Bridging Text and Knowledge with Multi-Prototype Embedding for Few-Shot Relational Triple Extraction

Haiyang Yu[1,2]*, **Ningyu Zhang**[1,2*†], **Shumin Deng**[1,2], **Hongbin Ye**[1,2],
Wei Zhang[3], **Huajun Chen**[1,2 †]
[1] Zhejiang University
[2] AZFT Joint Lab for Knowledge Engine
[3] Alibaba Group
{yuhaiyang,zhangningyu,231sm,yehb,huajunsir}@zju.edu.cn
lantu.zw@alibaba-inc.com

Abstract

Current supervised relational triple extraction approaches require huge amounts of labeled data and thus suffer from poor performance in few-shot settings. However, people can grasp new knowledge by learning a few instances. To this end, we take the first step to study the few-shot relational triple extraction, which has not been well understood. Unlike previous single-task few-shot problems, relational triple extraction is more challenging as the entities and relations have implicit correlations. In this paper, We propose a novel multi-prototype embedding network model to jointly extract the composition of relational triples, namely, entity pairs and corresponding relations. To be specific, we design a hybrid prototypical learning mechanism that bridges text and knowledge concerning both entities and relations. Thus, implicit correlations between entities and relations are injected. Additionally, we propose a prototype-aware regularization to learn more representative prototypes. Experimental results demonstrate that the proposed method can improve the performance of the few-shot triple extraction.

1 Introduction

Relational Triple Extraction is an essential task in Information Extraction for Natural Language Processing (NLP) and Knowledge Graph (KG) (Yu et al., 2017; Huang et al., 2020), which is aimed at detecting a pair of entities along with their relation from unstructured text. For instance, there is a sentence "*Paris is known as the romantic capital of France.*", and in this example, an ideal relational triple extraction system should extract the relational triple $\langle$*Paris, Capital_of, France*$\rangle$, in which *Capital_of* is the relation of *Paris* and *France*.

Current works in relational triple extraction typically employ traditional supervised learning based on feature engineering (Kambhatla, 2004; Reichartz et al., 2010) and neural networks (Zeng et al., 2014; Bekoulis et al., 2018a). The main problem with supervised learning models is that they can not perform well on unseen entity types or relation categories (e.g., train a model to extract knowledge triples from the economic text, then run this model to work on scientific articles). As a result, supervised relational triple extraction can not extend to the unseen entity or relation types. A trivial solution is to annotate more data for unseen triple types, then retraining the model with newly annotated data (Zhou et al., 2019). However, this method is usually impractical because of the extremely high cost of annotation.

Intuitively, humans can learn about a new concept with limited supervision, e.g., one can detect and classify new entities with 3-5 examples (Grishman et al., 2005). This motivates the setting that we aim at for relational triple extraction: Few-Shot Learning (FSL). In few-shot learning, a trained model rapidly learns a new concept from a few examples while keeping great generalization from observed examples (Vinyals et al., 2016; Deng et al., 2020b). Hence, if we need to extend relational triple extraction into a new domain, a few examples are needed to activate the system in the new domain without retraining the model. By formulating this FSL relational triple extraction, we can significantly reduce the annotation cost and training cost while maintaining highly accurate results.

* Equal contribution and shared co-first authorship.
† Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 6399–6410
Barcelona, Spain (Online), December 8-13, 2020

