Information Processing and Management 58 (2021) 102596

Contents lists available at ScienceDirect

Information Processing and Management

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

Enhanced prototypical network for few-shot relation extraction

Wen Wen [a], Yongbin Liu [a,*], Chunping Ouyang [a,c], Qiang Lin [a], Tonglee Chung [b]

[a] School of Computer, University Of South China, Hunan, China
[b] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[c] Hunan provincial base for scientific and technological innovation cooperation, Hunan, China

ARTICLE INFO

Keywords:
Few-shot learning
Transformer
Relation extraction

ABSTRACT

Most existing methods for relation extraction tasks depend heavily on large-scale annotated data; they cannot learn from existing knowledge and have low generalization ability. It is urgent for us to solve the above problems by further developing few-shot learning methods. Because of the limitations of the most commonly used CNN model which is not good at sequence labeling and capturing long-range dependencies, we proposed a novel model that integrates the transformer model into a prototypical network for more powerful relation-level feature extraction. The transformer connects tokens directly to adapt to long sequence learning without catastrophic forgetting and is able to gain more enhanced semantic information by learning from several representation subspaces in parallel for each word. We evaluate our method on three tasks, including in-domain, cross-domain and cross-sentence tasks. Our method achieves a trade-off between performance and computation and has an approximately 8% improvement in different settings over the state-of-the-art prototypical network. In addition, our experiments also show that our approach is competitive when considering cross-domain transfer and cross-sentence relation extraction in few-shot learning methods.

1. Introduction

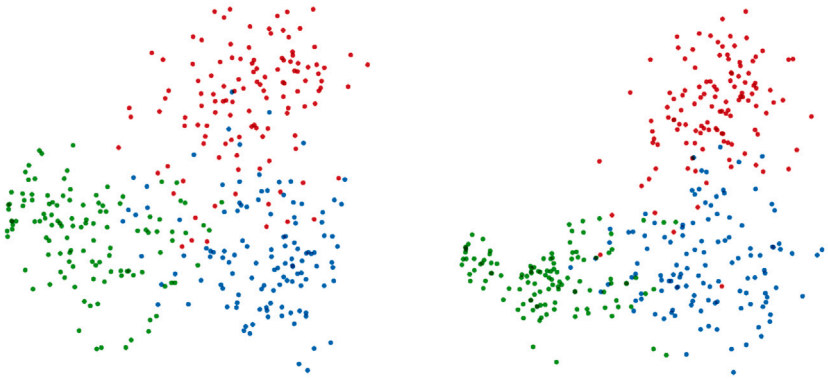
Relation extraction is a significant topic in NLP (Natural Language Processing), which aims to recognize the relation between two entities in a given text. Supervised models are commonly used, but they suffer from the limitation of the amount of high-quality annotated data which is not readily available because identification and manual annotation are arduous and expensive. Inspired by the strong ability exhibited by humans who can learn from the past and acquire new knowledge from a few examples, few-shot learning has been proposed and is rapidly emerging as a viable means for completing various tasks (Vo & Bagheri, 2019; Ye & Luo, 2019). Few-shot learning is able to reduce the burden of annotated data and quickly generalize to new tasks without training from scratch. In this paper, we focus on few-shot relation extraction tasks and aim to improve the performance of prototypical networks (Wang & Yao, 2019).

A prototypical network (Snell et al., 2017), which computes the Euclidean distance between each query instance and the prototype of each class, is an advanced method for few-shot learning tasks. In 2019, Gao et al. proposed hybrid attention-based prototypical networks (Gao, Han, Liu et al., 2019), which consist of an instance-level attention module and a feature-level attention module. The instance-level module aims to determine informative support instances, while the feature-level module extracts discriminative feature dimensions and then produces the distance metric for each relation by using the CNN architecture.

However, paper Tang et al. (2018) demonstrates that transformers can achieve better performance in extracting semantic features than CNNs. To show the advantage of Transformer, we utilize TensorBoard to compare the example embedding distribution of

* Corresponding author.
E-mail address: yongbinliu03@gmail.com (Y. Liu).

https://doi.org/10.1016/j.ipm.2021.102596
Received 18 December 2020; Received in revised form 23 February 2021; Accepted 18 March 2021
Available online 7 April 2021



(a) Hybrid Attention-Based Prototypical Network.

(b) TPN.

Fig. 1. Comparison between examples embedding with or without transformer on 3 classes and 60 examples for each which choose from Fewrel1.0 test dataset. The P4552 (mountain range) relation points have been colored blue. The P177 (crosses) relation is red, and the P59 (constellation) relation is green. The Hybrid Attention-Based Prototypical Network uses CNN as the feature extractor, while TPN is the model which we proposed in this paper, and the model utilizes transformer as the feature extractor instead of CNN. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Parameter statistics.

Model	Parameter Size
Hybrid Attention-Based Proto-typical Network (Gao, Han, Liu et al., 2019)	20.36 M
TPN	36.13 M

Table 2
Hyper-parameter settings.

Hyper-parameter	Initial Value
Batch Size	1
Query Size	5
Learning Rate	1e−1
Learning Rate Step Size	5000
Weight Decay	1e−5
Optimizer	SGD
Training Iterations	30,000
Hidden Layer Dimension	512

the Fewrel 1.0 dataset. Because the number of relations is too large, we choose a part of the dataset, as shown in Figs. 1(a) and 1(b). Different relations have been painted with different colors. From the two figures, two significant discoveries are made when comparing our proposed model integrated transformer and the model containing CNN: First, the points in the same relations aggregate better. Second, the points with different colors are more separable. To analyze these results, we calculate the numbers of parameters in each model and present them in Table 1. Due to the efficiency and extendibility of the transformer, the proposed model containing transformer is able to train on over $36M$ parameters and has approximately 1.5 times compared with the model with CNN. Thus, our proposed model can acquire more hidden features that lead directly to better performance. The parameters of the experiments are shown in Table 2.

In addition, Tang et al. (2018) also demonstrates that CNN is not good at capturing long-range dependencies between tokens for longer sentences compared to the transformer model (Vaswani et al., 2017). However, in practice, the path between the co-dependent tokens may be overlong or appear in different sentences, so predicting the relation between the given entity pair should consider long sentences and cross sentences, both of which need long-term memory. CNN has difficulty meeting the far away information because the length between two co-dependent tokens is limited by the size of the kernel and the number of layers, as shown in Fig. 2(a). Multi-head attention is a vital part of transformers, and it greatly helps transformer performance because the shorter path between co-dependent tokens helps to learn the dependencies more easily. A diagram of the attention mechanism is shown in Fig. 2(b). Attention parallel handles the input sequence, and in its self-attention, any two tokens in an example are connected directly (Zeng et al., 2014). The self-attention characteristic has great help in capturing long-range dependencies. In addition, multi-head attention focuses on different parts of the input and then encodes the objective word with the whole instance. Because of the attention mechanism potential, transformers allow for more parallelization and are able to gain more semantic information while requiring less time for training (Ringer et al., 2019; Vaswani et al., 2017).

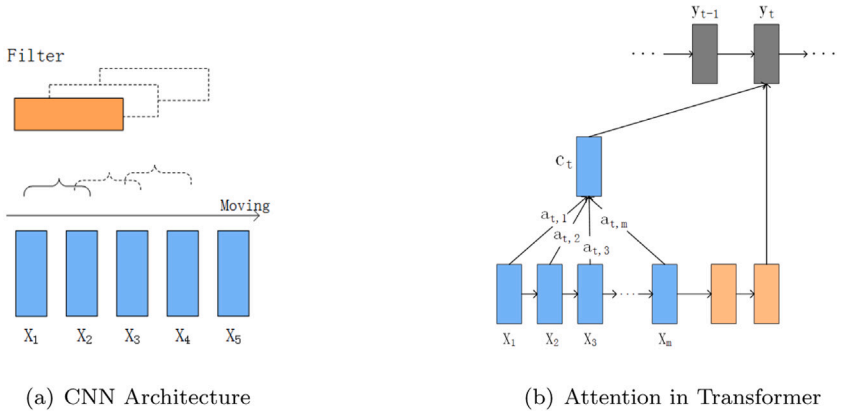


Fig. 2. Comparison between CNN architecture and attention mechanism in transformer architecture. When considering single layer and setting the size of kernel to k, CNN can only capture k-gram piece of information, while transformer can capture long-range dependencies because of the direct connection between all co-dependent tokens.

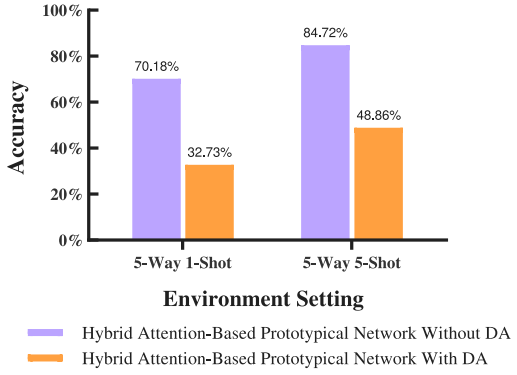


Fig. 3. Disaster of Domain Adaptation (DA).

