IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 August 2024, accepted 28 August 2024, date of publication 2 September 2024, date of current version 11 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3452720

RESEARCH ARTICLE

Few-Shot Relation Extraction Through Prompt With Relation Information and Multi-Level Contrastive Learning

YE DONG[1], RONG YANG[2], JUNBAO LIU[1], AND XIZHONG QIN[1,3]

[1]School of Computer Science and Technology, Xinjiang University, Ürömqi 830049, China
[2]Department of Nursing, Fifth Affiliated Hospital of Xinjiang Medical University, Ürömqi 830049, China
[3]Xinjiang Signal Detection and Processing Key Laboratory, Ürömqi 830049, China

Corresponding author: Rong Yang (292662644@qq.com)

This work was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region, Application Research on Assisted Intelligent Care for Inpatients Based on Electronic Medical Records under Grant 2021D01C428.

ABSTRACT Few-shot relation extraction uses only limited labeled data to predict relations between entities. Recently, several studies have introduced prompts to better guide models in understanding relations between entities. Although effective, these approaches ignore the hidden interaction information between support instances and relations, which causes the prompts without effective guidance. In addition, due to the limited labeled data, the model cannot get enough information for training, leading to the problems of relation confusion. In this paper, we propose **RelPromptCL**, a few-shot relation extraction method that consists of **Prompt** learning with **Rel**ation information and **C**ontrastive **L**earning. Specifically, RelPromptCL first gets more helpful information by utilizing prompt templates with relation information and then fuses the instance features with the relation features to obtain prototype representation. At the same time, the use of multi-level contrastive learning allows the model to discriminate more between different classes and improves the discriminative capability of the model. Finally, the similarity between the query instance and the prototypes is computed for relation classification. We carried out extensive experiments on both public datasets, the FewRel1.0 datasets and the FewRel2.0 datasets. The results clearly show the efficiency of RelPromptCL.

INDEX TERMS Few-shot relation extraction, prompt learning, contrastive learning, relation information, prototype network.

I. INTRODUCTION

Relation extraction (RE) [1] stands as a crucial task in Natural Language Processing (NLP), dedicated to extracting relations between entities from sentences, which can be leveraged for other advanced tasks [2], [3]. Most existing methods use supervised learning [4], which uses several labeled samples to train the model. However, labeling large-scale datasets is time-consuming and laborious, and the models are prone to insufficient data, leading to difficulties in effective generalization and overfitting. Hence, Han et al. [5] presented the few-shot relation extraction (FSRE) task that allows the model to be trained with the limited labeled data and gives terrific relation extraction results. Because of its validity, numerous innovative methods have been proposed in recent years to enhance the performance of few-shot relation extraction. Currently, mainstream FSRE approaches include optimization-based algorithms [6] and metric-based prototype network approaches [7], [8], [9], [10], [11], [12]. The optimization-based algorithms quickly adapt to new tasks by optimizing model parameters. The metric-based prototype network approach is designed to learn a metric space for classification by measuring the distance between instances in the query set and relation prototypes.

The associate editor coordinating the review of this manuscript and approving it for publication was Alicia Fornés.

Although these methods have achieved favorable results in the few-shot relation extraction task, they only connect the feature representations of the entity pairs as features of the relations between the pairs. This does not accurately express the feature representation of the relation between pairs of entities since even identical pairs of entities can have varied relations in different contexts. For example, ''John is in New York'' and ''John is from New York'' express different relations, although the entity pairs are the same.

Prompt learning can effectively leverage the knowledge of PLMs to predict the outputs required for downstream tasks. This makes the inclusion of prompt learning into the few-shot relation extraction became a new paradigm. These methods incorporate raw instances and prompts and then use the [mask] feature as part of the relation representation in the instance. For example, Li et al. [13] aggregated [cls] token embedded features and [mask] token embedded features as the original relation representation for each instance. Philipp Borchert et al. [14] add prompt templates for instance and relation sentences respectively, then use the [mask] feature as one of the sentence representations.

Although these methods have achieved good results, few-shot relation extraction, which is based on prompt learning, still has many problems. Firstly, these methods do not sufficiently use relation information in constructing the templates, which may make it difficult to guide the model in capturing useful relation features between entities. Secondly, in model learning, the amount of data obtained for each episode is limited, and the model cannot fully learn the features of each relation. This may make it difficult for the model to classify different relations during the classification process, causing the problem of relation confusion.

Demonstration learning [15] allows the model to output results that are more in line with expectations by giving the PLMs examples as prompts to tell the PLMs what task is being accomplished and how to accomplish it. Inspired by it, we propose an effective few-shot relation extraction model RelPromptCL, that combines prompt learning and contrastive learning. Specifically, the method adopts prompt learning to obtain more reliable instance representations by guiding the features of the model output [MASK] towards the real relation class by designing well-designed prompt templates incorporating the real relation class. Fig. 1 illustrates this paper's prompting method for infusing relation information. A richer representation of instances is obtained by combining the original instances with prompted templates, which guides the model in classifying more accurately through the prompts. In addition, to alleviate the relation confusion problem, we introduce instance- and prototype-level contrastive learning, where the first is to make the model focus on distinguishing different information between different relation instances, and the other focuses more on different information between different relation prototypes, as well as instance-prototype-level contrastive learning, which can pull instances closer to correspond relation prototype and push away other relation prototypes. The three types of contrastive learning interact with each other to make the model more discriminative across relations.

Our main contributions are listed below:

- We propose an effective few-shot relation extraction method based on prompt learning. (RelPromptCL). RelPromptCL enhances the model's representational capabilities by constructing prompt templates containing relation labels, enabling the model to obtain more accurate feature representations from prompt and true relation;
- We adopt the instance-, prototype-, and instance-prototype-level multi-level contrastive learning to enable the model to make better distinctions between different relations from various perspectives, which can improve classification accuracy;
- We conducted experiments on four settings of the Fewrel1.0 datasets [5], and the Fewrel2.0 datasets [16] to evaluate the effectiveness of RelPromptCL. A series of experiments demonstrated that RelPromptCL performs better than other FSRE methods.