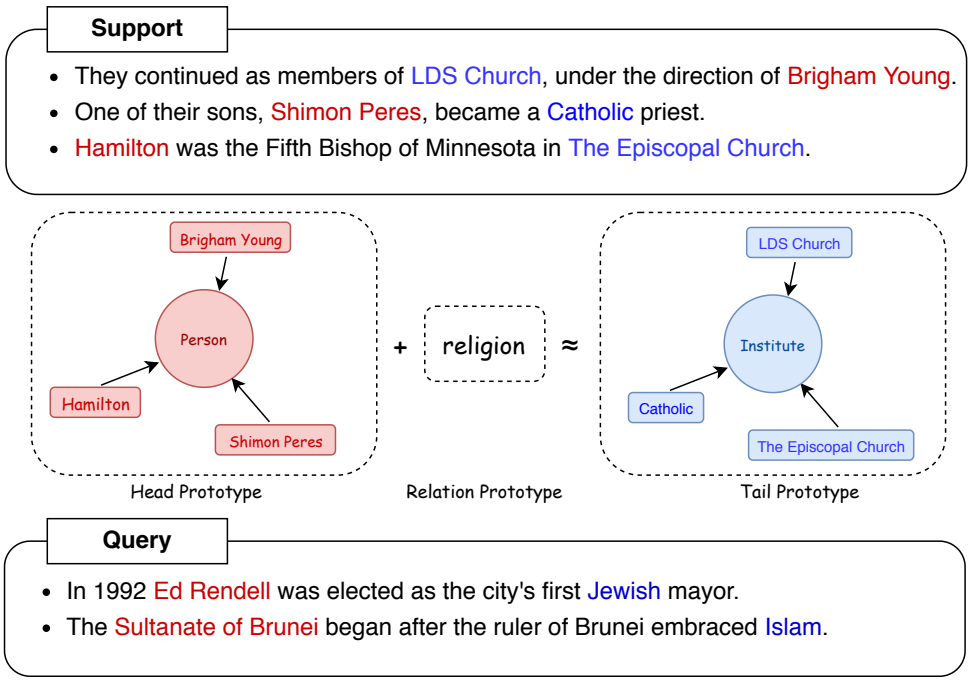


Figure 1: Illustration of our proposed model for relational triple extraction in the few-shot setting. The texts marked in red are head entities while in blue are tail entities. Head and tail entity prototypes are connected with the relation prototype.

Though methods of few-shot learning develop fast in recent yeas, most of these works concentrate on single tasks such as relation extraction and text classification (Geng et al., 2019; Ye and Ling, 2019). However, the effect of joint extraction of entities and relations on low-resource scenarios is still not well understood, which are two subtasks belonging to relational triple extraction. Unlike extraction for each single task, joint entity and relation extraction is more challenging, as entity and relations have implicit correlations, which cannot be ignored.

To address this issue, we propose a Multi-Prototype Embedding network (MPE) model to extract the few-shot relational triples, inspired by the prototypical network (Snell et al., 2017). To be specific, we utilize two kinds of prototypes regarding both entities and relations. Note that entity pairs and relations have explicit knowledge constraints (Bordes et al., 2013), such as the *Born_in* relation suggests that the type of *head entity* must be *PERSON*, and vice versa. Based on those observations and motivated by the knowledge graph embedding (Xie et al., 2016), we introduce the hybrid prototypical learning to explicitly inject knowledge constraints. We firstly learn entity and relation prototypes and then leverage translation constraint in hyperspace to regularize prototype embedding. Note that such knowledge-aware regularization not only injects prior knowledge from the external knowledge graph, but also leads to a more smooth and representative prototype for few-shot extraction. Moreover, we introduce prototype-regularization considering both intramural and mutual similarities between different prototypes. Experimental results on the FewRel dataset (Han et al., 2018) demonstrate that our approach outperforms baseline models in the few-shot setting.

To summarize, our main contributions include:

- We study the few-shot relational triple extraction problem and provide a baseline for this new research direction. To our best knowledge, this is a new branch of research that has not been explored.
- We propose a novel Multi-Prototype Embedding approach with hybrid prototype learning and prototype-aware regularization, which bridge text and knowledge for few-shot relational extraction.
- Extensive experimental results on the FewRel dataset demonstrate the effectiveness of our method.

2 Related Work

Two main directions have been proposed for relational triple extraction, which has two subtasks: entity extraction and relation extraction, namely, pipeline (Lin et al., 2016; Trisedya et al., 2019; Wang et al., 2020; Zhang et al., 2020b; Nan et al., 2020) and joint learning methods (Bekoulis et al., 2018b; Nayak and Ng, 2020; Ye et al., 2020). The pipeline model can be more flexible because it extracts entity pairs and relations sequentially, but this design will lead to error propagation (Zhang et al., 2018). Meanwhile, joint relational triple extraction models can solve this problem well by extracting triples end-to-end, and the interaction between entities and relations can be realized within the model, which makes the performance of the two mutually enhanced.

