CLoCE:Contrastive Learning Optimize Continous Prompt Embedding Space in Relation Extraction

Anonymous NAACL-HLT 2021 submission

Abstract

Recent studies have proved that prompt tuning can improve the performance of pre-trained language models on downstream tasks. However, in the task of relation extraction (RE), there are still a large number of confusing samples that hinder prompt-tuning method from achieving higher accuracy. Inspired by previous works, we innovatively utilize contrastive learning to solve this problem. We propose a prompt-tuning-based framework and apply contrastive learning to optimize the representation of input sentences in embedding space. At the same time, we design a more general template for RE task, and further use knowledge injection to improve performance of the model. Through extensive experiments on public datasets, the micro F_1-score(%) of our model exceeds the existing SOTA on the Re-TACRED and TACREV datasets by 0.5 and 1.0, respectively. Meanwhile, in the few-shot scenario, our model also has a more robust performance than fine-tune methods.

1 Introduction

The Relation Extraction (RE) task is a fundamental problem in Natural Language Processing (NLP). As the core task of information extraction (IE), RE extracts effective structured semantic information from unstructured text, which has a crucial impact on many downstream tasks.

Recently, the self-supervised PLMs, such as ELMo (Peters et al., 2018), GPT (Radford et al., 2019) and BERT (Devlin et al., 2018), have been widely used in NLP. One of the general paradigms of pre-trained language models (PLMs) is to transfer rich contextual knowledge to specific downstream tasks by fine-tuning model parameters. Although the PLMs obtain a wealth of semantic knowledge, it remains a challenge for the paradigm to extract the specific knowledge required and to improve the utilization of knowledge.

With the release of GPT-3 (Brown et al., 2020), the application of prompt-tuning PLMs has been widespread studied. Prompt-tuning bridges the gap between pre-training and fine-tuning as a new fine-tuning method, and makes task-specific pre-train models more concise under multiple scenarios. As shown in Figure 1, the RE task based on prompt-tuning is transformed into a cloze task, which is to predict the `[mask]` in the prompt, thereby inferring the implied relationships between entities. Each `[mask]` has a set of candidate words that make up the answer space. Some recent studies have shown that the prompt-tuning method can also achieve excellent performance in the few-shot setting (Gao et al., 2020; Schick and Schütze, 2021; Liu et al., 2021b).

For the RE task with prompt-tuning, a series of researchs have focused on automatic prompts generation (Schick et al., 2020; Schick and Schütze, 2021; Shin et al., 2020; Gao et al., 2021a) for handling labor-intensive human-picked constructs of prompts. However, automatically generated prompts do not have satisfactory performance compared with manually designed prompts and require additional computation cost for generation and verification. For manually designed prompts, a major challenge is how to construct appropriate templates with rich knowledge. By injecting additional information into the prompt template design and the answer construction (Han et al., 2021; Chen et al., 2021; Zhou and Chen, 2021), the templates will have semantic level knowledge of relation and entity types in the relevant domain to implement more precise RE. Further complicating the issue, for these multi-class classification tasks, the above methods are unable to distinguish between a pair of confusing relations.

To solve the above problems, we propose a novel model for RE that incorporates contrastive learning into the prompt-tuning paradigm. To make the model better understand the semantic information of the input samples, we take full advantage of the bias samples by constructing positive and negative samples, and use contrastive learning to optimize the semantic representation of inputs in embedding space. For the better effect of prompt, we inject entity types into prompt as additional knowledge. Through a great deal of experiments on public datasets, we observe that knowledge injection can further improve the performance of the prompt-tuning paradigm. We conduct extensive experiments on three popular sentence-level RE datasets. The results show that CLoCE can significantly outperform existing state-of-the-art baselines. Specifically, our model advances the previous SOTA on Re-TACRED and TACREV. Meanwhile, our model achieves better results with few training epochs. In addition, experiments under low-resource scenario show that the model still has robust effect in few-shot setting. Our contributions can be summarized as follows:

- We propose a prompt-based framework **C**ontrastive **L**earning **O**ptimize **C**ontinous prompt **E**mbedding(**CLoCE**): enlighteningly processing biased samples during training process to construct positive and negative samples, and applying contrastive learning to optimize the semantic representations in embedding space so as to distinguish confusing relation more accurately. To the best of our knowledge, it is the first work to introduce contrastive learning for the prompt-tuning method in RE tasks.