II. RELATED WORK

A. FEW-SHOT RELATION EXTRACTION

Few-shot relation extraction (FSRE) identifies relations between pairs of entities using only the learning of the limited labeled data. The dominant approaches are optimization-based algorithms and prototype network approaches based on metric learning. Among them, the optimization-based algorithms aim to enable rapid adaptation to new tasks by optimizing the model parameters. Dong et al. [6] proposed a meta-learning framework based on meta-information-guided few-shot learning for classification to alleviate the reliance of support instances on optimal adaptation parameters. Qu et al. [17] proposed a new stochastic gradient Lanzmann dynamics meta-learning method, which can deal with the uncertainty of the prototype vectors and efficiently learn the a posteriori distribution of the relation prototype vectors. Although optimisation-based approaches are straightforward and practical, they also face the problem of overfitting. Therefore, considering that instances within the same class are proximate while those in different classes are distant, some researchers have proposed based on metric approaches for few-shot relation extraction.

The prototype network approach based on metric learning refers to learning a metric space in which the distance between instances and each prototype is computed using a metric function that selects the nearest relation prototype as a predictive label. Ideally, the prototype representation should accurately express the relation semantics. However, this is still a challenge for existing models. In order to obtain more reliable prototypes, one idea is to modify the prototype using information from the query instance. For example, Wen et al. [18] proposed mining class information from unlabelled query instances through clustering,

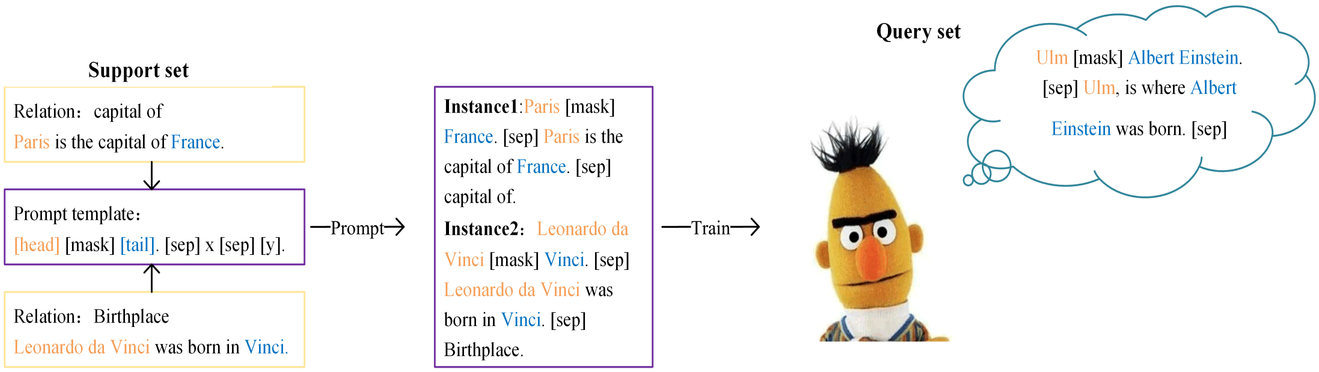


FIGURE 1. **Example of a 2-way 1-shot with a prompt template. Orange indicates the subject entity, and blue is the object entity. Instances are combined with relation information through prompt templates to lead the model to an understanding of the features of relation between entities. We use support instances for training and to predict the relation between entities in the query instance accurately.**

which modifies the initial prototype to obtain the final prototype representations. This improves the basic problem that the prototype acquired is always unreliable due to the limited number of support set instances. Another idea is to bring in external knowledge, such as relation knowledge or entity knowledge. Liu et al. [11] proposed an efficient and no-parameter required prototype generation method (PRM), which can explicitly utilize relation information along with instances to generate prototype representations and obtain more distinguishable prototypes. Zhang et al. [12] used generic and domain-specific Knowledge Graphs (KGs) for knowledge augmentation, which generated the prototype network and thus performed relation extraction to improve its domain adaptability.

B. PROMPT LEARNING

Prompt learning [19] has garnered significant attention recently, particularly with the introduction of large models like GPT-3 [20]. It efficiently guides PLMs in performing various NLP tasks by introducing well-designed or automatically generated prompt templates. Prompt templates are divided into hard prompt templates [21] and soft prompt templates [22] based on how they are input to the model. Whereas the hard prompt templates are designed with fixed text that does not change during training. Han et al. [23] proposed a rule-based approach to designing more efficient templates, which split the prompts in the template into several sub-prompts and combined them into a whole after constructing the sub-prompts. Soft prompt templates refer to trainable embedding vectors that can be dynamically adjusted. Li and Liang [24] replaced the prompt template discrete token with continuous virtual token embedding. Although these methods achieve promising results, relation class information is not used in constructing the template and may not adequately guide the model to capture key class features. We propose to add relation information to the hard prompt templates of the support set so that the model not only relies on the pre-trained model's prior knowledge during training but also learns by providing the correct answers to gain a much more accurate representation of the features.

C. CONTRASTIVE LEARNING

The successful use of unsupervised contrastive learning(CL) in computer vision(CV) [25], [26] has influenced research in NLP. The idea is to learn discriminative feature representations by constructing pairs of positive and negative samples, pulling pairs of positive samples to be close together, and pushing negative samples to be distant in the feature space. Han et al. [27] used a supervised contrast pre-training method to acquire more distinct representations during the pre-training phase. This allows semantically related representations to be closer to each other and vice versa. Peng et al. [28] proposed an entity-masked comparison pre-training framework that takes sentences in Wikidata that have the same relation as the input as positive samples and randomly selects sentences from other relations as negative samples. So that the model can better understand the text context and type information. Luo et al. [29] combined contrast loss for sentence anchoring and label anchoring to align the feature distributions between instances and relations. Borchert et al. [14] extracted multiple representations of a sample, and contrastive learning was used to extract additional discriminative information from these sentence representations. Inspired by these works, we designed multi-level contrastive learning to enhance the model's ability to distinguish between relations and alleviate the problem of relation confusion.