However, due to the "data-hungry" attribute of conventional neural networks, these relational triple extraction models need a large amount of data for training. Thus, lots of efforts (Zhang et al., 2019; Yu et al., 2020; Zhang et al., 2020a) have been devoted to few-shot learning, (Han et al., 2018) presents a few-shot relation extraction datasets to promote the research of information extraction in few-shot scenarios and adapt some few-shot learning methods (Munkhdalai and Yu, 2017; Satorras and Estrach, 2018; Mishra et al., 2017; Deng et al., 2020a) for this task. Among these models, the prototypical network (Snell et al., 2017) achieves comparable results on several few-shot learning benchmarks; meanwhile, it is simple and effective. This model assumes that each class exists a prototype, and it tries to find the prototypes for classes from supporting instances and compares the distance between the query instance under a particular distance metric. In natural language processing, (Gao et al., 2019) first proposes a hybrid attention-based prototypical network for few-shot relation extraction. (Fritzler et al., 2019) proposes to utilize the prototypical network to tackle the few-shot named entity recognition. (Hou et al., 2020) proposes a collapsed dependency transfer mechanism and a Label-enhanced Task-Adaptive Projection Network (L-TapNet) for few-shot slot filing. However, all previous few-shot works mainly consider single tasks, while relational triple extraction should take both entity and relation into consideration. To the best of our knowledge, we are the first approach for the few-shot relational triple extraction, which addresses both entities and relations.

Our work is motivated by knowledge graph embedding (Xie et al., 2016) such as TransE (Bordes et al., 2013) from Knowledge graph (KG), which is composed of many relational triples like $\langle$*head, relation, tail*$\rangle$. TransE is first proposed by (Bordes et al., 2013) to encode triples into a continuous low-dimensional space, which is based on the translation law $h + r \approx t$. Many follow-up works like TransH (Wang et al., 2014), DistMult (Yang et al., 2014), and TransR (Lin et al., 2015), propose advanced methods of translation by introducing different embedding spaces. In few-shot settings, it is extremely challenging to inject implicit knowledge constrains in vector space. Such simple yet effective knowledge constraints provide an intuitive solution.

3 Methodologies

3.1 Problem Definition

In few-shot relational triple extraction task, we are given two datesets, $\mathcal{D}_{meta-train}$ and $\mathcal{D}_{meta-test}$. Each dataset consists of a set of samples (*x, t*), where *x* is a sentence composed of *N* words, and *t* indicates relational triple extracted from *x*. The form of *t* is $\langle$*head, relation, tail*$\rangle$, where *head* and *tail* are entity pairs associated with the *relation*. These two datasets have their own relation domain spaces that are disjoint with each other. In few-shot settings, $\mathcal{D}_{meta-test}$ is split into two parts: $\mathcal{D}_{test-support}$ and $\mathcal{D}_{test-query}$. Due to entity pair types can be determined by the relation categories, e.g. the *Born_in* relation suggests that the type of *head* might be *PERSON* and *tail* might be *LOCATION*, we are able to determine the classification of triples only by specifying the categories of the relations. Therefore if $\mathcal{D}_{test-support}$ contains *K* labeled samples for each of *N* relation classes, this target few-shot problem is named *N*-way-*K*-shot. $\mathcal{D}_{test-query}$ contains test samples, each should be labeled with one of *N* relation classes, and associated entity pairs also need to be extracted correctly.

It is non-trivial to train a good model from scratch using $\mathcal{D}_{test-support}$ and evaluate its performance on $\mathcal{D}_{test-query}$, limited by the number of test-support samples (i.e.,., $N \times K$). Inspired by an important

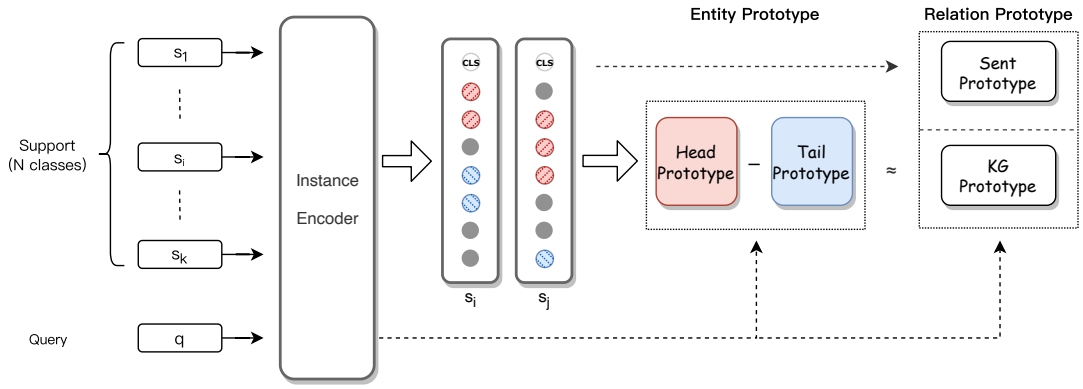


Figure 2: Overview of our proposed Multi-Prototype Embedding (MPE) model. Best view in color.