- We design a general template for the sentence-level RE to improve the generalization of the framework. Knowledge is injected into the prompt template to improve the performance of the model.

- To verify the effectiveness, we conduct experiments on the three most frequently used sentence-level relation extraction datasets and our model outperforms existing SOTA.

2 Related Work

Relation Extraction RE is a sub-task of IE which pays more attention to the relationship between specific entities. The pioneering approaches are pattern-based methods (Soderland et al., 1995; Califf and Mooney, 1997), which automatically construct pattern rules from grammatical elements. Feature-based methods (Zhou et al., 2005; Jiang and Zhai, 2007; Nguyen et al., 2007) use feature engineering on entities and contexts before classification tasks. Methods based on Convolutional Neural Networks (Zeng et al., 2014), Recurrent Neural Networks (Vu et al., 2016) and Long Short-Term Memory Networks (Zhou et al., 2016) introduce neural networks to relation extraction. Graph-based methods (Zhang et al., 2018; Guo et al., 2019, 2020) construct entities graph for inferencing. Recently, PLMs have achieved excellent success by labeling amounts of data. Impressive results are achieved by using limited annotated sentences to fine-tune PLMs (Han et al., 2018a; Gao et al., 2019). In contrast to the traditional methods, BERT-based models (Wu and He, 2019; Joshi et al., 2020; Yu et al., 2020) have become mainstreamed trend. On this basis, Lyu and Chen (2021) use entity type to constrain relation classification and achieve SOTA performance on TACRED(Zhang et al., 2017).

To avoid using a multitude of labor-intensive annotated instances, recent trend is few-shot settings. Han et al. (2018b) construct FewRel which is a few-shot relation extraction dataset based on the N-way K-shot method. Gao et al. (2020) focus on the application of snowball in Few-Shot Relation Learning. Han et al. (2021) achieve a balance between performance and cost based on manually selecting sub-prompts. Chen et al. (2021) propose a method to jointly optimize prompt templates and answer words in continuous space.

Contrastive Learning Contrastive learning concentrates on learning the common features between instances of the same class(Positive sample) and distinguishing the differences between instances of the different classes(Negative sample). For different tasks' loss functions, the contrastive learning can be effective, since some methods optimizing these loss functions that are combined with the contrastive learning loss function. Wu et al. (2020) combine word-level and sentence-level losses based on contrastive learning to optimize sentence-level PLMs. Giorgi et al. (2021) design a self-supervised objective for learning universal unlabeled sentence embeddings. Zhang et al. (2021) propose a contrastive learning framework to separate different categories that overlap with each other in the representation space better at the beginning of the learning process.

Other methods use contrastive learning to con-

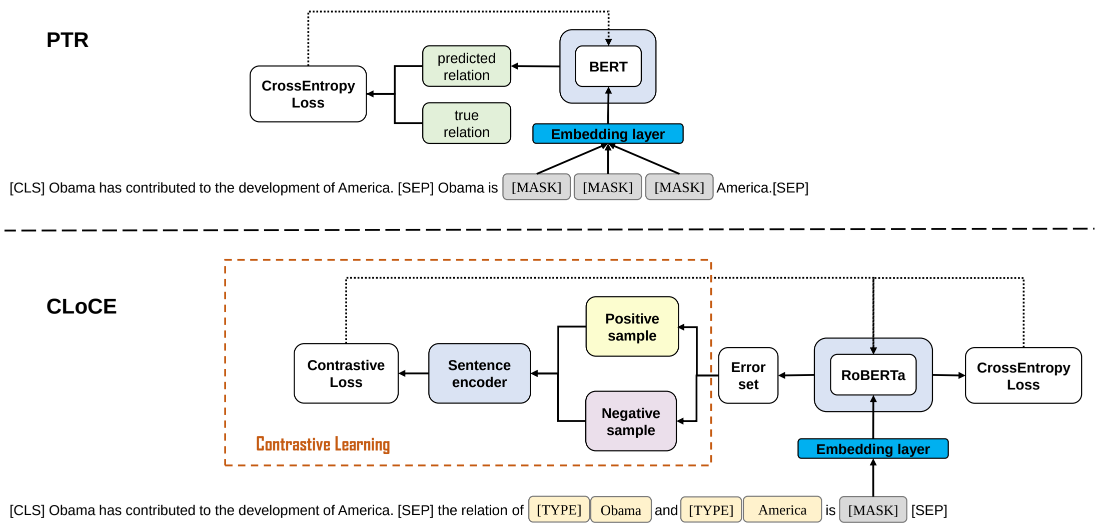
PTR
CrossEntropy Loss
predicted relation
true relation
BERT
Embedding layer
[CLS] Obama has contributed to the development of America. [SEP] Obama is [MASK] [MASK] [MASK] America.[SEP]
CLoCE
Contrastive Loss
Sentence encoder
Positive sample
Negative sample
Contrastive Learning
Error set
RoBERTa
CrossEntropy Loss
Embedding layer
[CLS] Obama has contributed to the development of America. [SEP] the relation of [TYPE] Obama and [TYPE] America is [MASK] [SEP]

