:religion	person	was member of	religion	religion	person	belief religion
per:schools_attended	person	's school was	organization	organization	person	graduated university
per:employee_of	person	's employee was	organization	organization	person	work place
per:age	person	's age was	number	number	person	year old
per:siblings	person	's sibling was	person	person	person	one sibling
per:parents	person	's parent was	person	person	person	own parent
per:children	person	's child was	person	person	person	one child
per:alternate_names	person	's alias was	person	person	person	alternate name
per:origin	person	's nationality was	country	country	person	nationality country
org:website	organization	's website was	url	url	organization	website url
org:founded_by	organization	was founded by	person	person	organization	one founder
org:founded	organization	was founded in	date	date	organization	establish time
org:shareholders	organization	was invested by	person	person	organization	one shareholder
org:dissolved	organization	was dissolved in	date	date	organization	dissolution time
org:state_of_headquarters	organization	was located in	state	state	organization	headquarter state
org:country_of_headquarters	organization	was located in	country	country	organization	headquarter country
org:city_of_headquarters	organization	was located in	city	city	organization	headquarter city
org:member_of	organization	was member of	organization	organization	organization	one number
org:political	organization	was member of	religion	religion	organization	belief religion
org:top_members	organization	's employer was	person	person	organization	one boss
org:number_of_employees	organization	's employer has	number	number	organization	boss number
org:alternate_names	organization	's alias was	organization	organization	organization	alternate name
org:members	organization	's member was	organization	organization	organization	one member
org:parents	organization	's parent was	organization	organization	organization	higher organization
org:subsidiaries	organization	's subsidiary was	organization	organization	organization	subordinate organization
org:website	organization	's website was	url	url	organization	website url
NA	entity	is irrelevant to	entity	entity	entity	unrelated entity

Table 10: The prompt learning templates on TACRED dataset. m_i indicates the i-th mask in templates and NA represents no relation.