machine learning principle that test and train conditions must match, we can also split $\mathcal{D}_{meta-train}$ into two parts, $\mathcal{D}_{train-support}$ and $\mathcal{D}_{train-query}$, and mimic the few-shot settings at the training stage. In each training iteration, *N* triple categories are randomly selected from $\mathcal{D}_{train-support}$, and *K* support instances are randomly selected from each of *N* triple categories. In this way, we construct the train-support set $S = \{s_k^i; i = 1, \ldots, N, k = 1, \ldots, K\}$, where s_k^i is the k-th instance in triple category i. Meanwhile, we randomly select *R* samples from the remaining samples of those *N* triple categories and construct the train-query set $Q = \{(q_j, t_j); j = 1, \ldots, R\}$, where t_j is the triple extracted from instance q_j. Our goal is to optimize the following function:

$$L = -\frac{1}{R} \sum_{(q,t) \in Q} P(t|S, q) \tag{1}$$

Where $P(t|S, q)$ is the probability of gold standard relational triples.

3.2 Framework Overview

In this section, we will introduce our proposed Multi-Prototype Embedding (MPE) model for few-shot relational triple extraction. For brevity, we will temporarily study a sentence with one relation and associated entity pairs. The framework of our proposed model is shown in Fig. 2, which has three main modules.

- **Instance Encoder.** We utilize the pre-trained language model BERT (Devlin et al., 2018) to encode sentence, which adopts multi-head attention to learn contextual representations. Note that any other encoders such Roberta (Liu et al., 2019) and XLNet (Yang et al., 2019) can also be applied.

- **Hybrid Prototype Learning.** After obtaining entity pairs representations of each sentence used by sequence labeling methods, we can get entity prototypes in support set, and then construct relation prototype based on knowledge graph constraint, which takes the interaction between entity pairs and relations into account.

- **Prototype-Aware Regularization.** To further enhance the prototype learning, we optimize the position of prototypes in representation spaces. We make the distance between each prototype and related instances closer and distract those prototypes with different types.

3.3 Instance Encoder

For each sentence $x = \{w_1, w_2, \ldots, w_n\}$ in the support or query dataset, where $w_i \in x$ is the word token in sentence x, we first construct input sentence in the form: $\{$[CLS], $w_1, w_2, \ldots, w_n$, [SEP]$\}$, in order to match the input of BERT (Devlin et al., 2018). The pre-trained language model has been shown to be effective in many NLP tasks. [CLS] token is used to represent the entire sentence information, and [SEP] is the end token of sentence. After multi-head attention (Vaswani et al., 2017) calculation, we

can get sentence contextual embeddings $\mathcal{B} = \{h_0, h_1, h_2, \ldots, h_n, h_{n+1}\}$, where $\mathcal{B} \in \mathbb{R}^{d_{n+2} \times d_b}$, d_b is BERT pre-defined hidden size, h_0 is [CLS] token embedding, h_{n+1} is [SEP] token embedding, and $h_i, i \in [1, n]$ is each token embedding in sentence. Note that n can be different from input sentence length because of tokenizer (e.g., byte-pair-encoding) might split words into sub-tokens.

3.4 Hybrid Prototypical Learning

Entity Prototype Learning. During training stages, sentence representations in support datasets are first used to construct the entity pairs prototypes. We build an entity labeling set S = {B-Head, I-Head, B-Tail, I-Tail, O, X} to label out each token in the sentence, where B-Head, I-Head indicate head entity positions, B-Tail, I-Tail indicate tail entity positions, O means other tagging labels, and X is any remaining fragments of tokens split by the tokenizer.