Figure 1: Model architecture of prompt-tuning and CLoCE. In CLoCE-D we use the Roberta-encoder and embedding layer as the sentence encoder in this figure with which parameters are shared during the classification process. In comparison, the sentence encoder of CLoCE-S is a single embedding layer sharing parameters.

struct augmentation samples and fine-tune the models. Yan et al. (2021) propose ConSERT, a contrastive framework for self-supervised sentence representation transfer. Gao et al. (2021b) propose another contrastive learning framework SimCSE by exploiting random-sampled dropout as minimal data augmentation.

Prompt-tuning With the emerging of GPT-3 (Brown et al., 2020), manually creating prompts to handle NLP tasks has become a novel paradigm for few-shot learning. Schick and Schütze (2020, 2021) use pre-defined manually crafted templates in a few-shot learning setting. Although manually crafted templates can be fairly accurate, it is still possible that the best performance prompt cannot be designed (Jiang et al., 2020). Many methods have been proposed (Wallace et al., 2019; Haviv et al., 2021) to automate discrete prompts search so as not to leverage labor-intensive prompt engineering. Shin et al. (2020) propose gradient-based search to automatically generate templates and label words. Gao et al. (2021a) introduce a pre-trained model T5 to generate template tokens. In relation to discrete prompts, several studies on continuous prompts (Qin and Eisner, 2021; Hambardzumyan et al., 2021) relax the pattern restrictions of the embeddings of template words. Li and Liang (2021) propose Prefix-Tuning to optimize a continuous task-specific vector with few parameters. Furthermore, Liu et al. (2021b) propose P-tuning to insert trainable variables into the embedding input. Liu et al. (2021a) ameliorate the original prefix-tuning through deep prompt tuning and introduce deeper representations for pseudo tokens.

For relation extraction, Han et al. (2021) devise prompt tuning with rules. This approach designs several simple sub-prompts and combines these sub-prompts according to logical rules to form task-specific prompts for multiple-class classification task. Chen et al. (2021) propose a novel knowledge-aware prompt-tuning to encode semantic knowledge among entity types and relations by prompt template design and answer construction with injected knowledge.

3 Method

In this section, we give the definition of sentence-level RE task and the general paradigm of prompt-tuning in this task in Section 3.1. In Section 3.2, template design and knowledge injection are introduced first. The second part of Section 3.2 introduces the complete process of relation classification and focus on how contrastive learning optimizes the semantic representation space. Finally, we introduce our training process "an alternate way of training" from a holistic perspective.

3.1 Relation Extraction

Definition of Relation Extraction Relation extraction is a critical task in NLP. Let $\mathcal{D} = \{\mathcal{X}, \mathcal{R}\}$ denote a RE dataset, where $\mathcal{X}$ is the set of sentences and $\mathcal{R}$ is defined as the set of relation labels. For each instance, the input of RE task is a token sequence $x = \{t_1, t_2, \cdots, t_n\} \in \mathcal{X}$, where $e_s = \{t_i, \cdots, t_j\}$ and $e_o = \{t_m, \cdots, t_n\}$ represent subject entity and object entity. The output of RE task is the prediction of relation $r \in \mathcal{R}$ between e_s and e_o, which is usually presented in the form of probability vector.