On the other hand, in real-world applications, we always would like to train on a general-domain corpus that has abundant annotated data and then test on other specific corpora that are difficult to annotate and suffer from data sparsity problems, such as medical domain data (Gao, Han, Zhu et al., 2019; Zeng et al., 2018). However, most of the existing algorithms breakdown when they apply to make predictions on datasets that have little information. When considering transfer across domains, the domain adaptation (DA) task comes into being (Baktashmotlagh et al., 2013; Zhao et al., 2020). We evaluate a hybrid attention-based prototypical network (Gao, Han, Liu et al., 2019), which is the start-of-the-art method of few-shot relation extraction tasks. First, we experiment on the Fewrel dataset (Han et al., 2018) which comes from wiki (without DA). Then, to show the effect of applying the trained model to other domains, that is, the testing data domain is different from training (with DA); therefore, we choose a new test set PubMed that comes from a database of biomedical literature and is annotated by Fewrel2.0 (Gao, Han, Zhu et al., 2019). The parameter setting are shown in Table 2, which are determined by experiments in Section 4. As Fig. 3 shows, the classification performance drops dramatically when taking DA into consideration such that in the 5-way 5-shot setting, the accuracy of the model falls from 71.93% to 32.73%, and in the 5-way 1-shot setting, the number drops from 85.44% to 48.86%, where has a great approximately of about 40%.

To address the above issues, we propose a novel model named TPN that integrates a transformer into a prototypical network (Chen et al., 2018; González et al., 2020). The TPN model focuses on enhancing the prototypical network (Gao, Han, Liu et al., 2019) by providing a more powerful extractor to highlight more discriminative features for relations since the few-shot tasks have only a few examples and the performance of the few-shot learning method depends strongly on the quality of the extracted features. We apply our approach to the few-shot relation extraction task and evaluate the model on three benchmark datasets. To further improve our proposed method, we adopt pre-trained BERT (Devlin et al., 2018) as the encoder of our model, which is signed by TPN(BERT) in our paper. The results of our experiments demonstrate that our method outperforms the state-of-the-art baseline model and is competitive for few-shot learning tasks (Ji et al., 2020).

In summary, the main contributions of our proposed TPN model can be summarized as:

- We proposed a novel TPN model for the few-shot relation extraction method. The model enhances the prototypical network (Gao, Han, Liu et al., 2019) by integrating the transformer model into it as the relation-level feature extractor, which

is able to extract more discriminative feature dimensions for relations. In addition, we employ TPN(BERT), which utilizes pre-trained BERT (Devlin et al., 2018) as the encoder of the proposed model.
- The proposed TPN model with a transformer is more suitable for cross-sentence datasets, which are more general in the real world, than other state-of-the-art baselines, especially BERT-PAIR.
- Both the number of transformer blocks and the number of heads in attention affect the performance of our model.
- Our experiments use three datasets to evaluate our method under several different scenarios. The experimental results demonstrate that our model not only strikes a balance between performance and consumption, and our model can achieve a competitive performance compared with existing state-of-the-art baselines when tested on the same domain as training, but is also competitive when considering domain adaptation and crossing-sentence relations.

The remainder of this paper is structured as follows. The related work is presented in Section 2. Section 3 defines the few-shot relation extraction task and then describes the proposed TPN method in detail. Experiments are presented and the results analyzed in Section 4. Finally, in Section 5, we conclude the work and list future work.

2. Related work

2.1. Supervised relation learning

The supervised model, which is the most common method for relation extraction tasks, only works efficiently when there are adequate samples of each relation. However, large-scale and high-quality annotated data are not readily available because of the high expense of labeling. In addition, the data are always labeled on a specific corpus that includes a limited number of domains and is far from infinite in the real world; thus, in practical applications, supervised models can only train from scratch to learn new tasks. Moreover, most existing models are not able to rapidly generalize data because they cannot learn by incorporating prior knowledge (Elsken et al., 2020; Jin et al., 2019; Liu et al., 2017; Xie et al., 2020; Zhang et al., 2019).

In 2009, Mintz et al. (2009) suggested a method that applies distant supervision to relation extraction. The main characteristic of this method is that it automatically annotated data instead of depending on manual annotation. It is assumed that if two entities contained in a sentence have a known relation label, the pair of entities would like to express the same relation in all sentences. Although it can produce a great amount of labeled data, noise is always introduced to the generated data, and noise data may guide to poorly-performing models because the pair of entities may have different relations in different sentences. Then, multiple instance learning (MIL) (Chung et al., 2019; Luo et al., 2017; Wan et al., 2019; Zeng et al., 2015) was proposed to alleviate the problem by relaxing the distant supervision assumption. MIL predicted the labels at the bag level which aggregated multiple instances. However, the label was unstable and may shift with only a tiny change in the data.

These supervised learning methods need enough labeled data for training, which is a very high cost. Sometimes it is tough to obtain large-scale annotation datasets for some fields. Although the distance supervision method does not require the scale of labeled data as much as the supervised learning method, it also needs a certain number of training datasets. When the training examples of some relationships are few, its performance will still decline sharply.

To alleviate the above issues, few-shot learning is proposed by simulating humans who have the natural ability to take into account previous knowledge and then recognize new classes when given a small number of examples. Thus, the purpose of few-shot learning is to retain prior knowledge and rapidly generalize to new classes from only a few examples of each class without retraining (Wang et al., 2020).

2.2. Parameters optimization learning

To utilize previously learned and then make rapid generalization come true, meta network (Munkhdalai & Yu, 2017; Ravi & Larochelle, 2017) was proposed based on gradient optimization (Elsken et al., 2020). Meta network consisted of a base learner and a meta learner. The meta learner acquired meta-knowledge across tasks, while the base learner gained meta-information presented in the loss gradient and then adaptively found a proper parameter to update the model in each task (Finn et al., 2017; Vanschoren, 2018).

Model agnostic method is proposed by Finn et al. (2017). The idea of MAML (Model Agnostic Meta Learning) approach is to learn an initial condition (set of initialization parameter) that is good for fine-tuning on few-shot problems. The few-shot optimization approach (Ravi & Larochelle, 2017) goes further in meta-learning not only a good initial condition, but also an LSTM based optimizer to help fine-tune. And then Bayesian Model-Agnostic Meta-Learning (Yoon et al., 2018) combines scalable gradient-based meta-learning with nonparametric variational inference in a principled probabilistic framework. Recently, the Bayesian Meta-learning based graph neural network and BERT is presented by Qu et al. (2020). The REGRAB (Few-shot Relation Extraction via Bayesian Meta-learning on Relation Graph) approach adopt the Bayesian meta-learning and parameterize the prior distribution of prototype vectors of relations by applying a graph neural network to the global graph. They use stochastic gradient Langevin dynamics to draw multiple samples from the posterior for optimization. However these approaches suffer from the need to fine-tune on the target problem, or depend on the pre-training model.

2.3. Metric based few-shot learning

Subsequent work on the Siamese neural network (Koch et al., 2015), which was applied to few-shot classification by Koch(2015), aimed to naturally rank the similarity between inputs by using a convolutional architecture. It is also called a twin neural network; just as its name implies, the Siamese network can only evaluate the similarity of two inputs once a time. Thus, it needs to iterate over all training examples one by one which is time-consuming work. Then, in 2017, Vinyals et al. formulated a matching network (Vinyals et al., 2016) that enhanced neural networks with external memories and used an attention mechanism to predict classes. It provided a new method named cosine distance as the metric of similarity. MLMAN model (Multi-level matching and aggregation network for few-shot relation classification) (Ye & Ling, 2019) goes further improving matching network. The approach encodes the query instance and each support set in an interactive way by considering their matching information at both local and instance levels. In 2018, Sung et al. (2018a) proposed a relation network for few-shot learning. The relation network learns an embedding and a deep non-linear distance metric for comparing query and sample items. In addition, since the squared Euclidean distance empirically outperforms the more commonly used cosine similarity, Snell et al. (2017) proposed a simpler and more efficient model prototypical network that used the simple Euclidean distance as the distance function in the naive approach. The prototypical network assumes that there exists a prototype where points cluster around for each class and the query point is classified by using the distance function to calculate the nearest class prototype, which is determined by the instances in its support set. Recently, Ren et al. (2020) proposed a two-phase prototypical network model with prototype attention alignment and triplet loss to dynamically recognize the novel relations with a few support instances meanwhile without catastrophic forgetting.

Most of these works have been developed in the image domain but are not widely used in the NLP domain, so we demand to apply them to few-shot relation classification tasks. In 2019, Gao et al. introduced a hybrid attention-based prototypical network (Gao, Han, Liu et al., 2019), which is our state-of-the-art baseline and has the same idea as a prototypical network. The model consisted of the instance-level attention module and the feature-level attention module. The instance-level attention module aims to choose the more informative instances in the support set, while the feature-level attention module uses the CNN model to extract the important dimensions of relation features, and then formulates the distance metric in a suitable way for different relations. The method demonstrates that the performance of models depends strongly on the quality of features, so we concentrate on it and aim to enhance the ability to extract discriminative feature dimensions by replacing CNN.