III. METHODS

In this section, we will begin by introducing the overall framework of RelPromptCL. Then, we will present the definition of the few-shot relation extraction task and specify the various parts of RelPromptCL.

A. OVERALL FRAMEWORK

As shown in Fig. 2, RelPromptCL consists of four components. (1) Prompt fusion: In this part, We blend original instances with well-designed prompt templates to get the enhanced instances. (2) Prototype generation: In this part, we first generate feature representations of the augmented instances through a shared sentence encoder.

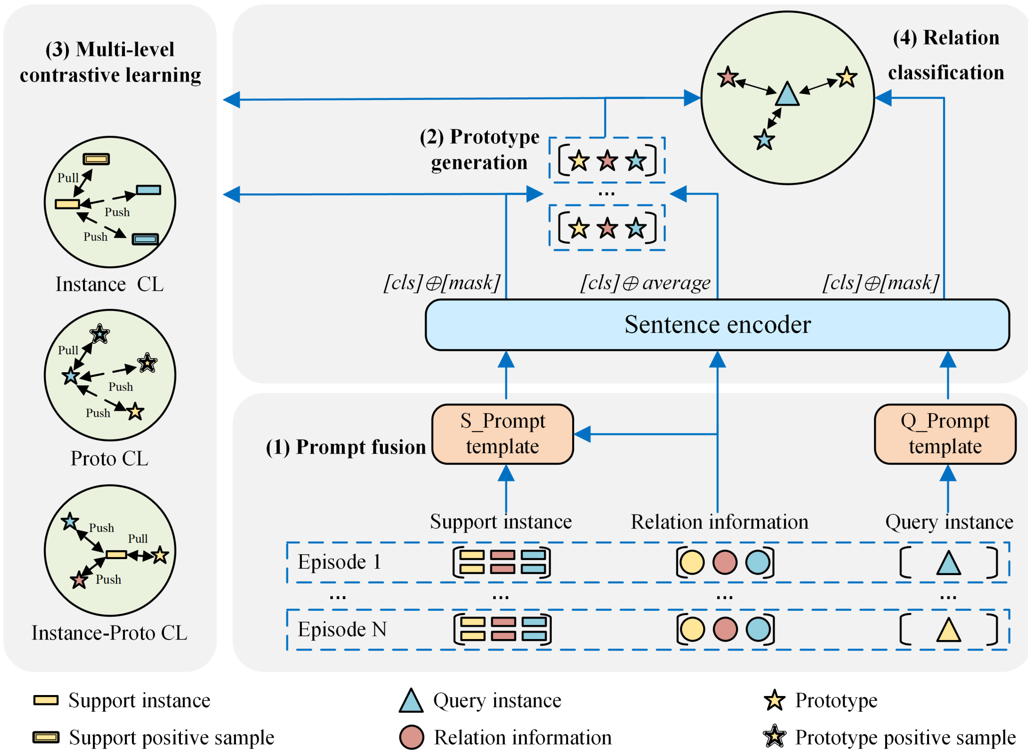


FIGURE 2. The architecture of RelPromptCL. RelPromptCL consists of four parts: (1) the Prompt fusion part, (2) the prototype generation part, (3) the multi-level contrastive learning part, and (4) the relation classification part. The $\oplus$ represents the connection operation.

Then, the prototype representation is obtained by combining the support instance features with the relation features. (3) Multi-level contrastive learning: In this part, we put the support instance features and prototype representations through multi-level contrastive learning so that RelPromptCL tends to get a more uniform feature distribution. (4) Relation classification: In this part, the relation between entities is identified by computing the similarity between the instance feature representation of the query set and the prototype feature representation.

Specifically, support and query instances are fused with the prompt template to get the instance representation with [mask] token by prompt function, respectively, and then generate the feature vector representation by the shared sentence encoder. Then, combine the support instance features with the relation features to get the final prototype feature representation. Meanwhile, the support instance features and final prototype features are then subjected to multi-level contrastive learning to obtain better-distributed instance and prototype representations. Finally, the query instance feature representation and prototype feature representation distances are computed to determine the relation between entity pairs.

B. TASK DEFINITION

The few-shot relation extraction task usually takes the N-way K-shot paradigm. Specifically, the entire dataset is divided into three sets: training set D_{val}, validation set D_{val}, and test set D_{test}, each of which has non-overlapping relations. Each set can be divided into episodes containing support set S and query set Q. The support set $S = \{s_k^n; n = 1, \ldots, N, k = 1, \ldots, K\}$ contains N different relations, each with K different labeled support instances. Moreover, the query set Q is the other instance in these N relations.

Each instance (x, e, y) consists of the contextual sentence $x = \{x_1, x_2, \ldots, x_n\}$, entity pairs $e = \{e_h, e_t\}$, and relation labels y. Where n is the length of sentence x; e_h denotes subject entity; e_t denotes object entity.

C. PROMPT FUSION

To steer the content of the model inference [mask] towards the correct relation, we incorporate relation labeling information in the constructed support set instance prompt template. That is, for a given support set instance x_s, it will be enhance by prompt template:

$$p_{support}(x_s) = \{[cls], e_h, [mask], e_t, [sep], x_s, [sep], y\} \quad (1)$$

where e_h denotes subject entity; e_t denotes object entity; x_s is the original query set instance; y is the relation between entities for this instance.

