Pattern Recognition 156 (2024) 110779

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Few-shot relational triple extraction with hierarchical prototype optimization

Chen Gao [a], Xuan Zhang [b,c,*], Zhi Jin [d,e], Weiyi Shang [f], Yubin Ma [b], Linyu Li [d,e], Zishuo Ding [g], Yuqin Liang [b]

[a] School of Information Science and Engineering, Yunnan University, Yunnan 650091, China
[b] School of Software, Yunnan University, Yunnan 650091, China
[c] Key Laboratory of Software Engineering of Yunnan Province, Yunnan 650091, China
[d] Key Lab. of High-Confidence Software Technologies (MoU), Peking University, Beijing 100871, China
[e] School of Computer Science, Peking University, Beijing 100871, China
[f] Department of Electrial and Computer Engineering, University of Waterloo, Ontario, Canada
[g] Data Science and Anaytics Thrust, Hong Kong University of Science and Technology (Guangzhou), Guangdong 511442, China

ARTICLE INFO

Keywords:
Few-shot learning
Relational triple extraction
Hierarchical prototype optimization
Contrastive learning
Prompt learning
Prototype network

ABSTRACT

Relational Triple Extraction (RTE) aims to extract relations and entities from unstructured text. Current RTE models using supervised learning require a large amount of labeled data, which presents a challenge for real-world applications. Therefore, the research work on Few-Shot Relational Triple Extraction (FS-RTE) has been proposed. However, the existing work cannot effectively construct accurate prototypes from a small number of samples, and it is difficult to model the dependencies between entities and relations, resulting in poor performance in relational triple extraction. In this paper, we propose a Hierarchical Prototype Optimized FS-RTE method (HPO). In particular, to mitigate prototype bias built on a small number of samples, HPO uses prompt learning to merge the information of relational labels into the text. Then, the entity-level prototypes are constructed using a span encoder to avoid label dependency between entity tokens. Finally, the hierarchical contrastive learning (HCL) method is introduced to improve the metric space between the prototypes of entities and relations, respectively. Experiments conducted on two public datasets show that HPO can significantly outperform previous state-of-the-art methods.

1. Introduction

Relational triple extraction (RTE) is a subtask of information extraction [1,2]. It plays a crucial role in the construction of knowledge graphs [3], which has important applications in fields such as machine translations [4], question answering systems [5], and recommender systems [6]. For example, given the sentence "*Beijing is the capital of China*", the expected output of the RTE system is the relational triple <*Beijing*, *capital_of*, *China*>, where "*capital_of*" denotes the relation present in the sentence, while "*China*" and "*Beijing*" correspond to the head and tail entity, respectively. Recently, although the relational triple extraction methods based on supervised learning have achieved good performance [7,8], these supervised learning approaches require a large amount of labeled data, which is difficult to meet the practical needs of the realistic application. In many domains, there is not enough labeled data to use supervised learning-based relational triple extraction methods directly. Furthermore, supervised learning cannot learn entities and relation types that have not been seen before. To solve the problem of lacking labeled data, researchers propose few-shot learning methods. Few-shot learning (FSL) [9] commits to learning with only a few samples. In FSL, the dataset is generally divided into support and query sets. The model learns a new concept from only a few instances in the support set while maintaining good generalization in the query set. Based on the transfer learning paradigm, Liu et al. [10] proposed a regenerative network of self-supervised label enhancement methods to reduce the generalization error in the learning process. In addition, to solve the noisy label problem in small sample learning, An et al. [11] designed a dual network structure based on contrastive network and meta-network to extract feature-related intra-class and inter-class information, respectively.

Few-shot relational triple extraction (FS-RTE) methods [12,13] aim to extract emerging relational triples based on a given small amount

* Corresponding author at: School of Software, Yunnan University, Yunnan 650091, China.
E-mail addresses: cg2@mail.yun.edu.cn (C. Gao), zhxuan@ynu.edu.cn (X. Zhang), zhijin@pku.edu.cn (Z. Jin), wshang@uwaterloo.ca (W. Shang), 1023826776@qq.com (Y. Ma), linyuli@stu.pku.edu.cn (L. Li), zishuoding@hkust-gz.edu.cn (Z. Ding), liangyq@mail.ynu.edu.cn (Y. Liang).

https://doi.org/10.1016/j.patcog.2024.110779
Received 11 January 2024; Received in revised form 9 June 2024; Accepted 10 July 2024
Available online 14 July 2024

Table 1
An example of 2-Way 2-Shot FS-RTE. The head entity is underlined and the tail entity is in wavy lines.

Support set	
R1: Born_in	**Instance1:** Donald Trump was born in New York. **Instance2:** Jay Chou, a famous singer from Taiwan, China.
R2: Capital_of	**Instance1:** The capital of the People's Republic of China was set in Bei Ping, which was renamed Beijing with immediate effect. **Instance2:** Washington is located in the northeastern United States and is the capital of the United States.
Query set	
R1 or **R2**	Adele Adkins is a famous pop singer from England. Published many popular singles.

of labeled data. By employing the FSL strategy, it can achieve better performance while significantly reducing the data annotation. Table 1 shows a 2-Way 2-Shot FS-RTE task. There are two types of relations in the support set, namely "*Born_in*" and "*Capital_of*", and each type of relation contains two instances. The head entity is highlighted in underlined, and the tail entity is wavy lines in the given example. The query set contains a sample to be classified, and the model needs to learn based on the examples in the support set to make predictions on the query set.

In recent years, FS-RTE methods have developed rapidly. Most of the previous work used a pipeline-based approach and modeled it as two subtasks to solve, namely few-shot named entity recognition (FS-NER) [14–16] and few-shot relation extraction (FS-RE) [17,18]. For example, the pipeline-based approach proposed by Nasar et al. [19] first extracts all entities in a sentence based on the FS-NER method, and then classifies the relations of the identified entity pairs using an FS-RE method. Such approaches require constructing two different models, leading to high time and space complexity. In addition, they separate the linkage between the two tasks and often result in the generation of redundant entities [20]. To address the shortcomings of the current pipeline paradigm, some related works have proposed joint FS-RTE methods [12,13], which do not model the two subtasks separately but employ a unified model to address both subtasks simultaneously. For example, Yu et al. [12] conducted the first study of FS-RTE in the N-Way K-Shot setting and proposed a multi-prototype embedding network (MPE) model to extract relational triples in sentences. The model adopts the entity-then-relation paradigm, which first performs NER using conditional random fields (CRF) [21]. After that, a prototype of the head and tail entities is constructed using a prototype network [22] approach to generate a relation representation and encode it to the sentence representation to generate a relation prototype. Finally, relational triples in the query set are identified using these prototypes. Cong et al. [13] proposed a joint FS-RTE method, which effectively alleviates the entity redundancy problem by constructing relation and entity prototypes to extract relations in sentences and then identify entities based on specific relations. He et al. [23] improved the FS-NER performance with the nearest neighbor matching strategy [24], which improved the overall performance of FS-RTE. Fei et al. [25] proposed a perspective transfer network (PTN) for FS-RTE.

The above methods have solved the problem of FS-RTE to some extent, but there are still shortcomings. Since the support set only contains a small number of samples under each relation, the prototypes built based on these samples are inaccurate, especially in the 1-shot task. The inaccurate prototype cannot represent the overall characteristics of the relation well, thus affecting the result of the model. In other words, the relation and entity prototypes are constructed directly based on several samples in the support set. Such prototype construction is very crude, and it is difficult to generate accurate prototypes. In addition, the token-based entity prototype construction does not consider the overall semantic information of entities. Such a prototype network suffers from a rough estimation of label dependency. Label dependencies are impossible when few-shot NER models are involved since some labeled data is not enough to learn reliable dependencies, and the label set may vary from domain to domain. It is difficult to accurately represent the original semantic information of entities by splitting the entity prototype into two parts: start and end. For example, "*new city*" and "*new york city*" have the same start and end words, but they represent different meanings. Finally, the existing methods do not model the association between entity and relation prototypes. Different relations have constraints on the types of entities, and vice versa [26]. For example, the relation "*Born in*" determines the probability of the entity types being "*Person*" and "*Location*". In the metric space, the types of these two entities should be distributed around the relations. Intuitively, the prototypes of different relations should be kept at a certain distance interval, and the same goes for entity prototypes. Prototypes of related relations and entities should be closely distributed in the metric space, otherwise distances should be maintained.

To solve the above problems, we propose a hierarchical prototype optimization-based FS-RTE method (HPO). HPO constructs relational prototypes by introducing relational information based on prompt templates and combining sentences in the support set. The query set and the relation prototype are then compared to detect the presence of predefined relations in the query set statements. Once the relation is detected, HPO constructs a span-encoded representation of the sentence from instances under the specific relation, thereby obtaining entity-level prototypical representations of the head and tail entities in the sentence. Finally, all potential entities in the query set sentences are obtained by comparing the query set and entity prototypes to extract the relational triples while using hierarchical contrastive learning(HCL) to optimize the representation of relations and entity prototypes. Fig. 1 shows an overview of our approach.

Relation label information can accurately represent the semantics of relations and related entities. Regarding RE, HPO complements the relation prototypes constructed from the support set by introducing knowledge of external relations information. For NER, HPO builds entity-level prototypes through span-based sentence encoding. It uses text spans in a sentence as candidate entities and then uses a similarity measure to determine whether a span is an entity. In this way, it alleviates the label dependency problem and takes full advantage of the complete semantic information of entities. Furthermore, HPO proposes a hierarchical contrastive learning approach to optimize the representation of entity and relation prototypes to jointly optimize the metric space of entity and relational prototypes and model the association between them. Specifically, it uses the distance between the relation and the entity prototype as the optimization target to perform contrastive learning in hierarchies. The specific contrastive learning strategy mainly consists of the following aspects: First, the distance of different prototypes of relations should be kept at suitable intervals. Second, the distance of prototypes of different types of entities should be kept at suitable intervals. Finally, interrelated relations and entities should be closely distributed in the metric space.