CNNs and RNNs are widely used as the feature extractors. In CNNs, the maximum length that CNN can capture is determined by the number of CNN layers and the size of the kernel, as shown on the left of Fig. 2(a). However, the number of CNN layers is under restriction. For a single CNN layer, the path between co-dependent tokens may be very long and it is difficult to find the correct dependencies when evaluating on long sentences. Moreover, RNNs predict the current output by using the previous sequential information, which hinders parallelization. Vaswani et al. proposed the transformer model (Vaswani et al., 2017), which solved the above problems and provided a powerful feature extractor. The self-attention mechanism in the transformer can handle the input sequence in parallel and make direct connection between co-dependent tokens, which play an important role in capturing long-range dependencies. Multi-head attention focuses on different subspace to acquire richer information.

Above all, we propose the novel TPN model, which integrates the transformer model into a prototypical network. Our proposed model has two modules: in the relation-level module, the transformer is the main component that aims to extract the more discriminative feature for different relations, while the example-level module is the same as the instance-level module in Gao, Han, Liu et al. (2019), which extracts the more informative example for each query example.

3. Methodology

In this section, we define the few-shot extraction task and then introduce our innovative solution that integrates the transformer model into a prototypical network, which aims to extract features for different relations and then obtain a task-adaptive distance metric for few-shot relation extraction tasks. We describe the detailed processes of our method in two parts below.

3.1. Task definition

Few-shot relation extraction is a task that proposes the production of a reliable classifier that can rapidly adapt to new tasks when given just a few examples. Formally, the objective of our experiments is to predict the label r_y of query example y that has not been trained upon, after seeing a small number of support examples S. The label of a query example y is the semantic relation between head entity e_h and tail entity e_t, both of which are mentioned in the query y (Devos & Grossglauser, 2019; Wang & Yao, 2019). Fig. 4 shows an example of the task.

As shown in Fig. 4, all the examples are inherently divided into three segments respectively, depending on the given head entity e_h and tail entity e_t. The two entities are bounded by the relation r, and then each example can be typically represented as an irreversible relationship $r(e_h, e_t)$. For the N-way K-shot task, the model classifies over N classes that are randomly sampled from the dataset and randomly samples $K+Q$ examples from each class where K should be quite small. Therefore, $N * K$ examples make up of the support set S, while the other $N * Q$ examples consist of the query set Y. Support set S is defined as Eq. (1), where r_i is the relation label between the pair of entities $(e_{h_{ij}}, e_{t_{ij}})$ in example x_j (Satorras & Estrach, 2018; Sung et al., 2018b), and similarly, query set Y can be defined as Eq. (2).

$$\begin{aligned} S = \{ & r_1\{(x_{11}, e_{h_{11}}, e_{t_{11}}), \ldots, (x_{1K}, e_{h_{1K}}, e_{t_{1K}})\}, \ldots, \\ & r_N\{(x_{N1}, e_{h_{N1}}, e_{t_{N1}}), \ldots, (x_{NK}, e_{h_{NK}}, e_{t_{NK}})\}\} \end{aligned} \tag{1}$$

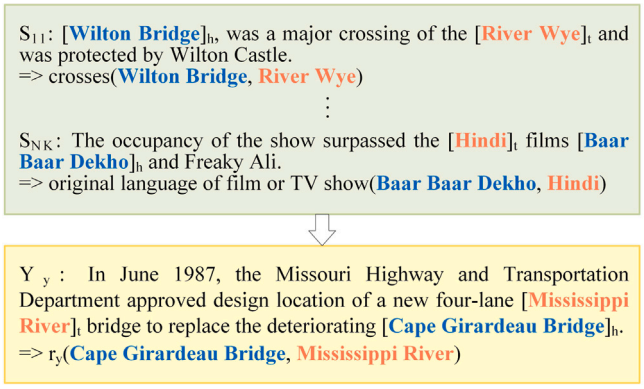


Fig. 4. An example of N-way K-shot relation classification task. r_y is the relation label of the unseen example y which come from query set Y. Few-shot relation extraction model predicts the relation label r_y belonging to which one of the N classes by only seeing the examples in support set S. To clear display the task, we assume that there has $Q = 1$ query example for each relation in Y and r_y is the relation "crosses".

$$\begin{aligned} Y = \{ & r_1\{(y_{11}, e_{h_{11}}, e_{t_{11}}), \ldots, (y_{1Q}, e_{h_{1Q}}, e_{t_{1Q}})\}, \ldots, \\ & r_y\{(y_{y1}, e_{h_{y1}}, e_{t_{y1}}), \ldots, (y_{yQ}, e_{h_{yQ}}, e_{t_{yQ}})\}, \ldots, \\ & r_N\{(y_{N1}, e_{h_{N1}}, e_{t_{N1}}), \ldots, (y_{NQ}, e_{h_{NQ}}, e_{t_{NQ}})\}\} \end{aligned} \tag{2}$$

Few-shot relation extraction models work with minibatches, each of which is formed from (R, S, Y) where R is the relation set and the labels of S and Y are belong to it. For one minibatch, the model learns features from S and then predicts the label r_y of each unseen query example y based on R.

3.2. Model

In this section, we introduce our proposed TPN model in detail. The framework of our TPN model is presented in Fig. 5. Our model is trained to learn a metric space and the classification can be performed by computing the distances between each query and prototypes of different relations.

3.2.1. Embedding and encoding

Let $x = (word_1, word_2, \ldots, word_{|x|})$ denote an example in which entities pair (e_h, e_t) are mentioned. Then, the lth word in the given example x maps to a low-dimensional dense-vector $w_l \epsilon R^{d_w}$ where R^{d_w} is the dimension of word embedding, according to the pre-trained dictionary word vector GloVe (Pennington et al., 2014). Each w_l is injective and contains the semantic and syntactic information of the context. We calculate the relative position pos_{lh} (pos_{lt}) to the entity e_h (e_t) because the relative position is a significant feature to guide our model to pay more attention to words that are close to the target entities. We show the process in Eqs. (3) and (4), where pos_h (pos_t) is the pos of e_h (e_t), and len is the length of the example. If ($|x| < len$), the example should be padded by blank, and when ($|x| > len$), the overlength would be cut (Gao, Han, Liu et al., 2019).

$$pos_{lh} = l - pos_h + len,\ pos_{lh} \epsilon R^{d_p}; \tag{3}$$

$$pos_{lt} = l - pos_t + len,\ pos_{lt} \epsilon R^{d_p}; \tag{4}$$

Then, we concatenate the word embedding $\{w_1, w_2, \ldots\}$ and the two position embeddings $\{(pos_{1h}, pos_{1t}), (pos_{2h}, pos_{2t}), \ldots\}$ as input embeddings of the model. As a result, we present the input sequence as follows:

$$\begin{aligned} \{I_1, I_2, \ldots, I_{len}\} = \{ & \\ & [w_1; pos_{1h}; pos_{1t}], \\ & [w_2; pos_{2h}; pos_{2t}], \\ & \cdots, \\ & [w_{len}; pos_{lenh}; pos_{lent}] \\ \}, \quad & I_m \epsilon R^{d_m}, d_m = d_w + 2 * d_p \end{aligned} \tag{5}$$

The encoding layer encodes the input $\{I_1, I_2, \ldots, I_{len}\}$ to obtain the final example representation. The formula is shown in Eq. (6) where E_j refers to the jth example vector, and $SE(\cdot)$ is the function of the sentence encoder. In our work, we follow (Gao, Han, Liu et al., 2019) to choose CNN as the encoder and then follow a max pooling layer to reduce the feature map first. Then, to further develop the performance of our model, we utilize pre-trained BERT as the encoder, instead of CNN.

$$E = SE(I_1, I_2, \ldots, I_{len}) \tag{6}$$

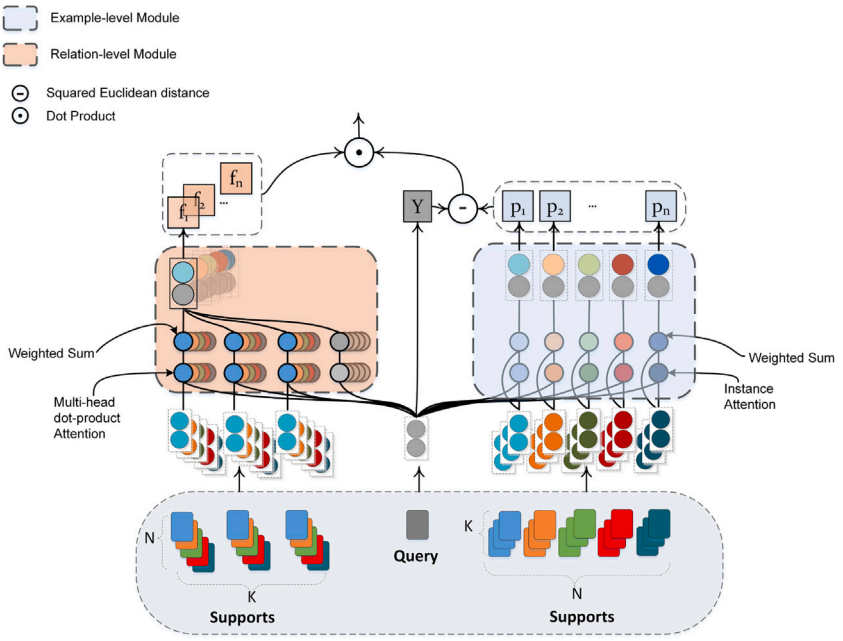


Fig. 5. The framework of TPN model. "Supports" is the support set, which has N*K examples for N-way K-shot setting. Different color in the "Supports" represent different relations. The model learn from "Supports" to predict the label of "Query". The model consist of the following two parts: relation-level module (left side) and example-level module (right side).