The prompt template for query set instances does not contain relation names, and we enhance the support set instance x_q with the prompt template as:

$$p_{query}(x_q) = \{[cls], e_h, [mask], e_t, [sep], x_q, [sep]\} \quad (2)$$

where e_h denotes subject entity; e_t denotes object entity; x_q is the original query set instance.

D. PROTOTYPE GENERATION

We encode the enhanced instances through the encoder and use the hidden state vector connection of the [cls] token and [mask] token as the feature representation of the input instances:

$$I(x) = h_{[cls]} \oplus h_{[mask]} \tag{3}$$

where $I(x) \in \mathbb{R}^{2\times d}$, $h_{[cls]}, h_{[mask]} \in \mathbb{R}^{d}$, d is the size of the encoder hidden state. $h_{[cls]}$ denotes the feature representation of [cls] in the instance, $h_{[mask]}$ denotes the feature representation of [mask] in the instance. The $\oplus$ represents the connection operation.

We concatenate the relation name and the relation description via [sep] as the relation representation and enter them into the encoder. We combine the special markers [cls] with the averaged values of all token features. That is, the relation i's feature representation is:

$$R^i = h_{[cls]} \oplus h_{[mean]} \tag{4}$$

where $R^i \in \mathbb{R}^{2\times d}$, $h_{[mean]} = \frac{1}{M}\sum_{m=1}^{M} h_{[m]}$ denotes the average of all token feature representations in relation i, M represents the number of tokens for the relation i.

We averaged each relation's K support instance feature representations to obtain the initial relation prototype.

$$P^i_{initial} = \frac{1}{K}\sum_{k=1}^{K} I(x^i_{sk}) \tag{5}$$

where $P^i_{initial} \in \mathbb{R}^{2d}, i = 1, \ldots, N$ is the initial relation prototype of relation i, and x^i_{sk} is the k-th supported instance of relation i.

Inspired by Liu et al. [30] proposed method of fusing the relation representation with the initial relation prototype of the instance, the final prototype representation is obtained by directly adding the initial relation prototype representation P and the relation representation R:

$$P^i = P^i_{initial} + R^i \tag{6}$$

where $P^i \in \mathbb{R}^{2d}, i = 1, \ldots, N$ is the final relation prototype of relation i.

E. CONTRASTIVE LEARNING

In order to make the model's representation of different relations of instances and prototypes more discriminative, we introduce instance-level CL, prototype-level CL, and instance-prototype-level CL to mitigate the relation confusion problem.

1) INSTANCE-LEVEL CONTRASTIVE LEARNING

In CL, the setting of positive and negative samples is extremely important. Inspired by Gao et al. [31] proposed using dropout for data augmentation. We use dropout for each instance feature of the support set to generate a new feature representation. And the instance features and dropout features of the same relation of that instance are used as positive samples for that instance. Instance features for other relations and dropout features are used as negative samples. Specifically, given an N-way K-shot episode with a total of $N \times K$ instances, $N \times K$ instance features are added after the dropout of each instance feature by instance-level contrastive learning. Finally, there are a total of $2 \times N \times K$ instance features. That is, for an instance feature $h(x_t)$ of a particular relation, we use a total of $2 \times K - 1$ features of the other instance features of the relation and the dropout feature corresponding to the instance of the relation as positive samples $h(x_p)$, while the remaining $2 \times (N-1) \times K$ instance features are used as negative samples $h(x_n)$. The 2-way 2-shot instance-level CL example is shown in Fig. 3. The instance-level CL loss is calculated as follows:

$$L_{SCL} = -\frac{1}{2NK}\sum_{t=1}^{2NK}\log \times \frac{\sum_{y_p=y_t}\exp(\frac{s(h(x_t),h(x_p))}{\tau})}{\sum_{y_p=y_t}\exp(\frac{s(h(x_t),h(x_p))}{\tau}) + \sum_{y_n\neq y_t}\exp(\frac{s(h(x_t),h(x_n))}{\tau})} \tag{7}$$

where $s(\cdot,\cdot)$ is the distance metric function, here we use dot product, which can be used as the similarity measure between two instance features, x_p means that its relation is the same as x_t, as a positive sample; x_n means that its relation is different from that of x_t, as a negative sample; y is the relation, and τ is the temperature factor.

Through instance-level contrastive learning, instance representations of the same relation are pulled closer together, and instance representations of different relations are drawn farther apart. This improves the model's ability to discriminate between different relations of instances.

2) PROTOTYPE-LEVEL CONTRASTIVE LEARNING

Similarly, we use dropout for each relation prototype P to generate new prototype features P^{i+} and consider its representation as the positive sample for that prototype, and the other relation prototypes and their dropout features as negative samples. The prototype-level CL loss is calculated as follows:

$$P^{i+} = dropout(P^i) \tag{8}$$

$$L_{PCL} = -\sum_{i=1}^{2N}\log\frac{\exp(\frac{s(P^i,P^{i+})}{\tau})}{\exp(\frac{s(P^i,P^{i+})}{\tau})) + \sum_{j\neq i,i+}\exp(\frac{s(P^i,P^j)}{\tau})} \tag{9}$$

where P^j means that its relation is different from that of P^i, as negative samples.

With prototype-level contrastive learning, the model can more easily distinguish between different relations of prototype representations.