Abstractly, RE task is to infer the relation $r \in \mathcal{R}$ between e_s and e_o from given sentence $x^i \in \mathcal{X}$, the location of the entities, the type of the entities and other annotation information of the entities. All we need to do is find a function f from the function space to fit the mapping $f(x^i) \rightarrow r_j(e_s^i, e_o^i) \in \mathcal{R}$ more accurately.

Fine-tuning of PLMs Given a PLM $\mathcal{F}_{PLM}$ for RE, general fine-tuning methods first convert the instance $x = \{t_1, t_2, \cdots, t_n\}$ into $\{\texttt{[CLS]}, t_1, t_2, \cdots, t_n, \texttt{[SEP]}\}$ as an input sequence of PLM. The PLM $\mathcal{F}_{PLM}$ encodes all the tokens of input sequence into the corresponding hidden vectors such as $h = \{h_{\texttt{[CLS]}}, h_1, h_2, \cdots, h_n, h_{\texttt{[SEP]}}\}$.

Normally, a [CLS] head is utilized to compute the probability distribution over the class set Y with the softmax function $p(\cdot \mid x) = Softmax(\mathbf{W}\mathbf{h}\texttt{[CLS]} + \mathbf{b})$, where $\mathbf{h}\texttt{[CLS]}$ is the embedding vector of [CLS], $\mathbf{W}$ is an original matrix that needs to be randomly initialized before fine-tuning, and $\mathbf{b}$ is a bias vector. The parameters of $\mathcal{F}_{PLM}$, $\mathbf{b}$ and $\mathbf{W}$ are tuned to the cross-entropy loss over $p(y \mid x)$ on the $\mathcal{X}$.

Prompt-tuning Prompt-tuning transforms RE task into the prediction task of mask with PLMs. Specifically, for each input sentence, the template function $\mathcal{T}$ maps the input $x \in \mathcal{X}$ to a sequence that fuses the original input and the template as $x_{prompt} = \mathcal{T}(x)$. This process adds additional information into the original template such as the entity name. $\mathcal{V}$ is a set of label words in the vocabulary of language model $\mathcal{F}_{\mathcal{PLM}}$, and $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$ is an injective mapping that connects task labels and label words $\mathcal{V}$.

In addition to keeping the original words in x, one or more [MASK] is placed into x_{prompt} to fill in the label words. Since $\mathcal{F}_{\mathcal{PLM}}$ can predict the correct token at the masked position, we can formalize $p(y \mid x)$ with the probability distribution over $\mathcal{V}$ at the masked position, namely $p(y \mid x) = p(\texttt{[MASK]} = \mathcal{M}(y) \mid x_{prompt})$. Taking the description in Figure 1 as an example, we map x to x_{prompt} =“[CLS] x [SEP] *Obama is* [MASK] [MASK] [MASK] *American* [SEP]".

We can use $\mathcal{F}_{\mathcal{PLM}}$ to encode x_{prompt} to gain the hidden vector of [MASK] and generate a probability distribution $p(\texttt{[MASK]} \mid x_{prompt})$ to describe which words in $\mathcal{V}$ are suitable to replace [MASK] words. Ultimately, we set $\mathcal{M}(y = \textit{“the residence of”}) \rightarrow$ “*per : countries_of_residence*” as one label, and $\mathcal{M}(y = \textit{“countries of birth”}) \rightarrow$ “*per : countries_of_birth*” as another label, respectively.

Depending on whether $\mathcal{F}_{\mathcal{PLM}}$ predicts “*was birth in*” or “*birth*”, we can determine whether the relation label of input x is either “*per : countries_of_birth*” or “*per : countries_of_residence*”.

3.2 Our Approach

Template Design and Knowledge Injection To make templates more generic on RE tasks, as shown in Figure 1, we design the template as: " *the relation of* [MASK]$_1$ [MASK]$_2$ *and* [MASK]$_3$ [MASK]$_4$ *is* [MASK]$_5$ ", where [MASK]$_2$ and [MASK]$_4$ are the head entity and tail entity. In contrast to PTR (Han et al., 2021), our template is more generic on different datasets. For more precise prediction of the relation [MASK]$_5$, we inject the knowledge of entity type into the template as [MASK]$_1$ and [MASK]$_3$. In contrast to KnowPrompt (Chen et al., 2021), we inject knowledge directly into templates instead of noisy input. Intuitively, this is more helpful for inference process.