In summary, the main contributions of this paper are as follows:

1. We propose an FS-RTE method based on hierarchical prototype optimization (HPO). Specifically, we optimize the relation prototype representation on the relation extraction task by introducing relation label knowledge using prompt learning. On the task of named entity recognition, we construct span-based entity-level prototypes, which can effectively alleviate the problem of label dependence and semantic incompleteness.

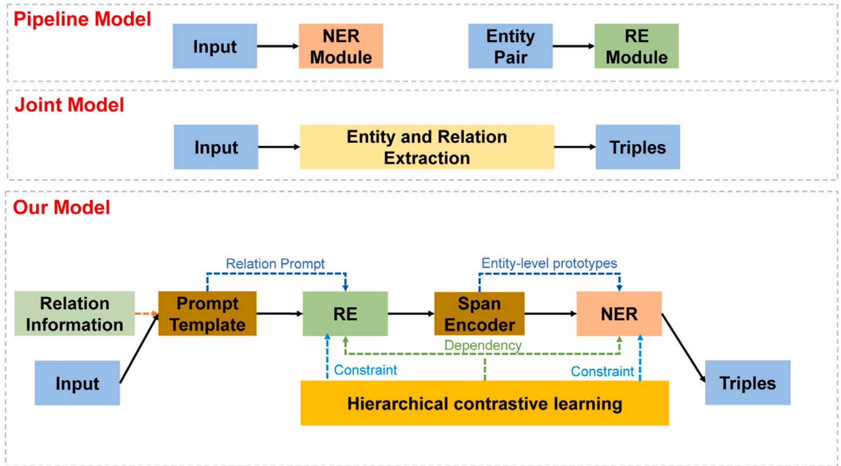


Fig. 1. The extraction processes of existing approaches and our method.

2. To better model the association between relation and entity prototypes, we introduce hierarchical contrastive learning to jointly refine the metric space for relation extraction and entity recognition, thus enhancing the decentralized distribution of prototypes. It can optimize the representation of prototypes from three levels: entities, relations, and interactions between entities and relations.
3. Extensive experiments on two datasets are conducted to compare the proposed HPO with state-of-the-art baseline models. The results show that HPO achieves a significant improvement of 1.9%~11.1% in terms of the F1 score for FS-RTE.

2. Related work

As an important subtask in information extraction, RTE is also known as entity and relation extraction, which refers to the extraction of the predefined relation between entities and entities from unstructured text. Based on extraction steps, RTE is mainly divided into two types: pipeline learning methods [27,28] and joint learning methods [7,8]. The pipeline-based methods usually contain two steps and require NER of text first and then RE based on the completion of entity recognition. Although the pipelined learning-based RTE approach simplifies the task process, it suffers from error propagation, missing interactions, and entity redundancy. Compared with the pipeline learning-based approach, joint learning-based methods use a single neural network model to perform triple extraction, simultaneously. Supervised learning [28] based RTE methods have shown satisfactory performance, but still have limitations. Since supervised learning requires a large amount of labeled data, which is often a very labor-intensive task when oriented to specific domains, and lack of model transferability to new domains. In addition, supervised learning models cannot handle invisible relations and entities. Therefore, it is essential to explore FS-RTE.

At present, pipeline-based FS-RTE is mainly decomposed into two sub-tasks, namely FS-NER [14–16] and FS-RE [17,29]. Fritzler et al. [30] proposed to use the prototype network for NER, by constructing entity prototypes and using the method of similarity matching to realize entity extraction in few-shot scenes. With the rise of prompt learning [31], methods based on prompt learning templates have also been applied to the field of NER. Cui et al. [32] proposed to build a prompt template to convert the NER task from the traditional sequence labeling task to the classification task so that the information in the pre-trained model can be more fully utilized. In the field of FS-RE, Han et al. [33] built the first large-scale few-shot relational extraction dataset based on Wikipedia data sources, which played a crucial role in promoting the development of related tasks. Gao et al. [34] first proposed an attention-based hybrid prototypical network for FS-RE. Some other works [35–37] used an attention mechanism to incorporate the knowledge of relation labels to alleviate the prototype representation bias. These methods provide ideas for the solution of FS-RTE, but they only consider a single task and model each task separately, which still cannot fundamentally solve the problems faced by FS-RTE.

To simultaneously utilize relations and entities in FS-RTE, Yu et al. [12] pioneered a multi-prototype embedding network model to extract relational triples jointly. They first use CRF to identify entities in sentences and then connect text and knowledge about entities and relations by designing a hybrid prototype learning mechanism. Finally, they perform relation classification on the extracted entities. This method unifies the two subtasks in RTE for the first time but still has limitations. Firstly, achieving the desired performance by directly using CRF for NER is difficult due to insufficient labeling data in the FS-RTE task. Secondly, the extracted entities have a large amount of redundancy, which will affect the subsequent RE. To solve the above problems, Cong et al. [13] proposed a relation-then-entity extraction paradigm, which constructs relation and entity prototypes from the data in the support set. Specifically, it first identifies the relations in the sentences and then extracts the entities contained in the identified relations. Wang et al. [38] introduce heterogeneous graph networks to address both error propagation and data defects. He et al. [23] propose to extract relational triplet sequences, which are based on prototype network and nearest neighbor matching [24], to identify head and tail entities in sentences based on the semantic similarity of words. To better use global information, Fei et al. [25] proposed a novel perspective transfer network to solve FS-RTE. Specifically, it first identifies relations in sentences. Then, move from the relational to the entity perspective to extract the entities in the sentence. Finally, it shifts to the triple perspective to verify the plausibility of the extracted relational triples.

To solve the problems in the above methods, we propose a prototype-optimized few-shot relational triple extraction method, which uses relational information and span encoding to optimize relational and entity prototypes, respectively, and introduces contrastive learning to optimize the distribution between prototypes.

3. Methodology

In this paper, we propose an FS-RTE method based on the hierarchical prototype optimization called HPO. The framework of our proposed model is shown in Fig. 2. HPO takes a support set and a query set as input and extracts relational triples for sentences in the query set.

HPO first uses a prompt-based learning approach to incorporate relational information to optimize the representation of relation prototypes and perform relation extraction. Then HPO constructs sentence

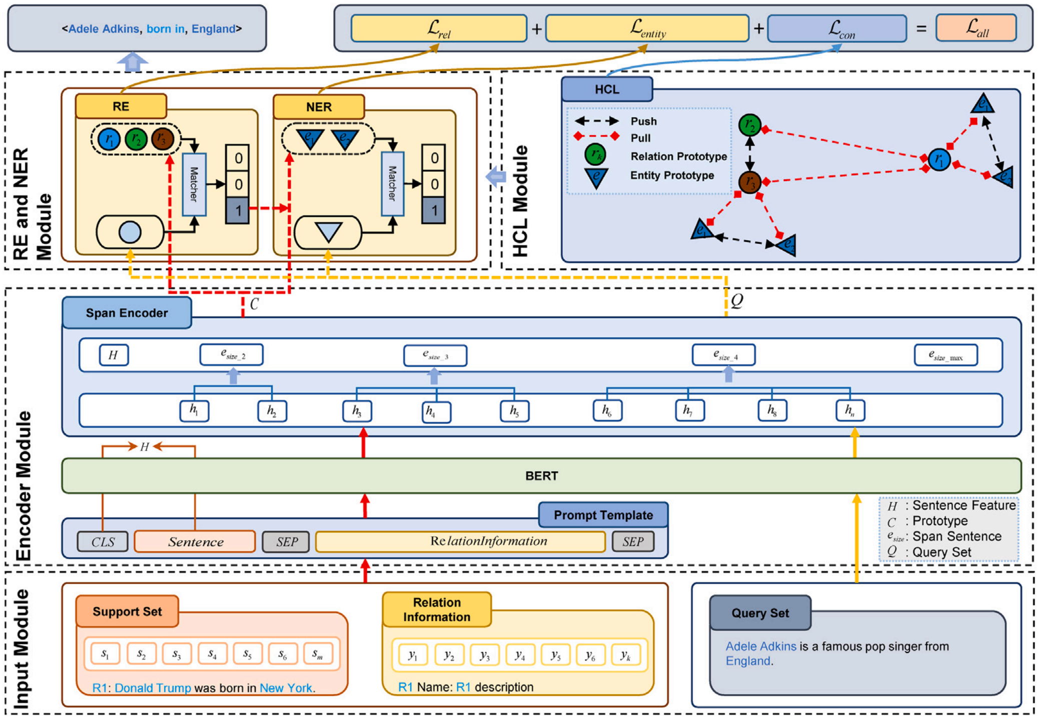


Fig. 2. The overall framework of our method.

span sequences based on token representations and constructs entity-level prototype representations. According to the relation detected in the first stage, the entity spans presented in the query set are detected using the specific entity prototypes under that relation to output the relational triples. Finally, an HCL approach is proposed to optimize the representation of prototypes better to model the connection between entity and relation prototypes. As shown in Fig. 2, given the support set sentence "*Donald Trump was born in New York*", it contains the relation "*R1*", the head entity "*Donald Trump*" and the tail entity "*New York*". According to the relation and entity information in the support set, a prototype representation is built and then the triple *<Adele Adkins, born in, England>* is identified in the query set sentence "*Adele Adkins is a famous pop singer from England.*"

3.1. Task definition

Given a dataset D, traditional methods based on supervised learning split all instances into two datasets D_{train} and D_{test}. They share the same label space. The model is trained on D_{train} and reports result on D_{test}. To be specific, given a sentence $S = \{s_1, s_2, \dots, s_i\}_{i=1}^{m}$ and a predefined relation set $R = \{r_1, r_2, \dots, r_j\}_{j=1}^{k}$, RTE model is aimed to detect relational triples $T = \{\langle h_i, r_j, t_i \rangle | h_i, t_i \in E, r_j \in R\}$ in the sentence, where E denotes the set of entities, h_i and t_i denote the head and tail entities, respectively.