We utilize Conditional Random Field (CRF) (Lafferty et al., 2001) for sequence labeling as it models the constraints between labels, which is more convenient in few-shot learning scenarios. Let $y = \{y_0, y_1, y_2, \ldots, y_n, y_{n+1}\}$, where y_0 is [CLS] token label which means the start of sentence, y_{n+1} is [SEP] token label which means the end of sentence, and $y_i, i \in [1, n]$ is each token label of sentence in entity labelling set. CRF uses emission and transition scores to combine local and global information, in our model, score of this sequence is evaluated as:

$$Score(x, y) = \sum_{i=0}^{N+1} E_{y_i, i} + \sum_{j=0}^{N} T_{y_i, y_{j+1}} \tag{2}$$

Let $\mathcal{Y}_\mathcal{X}$ indicate the exponential space of all possible labelings of this sequence x. The probability of a specific labeling $y \in \mathcal{Y}_\mathcal{X}$ is evaluated as:

$$p(y|x) = \frac{e^{Score(x,y)}}{\sum_{y \in \mathcal{Y}_\mathcal{X}} e^{Score(x,y)}} \tag{3}$$

We name the CRF-based sequence labeling loss as $loss_{crf}$ and minimize it during training stage.

After the above instance encoder and sequence labeling, we can obtain the head and tail representation to match the entities between the query and support set. Due to the variable length of entity words, we only use the first token representation of each entity word as head/tail embeddings, which is also used in (Soares et al., 2019). For measuring the distance between samples in query set and support set, we need compute a representative vector, called prototype, for each class $t \in T$ in the support set S from its instances' vectors. The original Prototypical Network (Snell et al., 2017) hypothesis that all instance vectors are equally important, so it aggregates all the representation vectors of the instance of class t_i, and then perform averaging over all vectors as follows:

$$head_{proto} = \frac{1}{|S_k|} \sum_{head_i \in S_k} head_i \qquad tail_{proto} = \frac{1}{|S_k|} \sum_{tail_i \in S_k} tail_i \tag{4}$$

where $head_i, tail_i$ are each sentence's entity pairs representations. Intuitively, the instances of a given relation may be quite different. Thus, we propose to adopt weighted sum prototype, named Proto+Att network inspired by (Gao et al., 2019). The weights are obtained by attention mechanism according to the representational vector of the query Q as follow:

$$head_{proto} = \frac{1}{|S_k|} \sum_{head_i \in S_k} \alpha_h head_i \qquad tail_{proto} = \frac{1}{|S_k|} \sum_{tail_i \in S_k} \alpha_t tail_i \tag{5}$$

where

$$\alpha_h = \frac{\exp(e_{h_i})}{\sum_{m=1}^{k} \exp(e_{h_m})} \qquad e_{h_i} = head_{proto}^T Q$$
$$\alpha_t = \frac{exp(e_{t_j})}{\sum_{n=1}^{k} exp(e_{t_n})} \qquad e_{t_j} = tail_{proto}^T Q \tag{6}$$

Specifically, we use Euclidean distance $d(\mathbf{z} - \mathbf{z}') = \|\mathbf{z} - \mathbf{z}'\|^2$, to calculate the distance between entity prototypes and instances in query set, and minimize this distance as $loss_{entity}$.

Relation Prototype Learning. This module computes relation prototypes associated with each entity pair. On the one hand, the first token [CLS] in the sentence representation can represent the whole sentence information. So like the above entity prototypes calculation, we can get sentence prototypes $sent_{proto}$, used by this sentence information in support set.

On the other hand, knowledge graph representation learning inspires us to learn a translation law $h + r \approx t$ (Bordes et al., 2013) on a continuous low-dimensional space, where h, r, t describe the head entity, the relation and the tail entity respectively. So we use $head_{proto}$ and $tail_{proto}$ to construct knowledge graph prototype kg_{proto}, which takes the interaction between entities and relations into consideration as follows:

$$kg_{proto} = |head_{proto} - tail_{proto}|W_r \tag{7}$$

Finally, we combine the prototype of sentence represent ions $sentproto$ and prototype from knowledge constrains between entity pairs kg_{proto} to form the relation prototype as follows:

$$relation_{proto} = [sent_{proto}; kg_{proto}], \tag{8}$$

Where $[;]$ refers to the feature vector concatenation. Similar to the entity prototype, we use Euclidean distance to calculate the distance between relation prototype $relation_{proto}$ and the sentence in the query set Q, and minimize this distance as $loss_{relation}$.