Relation Classification Different from using MLP for classification after obtaining feature via an encoder such as BERT, we take full advantage of the characteristics of the masked language model and fuse classification information into the soft labels for prediction. For each input sample, CLoCE will fill the entity name [MASK]$_2$ and [MASK]$_4$, entity type [MASK]$_1$ and [MASK]$_3$ into the designed template. After splicing the input sentence and the prompt, the number of input tokens keeps uniform by truncating or filling with hyperparameter as len_{max} which is defined as the max length of sequence. The token sequence is defined as S. Processing by embedding layer, the output tensor obtains a primary semantic representation, in which the embedding vector of each token can be trained.

Afterwards, we input the primary representation of token sequence $S^{'}$ into the RoBERTa-encoder for deeper semantic representations. The output tensor Y has the same shape as $S^{'}$. We define the vector Q as one-hot encode to locate the masked position. We use the vector Q to obtain the predicted representation of the `[MASK]` from Y and map it to a c-dimensional tensor by computing the similarity between soft labels and prediction to infer the probability of each class. Both the parameters of embedding layer and the RoBERTa-encoder are optimized by backpropagation.

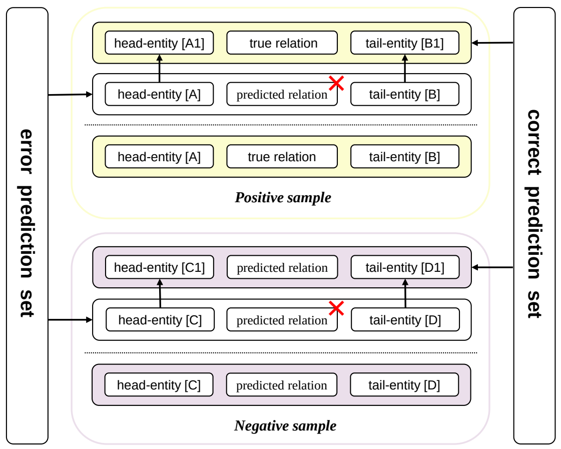

Figure 2: The process of constructing positive and negative sample pairs.

Contrastive Learning As shown in Figure 2, each annotated instance in training set has the true relation r_{true}. In each epoch of training, we obtain the predicted result. Compared with the true label,we construct two sets to collect samples with correct or incorrect predictions , which called correct prediction set $\mathcal{N}_{correct}$ and error prediction set $\mathcal{N}_{error}$.

For each instance x^{i}_{error} of $\mathcal{N}_{error}$ that has the predicted relation $r^{i}_{error} \in \mathcal{R}$, we select an annotated instance $x^{j}_{correct}$ from $\mathcal{N}_{correct}$ that has the same relation with x^{i}_{true}, e.g. $r^{j}_{correct} = r^{i}_{true}$. Further, we replace `[A1]` and `[B1]` that are the head and tail entity in $x^{j}_{correct}$ with `[A]` and `[B]` that are the head and tail entity in x^{i}_{error}. Therefore $x^{j}_{correct}$ with `[A]` and `[B]` becomes a positive sample.

Likewise, for each instance x^{i}_{error} of $\mathcal{N}_{error}$, unlike before, we select an annotated instance $x^{j}_{correct}$ from $\mathcal{N}_{correct}$ that has the same relation with x^{i}_{error}, i.e. $r^{j}_{correct} = r^{i}_{error}$. Then we substitute `[C1]` and `[D1]` that are the head and tail entity in $x^{j}_{correct}$ with `[C]` and `[D]` that are the head and tail entity in x^{i}_{error}. A positive sample is generated by $x^{j}_{correct}$ with `[C]` and `[D]`.