3.2.2. Feature extraction pipeline

Feature extraction is the main problem that hinders the performance of the few-shot learning model. In this section, we introduce the two key components of the feature extraction pipeline in our model below.

Example-level selector. Prototypical networks classifying the relation mainly depend on the prototype of each relation. Prototypical networks determine the prototype p_i of relation r_i by taking all the examples in relation r_i into consideration. The naive approach calculates the prototype p_i by averaging the support embedding $\{E_{i1}, E_{i2}, \dots, E_{ij}, \dots, E_{iK}\}$ for relation r_i, which is shown in Eq. (7).

$$p_i = \frac{1}{K}\sum_{j=1}^{K} E_{ij} \tag{7}$$

However, in practice, support examples should have different contributions to predict query labels, due to the diverse semantics of relations. Thus, the example selector we choose works in the same way as the instance-level attention module of (Gao, Han, Liu et al., 2019), which is shown on the right side of Fig. 5. The example-level selector focuses on the examples that have more similar features of the query for each relation r_i through the inner product between support and query. Thus, the prototype p_i of each relation r_i is calculated by Eq. (8), where α_j is the weight matrix of each example (Gao, Han, Liu et al., 2019), q is the query attention, and F is the linear layer.

$$p_i = \sum_{j=1}^{K} \alpha_j E_{ij} \tag{8}$$
$$\alpha_j = Example\_Score = SoftMax\{Sum(\tanh(F(E_{ij}) \times F(q)))\}$$

Relation-level extractor. To solve the problem of few-shot relation extraction tasks, we demand obtaining a task-adaptive distance function to compute the similarity between prototype p and query instances y. Then, our model outputs the minimum value of distance as the predictive label r_y. In our model, the distance function is based on Euclidean distance, which follows (Gao, Han, Liu et al., 2019). The distance function is shown in Eq. (9), where f_i is the weight of feature dimensions for relation r_i.

$$d(p_i, y) = f_i \cdot (p_i - y)^2 \tag{9}$$

To achieve better feature representation of relations, we propose utilizing the transformer model (Vaswani et al., 2017) as the relation feature extractor to enhance the hybrid attention-based prototypical network (Gao, Han, Liu et al., 2019), which is our state-of-the-art baseline, as shown on the left side of Fig. 5. Compared to CNN, which has been used in (Gao, Han, Liu et al., 2019) and most existing works, transformer is a strong feature extractor that is able to extract the more discriminative feature dimensions for each relation in our task. The transformer we used consists of $B = 3$ transformer blocks with $h = 8$ attention heads. In the transformer, we set the support instance embedding as the input of the source language and the query embedding as the input

of target language. To avoid the ambiguity of the sentence, the extra relative positions $(tpos_1, tpos_2, \ldots, tpos_{1en})$ for each sentence are required to the input of the transformer. Then, the transformer encoder maps $Z_0 = \{E_1TP_1; E_2TP_2; \cdots, E_{N*K}TP_{N*K}\}$, where $TP_t = (tpos_1, tpos_2, \ldots, tpos_{1en})$, to the context sensitive representations $C = (c_1, c_2, \ldots, c_{N*K})$ in the following way (LN is the layer normalization, $Atten$ is the function of multi-head attention, and PF is the position-wise feed forward network.):

$$\begin{aligned} M_b &= LN(Atten(Z_{b-1}) + Z_{b-1}), \quad b = b \cdots B \\ Z_b &= LN(PF(M_b) + M_b), \quad b = b \cdots B \end{aligned} \tag{10}$$

Finally, the transformer decodes the output sequence $F = (f_1, f_2, \ldots, f_N)$ in the same way as the encoder part, where $f_i = g(c_i, f_1, \ldots, f_{i-1})$ (Hu et al., 2019; Wang et al., 2019).

In addition, transformer mainly depends on multi-head attention and contributes significantly to the quality of the model. In multi-head attention, the attention is calculated respectively by $h = 8$ parallel heads that focus on different parts of the input and can learn different representations in different subspaces. Then concatenate the output of $h = 8$ heads to obtain the final vector which will be more semantic (Vaswani et al., 2017; Vig, 2019). The formulation is shown in Eq. (11), where $head_t$ is the result of the tth attention head and W is the weight parameter.

$$multi\_head(Y, K, V) = [head_1; \cdots; head_h]W \tag{11}$$

For each head in attention, a query vector y and a set of key–value pairs (k, v) are required, which are supposed to be produced by linear transformation of input embedding, and they are d_k, d_k, and d_v dimensions, respectively. Then, we perform the attention function which aims to output the weighted sum of values for the object w_l, as shown in Eq. (12). Similar to the operation shown in (Vaswani et al., 2017), we use the dot products of q with all corresponding k and then scale each by $\frac{1}{\sqrt{d_k}}$ to stabilize the gradient. Finally, a softmax function should be applied to obtain the weights on the values, and the result is the final attention-degree of all words in the whole instance (Domhan, 2018).

$$\begin{aligned} head_t &= Dot\_Atten(q_t, k_t, v_t, d_k) \\ &= SoftMax(\frac{q_t k_t^T}{\sqrt{d_k}})v_t, \\ q, k, v &= SW_Q, SW_k, SW_v \end{aligned} \tag{12}$$

4. Experiment

In this section, we compare our proposed TPN model with existing strong baselines to show the advantages. Then, we further demonstrate the effect of the number of transformer blocks and the heads of attention in the transformer.

4.1. Datasets

For each scenario, we evaluate our TPN model on three benchmarks, with details shown in Table 3. The FewRel (Han et al., 2018) dataset contains 100 relations, each with 700 instances. The test set, which has 20 relations, is hidden by the authors, so we need to build the test set by ourselves. The original training has 64 relations, and the original validation has 16 relations. To satisfy the experiment on 10-way tasks which need ten relations at least and try our best to keep the origin FewRel dataset to show the advantage of our proposed model directly, we randomly choose four relations from the train set and six relations from the validation to form the test set of ours. Thus, the dataset we used has 60 relations for training, 10 relations for validation and 10 relations for testing. Since all of the FewRel data are from the Wikipedia corpus, that is to say, they are included in the same domain, we evaluate our model on the FewRel2.0 (Gao, Han, Zhu et al., 2019) dataset, which considers the cross domain. The training set of FewRel2.0 comes from Wiki (Vrandečić & Krötzsch, 2014), which is the same as FewRel and has 64 relations. We use SemEval-2010 task 8 (Hendrickx et al., 2010) of FewRel2.0 as validation, which has 17 relations. Most of the data in the SemEval-2010 task 8 dataset come from news. Then, we tested on PubMed,[1] which has 10 relations and comes from the database of biomedical literature. Thus, the training set, validation set and test set are in different domains. FewSP is collected by us to focus on crossing-sentence relation extraction, and it consists of the following two parts: one part comes from Google Code Relation-extraction-corpus[2] which contains over 10,000 sentences. The sentences have the following five kinds of relations: institution, place of birth, place of death, date of birth and graduate degree, and the remainder part is extracted from TAC with over 5000 sentences also with five kinds of relations including personal social, general affiliation, physical, origin affiliation, part whole. We randomly choose 5 relations for the training set and the other 5 relations for the validation set. The relation of our test set is the same as the validation, but the example of our test is disjointed with the validation because we cannot find more cross-sentence relation datasets. Each of the datasets is divided into the following three parts: training set, validation set, and testing set. In addition, the training set has its own label space, which is disjoint with the others. Moreover, we use accuracy as our evaluation criteria.

[1] https://www.ncbi.nlm.nih.gov/pubmed/

[2] https://code.google.com/archive/p/relation-extraction-corpus/downloads

Table 3
Dataset.

Dataset	Source	Apply	Relation Number	Instances Number
FewRel	Wiki	Train	60	42,000
		Val	10	7000
		Test	10	7000
FewRel2.0	Wiki	Train	64	44,800
	SemEval	Val	17	8851
	PubMed[1]	Test	10	2500
FewSP	Google Code	Train	5	4575
	Relation-extraction-corpus[2]	Val	5	7832
	& TAC	Test	5	3359

4.2. Experimental settings

In this part, we study the impacts of the hyper-parameters in our model. BERT-PAIR is trained on a machine with 3 T V100 32GB GPUs, and the other models are trained on a device with single Tesla P100 16GB GPUs. First, we randomly choose various values for each hyper-parameters, and then focus on the most influential hyper-parameters. For example, the learning rate is one of the key metrics to improve the accuracy of our model and is shown in Fig. 6.