3.5 Prototype-Aware Regularization

Previous few-shot learning approaches (Ye and Ling, 2019) have shown that if the representations of all support instances in a class are far away from each other, it could become difficult for the derived class prototype to capture the common characteristics of all support instances. Therefore, we propose prototype-aware regularization to optimize prototype learning. Intuitively, we argue that the representational vectors (e.g, sentence representations/prototypes) of the same class should be close to each other; the prototypes of different types should be located far from each other in the prototypical space. Specifically, We use Euclidean and Cosine distance to measure these similarities, and optimize the prototype represetations as follows:

$$loss_{intra} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \|x_i^k - p_i^k\|_2^2 \qquad loss_{inter} = 1 - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=i+1}^{N} cosine(p_i, p_j) \tag{9}$$

where x_i is each sentence representation, p_i is associated prototypes, $loss_{intra}$ and $loss_{inter}$ are two different prototype-aware regularization functions. The overall regularizationn loss is: $loss_{regular}$ = $loss_{intra}$ + $\alpha loss_{inter}$, and α is hyperparameter.

The overall objective of the optimization is as follows:

$$L = loss_{crf} + \beta loss_{entity} + \gamma loss_{relation} + \delta loss_{regular} \tag{10}$$

where β, γ and δ are the trade-off parameters.

4 Experiments

4.1 Datasets

We conduct experiments on the public dataset FewRel[1] (Han et al., 2018), which is derived from Wikipedia and annotated by crowd workers. FewRel releases 80 relation categories, and each relation has 700 samples. We reconstruct the FewRel dataset to satisfy the few-shot relational triple extraction task. Our input information has only one sentence, and the required output is the relation and related entity pairs, which is a complete knowledge triple in the scheme of $\langle$*head, relation, tail*$\rangle$. In our experiments, we randomly select 50 relations for training, 15 for validation, and the rest 15 relation types for testing. Note that there are no overlapping types between these three datasets.

4.2 Settings

We implement our approach with Pytorch (Paszke et al., 2019). We employed mini-batch stochastic gradient descent (SGD) (Bottou, 2010) with the initial learning rate of $1e^{-1}$. The learning rate was decayed to one third with every 2000 steps, and we train 30,000 iterations. The dropout rate of 0.2 is used to avoid overfitting. Previous study (Snell et al., 2017) found that models trained on more laborious tasks may achieve better performances than using the same configurations at both training and test stages. Therefore, we set $N = 20$ to construct the train-support sets for 5-way and 10-way tasks. Furthermore, in each step, we sample 5 instances for query datasets. We utilize grid-search on valid set to tune hyperparameters. All of the hyper-parameters used in our experiments are listed in Table 1.

Component	Parameter	Value
BERT	type	base-uncased
	hidden size	768
Dataset	N_{train}	20
	R_{query}	5
learning rate	init (proto)	0.1
	init (BERT)	0.0005
	weight decay	1/3
	decay steps	2000
loss	α	0.75
	β	0.5
	γ	0.8
	δ	1

Table 1: Hyper-parameters of our approach.

We consider two types of few-shot relational triple extraction tasks in our experiments: 5-way 5-shot and 10-way 10-shot. We evaluate the performance of the entity, relation, and triple with the micro F1 score. To be specific, the entity performance refers to that the entity's span and span type are correctly predicted, the relation performance means that the relation of the entity pairs is correctly classified. Moreover, the triple performance means that the entity pair and associated relation are all matched correctly.

4.3 Baselines

We compared our model with baselines of supervised approaches and few-shot learning methods:

Supervised Learning. We utilize BERT (Devlin et al., 2018) with fine-tuning (Finetune) as supervised learning baselines. We finetune BERT with a batch size of 16 for 100 iterations.

Few-shot Leaning. We apply Matching Network (MatchingNet) (Vinyals et al., 2016), Relation Network (RelationNet) (Sung et al., 2018), vanilla Prototypical Network (Proto) (Snell et al., 2017) and Prototypical Network with attention (Proto+Att) (Ye and Ling, 2019) as few-shot baselines. We only utilize the sentence prototypes sen_{proto} in few-shot baselines which do not take the implicit knowledge into consideration.