In contrastive learning module, CLoCE input the positive example pairs and the negative example pairs, and then minimize the contrastive learning loss ℓ_{CL} for backpropagation after obtaining the sentence-level feature expression. The contrastive loss function is shown below, where m is a trainable parameter. The purpose of this module is to close the semantic feature representation of the same relationship and push away the semantic feature representation of different relationships.

$$\begin{aligned}\ell_{CL} &= L(\theta, Y, S_i, S_j)\\ &= \frac{1}{M} \sum_{(S_i,S_j,y_i)\in N_{total}} [(1-y_i)D^2_\theta(S_i,S_j)\\ &\quad + y_i \cdot \frac{1}{2}(\max\{0, m - D_\theta(S_i,S_j)\})^2]\end{aligned}$$

The contrastive learning loss function is shown above where m is a trainable parameter. If the distance between negative examples is greater than m, the loss is not calculated. M is the number of samples in $\mathcal{N}_{correct}$ and $\mathcal{N}_{error}$. CLoCE uses cosine distance to represent distance between sentences in the sample as following. From the input sequence to the embedding space, embedding layer or RoBERTa-encoder can be used.

$$\vec{S_i} = \mathcal{F}^{\theta}_{PLM}(S_i), \vec{S_j} = \mathcal{F}^{\theta}_{PLM}(S_j)$$
$$D_\theta(\vec{S_i}, \vec{S_j}) = 1 - \frac{\vec{S_i}\cdot\vec{S_j}}{\|S_i\|_2\|S_j\|_2}$$

Alternate Training Since our algorithm is based on the wrong predicted samples in each epoch from classification, we adopt an alternate training method. In each epoch, we first use the cross-entropy loss ℓ_{ce} to optimize the model end-to-end. At the same time, we collect bias samples to construct positive and negative samples as the input of contrastive learning module. The training of comparative learning module is carried out after each epoch.

Two versions of the CLoCE are provided to optimize primary semantic representation (CLoCE-S) or deeper semantic representation (CLoCE-D) respectively. CLoCE-S only trains the embedding layer in contrastive learning module. In contrast, CLoCE-D trains both the embedding layer and RoBERTa-encoder during each epoch. On different datasets, CLoCE-S and CLoCE-D have their own merits.

Model	TACRED	TACREV	Re-TACRED
PA-LSTM(Zhang et al., 2017)	65.1	73.3	79.4
GCN(Zhang et al., 2018)	64.0	-	-
CGCN(Zhang et al., 2018)	66.4	74.6	80.3
C-AGGCN(Guo et al., 2019)	69.0	-	-
BERT-LSTM-BASE(Shi and Lin, 2019)	67.8	-	-
R-BERT(Wu and He, 2019)	69.4	-	-
ROBERTA-LARGE(Liu et al., 2019)	68.7	76.0	84.9
SPANBERT(Joshi et al., 2020)	70.8	78.0	85.3
GDPNET(Xue et al., 2021)	70.5	80.2	-
LUKE(Yamada et al., 2020)	**72.7**	80.6	90.3
PTR(Han et al., 2021)†	71.9	81.5	90.6
KNOWPROMPT(Chen et al., 2021)†	70.9	81.2	89.6
CLoCE-S(our)	69.9	**82.0**	89.6
CLoCE-D(our)	70.2	81.9	**91.9**

Table 1: Results on TACREDTACREV and Re-TACRED datasets with micro F_1-score(%) as metrics. To ensure a fair comparison, following the previous work, we exclude the influence of the no-relation samples when calculating F_1-score(%). The bold font is the current SOTA. The data that is not specially marked is the experimental results provided by the original paper. The data marked as "†" is the results of the code reproduction provided by the original author. Due to some random factors, the results may be different from the original paper. In the column, CLOCE-S is our model for optimizing shallow semantic representation by contrastive learning. Relatively CLOCE-D is our model for optimizing deep semantic information.

4 Expriment

In this section, we detail our experiments from three parts: datasets, basic settings and result analysis. In section 4.1, the information of datasets used in experiments is shown. In section 4.2, We introduce the baselines, metrics and parameter settings in experiments. The results on the full-dataset and few-shot dataset are introduced in section 4.3. The last part of this section analyses the training process.

4.1 Datasets

Following the previous work, we select three of the most popular datasets for sentence-level RE tasks: TACRED (Zhang et al., 2017), TACREV (Alt et al., 2020) and ReTACRED (Stoica et al., 2021). Table 2 shows the number of samples and relations in the three datasets. TACRED is a large-scale relation extraction dataset developed by the Stanford NLP group. This dataset is widely used for sentence-level tasks of relation extraction. TACREV revises the validation set and test set of original TACRED, meanwhile retains the training set. Re-TACRED makes some corrections to all sets of TACRED, including labeling bias and simplifying the classification of entity relation.