In the FS-RTE task, given two datasets $\tilde{D}_{train}$ and $\tilde{D}_{test}$, it should be noted that the label spaces of these two datasets do not intersect. To better adapt to the needs of the task, most of the current FSL algorithms adopt the "episode" training strategy [39]. Therefore, in this paper, $\tilde{D}_{train}$ and $\tilde{D}_{test}$ are further divided into $\{\tilde{D}_{train}^{support}, \tilde{D}_{train}^{query}, \tilde{D}_{test}^{support}, \tilde{D}_{test}^{query}\}$. In each training episode, N triple categories are randomly selected from $\tilde{D}_{train}^{support}$, and K support instances are randomly selected from each of N triple categories. In this way, we construct the train-support set $\tilde{D}_{train}^{support} = \{(s_i, \langle h_i, r_j, t_i \rangle | s_i \in S, r_j \in R, h_i, t_i \in E)_{i=1}^{NK}\}$. Meanwhile, we randomly select G samples from the remaining samples of those N triple categories and construct the train-query set $\tilde{D}_{train}^{query} = \{(s_g, \langle h_g, r_j, t_g \rangle | s_g \in S, r_j \in R, h_g, t_g \in E)_{g=1}^{NG}\}$, where S denotes all sentences, R denotes predefined relation set and E denotes an entity set. $\tilde{D}_{test}^{support}$ and $\tilde{D}_{test}^{query}$ are constructed in the same way. We refer to such an FSL problem as the N-Way K-Shot problem, and the goal of FS-RTE is to extract relational triples from an unlabeled query set $\tilde{D}_{query}$ based on the support set $\tilde{D}_{support}$.

3.2. Encoder module

3.2.1. Token encoder

Given an input sentence $S = \{s_1, s_2, \dots, s_i\}_{i=1}^{m}$ in the support set $\tilde{D}_{train}^{support}$ and relation label information $Y = \{y_1, y_2, \dots, y_j\}_{j=1}^{k}$, some research work uses an additional BERT [40] to encode the relational information. In our case, $X = \{[CLS], s_1, \dots s_m, [SEP], y_1, \dots, y_k, [SEP]\}$ based on the predefined prompt template $Temple = \{[CLS], Sentence, [SEP], RelationInformation, [SEP]\}$. $[CLS]$ token represents the whole sentence information, and $[SEP]$ is the segmentation and ending token of the sentence. We used pre-trained BERT as the sentence encoder to capture the sequence feature embedding $A = \{a_1, a_2, \dots, a_i\}_{i=1}^{o}$ and the sentence feature embedding $H = \{h_1, h_2, \dots, h_i\}_{i=1}^{n}$ for each token. As shown in the following formulas (1) and (2):

$$a^{cls}, A = BERT(X) \tag{1}$$

$$H = A[0 : m + 1] \tag{2}$$

where $A \in R^{o \times d_h}$, $a_{cls} \in R^{1 \times d_h}$, $H \in R^{n \times d_h}$, h_i is the embedded representation of each token, o is the sequence length after embedding using WordPiece [40], d_h is the embedding dimension.

This direct encoding of relations to sentences using prompted-learned templates has the following advantages. First, we only need one encoder to encode the relation information into the sentence, greatly reducing the number of parameters. Second, BERT uses the next sentence prediction (NSP) task for pre-training, and in our task, we replace the next sentence with relational label information, so that BERT can extract information related to the class name from the input sentence, we call this method label prompt feature enhancement (LPFE). Finally, since the support set in FS-RTE has a few samples, the

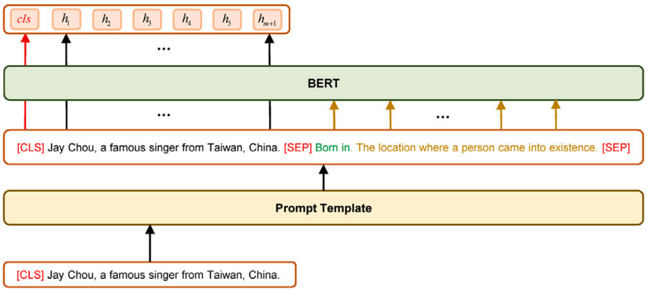


Fig. 3. The construction process of prompt templates.

prototypes generated based on these samples have semantic bias. In contrast, relational label information can be used as external knowledge to correct the prototypes, resulting in a more accurate prototype representation. Relation information can be obtained directly from the dataset, which contains the name and description of the relation. For example, for the "*born in*" relation, the description given is "*the location where a person came into existence*". The details are shown in following Fig. 3.

3.2.2. Span encoder

To better represent the complete semantic information of entities and construct entity-level prototypes instead of token-level entity prototypes, we further obtain the span encoding sequence of the sentence on top of obtaining the token-level encoding of the sentence. The token sequence is recorded into the corresponding span sequence according to the span size. Taking span size 2 as an example, the token sequence "*Donald, Trump, was, born, in, New, York*" is mapped to the span sequence "*Donald, Trump, was, born, in, New, York, Donald Trump, Trump was, …, New York*".

We use the following steps to construct a span-based sentence representation. First, given the sentence token encoding $H = \{h_1, h_2, \dots, h_i\}_{i=1}^{n}$, we can obtain different span representations $E = \{e_1, e_2, \dots, e_i\}_{i=1}^{l}$ depending on the size of the span. The embedding of the span is combined using the maximum pooling function f.

$$E = \{e_1, e_2, \dots, e_i\}_{i=1}^{l} = f[(h_1), \dots, (h_n), \dots, (h_1, h_2), \dots (h_{n-span\_size}, \dots, h_n)] \quad (3)$$

In addition, the width of the span has an impact on NER. Intuitively, a too long span is less likely to represent an entity. Therefore, we introduce span-width embedding to refine span-based sentence representations. Given a sentence span width sequence $W = \{w_1, w_2, \dots, w_i\}_{i=1}^{l}$, we randomly initialize a width embedding matrix to obtain a width embedding vector $G = \{g_1, g_2, \dots, g_i\}_{i=1}^{l}$. The width embedding matrix passes through the inverse of the model and continuously updates the propagation parameters.

$$G = \{g_1, g_2, \dots, g_i\}_{i=1}^{l} = Embedding(W = \{w_1, w_2, \dots, w_i\}_{i=1}^{l}) \quad (4)$$

After obtaining the span embedding and the span width embedding, we concat the two parts to get the final span-based sentence representation $S = \{s_1, s_2, \dots, s_i\}_{i=1}^{l}$.

$$\tilde{S} = Concat[E, G, h_{cls}] \quad (5)$$

$$S = W_s \tilde{S} + b \quad (6)$$

where $E \in R^{l \times d_h}$, $G \in R^{l \times d_g}$, $S \in R^{l \times d_h}$, l is the length of the span token sequence and d_g is the dimension of the span width embedding.

3.3. Relation extraction module

In this module, we aim to extract the relations present in the sentences. Since we only consider the case where there is only one relation in the sentence, the relation extraction task is essentially a classification task. Each relation present in the sentence is considered a category and the classifier is trained to learn from the data to classify the sentence to the correct relation type. Specifically, we use a prototype network-based approach for the FS-RE. First, we need to construct the prototype representation of the relations. We organize the sentences under all specific relations in the support set. Each relation contains K related sentences, and the entity information in the sentences is also labeled. We use the information of the sentence and combine the information of the head entity and the tail entity to construct the relation prototype because the types of entities and relations are mutually constrained. For example, given a relation "*born_in*", the corresponding head entity type is often "*Person*", and the tail entity type is "*Location*". Given the sentence span level representation s_i^k, token sentence cls representation $h_{cls}^{i,k}$, head entity $e_h^{i,k}$, and tail entity $e_t^{i,k}$ under relation k instance i in the support set, we construct the relation prototype according to the following formula:

$$e_h^{i,k}, e_t^{i,k} = f\_span(s_i^k) \quad (7)$$

$$s_{rel}^{i,k} = h_{cls}^{i,k} + e_h^{i,k} + e_t^{i,k} \quad (8)$$

where $h_{cls}^{i,k}, e_h^{i,k}, e_t^{i,k} \in {}^{1 \times d_h}$, $s_i^k \in {}^{l \times d_h}$, $s_{rel}^{i,k} \in {}^{l \times d_h}$, $s_{rel}^{i,k}$ denotes a relation representation that incorporates information about sentences, head entities, and tail entities. f_span function gets entity representation from sentences.

Inspired by Cong et al. [13], we should capture the information based on specific relations in the original sentences by introducing a support set-based attention mechanism.

$$s_{rel}^{i,k} = soft\max(\tilde{s}_{rel}^{i,k} H^k) H^k \quad (9)$$

Since a relation in the support set may contain multiple sentence instances, we integrate all the sentence and entity information in the sentence and average them to get the relation prototype c_k^r.

$$c_k^r = \frac{1}{K} \sum_{i=1}^{K} \tilde{s}_{rel}^{i,k} \quad (10)$$

where K denotes the number of sentences under each specific relation, N denotes the relation class and $c_k^r \in C^r$ denotes the set of all relation prototypes. Finally, using the same token encoder we can obtain the encoding h_i^q of the sentences in the query set, and then use a learnable similarity metric function [41] to compute the similarity between it and the relation prototype C^r.

$$\mathrm{p}_{r,i} = Similarity\left(h_i^q, C^r\right) \quad (11)$$

$$t^r = \mathrm{argmax}\left(\mathrm{p}_{r,i}\right) \quad (12)$$

where $t^r \in R$ denotes the set of all relation types, $p_{r,i}$ represents the probability that instance h_i^q is identified as specific relation. We construct a cross-entropy loss for parameter updating by following the formula.

$$\mathcal{L}_{\mathrm{rel}} = -\frac{1}{N} \sum_{i=1}^{N} y_{r,i} \log\left(p_{r,i}\right) \quad (13)$$

where $y_{r,i}$ indicates the golden label of the relations, N denotes the total number of relations.