The figure shows that on the FewRel, FewRel2.0 and FewSP datasets, our TPN model acquires the highest accuracy when the learning rate is 1e−1. The learning rate setting that is able to achieve the best performance for the TPN(BERT) model is different for the three datasets. Overall, we set the learning rate to 1e−1 in our experiments to compare different models fairly. Since memory constraints, we try our best to set these parameters which is shown in Table 2 as described above. We use the same parameter settings for all models and use the early stopping strategy to avoid overfitting. In our experiments, a group of query sets contains 5 examples, while the number for the support set is determined by our tasks. The batch size is set to 1, so each mini-batch contains one group of support sets and query sets. The initial learning rate is set to 1e−1, and we decay the learning rate after 5000 iterations. The parameters in our model are tuned by stochastic gradient descent (SGD) with a weight decay of 1e−5. Model checkpoint is saved every 2000 updates. For our transformer layer, there are 3 stacked blocks and the number of attention heads is set to 8 (Vaswani et al., 2017). For other hyper-parameters, the values are set according to the situation. In addition, to enable rapid learning and be fair to all models, we have the same setting for both training and testing in our experiments.

4.3. Convergence of TPN method

To determine how many steps our TPN model converges, we choose several points to present the changes in the accuracy of the model when training on the three datasets, as shown in Fig. 7. This demonstrates that the value of accuracy increases and tends to stabilize as the number of iterations increases. We can see that the models converge on all datasets at 30,000 steps.

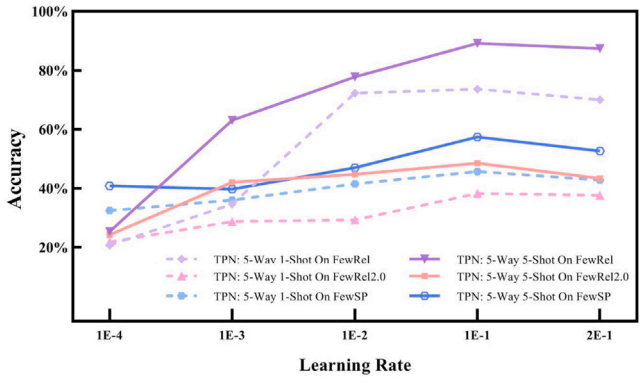
4.4. Comparison method

In this section, we select several typical few-shot learning methods as the baselines to evaluate the effectiveness of our TPN model.

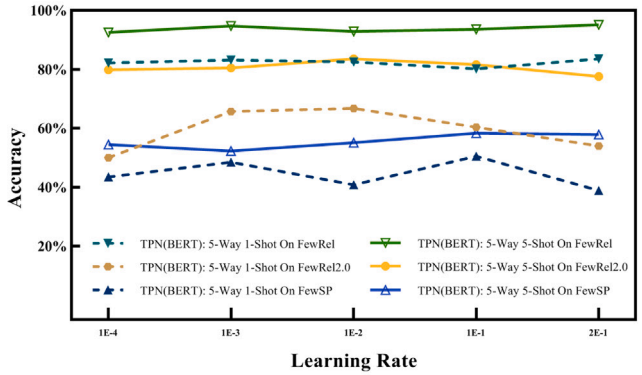
- GNN (Satorras et al. 2017): The method considers the support examples and query examples as the nodes in the graph, and the label of each support node is embedded into the node representations (Satorras & Estrach, 2018).
- Siamese Network(Koch et al. 2015): A method rank similarity between inputs. The model evaluates the similarity of two unlabeled inputs (Koch et al., 2015).
- Original Prototypical Network (Snell et al. 2017): It assumes there is a prototype for each relation, and the classifier works by computing the distances between the query and the prototype of each class. In the naive prototypical network, the prototype of each relation is the average of supports that have the same relation (Snell et al., 2017).
- Hybrid Attention-based Prototypical Network (Gao et al. 2019): The work is based on prototypical network, and it uses hybrid attention to address the diversity and noise of text (Gao, Han, Liu et al., 2019).
- BERT-PAIR (Gao et al. 2019): The model pairs each example in the query set with support examples, and then utilizes the BERT sequence classification model to determine whether the two examples have the same label (Gao, Han, Zhu et al., 2019).

4.5. Pre-trained model

To further improve our TPN model, we investigate the popular pre-trained model BERT as the encoder of our model, which produces the representations of the example embedding. BERT (Devlin et al. 2018) is a pre-trained language model that has been trained on a large and general domain dataset, and researchers utilize it as a benchmark to make the downstream tasks come true. BERT is the bidirectional encoder representation from transformers, and it is the first model that is able to gain bidirectional information at the same time. Fine-tuning on the pre-trained BERT has achieved start-of-the-art performances in many tasks (Devlin et al., 2018; Rothe et al., 2020). In our proposed model TPN(BERT), we use the uncased $BERT_{base}$. Its input is a 768 embedding vector, and it has 12 layers, each of which has 12 attention heads.



(a) TPN



(b) TPN(BERT)

Fig. 6. Accuracy with the increase of learning rate LR on the three datasets.

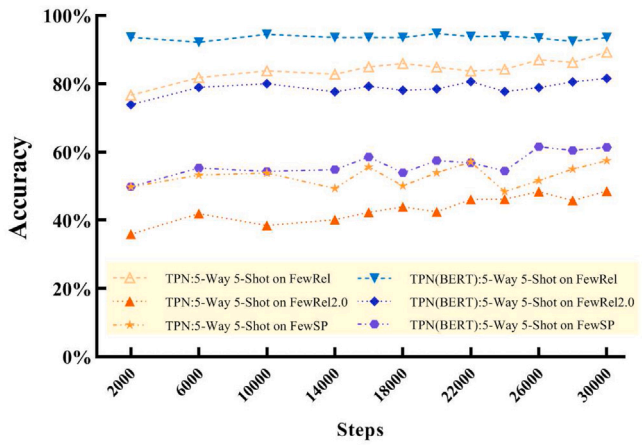


Fig. 7. Test accuracy curve with the number of steps changed on 5-way 5-shot task.

4.6. Experimental result and discussion

In this part, we show the comparison results between our proposed method and the typical approaches under the same hyper-parameters which are given in Table 2.

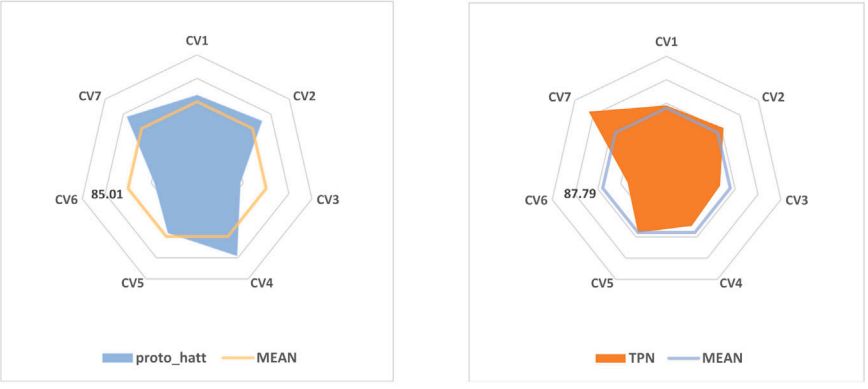
Table 4
Overall results.

Task		Hybrid Attention-Based Prototypical Network	TPN	TPN(BERT)
In domain (FewRel)	5-Way 1-Shot	71.93	73.69	80.14
	5-Way 5-Shot	85.80	89.24	93.60
	5-Way 10-Shot	89.42	90.29	95.08
	10-Way 1-Shot	61.91	63.07	72.67
	10-Way 5-Shot	79.84	80.31	89.83
Cross domain (FewRel2.0)	5-Way 1-Shot	35.08	38.32	60.35
	5-Way 5-Shot	47.03	48.55	81.60
	5-Way 10-Shot	51.57	53.29	83.77
	10-Way 1-Shot	25.89	26.74	38.12
	10-Way 5-Shot	35.20	35.40	76.91
Cross-sentence (FewSP)	5-Way 1-Shot	41.78	45.70	50.50
	5-Way 5-Shot	52.47	57.48	60.63
	5-Way 10-Shot	57.39	43.17@	60.09@
	10-Way 1-Shot	-	-	-
	10-Way 5-Shot	-	-	-

'-': The number of relations is not enough.
'@': Dropout increases to 0.9 or loss will explode, which makes the accuracy drop off.

Table 5
The comparison results on FewRel.

Model	5-Way 1-Shot	5-Way 5-Shot
GNN	61.41	76.16
Siamese Network	67.67	78.14
Original Prototypical Network	74.57	87.16
Hybrid Attention-Based Prototypical Network	71.93	85.44
BERT-PAIR	88.12	91.13
TPN	**73.69**	**89.24**
TPN(BERT)	**80.14**	**93.60**



(a) Cross validation on hybrid attention-based prototypical network.

(b) Cross validation on TPN.

Fig. 8. Radar plots on cross validation.

The overall results are shown in Table 4. For the three scenarios (in domain, cross domain, and cross-sentence few-shot relation extraction), the TPN model has over 1% improvement. Simultaneously, TPN(BERT) achieves approximately 8% improvement for different tasks than the hybrid attention-based prototypical network, a state-of-the-art model in few-shot learning methods. In Fig. 8, we present the results of cross validation. Our proposed model TPN can improve 2.5% on average. The relations are randomly selected for train and validation set on FewRel dataset.