Dataset	#Train	#Val	#Test	#Rel
TACRED	68124	22631	15509	42
TACREV	68124	22631	15509	42
RE-TACRED	58465	19584	13418	40

Table 2: Statistics of different datasets.

In order to observe the performance of the model in the low-resource scenario, we refer to the method of KnowPrompt (Chen et al., 2021) to construct few-shot datasets. Specifically, we construct the training set with K = 8, 16 and 32 samples which are randomly selected from each class in training set.

4.2 Experimental Settings

Baselines and Metrics We choose baselines considering representative works of RNN, GCN and PLMs. KnowPrompt (Chen et al., 2021) and PTR (Han et al., 2021) are the representative works of prompt-tuning for RE. To be consistent with the previous work, we select F_1-micro(%) as the evaluation index, and keep unrelated entity pairs outside the calculation.

Hyperparameters In order to reduce the bias of the results of experiment which caused by the hyperparameters, we keep the hyperparameters neutral in the experiment environment settings. The hyperparameters of all CLoCE model experiments remain unchanged and other models preserve the original settings of the relevant papers.

In the full-dataset experiments, our batchsize is set to 16, and the learning rates of the classification module and the contrastive learning module are set

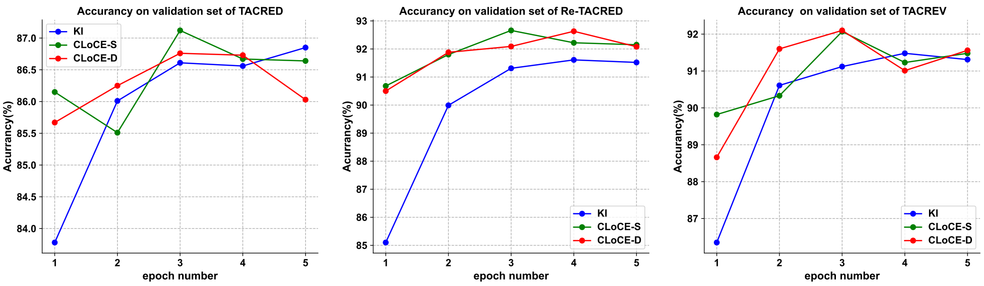

Figure 3: Results on validation set during training process. Accurancy(%) is treated as the evaluation index.

to 3e-5 and 1e-6, respectively. After each epoch of the classification module, contrastive learning module is executed. There 5 epochs are trained in total. In the few-shot experiment, we set random seeds 1-5 to generate a few-shot training set with K=8, 16, 32, the test set and validation set remain unchanged from the original dataset, and other parameters are consistent with the full-dataset experiment.

Enviroment The experimental environment uses two V100 GPU distributed to train a single model.The cuda vision is 10.2.

4.3 Experimental Results

Main Results For full-dataset experiments, the results show that the prompt-tuning method significantly outperforms the pre-training based approaches and other neural network approaches. It can be concluded that prompt-tuning can better improve the knowledge utilization of PLMs. In the comparison with prompt-tuning methods, CLoCE surpasses the current SOTA model in F_1 on both Re-TACRED and TACREV, and the performance on TACRED is close to SOTA, indicating the effectiveness of our model. By contrast, the optimization of deep or primary semantic features by contrastive learning has different performance on different datasets.

On small-scale datasets, the prompt-tuning methods are generally better than the pre-training based methods. The result indicates that prompt-tuning still has strong robustness in the scene of low-resource setting. The result shows that CLoCE performs better than FINE-TUNING method in most cases but weaker than PTR(Han et al., 2021) in K=8 settings. Extreme low-resource settings is still a challenge for CLoCE.