3.4. Relation-specific entity recognition module

As shown in Fig. 2, if a relation is detected from a sentence, we identify the head and tail entities under the corresponding relation. Since we construct a span-level sentence-based representation that integrates the complete information of the entity, we can directly construct the complete prototype head and tail entity while avoiding splitting the prototype of the entity. In the process of entity prototype construction, we integrate the information of entities under a specific relation for prototype construction to avoid interfering with information generated by entities in other relations. The specific construction method is shown in the formula (14).

$$c_k^a = \frac{1}{K}\sum_{i=1}^{K} e_a^{i,k}, a \in \{h,t\} \tag{14}$$

where K denotes the number of sentences under specific relation, $e_a^{i,k}$ denotes the vector representation of entities in sentence i under relation k, and $c_k^a \in C^a$ denotes the entity prototypes under relation k, which includes the head entity prototype c_k^h and the tail entity prototype c_k^t. Then we obtain the span-based query set sentence vector representation s_i^q and apply the similarity metric function to determine the category to which each span in the sentence.

$$p_{a,i} = Similarity\left(s_i^q, C^a\right) \tag{15}$$

$$t^e = \text{argmax}\left(\text{p}_{a,i}\right), \quad a \in \{h,t\} \tag{16}$$

where $t^e \in E$ denotes the head entity, tail entity, and other types. We use the cross-entropy loss for parameter updating as the following formula (17), where a denotes the entity type, h and t are the head entity and tail entity, $y_{a,i}$ indicates the golden label of the entities, N denotes the total number of relations.

$$\mathcal{L}_{\text{entity}} = -\frac{1}{N}\sum_{i=1}^{N} y_{a,i}\log\left(\tilde{p}_i^a\right) \tag{17}$$

Since the sentence has many negative sample spans, the true spans are often very few. Therefore, we use a random span sampling method in the training process to first obtain positive sample spans, and then fill the negative sample spans until the maximum span length is set. In the test set, we enumerate all possible spans for prediction.

3.5. Hierarchical contrastive learning

Prototype-based FSL methods rely heavily on a good metric space in which different classes should be well separated from each other and identical classes should be drawn closer together. Previous approaches have not fully considered optimizing these metric spaces, so we use an HCL-based approach to optimize prototype representations of entities and relations. Specifically, we work on three levels of prototype representation optimization. (1) different types of relation prototypes should be better separated in the representation of relation prototypes. (2) In the representation of entity prototypes, the prototypes of head and tail entities should be better separated from each other. (3) Entities related to relations should have their archetypal distances close, while entities not related to relations should separate their relation and entity distances. In addition, all prototypes should be distributed in a reasonable metric space, and therefore the distance between prototypes should not be distributed too far. Based on these ideas, we applied contrastive learning to optimize the representation of prototypes at three levels: entity, relation, entity and relation.

First, to better separate entity and relation prototypes, we standardize the learning of entity and relation prototypes, with a distance metric loss for optimizing their prototype representation. In addition, we argue that all prototypes should be distributed in an appropriately sized embedding space that is neither too large nor too small (Section 4.4). Therefore, we average the maximum and minimum distances as the final prototype distance constraint. The formula for the relational prototype constraint is shown in the following formula (18).

$$\begin{aligned}\mathcal{L}_{\text{conRel}} = \frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big(&\max\left(0, m_1 - \left\|c_i^r - c_j^r\right\|\right)^2 \\ &+ \min\left(0, m_2 - \left\|c_i^r - c_j^r\right\|\right)^2\Big)\end{aligned} \tag{18}$$

where m_1 denotes the minimum distance boundary value and m_2 denotes the maximum distance boundary value, N is the number of relations and c_i^r denotes the prototype of relation i.

Then we use the same method to constrain the entity prototypes under the current relation, as shown in the following formula (19).

$$\begin{aligned}\mathcal{L}_{\text{conE}} = \frac{1}{2N^2}\sum_{k=1}^{N}\sum_{i=1}^{|E|}\sum_{j=1}^{|E|}\Big(&\max\left(0, m_1 - \left\|c_{i,k}^a - c_{j,k}^a\right\|\right)^2 \\ &+ \min\left(0, m_2 - \left\|c_{i,k}^a - c_{j,k}^a\right\|\right)^2\Big)\end{aligned} \tag{19}$$

where m_1 denotes the minimum distance boundary value and m_2 denotes the maximum distance boundary value, N is the number of relations, E denotes the type of entities, the head entity and the tail entity, $c_{i,k}$ denotes the prototype of the *ith* class of entities under relation k.

Finally, we learn the alignment between the relation and entity prototypes. We expect that relations and entities should be mutually constrained to each other. Therefore, the relation prototype and its corresponding entity prototype should be close. We model this dependency by the relation-entity comparative loss, shown in the following formula (20).

$$\mathcal{L}_{\text{conRelEntity}} = \frac{1}{N|E|}\sum_{i=1}^{N}\sum_{j\in R_i}^{|E|}\left(\left\|c_i^r - c_j^a\right\|^2\right) \tag{20}$$

where R denotes the set of relations, N is the number of relations, c_j^r and c_j^a denote the prototype of relation i and the corresponding entity prototype under relation i, respectively.

3.6. Model training process

To explain our model more clearly, we describe our algorithm flow in pseudocode in Algorithm 1. Given the support set input, we first integrate the relation information into the sentence based on the prompt template for uniform encoding. Then, we get the sentence's token vector representation and construct the sentence's span vector representation based on the token representation. Next, the relation prototype is constructed and the relations in the query set are discriminated using the similarity matching function. Once a relation is detected, an entity prototype is built based on samples under a specific relation to determine whether each span in the sentence is an entity. Finally, we define distance-based hierarchical contrastive learning to jointly optimize prototypical representations of relations and entities.

4. Experiment

In this section, we present our experiments to evaluate our approach.

4.1. Datasets

We evaluate the model using two datasets. FewRel [33] is a general public dataset in the FS-RTE field, consisting of 70,000 sentences on 100 relations from Wikipedia and annotated by crowdsourced workers. To ensure a fair comparison with previous work, we follow the setting of the data in Yu et al. [12], where 80 classes of relation data from the FewRel dataset are selected and divided into the training dataset, dev dataset, and test dataset, based on the number of different relations. Relational triples are outputted through the sentences provided in the query set. The training dataset contains data from 50 relations classes, and the dev and test datasets contain data from 15 relations classes. Note that the relations classes in these three datasets are independent.

In addition, for a more comprehensive analysis and evaluation of our model, we introduced the FewNYT dataset, which was constructed based on the NYT [42] dataset. NYT is a large-scale dataset constructed using a remotely supervised approach based on the “New York Times” corpus, widely used to evaluate supervised learning relational triple extraction tasks. We selected sentences containing only one relation and chose 50 instances for each relation, and finally constructed a dataset

Algorithm 1 The training process of HPO

Input: $\tilde{D}^{support}_{train} = \{(s_i, < h_i, r_j, t_i > | s_i \in S, r_j \in R, h_i, t_i \in E)^{NK}_{i=1}\}$; $\tilde{D}^{query}_{train} = \{(s_g, < h_g, r_j, t_g >)^{NG}_{g=1}\}$

Output: $T = \{< h_i, r_j, t_i > | h_i, t_i \in E, r_j \in R\}$

Obtain input $X = \{[CLS], s, [SEP], y, [SEP]\}$ for integration relation information;
Encode the input by Equation (1), obtain sentence embedding by Equation (2);
Obtain span width embedding $G = \{g_1, g_2, ..., g_i\}^{l}_{i=1}$ by Equation (4) and construction span-based sentence embedding $E = \{e_1, e_2, ..., e_i\}^{l}_{i=1}$ by Equation (3)(5)(6);
for episode in episodes **do**
  for $i = 1 \rightarrow NK$ **do**
    Construct relation prototype based on support set $\tilde{D}^{support}_{train}$ by Equation (7)-(10);
    Determine the relation category of the sentences in the query set $\tilde{D}^{query}_{train}$ by Equation (11)-(12);
    Calculate relation extraction loss $\mathcal{L}_{\mathrm{rel}}$ by Equation (13) and relation prototype comparative loss $\mathcal{L}_{\mathrm{conRel}}$ by Equation (18);
  end for
  for $i = 1 \rightarrow NK$ **do**
    Construct entity prototype based on the support set $\tilde{D}^{support}_{train}$ under specific relation by Equation (14);
    Identifying entities in sentences by Equation (15)-(16);
    Calculate entity recognition loss $\mathcal{L}_{entity}$ by Equation (17) and entity prototype comparative loss $\mathcal{L}_{\mathrm{conEntity}}$ by Equation (19);
    Calculate relation-entity comparative loss $\mathcal{L}_{conRelEntity}$ by Equation (20);
  end for
end for
Let $\mathcal{L}$ to be minimized in the next episode.

Table 2
Statistics of FewRel and FewNYT datasets.

Category	FewRel			FewNYT		
	Train	Dev	Test	Train	Dev	Test
Sentence	35,000	10,500	10,500	35,000	10,500	800
Entity	70,000	21,000	21,000	70,000	21,000	1600
Relation	50	15	15	50	15	16

FewNYT containing 16 relation classes and 800 instances. because of the small amount of data in the FewNYT dataset, we used the whole dataset as a test set, using the training and dev set in FewRel. The two datasets' statistical information is shown in Table 2.

We also conducted statistics on the entity span information in FewRel, as shown in Table 3. Entity spans are concentrated in the range of 1 to 3, and the number of entities with a span greater than 7 is very small. This aligns with our intuition that entities are generally not too long. Therefore, we selected span 7 as the span setting for the experiment.

4.2. Experimental settings

To make a fair comparison with the previous work, we conducted several different sets of experiments on the FewRel and FewNYT datasets under N-Way K-Shot settings, where a training dataset contains $N \times K$ support instances and one query instance. We train the model on the training dataset, use the dev dataset to determine the best hyperparameters and evaluate the model on the test dataset. We consider five types of FS-RTE tasks in our experiments: 5-Way 1-Shot, 5-Way 5-Shot, 10-Way 1-Shot, 10-Way 5-Shot, and 10-Way 10-Shot. Our approach is implemented based on PyTorch, and all experiments are conducted on a machine with an RTX3090 GPU with 24 GB of memory. Due to memory limitations, the 10-Way 10-Shot was conducted on A100, which has 40 GB of memory. We employ AdamW [43] optimizer with an initial learning rate 1e−5 and weight decay of 1e−3. We set the batch size to 1, the number of training episodes to 40 000, the number of validation episodes to 500, and the number of testing episodes to 3000. We use the pre-trained language model BERT-base as in the previous baseline model, the sentence output feature dimension is 768, and the span width embedding is encoded using a randomly initialized embedding layer with a dimension of 25. We set the maximum length of the sentence to 128, the maximum length of the span constructed by the sentence to 170, and the maximum width of each span to 7. In the HCL section, we set the distance range from 5 to 10.