First of all, we evaluate on the FewRel dataset, and the results are shown in Table 5. From the table, it can be seen that our proposed model outperforms most of the existing state-of-art baselines. Although BERT-PAIT sometimes achieves higher accuracy than our model, BERT-PAIR depends entirely on BERT and works by concatenating each support example to each query example, which consumes very large amounts of memory with high time complexity. Thus it is unfriendly to most researchers. Table 6 presents

Table 6
Comparison of FLOPs.

Model	FLOPs
Proto_Hatt	0.648G
TPN	0.670G
BERT-PAIR	6795.142G
TPN(BERT)	544.584G

'Proto_Hatt': Hybrid Attention-Based Prototypical Network.

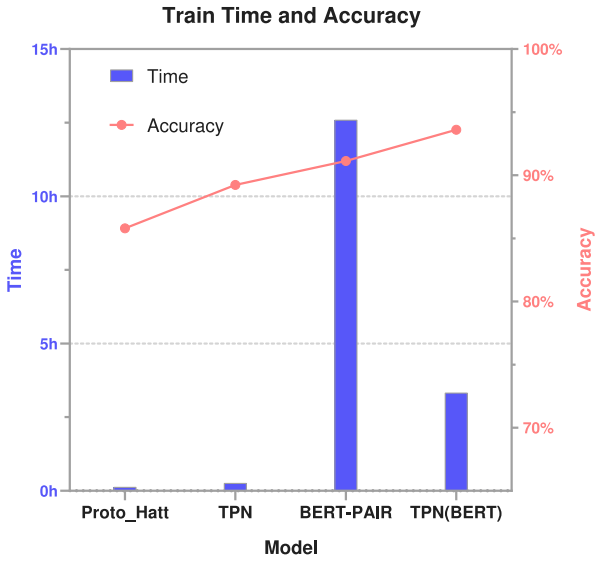


Fig. 9. The comparison of time and accuracy on 5-way 5-shot task. The blue bar is the time which consumes in the train, and the crease line is the accuracy.

the FLOPs of the four competitive models: Hybrid Attention-based Prototypical Network, BERT-PAIR, TPN, TPN(BERT), and divide them into two parts (with or without BERT) to compare. To show more clear, in Fig. 9, we compare the train time and the test accuracy of them. Compared with the hybrid attention-based prototypical network, our proposed model TPN improves 3.44% with a similar time complexity, and our proposed model TPN(BERT) achieves higher accuracy. Moreover, the performance of TPN(BERT) is competitive with BERT-PAIR while it only requires 1/5 time of BERT-PAIR, because TPN(BERT) utilizes only one BERT layer as the pre-trained model to encode the examples. As a result, our model TPN\TPN(BERT) achieves a trade-off between effectiveness and efficiency, which outperforms BERT-PAIR, and is more likely to recur and further improve the performance. Thus, we compare the results of different models, except BERT-PAIR.

For the 5-way 1-shot task, the accuracy for the TPN method achieves 73.69%, and TPN(BERT) achieves 80.14%, which obtains 1.76% and 8.21% improvements when compared to the SOTA prototypical network. For the 5-way 5-shot task, the accuracy reaches 89.24% and 93.60% for TPN and TPN(BERT), which has 3.80% and 8.16% improvement from the existing highest score(except BERT-PAIR). The result means that our proposed model manages to correctly choose more informative features to compose a more confident distance metric.

In addition, we present the results of each relation in Fig. 10. We find that the accuracy balance for every relation, instead of only contributed by one. As we can see, our proposed model outperforms the baselines at most relations. Moreover, we notice that in Fig. 10, the P4552 (mountain range) relation advances most significantly. As Figs. 1(a) and 1(b) respectively show, the hybrid attention-based prototypical network mixes the P4552 (mountain range) and P59 (constellation) relations with a higher probability than our proposed TPN model. To concretely demonstrate the results, we randomly choose 120 examples from the Fewrel1.0 test dataset and count the incorrect labels, which are presented in Table 7. As shown in this table, we find that the number of incorrect labels in our proposed TPN model is lower than that in the baseline model for both the total number and the single relation. Then, we choose some examples that come from the P4552 relation, as shown in Fig. 11. The hybrid attention-based prototypical network predicts the label of the two examples as P59 (constellation), while our proposed TPN model can obtain the correct label.

In Fig. 12, we present the visualization analysis of the first example vector and the prototype of the correct relation, both of which have 512 dimensions. We compare the difference between these results to show the advancement of our feature extractor. As the figure shows, in the hybrid attention-based prototypical network, there is a large difference between the predicted representation and the prototype representation, while in our proposed TPN model, the predicted vector of the query example is similar to the prototype representation. Therefore, the experiment demonstrates that our proposed TPN model, which integrates the transformer, has a more powerful feature extraction ability.

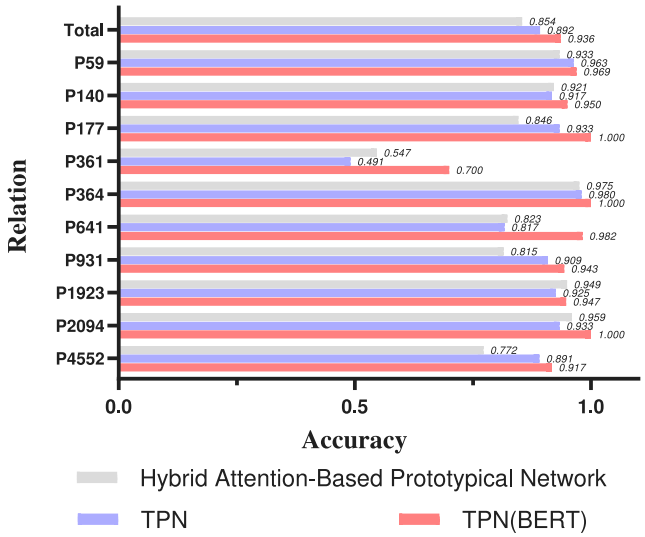


Fig. 10. Accuracy of each relation in test data. P59 is "constellation" relation, P140 is "religion" relation, P177 is "crosses" relation, P361 is "part of" relation, P364 is "original language of film or TV show" relation, P641 is "sport" relation, P931 is "place served by transport hub" relation, P1923 is "participating team" relation, P2094 is "competition class" relation, and P4552 is "mountain range" relation.

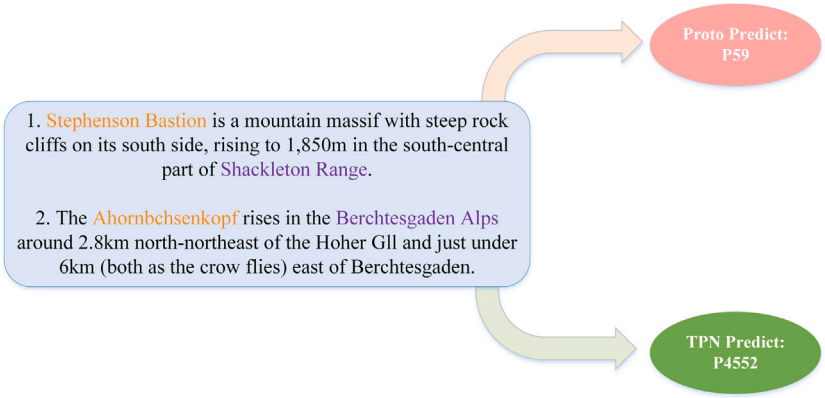


Fig. 11. Examples of P4552 (mountain range) relation.

Table 7
Wrong label statistics.

Predict Label	P59	P140	P177	P361	P931	P2094	total
Proto	11	2	16	1	6	1	37
TPN	7	0	13	3	5	0	28

* "Proto" is the hybrid attention-based prototypical network.

Table 8
The comparison results on FewRel2.0.

Model	5-Way 1-Shot	5-Way 5-Shot
GNN	24.94	42.67
Siamese Network	34.68	41.10
Original Prototypical Network	40.79	45.35
Hybrid Attention-Based Prototypical Network	35.09	47.03
BERT-PAIR	69.86	75.71
TPN	**38.32**	**48.55**
TPN(BERT)	**60.35**	**81.60**

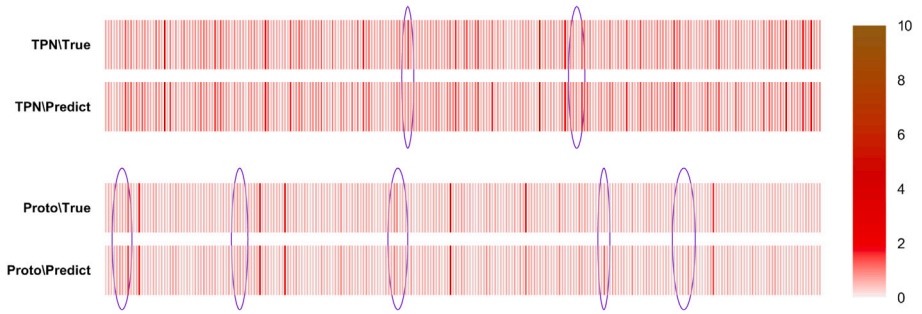


Fig. 12. Example visualization. "Proto" is the hybrid attention-based prototypical network. "True" means the prototype vector of true relation (P4552), "Predict" means the embedding of the first example in Fig. 11. The places where have been marked with purple ellipses are the differences between "True" and "Predict".