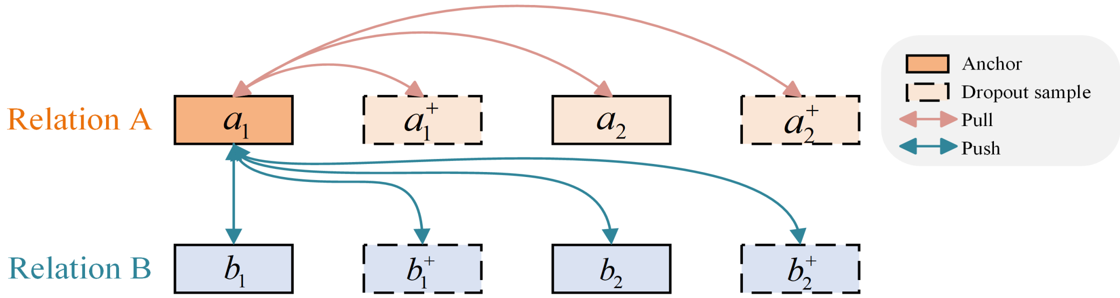


FIGURE 3. 2-way 2-shot Example of instance-level contrastive learning. Relation a has two instances, a_1 and a_2; Relation b has two instances, b_1 and b_2. The instance a_1 is passed through dropout to get its new feature representation a_1^+, and the instance a_2 is passed through dropout to get its new feature representation a_2^+. Assuming we use a_1 as the anchor, its positive sample are a_1^+, a_2 and a_2^+; its negative samples are b_1, b_1^+, b_2 and b_2^+. Contrastive learning is used to bring the a_1 instance features closer to the a_1^+, a_2 and a_2^+ features and push away the b_1, b_1^+, b_2 and b_2^+ features.

3) INSTANCE-PROTOTYPE-LEVEL CONTRASTIVE LEARNING

We use instance $I(x_i)$ as the anchor point, its relation prototype P^i as the positive sample, and the other relation prototype as the negative sample. Instance-prototype-level CL loss is calculated as follows:

$$L_{SPCL} = -\sum_{i=1}^{N} \log \frac{\exp(\frac{s(I(x_i),P^i)}{\tau})}{\exp(\frac{s(I(x_i),P^i)}{\tau}) + \sum_{j\neq i} \exp(\frac{s(I(x_i),P^j)}{\tau})} \tag{10}$$

where P^i means that its relation is the same as x_i, as positive samples; P^j means that its relation is different from that of x_i, as negative samples.

With instance-prototype-level contrastive learning, instances are pulled closer to their relation prototypes and pushed away from negative relation prototypes.

F. RELATION CLASSIFICATION

After obtaining N final relation prototypes, we calculate the similarity between the query instance and each prototype and use the relation class with the highest similarity as the prediction result. Then, label y for the j-th query instance as:

$$y_j = \arg\max_i \left(P^i \cdot I(q_j)\right) \tag{11}$$

We utilize the cross-entropy loss as the classification loss:

$$L_{CE} = -\sum_{i=1}^{N} (y_{ji} \times \log(\hat{y}_{ji})) \tag{12}$$

where i is the relation, j is the j-th query instance, $y_{ji} \in \{0, 1\}$ is the label, and $\hat{y}_{ji}$ is the probability of the j-th instance belongs to relation i.

We combine the classification loss with the multi-level contrastive learning losses to obtain the final loss.

$$L = L_{CE} + \lambda L_{SCL} + \lambda L_{PCL} + \lambda L_{SPCL} \tag{13}$$

where λ is the hyperparameter.

TABLE 1. Specific information of datasets.

Dataset	Source	Apply	Relation number	Instance number
FewRel1.0 & 2.0	Wiki	Train	64	44800
FewRel1.0	Wiki	Val	16	11200
		Test	20	14000
FewRel2.0	PubMed	Val	10	1000
		Test	15	1500

IV. EXPERIMENTS

A. DATASETS

The dataset used in the experiments is described below:

FewRel1.0 [5]: The FewRel1.0 dataset from Wikipedia includes 100 relations, each containing 700 pieces of training data. The entire dataset is divided into 64 relations for training, 16 relations for validation, and 20 unpublished annotated relations for testing.

FewRel2.0 [16]: The training set for FewRel 2.0 is consistent with that of FewRel 1.0. However, the validation and test sets are from the PubMed medical database to explore the cross-domain adaptability of the model. The entire dataset is divided into 64 relations for training, 10 relations for validation, and 15 non-publicly labeled relations for testing.

We used the FewRel 1.0 and FewRel 2.0 datasets officially provided by CodaLab Competitions at https://codalab.lisn.upsaclay.fr/competitions/7395 and https://codalab.lisn.upsaclay.fr/competitions/7397 to ensure a fair comparison. Table 1 gives specific information about each dataset.

B. EXPERIMENTAL SETTINGS

As with the official benchmark method, we evaluated the effectiveness of RelPromptCL on four different data setup methods, including 5-way 1-shot, 5-way 5-shot, 10-way 1-shot, and 10-way 5-shot. For a fair comparison, We respectively use Bert-base-uncased [32] and CP [28] as the sentence encoder. Where CP is a model obtained by BERT that underwent additional training using contrastive learning. We have set the number of training iterations to 30000 and 1000 validation iterations and the optimizer to Adam. The learning rate was set to 1e-5 in BERT and 3e-6 in CP. The dropout rate of the contrastive loss is set to 0.01, the temperature parameter τ is set to 0.03 in Bert, 0.08 in CP, λ is set to 0.1, and the batch size is set to 4. We evaluate the experimental results with accuracy(%).

Since CodaLab Competitions does not publish test set labels to the public, we need to submit our predictions on the evaluation page to get the accuracy of our predictions. The main results of RelPromptCL on the test set can be found in the CodaLab Competitions at https://codalab.lisn.upsaclay.fr/competitions/7395#results and https://codalab.lisn.upsaclay.fr/competitions/7397#results (User: dy1).