We evaluate the model using Precision, Recall, and F1-score. For each test instance, only an exact match of the triples is considered correct. In addition, we evaluate the performance of entities and relations for a more fine-grained comparison.

4.3. Experimental results

On the FewRel dataset, we compared our model with the following baseline models. (1) CasRel [7], a joint entity and relations extraction model based on supervised learning methods. It proposes a novel annotation framework that first extracts the head entities and then finds the tail entities corresponding to the head entities according to each relation type. (2) MatchNet [41], a Nearest Neighbor Method with Embedded Feature Extractors for Few-Shot Classification Tasks. (3) Proto [22], a prototype network-based FS-RE method. (4) FS-GNN [44], a method for few-shot learning based on graph neural network. (5)MPE [12], an FS-RTE model that first extracts entities and then classifies relations. It uses CRF for NER, followed by a translation-based approach for RE. (6) MLMAN [45], a few-shot learning method that introduces native matching and aggregation algorithms to enhance the representation of support and query instances. (7) NNM [23], a model that uses the nearest neighbor matching method for FS-RTE, which identifies head entities and tail entities in sentences based on the semantic similarity of words. (8) TGIN [38], a model for end-to-end FS-RTE based on multi-layer heterogeneous graphs. The heterogeneous graph contains two types of nodes and three types of edges. It uses translational algebra operations to mine semantic features. (9) PTN [25], a model that uses a perspective transfer network. The heterogeneous graph first identifies the relations in the sentence, then extracts the corresponding head and tail entities according to the relations, and finally verifies the relational triples. (10) RelATE [13], a model for FS-RTE based on relational decomposition. It adopts a similar strategy to PTN, both of which are relation-then-entity recognition paradigms. Table 4 shows the F1 scores of our HPO model compared with all baseline models on the FewRel dataset. It should be noted that the F1 score combines precision and recall, so most models only give the experimental F1 score. For a more convenient comparison, we follow previous work and use the F1 score to represent the final Experimental results.

As shown by the results in Table 4, the performance of our proposed method is considerably better than that of the baseline models. CasRel shows very poor experimental results in the few-shot setting. These poor results indicate that the models constructed by the traditional supervised learning-based paradigm cannot solve the RTE task in the few-shot scenario. MPE performs NER first and then RE. They use traditional entity extractors to identify entities in sentences and construct relation prototypes based on the acquired entities and external knowledge to achieve relational triple extraction. Due to traditional entity extraction methods, it is difficult for them to obtain ideal entity recognition results under a small number of training samples, which further affects the performance of relational triplets. In addition, MatchNet, Proto, FS-GNN, MLMAN, and TGIN are also based on the entity-then-relation paradigm. Although their relation extraction performance is high, the entity recognition effect is poor, which leads to

Table 3
Statistics of entity span information in the FewRel dataset. SW refers to the width of the span, [1, 3] denotes the span interval $1 \leq SW \leq 3$, (3, 5] denotes the span interval $3 < SW \leq 5$. It should be noted that the number of entities counted includes repeated occurrences.

Model	Train				Dev				Test			
	[1, 3]	(3, 5]	(5, 7]	SW > 7	[1, 3]	(3, 5]	(5, 7]	SW > 7	[1, 3]	(3, 5]	(5, 7]	SW > 7
Entity num	63,844	5302	751	103	19,534	1233	193	40	19,730	1151	107	12
Entity ratio	91.2%	7.57%	1.07%	0.15%	93.02%	5.87%	0.92%	0.19%	93.95%	5.48%	0.51%	0.06%

Table 4
F1 Results of the compared models on FewRel. Note that the best and second-best results are marked in **bold** and underlined, respectively.

Model	5W1S	5W5S	10W1S	10W5S	10W10S
CasRel [7]	–	2.1	–	–	2.0
MatchNet [41]	15.4	18.7	8.2	–	16.3
Proto [22]	15.9	21.2	10.4	–	15.4
FS-GNN [44]	17.8	24.5	11.4	–	16.1
MPE [12]	–	23.3	–	–	12.1
MLMAN [45]	20.4	28.5	15.3	–	19.2
NNM [23]	–	32.1	–	–	25.0
TGIN [38]	24.0	32.3	17.3	–	22.8
PTN [25]	30.0	40.0	25.3	33.6	36.2
RelATE [13]	28.7	42.3	20.3	34.8	40.9
HPO (Ours)	**41.1**	**48.3**	**34.4**	**41.5**	**42.8**
Improvement	(+11.1%)	(+6.0%)	(+9.1%)	(+6.7%)	(+1.9%)

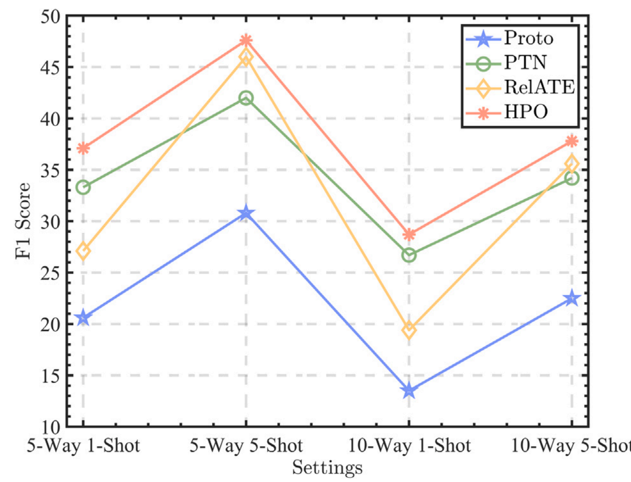


Fig. 4. F1 results of the compared models on the FewNYT dataset.

the unsatisfactory performance of the relational triplet extraction. More detailed comparison results are shown in Table 5.

PTN and RelATE are the most recent and effective baseline models. Therefore, we focus on the comparisons with them. Our HPO improved over PTN by 8.3%, 7.9%, and 6.6% on 5-Way 5-Shot, 10-Way 5-Shot and 10-Way 10-Shot, respectively. On 5-Way 1-Shot and 10-Way 1-Shot, HPO improves 11.1% and 9.1% over PTN, respectively. Compared with RelATE, HPO improves 6.0%, 6.7% and 5% on 5-Way 5-Shot, 10-Way 5-Shot and 10Way10Shot, respectively. On 5-Way 1-Shot and 10-Way 1-Shot, HPO improves 12.4% and 14.1% over RelATE, respectively. This fully illustrates that our HPO can be better applied to FS-RTE. PTN uses a triple perspective transfer method for relation extraction. It sequentially matches each instance in the query set with the support set sentence to calculate the probability and then builds a token-level entity prototype for entity recognition. RelATE recognizes the relations present in sentences based on relation prototypes. However, they all simply construct relational and entity prototypes based on the support set and do not consider the guiding role of existing relational information on prototype construction, especially in the case of a few support-set samples, so their experimental performance is worse on the 1-Shot setting.

Based on the above analysis, we speculate that the performance gain of our HPO model comes from three main aspects. (1) To compensate for the prototype bias problem caused by insufficient support set samples, we use a prompt-based learning method to introduce external relational information further to optimize the prototypes' representation. (2) We build entity prototypes based on spans instead of tokens. On the one hand, we can better and more comprehensively represent the entity prototypes. On the other hand, we do not need to divide the entity prototypes, reducing the classification difficulty. The experimental results in Table 5 further confirm our point of view. (3) We propose contrastive learning at three levels, relations, entities, entity and relation, to better optimize the representation of prototypes, which improves the recognition accuracy of each subtask and ultimately improves the effectiveness of relation extraction.

To further evaluate the model, we selected some of the latest baselines to conduct experiments on the FewNYT dataset, and the specific F1 results are shown in Fig. 4. The difference from FewRel is that the relations in FewNYT do not have labels and description information. Therefore, we artificially construct the information of the relations among them. Our HPO still achieves the best experimental results, and the results of PTN and RelATE are also consistent with the performance of FewRel. In the 1-Shot setting, PTN is better than RelATE, while under the 5-Shot setting, RelATE outperforms PTN.

To further analyze the performance of the models on each subtask, we evaluate the performance of NER, RE, and RTE on the FewRel dataset under 5-Way 1-Shot, 5-Way 5-Shot, 10-Way 1-Shot, and 10-Way 10-Shot settings. As shown in Table 5, the performance of RE is significantly better than NER, which indicates that NER is more difficult than RE in few-shot setting. Among all the baseline models, MPE achieves the best RE performance. It adopts the extraction framework of entity-then-relation to use entity information to construct relational prototypes. However, since MPE uses a traditional entity extractor for NER, the accuracy of entity recognition is very low, which leads to unsatisfactory results in the RTE. PTN transforms RE into a binary classification problem, reducing the complexity of the RE task. As a result, PTN demonstrates superior performance in RE compared to RelATE and our HPO. However, PTN requires a loop to traverse each sentence for RE, which greatly increases the time complexity of the model. Our HPO is significantly better than RelATE and PTN in NER, which is a good indication that entity prototypes constructed based on span can achieve more effective entity extraction. Compared with RelATE, HPO has also achieved an overwhelming advantage in RE, especially in the 1-Shot task, which shows that introducing relation information can better construct prototype representations. With the increase of support set samples, the model improvement brought by relational information decreases, which also shows that the prototypes built with a small number of samples have great differences and cannot represent the representation of the overall sample well. Finally, HPO improves the performance of both subtasks by learning to model the representations of entity prototypes and relation prototypes by HCL, thereby improving the performance of the RTE.