Table 9
The comparison results on FewSP.

Model	5-Way 1-Shot	5-Way 5-Shot
GNN	39.99	40.91
Siamese Network	40.01	43.36
Original Prototypical Network	45.28	53.82
Hybrid Attention-Based Prototypical Network	41.78	52.47
BERT-PAIR	38.94	39.93
TPN	**45.70**	**57.48**
TPN(BERT)	**50.50**	**60.63**

To further demonstrate the effectiveness of our method on the domain adaptation (DA) task, we use our model and the baselines on the FewRel2.0 dataset, and the results are shown in Table 8. As Table 8 shows, the accuracy of our TPN method has over 3% improvement for 5-way 1-shot and 1.5% for 5-way 5-shot. TPN(BERT) improves over 20% for both 5-way 1-shot and 5-way 5-shot tasks. The results demonstrate that extracting more significant feature dimensions has great effectiveness in improving the performance of the model when considering the cross-domain; that is, the feature extractor of the baselines is not powerful enough for cross-domain scenario and our work is meaningful.

Finally, to test the ability of crossing-sentence relation extraction, we evaluate our model on the FewSP dataset and as Table 9 reports that the accuracy increases over 5% for both 5-way 1-shot and 5-way 5-shot settings compared to the result of the baselines. In addition, we find that BERT-PAIR is not good at extracting cross-sentence relations, while our model greatly improves the performance in this scenario. BERT-PAIR pairs each query with support examples and then predicts the query label by discriminating whether the two examples have the same label, however, the max-length setting limits the length of example embedding that is produced by query add support, and it will lose much semantic information, which directly influences the performance.

From the discussion above, we notice that our proposed TPN/TPN(BERT) model can effectively improve the performance on all three datasets, and it would be reasonable to assume that the performance of our TPN model would further develop with larger GPU memory.

Effect of the transformer in our TPN model. To demonstrate the effect of the transformer model used in our proposed model, we evaluate on FewRel. The test data have ten relations, and we choose five relations (P140, P361, P364, P641, P931, P1923) to show the effect. We present the average relative distances between the query point and the prototype of each relation in Fig. 13. The red color means that the query and prototype are close, while the blue color demonstrates that the query is far away from the prototype. As Fig. 13 shows, in our proposed model, the distance between query and wrong relation prototype is farther compared to our state-of-the-art baseline hybrid attention-based prototypical network, which demonstrates that our model extracts more discriminative features and has more confidence in determining the correct relation label of query. Thus, we can believe that the transformer layer in our model TPN/TPN(BERT) plays an important role in relation-level feature extraction.

Effect of the number of transformer blocks and attention heads in the transformer. Transformer models usually have multiple blocks with multi-head attention, which is more fine-grained than single blocks. However, it is questionable if the number of blocks and attention heads are properly limited by our hardware. First, we test on our server and find that only one transformer layer can be set at most without reporting an error. Second, we evaluate our model on different transformer shapes. The result of different numbers of transformer blocks is shown in Fig. 14(a), while the result of heads in attention is shown in Fig. 14(b). The two figures demonstrate that both the number of blocks and attention heads affect the ability of the transformer to model long-range dependencies and extract semantic features. Multi-head methods are able to focus on different parts of context and then acquire a better representation than single-head methods. Some heads or blocks using more model parameters and are able to advance the feature learning ability of the model. However, the dataset of few-shot learning, including our Fewrel1.0, Fewrel2.0 and FewSP datasets, is not large enough to make the parameters learn fully. As we can see in Figs. 14(a) and 14(b), the accuracy decreases dramatically when there are over 3

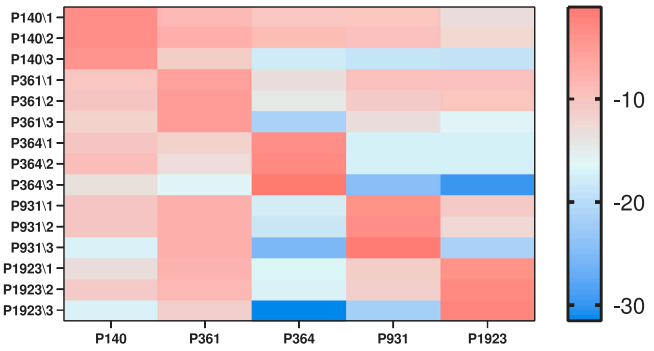
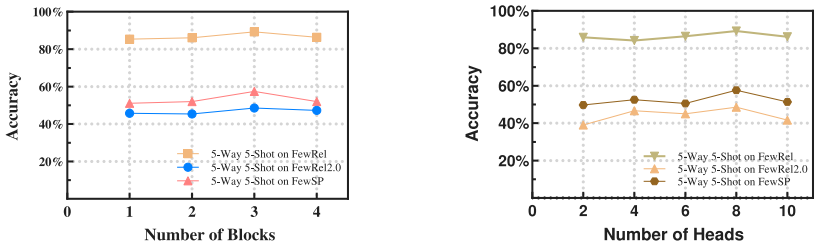


Fig. 13. Visualization of relative distance between query instance and prototype in each classes. "1" refer to the Hybrid Attention-based Prototypical Network, "2" refer to our model TPN, "3" refer to TPN(BERT). The horizontal axis represents the prototype of each class, while the vertical axis is the relation label of query instance. The color in red, the closer between the query and the prototype, and the more likely query instance will be labeled.



(a) Effect of the number of transformer blocks.

(b) Effect of the number of attention heads in transformer.

Fig. 14. Effect of the number of blocks and attention heads in the transformer layer on the 5-way 5-shot task.

blocks or 8 attention heads. Excess attention heads and transformer blocks increase the complexity of our model which wastes the resources, and also causes over-saturated parameters that aggravate the overfitting problem and causes the model to not generalize to new tasks well, especially for the cross-domain dataset Fewrel2.0. In addition, improper numbers of heads may cause the quality to drop off because some heads may pay more attention to some rare words, and then cause the overhigh weights of these rare words, which is the noise for our model. As a result, we find that utilizing 3 stacked transformer blocks with 8-head attention as the feature extractor of our proposed model is able to achieve the best performance under our environmental settings (Domhan, 2018; Voita et al., 2019).

4.7. Summary

From the above experiments, we analyze how the number of transformer blocks and attention heads affects the performance of our proposed TPN/ TPN(BERT) model, and demonstrate that our proposed model TPN/ TPN(BERT) can achieve a competitive result in few-shot relation extraction tasks, and extreme shortens the time of training compared with the strongest baseline, especially for cross-sentence relation extraction situations. GNN (Satorras & Estrach, 2018) achieves the worst performance. It is not suitable for few-shot datasets, which can only see a small number of examples in a batch, because the number of parameters is very large in GNN and cannot learn sufficiently, especially for the cross-domain dataset Fewrel2.0. Siamese network (Koch et al., 2015) only compares the similarity of two unlabeled inputs at one time, which is too slow to converge and is heavily affected by noise, while our TPN/TPN(BERT) model handles the input parallel and trains a batch at one time, which is highly efficient and learns a weight matrix of support examples that can alleviate the noise problem. The original prototypical network (Snell et al., 2017) averages the examples to obtain the prototype of each relation and uses the naive Euclidean distance function. Since the prototypical network works by computing the distance between the query and the prototype, the positions of the prototypes are deeply influenced by the noise points. The hybrid attention-based prototypical network (Gao, Han, Liu et al., 2019) not only considers the different contributions of support examples for different queries but also utilizes a dynamic Euclidean distance that is able to select the more discriminative features. However, when compared with our proposed TPN/TPN(BERT) model, the hybrid attention-based prototypical network uses CNN as the feature extractor which is not strong enough and cannot capture the long-range dependencies. Moreover, the transformer feature extractor in our proposed model handles the input parallel which requires less time, and has more learnable parameters to gain more features which alleviates the overfitting problem. Even though BERT-PAIR can obtain competitive accuracy, its performance drops off dramatically when applied to cross-sentence relation extraction tasks, while our proposed TPN/TPN(BERT) model achieves encouraging results.

5. Conclusions

In this paper we propose a few-shot relation extraction TPN model, which integrates transformer architecture into a prototypical network. While most existing work uses the CNN model to extract features, our model mainly depends on the multi-head attention mechanism in the transformer to achieve better feature extraction. Then we find that the 3-block 8-head transformer achieves the best performance for this task with hardware limitations. We evaluate the TPN model and utilize the pre-trained BERT to further improve on three benchmarks, demonstrating that our model is able to achieve a trade-off between performance and computation with capable of state-of-the-art performance on the few-shot relation extraction task. In future work, we may concentrate on utilizing the co-dependence entities in a paragraph to improve the performance of the crossing-sentence relation extraction task.

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.