C. COMPARATIVE MODELS

We carried out comparison studies using the following approaches to confirm the validity of the suggested method:

- Proto-BERT [7]: A conventional prototype network model that utilizes BERT as its encoder.
- BERT-PAIR [16]: The BERT sequence classification model receives the linked sequence, which comprises every query instance coupled with every supporting instance, and classifies the sequence.
- REGRAB [17]: Efficiently learn relation prototype vectors for relation classification by combining global relational graphs and Bayesian meta-learning.
- TD-proto [9]: Design a co-attentive module to extract useful and instructive information for sentences and entities, respectively. A gating mechanism is used to dynamically fuse the two types of information to obtain a knowledge-aware instance.
- ConceptFERE [10]: By introducing the concept of entity intrinsic for few-shot relation extraction.
- HCRP [33]: Learn better representations using relation labeling information. The task Adaptive Focal Loss will be designed so that the model can learn adaptively and focus on challenging tasks.
- SimpleFSRE [30]: A method for few-shot relation extraction that creates the final prototype by immediately appending relation information to the initial prototype.
- MultiRep [14]: A few-shot relation extraction method combining multi-sentence representation and contrastive learning.
- MTB [34]: Propose a relation extraction model trained by matching the blanks and tuning on labeled data.
- CP [28]: A comparative pre-training model for entity masks for relation extraction.

D. EXPERIMENTAL RESULTS

The experimental results on the dataset FewRel 1.0 validation and test sets are shown in Table 2, and the experimental results on the dataset FewRel 2.0 test set are shown in Table 3. The results of the compared models are taken from the original paper.

TABLE 2. Results of FewRel 1.0 validation/test set. The table is partitioned based on the encoder, with BERT utilized for the upper section and CP for the lower section. Where '– –' indicates no experimental results, N-K is an abbreviation for N-way K-shot, and avg refers to average. Bold results indicate the best ones.

Model	5-1	5-5	10-1	10-5
BERT-PAIR	85.66/88.32	89.48/93.22	76.84/80.63	81.76/87.02
REGRAB	87.95/90.30	92.54/94.25	80.26/84.09	86.72/89.93
TD-proto	——/84.76	——/92.38	——/74.32	——/85.92
ConceptFERE	——/89.21	——/90.34	——/75.72	——/81.82
HCRP	90.90/93.76	93.22/95.66	84.11/89.95	87.79/92.10
SimpleFSRE	91.29/94.42	94.05/96.37	86.09/90.73	89.68/93.47
MultiRep	**92.73**/94.18	93.79/96.29	**86.12**/91.07	88.80/91.98
RelPromptCL	92.12/**94.71**	**94.42/97.38**	85.77/**91.18**	**89.28/94.48**
MTB	——/91.10	——/95.40	——/84.30	——/91.80
CP	——/95.10	——/97.10	——/91.20	——/94.70
HCRP(CP)	94.10/96.42	96.05/97.96	89.13/93.97	93.10/96.46
SimpleFSRE(CP)	**96.21**/96.63	97.07/97.93	**93.38**/94.94	95.11/96.39
RelPromptCL(CP)	**96.21/96.76**	**97.80/98.33**	92.95/**94.99**	**95.80/96.70**

TABLE 3. Results of FewRel2.0 test set. Bold results indicate the best ones.

Model	5-1	5-5	10-1	10-5	Avg
Proto-BERT	40.12	51.50	26.45	36.93	38.75
BERT-PAIR	67.41	78.57	54.89	66.85	66.93
MTB	74.70	87.90	62.50	81.10	76.55
CP	79.70	84.90	68.10	79.80	78.13
HCRP	76.34	83.03	63.77	72.94	74.02
RelPromptCL	**81.24**	**91.54**	**69.04**	**84.66**	**81.62**

As seen in Table 2, using Bert as the encoder, RelPromptCL improves significantly compared to previous models, achieving the best results on all tasks. To be specific, RelPromptCL has a test set accuracy of 94.71% in task 5-way 1-shot, 97.38% in task 5-way 5-shot, 91.18% in task 10-way 1-shot, and 94.48% in task 10-way 5-shot, for an average test accuracy of 94.44%. Compared to MultiRep, the state-of-the-art prompt approach, which uses five sentence representations, RelPromptCL performs better using only [cls] and [mask] features. This illustrates the effectiveness of incorporating relation information into prompt templates. 10-way 1-shot is slightly higher than MultiRep because MultiRep uses multiple feature representations, which can capture more instance features. Using CP as the encoder, RelPromptCL produces the finest outcomes on all tasks, especially on the 5-way 5-shot set, which is 0.4% higher than the optimal method. This proves that RelPromptCL is effective. As can be seen in Table 3, RelPromptCL also achieves good results in the cross-domain task. The average improvement over the current state-of-the-art CP method was 3.49%, especially on the 5-way 5-shot setting, which was 6.76% higher than CP. It is shown that RelPromptCL guides the model to utilize the knowledge contained in PLMs for downstream tasks through prompt learning, which greatly enhances the model's generalization performance to handle cross-domain tasks effectively.

As demonstrated by Tables 2 and Table 3's results, RelPromptCL performs well compared to the above method. The performance improvement comes from two main aspects: (1) Incorporating relation names in the prompt template can make the prompt template more effective and guide

TABLE 4. Ablation study of RelPromptCL on each component.

Module	5-1	5-5	10-1	10-5
RelPromptCL	**94.71**	**97.38**	**91.18**	**94.48**
w/o SCL	94.65	96.64	90.73	93.86
w/o PCL	94.17	96.41	90.27	94.02
w/o SPCL	94.23	97.07	90.19	94.26
w/o Rel prompt	94.33	96.83	90.47	93.29

the model to get more reliable feature representations. (2) The introduction of multi-lever contrastive learning makes instances of different relations more distinguishable, alleviating the problem of relation confusion.

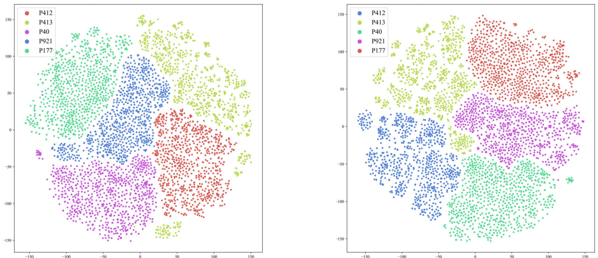
E. ABLATION STUDY

To verify the effectiveness of RelPromptCL, we perform ablation experiments on the FewRel 1.0 dataset by removing individual parts from the model. Where 'w/o SCL' denotes the removal of instance-level contrastive learning on top of RelPromptCL, 'w/o PCL' denotes the removal of prototype-level contrastive learning, 'w/o SPCL' denotes the removal of instance-prototype level contrastive learning, and 'w/o Rel prompt' denotes the removal of relation information in the prompt template. Table 4 shows the test set's accuracy results.