To further analyze the working efficiency of the model, we experimentally analyzed the training and inference time of the model in Table 6. All experiments are conducted on a machine with an

Table 5
F1 scores of entities, relations, and triples on FewRel. Note that the best and second-best results are marked in **bold** and underlined, respectively.

Model	5-Way 1-Shot			5-Way 5-Shot			10-Way 1-Shot			10-Way 10-Shot		
	Relation	Entity	Triple	Relation	Entity	Triple	Relation	Entity	Triple	Relation	Entity	Triple
MatchNet [41]	75.8	18.7	15.4	84.6	20.7	18.7	59.4	12.5	8.2	77.8	20.0	16.3
Proto [22]	77.6	19.4	15.9	87.4	25.1	21.2	65.7	14.5	10.4	76.0	19.8	15.4
FS-GNN [44]	78.4	21.6	17.8	88.4	26.0	24.5	66.9	15.7	11.4	77.7	20.5	16.1
MPE [12]	–	–	–	**93.8**	25.0	23.3	–	–	–	**84.6**	14.9	12.1
MLMAN [45]	82.5	23.4	20.4	91.8	30.5	28.5	70.7	20.4	15.4	81.9	23.3	19.2
NNM [23]	–	–	–	88.7	32.6	32.2	–	–	–	75.1	26.6	25.0
TGIN [38]	**83.7**	27.5	24.0	93.1	33.6	32.3	**72.3**	22.8	17.3	83.7	26.6	22.8
PTN [25]	81.1	47.2	30.0	84.2	56.9	40.0	68.7	39.4	25.3	77.6	52.4	36.2
RelATE [13]	66.3	46.5	28.7	79.9	59.6	42.3	54.7	37.9	20.3	75.4	57.0	40.9
HPO (Ours)	80.2	**57.4**	**41.1**	84.1	**65.0**	**48.3**	68.0	**51.4**	**34.4**	76.8	**60.0**	**42.8**

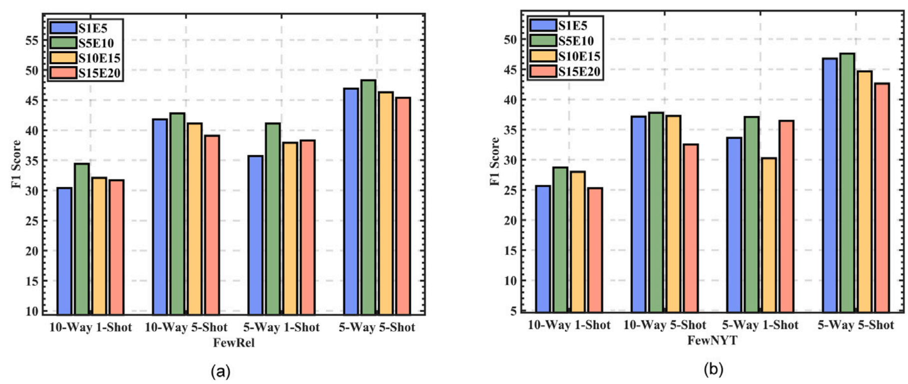


Fig. 5. Prototype distance experiments on the FewRel and FewNYT datasets.

Table 6
HPO training and inference time on FesRel dataset.

Setting	Training		Inferencing	
	All time (m)	Epoch time (ms)	All time (m)	Epoch time (ms)
5-Way 1-Shot	395	592	346	115
5-Way 5-Shot	466	699	421	140
10-Way 1-Shot	412	618	357	119
10-Way 5-Shot	578	867	553	184

RTX3090 GPU which has 24 GB of memory. We set the batch size to 1, the number of training episodes to 40 000, the number of validation episodes to 500, and the number of testing episodes to 3000.

4.4. Parameters sensitivity analysis

We study and analyze the hyperparameters of the model on the FewRel and FewNYT datasets. Fig. 5 presents the experimental results of our HPO under different prototype distance settings, where Fig.(a) represents the FewRel dataset and Fig.(b) represents the FewNYT dataset. We set four different distance intervals, namely 1–5, 5–10, 10–15, and 15–20. Then use S1E5, S5E10, S10E15, and S15E20 to represent them. We can observe that (1) the model has poor results with S1E5 and S15E20 settings, and the distance of the prototype is not suitable, either too close or too far. (2) The best results are under the S5E10 setting. This result validates our argument that the prototypes should be distributed in an appropriately sized embedding space that is neither too large nor too small.

4.5. Ablation studies

To analyze the impact of each different component in our model, we perform ablation studies on the FewRel dataset with 5-Way 1-Shot and 10-Way 1-Shot settings. Note that the setting is the same in each experiment except for the variables studied. The experimental results are shown in Table 7, where the removal of each component of the model results in degraded performance for relations, entity, and relational triple extraction.

Table 7
Ablation studies on FewRel. We report the F1 score under two settings. w/o RI refer to no relational information being introduced. w/o SPAN refer to instead of using the span-based approach to construct entity prototypes. w/o HCL refer to the removed hierarchical contrastive learning.

Category	FewRel			FewNYT		
	Relation	Entity	Triple	Relation	Entity	Triple
HPO	80.2	57.4	41.1	68.0	51.4	34.4
-w/o RI	70.5	54.7	35.4	58.4	49.6	27.8
-w/o SPAN	76.2	49.1	33.5	67.4	44.7	29.6
-w/o HCL	78.1	55.8	37.8	65.7	47.8	31.5

In Table 7, w/o RI means that no relational information is introduced. Similarly, w/o SPAN means that instead of using the span-based approach to construct entity prototypes, an entity prototype contains an entity start prototype and an entity end prototype, which are constructed based on the token. The w/o HCL indicates that the contrastive learning module is removed and the prototype representation of entities and relations is constructed directly based on the vector representation in the sentence. In both few-shot settings, w/o RI leads to a dramatic decrease in relational performance. We find that the reason may be mainly because the relation prototype constructed under 1-Shot carries a strong sample paranoia while introducing relation information can construct a better relation prototype. In addition, prompt-based learning for encoding relational information and HCL interacts with entities and relations, such that the reduced relation extraction performance also somewhat affects entity extraction. The main impact of w/o SPAN is the performance of entities. Compared with token level-based prototypes, span-based entity-level prototypes incorporate the complete semantic and span information of entities, which can better represent entities and thus improve the performance of entity recognition. w/o

Table 8
Ablation studies on FewRel about relation prompt template. We report the F1 score under four settings. w/o Relation Dec refer to use the relation name information. w/o Relation Name means only use the relation description information.

Model	5W1S	5W5S	10W1S	10W5S
HPO	41.1	48.3	34.4	42.3
-w/o Relation Dec	31.8	46.1	31.5	38.2
-w/o Relation Name	38.1	46.4	33.6	38.7

HCL leads to a decrease in the performance of both subtasks, indicating that HCL can effectively constrain the representation of entity and relation prototypes. The comparison of relation prototypes can constrain the distance between different relations, and the comparison of entity prototypes can constrain the distance between head entities and tail entities. The comparison between the relations and the entity prototype can constrain the distance between the relation and the entity to keep them within an appropriate range.

To further analyze the usefulness of the relation prompt template, we conducted the following experiments, as shown in Table 8. We set two baselines, w/o Relation Dec and w/o Relation Name, which indicate the addition of relation name and description to the prompt template, respectively. Compared with HPO, the performance of the two baselines has decreased, and the performance degradation under the w/o Relation Dec setting is the more obvious, which shows that it is difficult to fully and specifically describe the content of the relation using only the relation name because the relation name is often relatively concise. However, adding the relations name to the relation description can better describe and condense the relations. Therefore, we finally chose the template with the combination of the relation name and relation description.

To specifically analyze the effect of relation templates on HPO, we randomly construct different proportions of error sample relation templates for experiments on FewRel. The specific method uses a fixed ratio to randomly replace the correct relations in the sentence with the relation of other sentences, which was set to 20%, 40%, 60%, and 80%, respectively. We conducted experiments under the settings of 5-Way 1-Shot, 5-Way 5-Shot, 10-Way 1-Shot, 10-Way 5-Shot and 10-Way 10-Shot. Experimental results are shown in Fig. 6. As the proportion of incorrect relation information increases, the experimental results of the model gradually decrease, which shows that the introduction of complete and correct relation information can better help the model to learn.

4.6. Prototype visualization

To visualize the distribution of the prototype representation of our proposed method, we selected the 5-Way 1-Shot and 5-Way 5-Shot settings and used t-SNE [46] to visualize the prototypes of the relation categories of our HPO and the best baseline RelATE on FewRel and FewNYT. We selected relation categories from the test dataset and generated corresponding prototype embeddings of sentence relations based on the parameters determined by the model on the training set, with 400 instances selected for each relation category. Since there are only 50 instances of each relation in FewNYT, it contains duplicate instances. As shown in Fig. 7, we can observe that: (1) From Fig.(a) and Fig.(b) we can see that the 5-Way 5-Shot of RelATE has a great improvement over the 5-Way 1-Shot and the increase in sample size can greatly alleviate the problem of prototype bias. (2) In the 5-Way 1-Shot setting, the relation prototypes in Fig.(c) are far superior to Fig.(a). Our HPO introduces relational information and the contrastive learning mechanism, which makes the prototypes of each category keep a suitable distance from each other and can effectively alleviate the prototype bias problem arising from too few samples. In the 5-Way 5-Shot setting, Fig.(d) also outperforms the prototype distribution of Fig.(b). (3) The prototype distribution of Fig.(d) is more clustered than

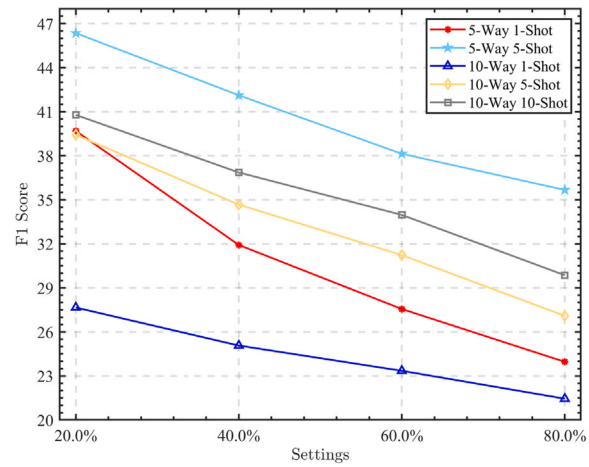


Fig. 6. Experimental results of relation templates with different error ratios on the FewRel dataset.

that of Fig.(c), further illustrating that the increase in sample size can improve the construction of prototypes. (4) The relations prototypes on FewNYT are more closely distributed, mainly because there are many similar relations in FewNYT, such as the relations "*/business/location*" and "*/business/company/place_founded*".