4.4 Analysis of Training Process

From Figure 3, the module of contrastive learning achieve higher accuracy and F_1-score with fewer

Few-shot Experiments				
Splite	Model	TACRED	TACREV	Re-TACRED
K=8	FINE-TUNING PTR KNOWPROMPT	- 28.1 -	10.5 25.3 28.6	20.1 43.6 45.8
	CLOCE-S CLOCE-D	10.5 9.6	13.2 12.6	13.8 10.9
K=16	FINE-TUNING PTR KNOWPROMPT	- 30.7 -	19.2 27.2 30.8	47.4 51.8 53.8
	CLOCE-S CLOCE-D	16.1 19.0	22.8 28.5	48.4 51.2
K=32	FINE-TUNING PTR KNOWPROMPT	- 32.1 -	26.0 33.1 34.2	53.6 54.8 55.2
	CLOCE-S CLOCE-D	29.2 27.7	26.2 29.2	54.3 52.6

Table 3: The results in the few-shot situation use micro F_1-score(%) as the evaluation index. K is defined as the number of samples randomly selected for each category in the training set of the original datasets. Referring to KNOWPROMPT(Chen et al., 2021), each experiment uses a random seed of 1-5 to generate the K-shot dataset. The experimental results of FINE-TUNING method, PTR and KNOWPROMPT are all from the original paper.The FINE-TUNING method refers to the BERT-BASE model.

	TACRED		**TACREV**		**Re-TACRED**	
	NUM	**MAX**	**NUM**	**MAX**	**NUM**	**MAX**
KI	4	86.9	5	91.5	4	92.0
CLOCE-S	4	**87.1**	3	**92.1**	3	**92.7**
CLOCE-D	4	86.8	3	**92.1**	4	92.6

Table 4: The best performance and epoch number on the validation set during training. The evaluation index uses micro F_1-score(%). In the first column, the three models are Prompt-tuning with knowledge injected(KI), CLoCE for optimizing primary semantic embedding(CLOCE-S), and CLoCE for optimizing deeper semantic representation(CLOCE-D).

epochs. In the three datasets, basically, CLoCE performs best after 3 epochs on the validation set on average. In contrast, KI need more epochs, 4 on average or more, for better performance.

At the same time, in order to verify the effectiveness of the contrastive learning and keep other settings unchanged, we use the template after knowledge injection and decouple the contrastive learning module. By comparison, it is found that our contrastive learning module can improve the semantic representation of prompt.

5 Conclusions

Prompt-tuning is a powerful tool for solving RE task. CLoCE provides a new framework for RE. In the full-dataset experiments, CLoCE outperforms most of the existing methods. However, it is still a big challenge to introduce contrastive learning into prompt-tuning framework in the few-shot setting. Especially in some extreme scenario (e.g. K = 8), the performance of CLoCE is still close to fine-tune method. At the same time, contrastive learning module can improve the speed of global optimization to a certain extent. CLoCE utilizes fewer epochs to achieve better results with less training times, which is also one of its competitiveness in low-resource scenario.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. *ArXiv*, abs/2004.14855.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Mary Elaine Califf and Raymond J. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *CoNLL*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *ArXiv*, abs/2104.07650.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. *ArXiv*, abs/2012.15723.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *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 the AAAI Conference on Artificial Intelligence*, volume 34, pages 7772–7779.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*.

John Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. *ArXiv*, abs/2006.03659.

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

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. *ArXiv*, abs/1906.07510.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. In *ACL/IJCNLP*.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018a. Hierarchical relation extraction with coarse-to-fine grained attention. In *EMNLP*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

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

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. Bertese: Learning to speak to bert. In *EACL*.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *NAACL*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *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)*, abs/2101.00190.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *ArXiv*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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*, abs/1907.11692.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *FINDINGS*.

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Relation extraction from wikipedia using subtree mining. In *AAAI*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Guanghui Qin and Jas' Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *ArXiv*, abs/2104.06599.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. *arXiv preprint arXiv:2010.13641*.

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

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

Peng Shi and Jimmy J. Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. *ArXiv*, abs/2010.15980.

Stephen Soderland, David Fisher, J. A. Aseltine, and Wendy G. Lehnert. 1995. Crystal: Inducing a conceptual dictionary. *ArXiv*, cmp-lg/9505020.

George Stoica, Emmanouil Antonios Platanios, and Barnab'as P'oczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *AAAI*.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *NAACL*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP*.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *ArXiv*, abs/2012.15466.

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

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *EMNLP*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *ArXiv*, abs/2105.11741.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. *ArXiv*, abs/2004.08056.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. In *NAACL*.

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

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

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*.

P. 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 *ACL*.

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