From Table 4, (1) Compared to RelPromptCL, removing one of the contrastive learning methods decreases the effectiveness of all of them, suggesting that the model can capture different features in the data using three different contrastive learning methods and that by combining these methods and utilizing their individual strengths, a more comprehensive and enriched representation can be obtained. (2) It is important to note that the accuracy of "w/o PCL" significantly decreases, probably because it only focuses on the differentiation between instances versus the differentiation between instances and prototypes, ignoring the differentiation between prototype relations. Demonstrate that enhanced discrimination between prototype relations is key to mitigating relation confusion. (3) In addition, the average decrease in accuracy under the "w/o Rel prompt" setting was 0.71, demonstrating that the prompt templates incorporating relation names provide the model with additional semantic information that can more explicitly guide the model in making predictions, thus enhancing the accuracy of the model. In summary, every module inside the model contributes to improving the few-shot relation extraction task's performance.

F. VISUALIZATION

In order to fully demonstrate the validity of the proposed RelPromptCL, we use t-distributed Stochastic Neighbor Embedding (t-SNE) [35] to visualize the instances in the FewRel 1.0 validation set in a 5-way 5-shot setup to show the difference in effectiveness between RelPromptCL and MultiRep.



(a) t-SNE of MultiRep in random relations

(b) t-SNE of the RelPromptCL in random relations

FIGURE 4. The t-SNE plot of the MultiRep and RelPromptCL in random relations.

TABLE 5. Silhouette coefficient of the MultiRep and RelPromptCL in random relations.

Model	Silhouette coefficient
MultiRep	0.30356
RelPromptCL	0.33306

First, we randomly select five relations (P412: voice type voice type, P413:position played on team/speciality, P40: child, P921: has facility, P177:crosses), and for each relation, we randomly sample 1000 instances of the features for t-SNE visualization. As shown in Fig. 4, (a) represents the result of the MultiRep, while (b) represents the result of RelPromptCL. The matching colors indicate the same relation. It is clear from Fig. 4 that for relation P413, the sample features of MultiRep are more dispersed than those of RelPromptCL.

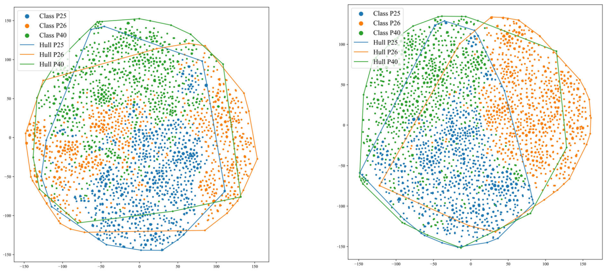
In order to evaluate the aggregation ability of MultiRep and RelPromptCL more intuitively, we calculate the silhouette coefficient [36] to measure the effect of clustering. A higher silhouette coefficient indicates denser clusters and more dispersed clusters. The calculation method is shown below:

$$\text{silhouette coefficient} = \frac{b_i - a_i}{\max\left(a_i, b_i\right)} \tag{14}$$

where a denotes the average distance of the sample to other samples in the same class, and b denotes the average distance of the sample to other classes.

Table 5 gives the silhouette coefficient of MultiRep and RelPromptCL, and it can be found that the silhouette coefficient of MultiRep is lower than that of RelPromptCL. Combined with Fig. 4 and Table 5, it can be shown that the RelPromptCL proposed in this paper has better clustering performance than the features generated by MultiRep.

We then selected three more confusing relations (P25: mother, P26: spouse, and P40: child) and visualized 1000 randomly selected instances of each relation for comparison. As shown in Fig. 5, (a) is the result of MultiRep, and (b) is the result of RelPromptCL. Meanwhile, in order to more intuitively assess the degree of overlap between similar relations for MultiRep and RelPromptCL, we calculate the ratio of the number of samples in the overlap region between different relations to the total number of relation samples to



(a) t-SNE of the MultiRep in similar relations

(b) t-SNE of the RelPromptCL in similar relations

FIGURE 5. **The t-SNE plot of the MultiRep and RelPromptCL in similar relations with boundaries.**

TABLE 6. **Percentage of overlap between MultiRep and RelPromptCL in similar relations.**

Model	MultiRep	RelPromptCL
Overlap rate between P25 and P26	74.90%	51.40%
Overlap rate between P25 and P40	78.60%	83.83%
Overlap rate between P26 and P40	77.44%	57.75%
Average overlap rate	76.98%	64.33%

obtain the percentage of overlap. Table 6 gives the overlap percentage between the different relations of MultiRep and RelPromptCL. Although RelPromptCL has a slightly higher percentage of overlap than MultiRep between relation P25 and P40, on average, RelPromptCL has a lower percentage of overlap than MultiRep. Combined with Fig. 5 and Table 6, it can be seen that in these 3 confusing relations, the RelPromptCL proposed in this paper has less overlap between the different relations than the MultiRep. This suggests that the RelPromptCL is effective in mitigating relation confusion problems by distinguishing more between these confusing relation instances.

V. CONCLUSION

This paper proposes a few-shot relation extraction method based on prompt and contrastive learning (RelPromptCL). RelPromptCL incorporates relation information into the prompt template to steer the model toward a more accurate prototype representation. Meanwhile, multi-level contrastive learning can improve the ability of the model to distinguish between different relations. The experimental results show that the method proposed in this paper outperforms other few-shot relation extraction methods, and the ablation analysis proved the validity of each part of RelPromptCL. We intend to extend this methodology to other tasks in the future, such as continuous few-shot relation extraction.