4.4 Overall Evaluation Results

The first line of Table 2 shows the performance of our model on the FewRel test set. From the results, we observe that:

[1]https://www.zhuhao.me/fewrel/

Model	5-Way-5-Shot			10-Way-10-Shot		
	Entity	Relation	Triple	Entity	Relation	Triple
Finetune	6.57±0.52	71.36±0.49	4.71±0.96	4.36±0.63	63.83±0.60	2.94±0.77
MatchNet	12.30±0.74	84.15±0.28	10.13±0.43	5.94±1.23	79.34±0.51	4.40±1.02
RelationNet	11.87±1.61	88.73±0.11	9.91±0.28	8.24±0.79	82.40±0.37	6.65±0.33
Proto	15.43±0.50	87.10±0.25	14.18±0.25	7.76±0.51	80.46±0.33	6.53±0.60
Proto+Att	19.84±0.83	89.29±0.36	18.20±0.46	11.66±0.39	82.95±0.19	10.55±0.31
MPE	**25.03±1.24**	**93.81±0.31**	**23.34±0.79**	**14.85±0.80**	**84.58±0.32**	**12.08±0.83**

Table 2: F1 score on the FewRel test set.

1) Our approach MPE achieve the best performance in few-shot setting compared with all baselines (about absolute 5% improvement than Proto+Att in 5-way-5-shot), which demonstrates that the multi-prototype leveraging both text and knowledge is effective.

2) Entity recognition performs much worse than relation extraction in few-shot settings, as sequence labeling is more challenging than classification tasks, and the empirical results also observed by (Hou et al., 2020). More studies need to be taken to handle the challenging few-shot entity recognition task.

3) Proto+Att achieve better performance than Proto, which reveals that different instances have different contribution to prototype learning.

4) The overall performance is far from satisfactory, which need more future works to be taken into consideration.

4.5 Ablation Study

We further analyze the different modules of our approach by taking ablation studies, as shown in Table 3. w/o CRF implied without the CRF decoder; w/o Att implied without the attention in prototypical learning; w/o intra implied without the intra- constrains between instances and prototypes ; w/o inter implied without the inter-constrains between prototypes. From Table 3, we observe that:

Muitl-Proto	10-Way-10-Shot		
	Entity	Relation	Triple
MPE	14.85	84.58	12.08
w/o CRF	8.03	83.55	6.96
w/o Att	12.27	82.42	10.32
w/o intra	11.33	80.49	9.35
w/o inter	12.86	81.22	10.50

Table 3: Ablation study.

1) All approaches without different modules obtain performance decays, and w/o CRF has significant performance decay than w/o Att, w/o intra, and w/o inter, which demonstrates that the efficacy of CRF is more critical in few-shot relational triple extraction.

2) w/o intra or w/o inter has more performance drop compared with w/o Att, which also illustrates that prototype-aware regularization benefits the prototype learning.

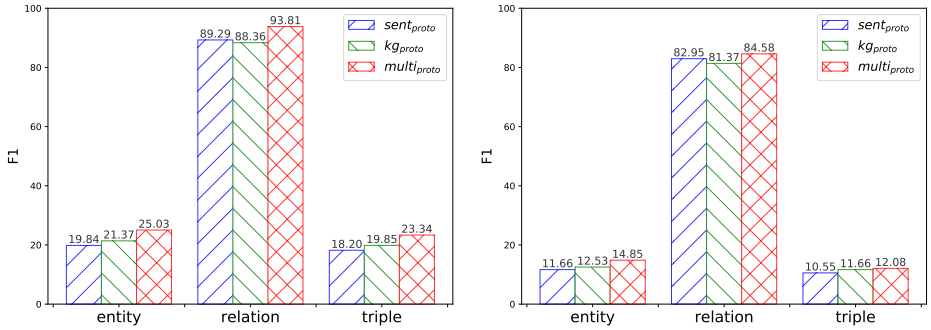
From Figure 3, we observe that the $multi_{proto}$ achieves better performance than sen_{proto} and kg_{proto}, and kg_{proto} is more advantageous than sen_{proto} for entity extraction, which further indicates that such knowledge constrains is beneficial.

In summary, we observe that entity recognition is more difficult than relation extraction in few-shot settings and the implicit correlation between them contribute to the performance.

4.6 Error Analysis

To further analyze the drawbacks of our approach and promote future works of few-shot relational extraction, we random select instances and conduct error analysis, as shown in Table 4:

Distract Context. As instance #1 shows, we observe that our approach may fail to those ambiguous contexts that may be expressed in a similar context but differ only in the fine-grained type of entities.



(a) 5-Way-5-Shot (b) 10-Way-10-Shot

Figure 3: Evaluation results of models with $sent_{proto}$, kg_{proto} and $multi_{proto}$.