4.7. Case study

To more intuitively demonstrate the advantages of our HPO in the FS-RTE, we use three cases to compare the gap between HPO and the current strongest baseline model RelATE in the 5-Way 1-Shot setting. The results are shown in Fig. 8. In the first case, due to too few sample instances, RelATE misidentifies the relation "*work location*" in the sentence as "*record label*", which directly leads to errors in the subsequent head entity and tail entity recognition. Unlike this, our HPO introduces relational information based on prompt templates, which effectively enhances the representation of sentences under specific relations. Therefore, relations in sentences can be correctly identified, and head entities and tail entities under corresponding relations can be extracted correctly. In the second case, RelATE correctly identifies the relation in the sentence, but does not fully identify the tail entity "*film director*". We speculate that the decrease in performance could be attributed to the similarity in semantics between the terms "*film director*" and "*director*". RelATE divides the entity prototype and builds the start and end prototypes for the entity based on the token. This segmentation method does not consider the overall semantics and span information of the entity, which is prone to span errors in the entity recognition process, resulting in wrong triplet extraction at the end. In the third case, RelATE also does not accurately identify the span of the head entity and wrongly identifies the tail entity. Based on sentence token encoding, our HPO further constructs sentence span encoding, and constructs entity-level prototypes instead of token-level, which can effectively use the complete semantics and span information of entities to improve the accuracy of entity recognition. In addition, HPO introduces HCL to optimize the prototype representations of relations and entities, so relational triples in sentences can be accurately identified.

We use a case study to analyze the ability of HPO to extract relation triplets. It aims to evaluate where and why errors occur in the extracted triples. We use the experimental results of the 5-Way 5-Shot in Fig. 9. In instance 1, HPO incorrectly identified the entity "*Tales from the Public Domain*", one of the most common errors. From Table 5, we can see that entity recognition is much more difficult than relation classification. HPO uses span encoding and hierarchical contrastive learning to optimize entity prototypes, but the recognition accuracy of

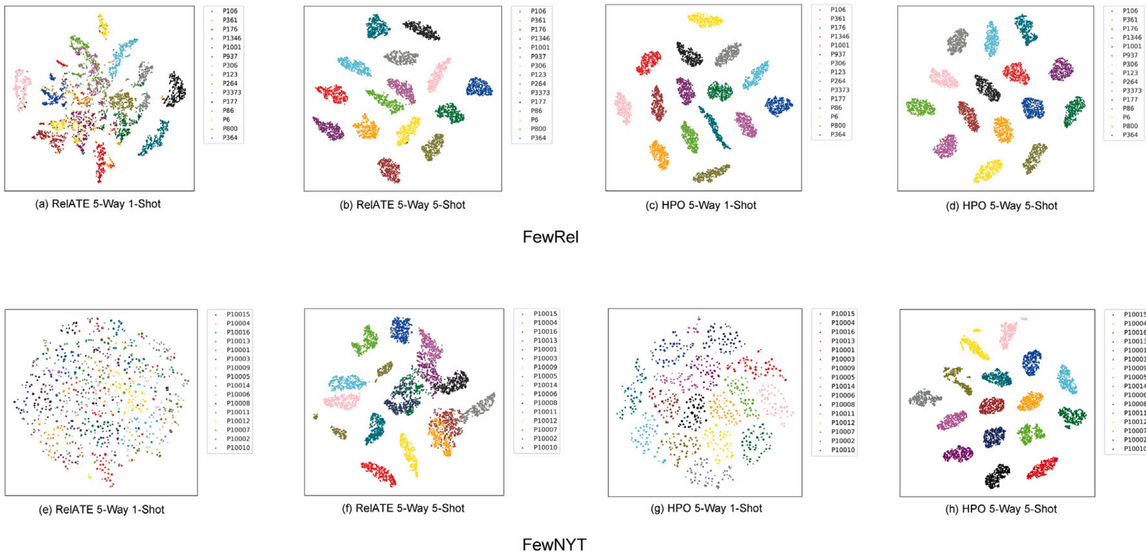


Fig. 7. The prototype embedding visualization of relations in the 5-Way 1-Shot and 5-Way 5-Shot settings on FewRel and FewNYT datasets using t-SNE. Where (a) and (b) denote RelATE on FewRel, (c) and (d) denote HPO on FewRel, (e) and (f) denote RelATE on FewNYT, (g) and (h) denote HPO on FewNYT. Each color denotes a class of relations.

Instance 1: He has studied at the Sibelius Academy University of Southern California and the tel aviv university with hagai Shaham.		
RelATE	<university, record label, aviv>	*[Relation error]][Entity recognition error]*
HPO	<hagai shaham, work location, tel aviv university>	
Instance 2: Dusan trancik (born 26 november 1946) is a slovak film director and screenwriter.		
RelATE	<Dusan trancik, occupation, director>	*[Span error]*
HPO	<Dusan trancik, occupation, film director>	
Instance 3: The album features vocal appearances from solomon grey, patrick baker, and ghostly international is matthew dear, among others.		
RelATE	<dear, record label, matthew>	*[Span error][Entity recognition error]*
HPO	<matthew dear, record label, ghostly international>	

Fig. 8. Instances of different models on the FewRel dataset under 5-Way 5-Shot. Green indicates the head entity in the sentence and blue indicates the tail entity in the sentence. Entities with errors in triplets are identified in red.

Instance 1: Following the release of The Simpsons thirteenth season, Tales from the Public Domain received mixed reviews from critics.				
True	<Tales from the Public Domain , part of, The Simpsons thirteenth season>	**HPO**	<Tales, part of, The Simpsons thirteenth season>	*[Entity recognition error]*
Instance 1: Quest publishes original articles written in either English or French each with a summary in the other language.				
True	<Quest, original language of film or TV show, English>	**HPO**	<Quest, original language of film or TV show, French>	*[Multi-triplet problem]*
Instance 2: Use byobu for extended features in your terminal window ghacks.net Byobu 3.0 reworked the build system to use automake and allow for porting to other Unix - like operating systems.				
True	<automake, operating system, Unix-like>	**HPO**	<automake, operating system, Unix>	*[Data annotation error]*
Instance 3: Penzance was the birthplace of Maria Branwell mother of three famous novelists – Charlotte Bronte, Emily Bronte and Anne Bronte.				
True	<Anne Bronte, sibling, Emily Bronte>	**HPO**	<Anne Bronte, sibling, Maria Branwell>	*[Reasoning error]*

Fig. 9. Common scenarios for errors of relational triple on FewRel dataset under 5-Way 5-Shot. Green indicates the head entity in the sentence and blue indicates the tail entity in the sentence. Entities with errors in triplets are identified in red.

more complex entities still needs to be improved. In instance 2, HPO identified the triple <*Quest, original language of film or TV show, French*>, which is correct. Still, it is judged as wrong because the annotation in the sentence is <*Quest, original language of film or TV show, English*>. This shows multiple correct triples in the sentence in the dataset, but because FewRel is a dataset extracted from single triples, no further annotation is performed. In instance 3, the recognition result of HPO is correct, but due to the errors in the annotations in the dataset, the recognition result is judged to be wrong. In instance 4, the triples in the sentence are relatively complex and require reasoning to get the correct result. HPO only recognized the head entity and the relationship and incorrectly identified the tail entity. This shows that HPO still needs to gain the recognition of triples that require reasoning.

5. Conclusion

In this paper, we propose a hierarchical prototype optimization method HPO for few-shot relational triplet extraction. HPO models the task in three stages and optimizes the prototype through three levels: entities, relations, and interactions between entities and relations. First it uses prompt learning to bring relational label information into the text. Then the text which is encoded across the text and entity-level prototypes are constructed, avoiding label dependencies between entities. Finally, a hierarchical contrastive learning approach is introduced to refine the metric space between entity, relation, and entity-relations prototypes. Experiments on public datasets show that our model outperforms state-of-the-art methods in different settings. Few-shot relational triplet extraction can effectively alleviate the problem of data deficiency for labeling, so it can be generalized better and be practicable in specific fields. However, HPO still has some limitations. For example, HPO has particular difficulties in processing multiple triples in a sentence, faces problems in computational performance and efficiency problems when processing entities with longer spans, and depends on external knowledge. In the future, we will explore few-shot triple extraction tasks in more complex scenarios, such as multi-triplet extraction and more complex interactions between entities and relations. It requires the model to have more robust feature extraction and learning capabilities, to sport more complex sentences, and to extract triples that meet the requirements from the sentences. While improving model performance, we also need to consider model efficiency and explore more flexible ways to utilize external knowledge.