REFERENCES

[1] X. Zhao, Y. Deng, M. Yang, L. Wang, R. Zhang, H. Cheng, W. Lam, Y. Shen, and R. Xu, "A comprehensive survey on relation extraction: Recent advances and new frontiers," *ACM Comput. Surveys*, vol. 56, no. 11, pp. 1–39, Nov. 2024.

[2] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 615–620.

[3] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3961–3973, Aug. 2022.

[4] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1298–1307.

[5] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun, "FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4803–4809.

[6] B. Dong, Y. Yao, R. Xie, T. Gao, X. Han, Z. Liu, F. Lin, L. Lin, and M. Sun, "Meta-information guided meta-learning for few-shot relation classification," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 1594–1605.

[7] J. Wang and Y. Zhai, "Prototypical Siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 178–181.

[8] T. Gao, X. Han, Z. Liu, and M. Sun, "Hybrid attention-based prototypical networks for noisy few-shot relation classification," in *Proc. 33rd Conf. Artif. Intell. (AAAI)*, 2019, pp. 6407–6414.

[9] K. Yang, N. Zheng, X. Dai, L. He, S. Huang, and J. Chen, "Enhance prototypical network with text descriptions for few-shot relation classification," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2273–2276.

[10] S. Yang, Y. Zhang, G. Niu, Q. Zhao, and S. Pu, "Entity concept-enhanced few-shot relation extraction," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 987–991.

[11] Y. Liu, J. Hu, X. Wan, and T.-H. Chang, "Learn from relation information: Towards prototype representation rectification for few-shot relation extraction," in *Findings of the Association for Computational Linguistics*. Seattle, WA, USA: Association for Computational Linguistics, 2022, pp. 1822–1831. [Online]. Available: https://aclanthology.org/2022.findings-naacl.139/

[12] J. Zhang, J. Zhu, Y. Yang, W. Shi, C. Zhang, and H. Wang, "Knowledge-enhanced domain adaptation in few-shot relation classification," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2183–2191.

[13] L. Li, Y. Zhang, J. Zou, and Y. Huang, "PNPT: Prototypical network with prompt template for few-shot relation extraction," in *Proc. Chin. Nat. Conf. Social Media Process.*, 2023, pp. 161–173.

[14] P. Borchert, J. De Weerdt, and M.-F. Moens, "Efficient information extraction in few-shot relation classification through contrastive representation learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2024, pp. 638–646.

[15] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 3816–3830.

[16] T. Gao, X. Han, H. Zhu, Z. Liu, P. Li, M. Sun, and J. Zhou, "FewRel 2.0: Towards more challenging few-shot relation classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6250–6255.

[17] M. Qu, T. Gao, L.-P. Xhonneux, and J. Tang, "Few-shot relation extraction via Bayesian meta-learning on relation graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7867–7876.

[18] M. Wen, T. Xia, B. Liao, and Y. Tian, "Few-shot relation classification using clustering-based prototype modification," *Knowl.-Based Syst.*, vol. 268, May 2023, Art. no. 110477.

[19] Y. Gu, X. Han, Z. Liu, and M. Huang, "PPT: Pre-trained prompt tuning for few-shot learning," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 8410–8423.

[20] T. B. Brown, "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[21] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–20.

[22] J. Wu, T. Yu, R. Wang, Z. Song, R. Zhang, H. Zhao, C. Lu, S. Li, and R. Henao, "Infoprompt: Information-theoretic soft prompt tuning for natural language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–26.

[23] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "PTR: Prompt tuning with rules for text classification," *AI Open*, vol. 3, pp. 182–192, Jun. 2022.

[24] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4582–4597.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[27] J. Han, B. Cheng, and G. Nan, "Learning discriminative and unbiased representations for few-shot relation extraction," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 638–648.

[28] H. Peng, T. Gao, X. Han, Y. Lin, P. Li, Z. Liu, M. Sun, and J. Zhou, "Learning from context or names? An empirical study on neural relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 3661–3672.

[29] D. Luo, Y. Gan, R. Hou, R. Lin, Q. Liu, Y. Cai, and W. Gao, "Synergistic anchored contrastive pre-training for few-shot relation extraction," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 17, pp. 18742–18750.

[30] Y. Liu, J. Hu, X. Wan, and T.-H. Chang, "A simple yet effective relation information guided approach for few-shot relation extraction," in *Findings of the Association for Computational Linguistics*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 757–763. [Online]. Available: https://aclanthology.org/2022.findings-acl.62/

[31] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 1–14.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2019, pp. 4171–4186.

[33] J. Han, B. Cheng, and W. Lu, "Exploring task difficulty for few-shot relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 2605–2616.

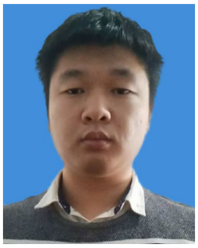
[34] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2895–2905.

[35] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–24, 2008.

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

YE DONG is currently pursuing the master's degree with the School of Computer Science and Technology, Xinjiang University. Her research interests include natural language processing and few-shot relation extraction.

RONG YANG is currently working as the Chief Nurse with the Fifth Affiliated Hospital of Xinjiang Medical University. Her research interests include clinical nursing, nursing management, and clinical nursing applications.



JUNBAO LIU received the B.S. degree in communication engineering from Huaibei Normal University, China. He is currently pursuing the master's degree in computer science and engineering with Xinjiang University. His research interests include deep learning and natural language processing.

XIZHONG QIN received the master's degree in radio engineering from Southeast University, China. He is currently working as an Associate Professor and the Master's Supervisor with the School of Computer Science and Technology, Xinjiang University. His research interests include machine learning, natural language processing, and deep learning.