Query Instance
instance #1 Delias mandaya is a species of pierine butterfly endemic to Mindanao, in the Philippines. extracted triplet: ⟨*Mindanao*, *contains administrative territorial entities*, *Philippines*⟩ ground truth: ⟨*Mindanao*, *country*, *Philippines*⟩
instance #2 Hamilton Hyde Kellogg was the Fifth Bishop of Minnesota in The Episcopal Church. extracted triplet: ⟨*Hamilton Hyde*, *religion*, *Church*⟩ ground truth: ⟨*Hamilton Hyde Kellogg*, *religion*, *The Episcopal Church*⟩
instance #3 His family has roots in the earliest Catholic presence in the United States west of the Appalachian Mountains; among his relatives are Martin John Spalding and John Lancaster Spalding. extracted triplet: ⟨*John Lancaster Spalding*, *religion*, *Catholic*⟩ ground truth: ⟨*Martin John Spalding*, *religion*, *Catholic*⟩

Table 4: Error analysis.

We argue that this may be caused by the unbalanced learning problems that models tend to classify the sentence with similar context to high-frequency relations.

Wrong Boundaries. As instance #2 shows, we observe that lots of extracted triples have incorrect boundaries, which further demonstrates the difficulty of entity recognition in the few-shot setting. More future works should be focused on the direction of few-shot sequence labeling.

Wrong Triples. As instance #3 shows, we observe that lots of extracted triples have entities that do not exist in the gold standard set. Generally, this is mostly happening in the sentence with multiple triples. Note the FewRel dataset does not have those labeled triples, and part of those cases is correct.

5 Conclusion and Future Work

In this paper, we study the few-shot relational triple extraction problem and propose a novel multi-prototype embedding network that bridge text representation learning and knowledge constraints. Extensive experimental results prove that our model is effective, but remains challenges. Those empirical findings shed light on promising future directions, including 1) enhancing entity recognition with effective sequence decoders; 2) studying few-shot relational triple extraction with more triples in a single sentence; 3) injecting logic rules to enable robust extraction; and 4) developing few-shot relational triple extraction benchmarks.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future. This work is funded by NSFCU19B2027/91846204/61473260, national key research program SQ2018YFC000004/2018YFB1402800, Alibaba CangJingGe (Knowledge Engine) Research Plan.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018a. Adversarial training for multi-context joint entity and relation extraction. *ArXiv*, abs/1808.06876.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018b. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst. Appl.*, 114:34–45.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020a. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. 2020b. When low resource nlp meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract). In *AAAI*, pages 13773–13774.

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

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 993–1000.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.

Ruiying Geng, Binhua Li, Yongbin Li, Xiao-Dan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *EMNLP/IJCNLP*.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. *arXiv preprint arXiv:2006.05702*.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. *ArXiv*, abs/2005.01159.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *ACL*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.

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

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org.

Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *In ACL*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *ArXiv*, abs/1911.09886.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2010. Semantic relation extraction with kernels over typed dependency trees. In *KDD '10*.

Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-shot learning with graph neural networks.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NIPS*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

Zifeng Wang, Rui Wen, Xi Chen, Shao-Lun Huang, Ningyu Zhang, and Yefeng Zheng. 2020. Finding influential instances for distantly supervised relation extraction. *arXiv preprint arXiv:2009.09841*.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. *ArXiv*, abs/1906.06678.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2020. Contrastive triple extraction with generative transformer. *arXiv preprint arXiv:2009.06207*.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. *ArXiv*, abs/1704.06194.

Haiyang Yu, Ningyu Zhang, Shumin Deng, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2020. The devil is the classifier: Investigating long tail relation classification with decoupling analysis. *arXiv preprint arXiv:2009.07022*.

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

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018. Attention-based capsule networks with dynamic routing for relation extraction. *arXiv preprint arXiv:1812.11321*.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of the NAACL*, pages 3016–3025.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen. 2020a. Relation adversarial network for low resource knowledge graph completion. In *Proceedings of The Web Conference 2020*, pages 1–12.

Ningyu Zhang, Luoqiu Li, Shumin Deng, Haiyang Yu, Xu Cheng, Wei Zhang, and Huajun Chen. 2020b. Can fine-tuning pre-trained models lead to perfect nlp? a study of the generalizability of relation extraction. *arXiv preprint arXiv:2009.06206*.

Xin Zhou, Luping Liu, Xiaodong Luo, Haiqiang Chen, Linbo Qing, and Xiaohai He. 2019. Joint entity and relation extraction based on reinforcement learning. *IEEE Access*, 7:125688–125699.