CRediT authorship contribution statement

Chen Gao: Writing – original draft, Validation, Software, Methodology. **Xuan Zhang:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Zhi Jin:** Writing – review & editing, Conceptualization. **Weiyi Shang:** Writing – review & editing. **Yubin Ma:** Visualization. **Linyu Li:** Visualization. **Zishuo Ding:** Data curation. **Yuqin Liang:** Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The National Natural Science Foundation of China under Grant No. 62192731; Science Foundation of Young and Middle-aged Academic and Technical Leaders of Yunnan, China under Grant No. 202205AC 160040; Science Foundation of Yunnan Jinzhi Expert Workstation, China under Grant No. 202205AF150006; Major Project of Yunnan Natural Science Foundation, China under Grant No. 202302AE09002003; Knowledge-driven Smart Energy Science and Technology Innovation Team of Yunnan Provincial Department of Education, China; Open Foundation of Yunnan Key Laboratory of Software Engineering, China under Grant No. 2023SE101; Yunnan University Graduate Research Innovation Foundation Project, China under Grant No. KC-23233953;

References

[1] S. Sarawagi, et al., Information extraction, Found. Trends® Databases 1 (3) (2008) 261–377.
[2] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Trans. Knowl. Data Eng. 34 (1) (2020) 50–70.
[3] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Comput. Surv. 54 (4) (2021) 1–37.
[4] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.
[5] S.K. Dwivedi, V. Singh, Research and reviews in question answering system, Proc. Technol. 10 (2013) 417–424.
[6] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. 34 (8) (2020) 3549–3568.
[7] Z. Wei, J. Su, Y. Wang, Y. Tian, Y. Chang, A novel cascade binary tagging framework for relational triple extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1476–1488.
[8] D. Sui, X. Zeng, Y. Chen, K. Liu, J. Zhao, Joint entity and relation extraction with set prediction networks, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–12.
[9] J. Lu, P. Gong, J. Ye, J. Zhang, C. Zhang, A survey on machine learning from few samples, Pattern Recognit. 139 (2023) 109480.
[10] F. Liu, S. Yang, D. Chen, H. Huang, J. Zhou, Few-shot classification guided by generalization error bound, Pattern Recognit. 145 (2024) 109904.
[11] Y. An, H. Xue, X. Zhao, J. Wang, From instance to metric calibration: A unified framework for open-world few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[12] H. Yu, N. Zhang, S. Deng, H. Ye, W. Zhang, H. Chen, Bridging text and knowledge with multi-prototype embedding for few-shot relational triple extraction, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 6399–6410.
[13] X. Cong, J. Sheng, S. Cui, B. Yu, T. Liu, B. Wang, Relation-guided few-shot relational triple extraction, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2206–2213.
[14] J. Huang, C. Li, K. Subudhi, D. Jose, S. Balakrishnan, W. Chen, B. Peng, J. Gao, J. Han, Few-shot named entity recognition: An empirical baseline study, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 10408–10423.
[15] J. Ma, M. Ballesteros, S. Doss, R. Anubhai, S. Mallya, Y. Al-Onaizan, D. Roth, Label semantics for few shot named entity recognition, in: Findings of the Association for Computational Linguistics, 2022, pp. 1956–1971.
[16] K. He, R. Mao, Y. Huang, T. Gong, C. Li, E. Cambria, Template-free prompting for few-shot named entity recognition via semantic-enhanced contrastive learning, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[17] T. Gao, X. Han, H. Zhu, Z. Liu, P. Li, M. Sun, J. Zhou, FewRel 2.0: Towards more challenging few-shot relation classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 6250–6255.
[18] S. Wadhwa, S. Amir, B.C. Wallace, Revisiting relation extraction in the era of large language models, in: Proceedings of the Conference. Association for Computational Linguistics. Meeting, Vol. 2023, NIH Public Access, 2023, p. 15566.
[19] Z. Nasar, S.W. Jaffry, M.K. Malik, Named entity recognition and relation extraction: State-of-the-art, ACM Comput. Surv. 54 (1) (2021) 1–39.
[20] H. E, W. Zhang, S. Xiao, R. Cheng, Y. Hu, Y. Zhou, P. Niu, Survey of entity relationship extraction based on deep learning, J. Softw. 30 (6) (2019) 1793–1818.
[21] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications, 2004, pp. 107–110.
[22] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4080–4090.
[23] X. He, H. Song, D. Cheng, B. Xu, Few-shot relational triple extraction with nearest neighbor matching, in: International Conference on Computer Graphics, Artificial Intelligence, and Data Processing, 2022, pp. 262–266.
[24] Y. Yang, A. Katiyar, Simple and effective few-shot named entity recognition with structured nearest neighbor learning, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 6365–6375.
[25] J. Fei, W. Zeng, X. Zhao, X. Li, W. Xiao, Few-shot relational triple extraction with perspective transfer network, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 488–498.
[26] Z. Zhang, B. Yu, X. Shu, X. Mengge, T. Liu, L. Guo, From what to why: Improving relation extraction with rationale graph, in: Findings of the Association for Computational Linguistics:The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 86–95.
[27] P. Qin, W. Xu, W.Y. Wang, Robust distant supervision relation extraction via deep reinforcement learning, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 2137–2147.
[28] Z. Zhong, D. Chen, A frustratingly easy approach for entity and relation extraction, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 50–61.
[29] T. Wu, H. Ma, C. Wang, S. Qiao, L. Zhang, S. Yu, Heterogeneous representation learning and matching for few-shot relation prediction, Pattern Recognit. 131 (2022) 108830.

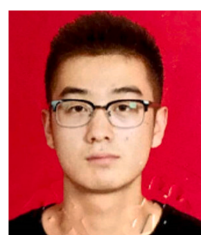
[30] A. Fritzler, V. Logacheva, M. Kretov, Few-shot classification in named entity recognition task, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 993–1000.
[31] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (9) (2023) 1–35.
[32] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using BART, in: Findings of the Association for Computational Linguistics: Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 1835–1845.
[33] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, M. Sun, FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4803–4809.
[34] T. Gao, X. Han, Z. Liu, M. Sun, Hybrid attention-based prototypical networks for noisy few-shot relation classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 6407–6414.
[35] L. Zhenzhen, Y. Zhang, J.-Y. Nie, D. Li, Improving few-shot relation classification by prototypical representation learning with definition text, in: Findings of the Association for Computational Linguistics: NAACL 2022, 2022, pp. 454–464.
[36] F. Zhao, Y. Shen, Z. Wu, X. Dai, Label-driven denoising framework for multi-label few-shot aspect category detection, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 2390–2402.
[37] H. Liu, F. Zhang, X. Zhang, S. Zhao, J. Sun, H. Yu, X. Zhang, Label-enhanced prototypical network with contrastive learning for multi-label few-shot aspect category detection, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1079–1087.
[38] J. Wang, L. Zhang, J. Liu, K. Ma, W. Wu, X. Zhao, Y. Wu, Y. Huang, TGIN: Translation-based graph inference network for few-shot relational triplet extraction, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[39] Y. Yu, Z. Ji, J. Han, Z. Zhang, Episode-based prototype generating network for zero-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14035–14044.
[40] J.D.M.-W.C. Kenton, L.K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.
[41] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016).
[42] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Machine Learning and Knowledge Discovery in Databases: European Conference, Springer, 2010, pp. 148–163.
[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[44] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, in: Proceedings of the 6th International Conference on Learning Representations, 2018, pp. 1–12.
[45] Y. Zhixiu, L. Zhenhua, Multi-level matching and aggregation network for few-shot relation classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2872–2881.
[46] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

Chen Gao received the B.S. and M.S. degrees in School of Software from East China University of Science and Yunnan University, China, in 2017 and 2021, respectively. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering,Yunnan University. His research interests include knowledge graphs, information extraction, and few-shot learning.

Xuan Zhang received the B.S. and M.S. degrees in computer science, and the Ph.D. degree in system analysis and integration from Yunnan University, Kunming, China. She is a professor with the School of Software, Yunnan University, Kunming, China. She is author of 4 books and more than 100 articles. She has been principal investigator for more than 40 national, provincial, and private grants and contracts. She is the core scientist of Yunnan Key Laboratory of Software Engineering and Yunnan Software Engineering Academic Team. Her research interests include blockchain, knowledge graph, natural language processing, recommendation system, and computer vision.

Zhi Jin received the Ph.D. degree in computer science from the Changsha Institute of Technology, China, in 1992. She is a Full Professor of Computer Science with Peking University, where she is the Deputy Director of the Key Laboratory of High Confidence Software Technologies (Ministry of Education). She is/was the Principal Investigator of more than 15 national competitive grants. Her research interests include software engineering, requirements engineering, knowledge engineering, and machine learning. She is a Standing Board Member of China Computer Federation (CCF) and the Chair of CCF Technical Committee of System Software. She was elected as a fellow of CCF in 2012.

Weiyi Shang is an Associate Professor at the University of Waterloo. His research interests include AIOps, big data software engineering, software log analytics and software performance engineering. He serves as a Steering committee member of the SPEC Research Group. He is ranked top worldwide SE research stars in a recent bibliometrics assessment of software engineering scholars. He is a recipient of various premium awards, including the SIGSOFT Distinguished paper award at ICSE 2013 and ICSE 2020, best paper award at WCRE 2011 and the Distinguished reviewer award for the Empirical Software Engineering journal. His research has been adopted by industrial collaborators (e.g., BlackBerry and Ericsson) to improve the quality and performance of their software systems that are used by millions of users worldwide. Contact him at wshang@uwaterloo.ca, https://ece.uwaterloo.ca/∼wshang/.



Yubin Ma is currently pursing for a master's degree in software engineering at Yunnan University. His current research interests include recommender systems, graph neural network and knowledge graphs.

Linyu Li is currently studying for a master's degree in software engineering at the National Model Software College of Yunnan University, and is expected to receive a master's degree in software engineering in June 2024. His current research interests include: knowledge graph completion, knowledge graph representation learning, knowledge graph embedding, knowledge graph reasoning.

Zishuo Ding is an Assistant Professor at the Hong Kong University of Science and Technology (Guangzhou). His research primarily revolves around the intersection of natural language processing and software engineering, with a focus on areas such as software performance engineering, software log analytics, and code representation learning. His work has been published in flagship conferences and journals including ICSE, FSE, ASE, TOSEM, and EMSE, and recognized with the SIGSOFT Distinguished Paper Award at ICSE 2020. More information at https://ece.uwaterloo.ca/∼z8ding/.

Yuqin Liang is currently studying for a master's degree in software engineering at the National Model Software College of Yunnan University, and is expected to receive a master's degree in software engineering in June 2025. Her current research interests include: natural language processing, information extraction and event extraction.