Acknowledgments

This work is supported by 973 Program (No. 2014CB340504), the State Key Program of National Natural Science of China (No. 61533018), National Natural Science Foundation of China (No. 61402220), the Philosophy and Social Science Foundation of Hunan Province, China (No. 16YBA323), Science and Technology Support Program, China (No. 2014BAK04B00), Natural Science Foundation of Hunan Province, China (No. 2020JJ4525) and the Scientific Research Fund of Hunan Provincial Education Department, China (No. 18B279 and No. 19A439).

References

Baktashmotlagh, Mahsa, Harandi, Mehrtash T., Lovell, Brian C., & Salzmann, Mathieu (2013). Unsupervised domain adaptation by domain invariant projection. In *2013 IEEE international conference on computer vision* (pp. 769–776). IEEE.

Chen, Mia Xu, Firat, Orhan, Bapna, Ankur, Johnson, Melvin, Macherey, Wolfgang, Foster, George, Jones, Llion, Schuster, Mike, Shazeer, Noam, Parmar, Niki, et al. (2018). The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 76–86).

Chung, Tonglee, Xu, Bin, Liu, Yongbin, Ouyang, Chunping, Li, Siliang, & Luo, Lingyun (2019). Empirical study on character level neural network classifier for chinese text. *Engineering Applications of Artificial Intelligence*, *80*, 1–7.

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

Devos, Arnout, & Grossglauser, Matthias (2019). Subspace networks for few-shot classification. arXiv preprint arXiv:1905.13613.

Domhan, Tobias (2018). How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 1799–1808).

Elsken, Thomas, Staffler, Benedikt, Metzen, Jan Hendrik, & Hutter, Frank (2020). Meta-learning of neural architectures for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12365–12375).

Finn, Chelsea, Abbeel, Pieter, & Levine, Sergey (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1126–1135). JMLR.org.

Gao, Tianyu, Han, Xu, Liu, Zhiyuan, & Sun, Maosong 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI conference on artificial intelligence, volume 33* (pp. 6407–6414).

Gao, Tianyu, Han, Xu, Zhu, Hao, Liu, Zhiyuan, Li, Peng, Sun, Maosong, & Zhou, Jie 2019. 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 (EMNLP-IJCNLP)* (pp. 6251–6256).

González, José Ángel, Hurtado, Lluís-F., & Pla, Ferran (2020). Transformer based contextualization of pre-trained word embeddings for irony detection in twitter. *Information Processing & Management*, *57*(4), Article 102262.

Han, Xu, Zhu, Hao, Yu, Pengfei, Wang, Ziyun, Yao, Yuan, Liu, Zhiyuan, & Sun, Maosong (2018). 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* (pp. 4803–4809).

Hendrickx, Iris, Kim, Su Nam, Kozareva, Zornitsa, Nakov, Preslav, Séaghdha, Diarmuid O., Padó, Sebastian, Pennacchiotti, Marco, Romano, Lorenza, & Szpakowicz, Stan (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 33–38). Association for Computational Linguistics.

Hu, Ziniu, Chen, Ting, Chang, Kaiwei, & Sun, Yizhou (2019). Few-shot representation learning for out-of-vocabulary words. Meeting of the association for computational linguistics (pp. 4102–4112).

Ji, Zhong, Chai, Xingliang, Yu, Yunlong, Pang, Yanwei, & Zhang, Zhongfei (2020). Improved prototypical networks for few-shot learning. *Pattern Recognition Letters*.

Jin, Hailong, Li, Chengjiang, Zhang, Jing, Hou, Lei, Li, Juanzi, & Zhang, Peng (2019). Xlore2: Large-scale cross-lingual knowledge graph construction and application. *Data Intelligence*, *1*(1), 77–98.

Koch, Gregory, Zemel, Richard, & Salakhutdinov, Ruslan (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop, volume 2. Lille*.

Liu, Yongbin, Ouyang, Chunping, & Li, Juanzi (2017). Ensemble method to joint inference for knowledge extraction. *Expert Systems with Applications*, *83*, 114–121.

Luo, Bingfeng, Feng, Yansong, Wang, Zheng, Zhu, Zhanxing, Huang, Songfang, Yan, Rui, & Zhao, Dongyan (2017). Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 430–439).

Mintz, Mike, Bills, Steven, Snow, Rion, & Jurafsky, Dan (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP: Volume 2-volume 2* (pp. 1003–1011). Association for Computational Linguistics.

Munkhdalai, Tsendsuren, & Yu, Hong (2017). Meta networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 2554–2563). JMLR.org.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Qu, Meng, Gao, Tianyu, Xhonneux, Louis-Pascal, & Tang, Jian (2020). Few-shot relation extraction via bayesian meta-learning on relation graphs. In *International conference on machine learning* (pp. 7867–7876). PMLR.

Ravi, Sachin, & Larochelle, Hugo (2017). Optimization as a model for few-shot learning.

Ren, Haopeng, Cai, Yi, Chen, Xiaofeng, Wang, Guohua, & Li, Qing (2020). A two-phase prototypical network model for incremental few-shot relation classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1618–1629).

Ringer, Sam, Williams, Will, Ash, Tom, Francis, Remi, & MacLeod, David (2019). Texture bias of cnns limits few-shot classification performance. arXiv preprint arXiv:1910.08519.

Rothe, Sascha, Narayan, Shashi, & Severyn, Aliaksei (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, *8*, 264–280.

Satorras, Victor Garcia, & Estrach, Joan Bruna (2018). Few-shot learning with graph neural networks. In *International conference on learning representations*.

Snell, Jake, Swersky, Kevin, & Zemel, Richard (2017). Prototypical networks for few-shot learning. In *Advances in neural information processing systems* (pp. 4077–4087).

Sung, Flood, Yang, Yongxin, Zhang, Li, Xiang, Tao, Torr, Philip H. S., & Hospedales, Timothy M. 2018a. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1199–1208).

Sung, Flood, Yang, Yongxin, Zhang, Li, Xiang, Tao, Torr, Philip H. S., & Hospedales, Timothy M 2018b. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1199–1208).

Tang, Gongbo, Müller, Mathias, Gonzales, Annette Rios, & Sennrich, Rico (2018). Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4263–4272).

Vanschoren, Joaquin (2018). Meta-learning: A survey. arXiv preprint arXiv:1810.03548.

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

Vig, Jesse (2019). A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th annual meeting of the association for computational linguistics: System demonstrations* (pp. 37–42).

Vinyals, Oriol, Blundell, Charles, Lillicrap, Timothy, Wierstra, Daan, et al. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems* (pp. 3630–3638).

Vo, Ducthuan, & Bagheri, Ebrahim (2019). Feature-enriched matrix factorization for relation extraction. *Information Processing and Management*, *56*(3), 424–444.

Voita, Elena, Talbot, David, Moiseev, Fedor, Sennrich, Rico, & Titov, Ivan (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5797–5808).

Vrandečić, Denny, & Krötzsch, Markus (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85.

Wan, Huaiyu, Zhang, Yutao, Zhang, Jing, & Tang, Jie (2019). Aminer: Search and mining of academic social networks. *Data Intelligence*, *1*(1), 58–76.

Wang, Qiang, Li, Bei, Xiao, Tong, Zhu, Jingbo, Li, Changliang, Wong, Derek F., & Chao, Lidia S. (2019). Learning deep transformer models for machine translation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1810–1822).

Wang, Yaqing, & Yao, Quanming (2019). Few-shot learning: A survey. arXiv preprint arXiv:1904.05046.

Wang, Yaqing, Yao, Quanming, Kwok, James T., & Ni, Lionel M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, *53*(3), 1–34.

Xie, Yuxiang, Xu, Hua, Li, Jiaoe, Yang, Congcong, & Gao, Kai (2020). Heterogeneous graph neural networks for noisy few-shot relation classification. *Knowledge-Based Systems*, Article 105548.

Ye, Zhi-Xiu, & Ling, Zhen-Hua (2019). Multi-level matching and aggregation network for few-shot relation classification. arXiv preprint arXiv:1906.06678.

Ye, Hai, & Luo, Zhunchen (2019). Deep ranking based cost-sensitive multi-label learning for distant supervision relation extraction. *Information Processing and Management*, Article 102096.

Yoon, Jaesik, Kim, Taesup, Dia, Ousmane, Kim, Sungwoong, Bengio, Yoshua, & Ahn, Sungjin (2018). Bayesian model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, *31*, 7332–7342.

Zeng, Xiangrong, He, Shizhu, Liu, Kang, & Zhao, Jun (2018). Large scaled relation extraction with reinforcement learning. In *AAAI* (pp. 5658–5665).

Zeng, Daojian, Liu, Kang, Chen, Yubo, & Zhao, Jun (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1753–1762).

Zeng, Daojian, Liu, Kang, Lai, Siwei, Zhou, Guangyou, & Zhao, Jun (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 2335–2344).

Zhang, Tongtao, Ji, Heng, & Sil, Avirup (2019). Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, *1*, 99–120.

Zhao, An, Ding, Mingyu, Lu, Zhiwu, Xiang, Tao, Niu, Yulei, Guan, Jiechao, Wen, Ji-Rong, & Luo, Ping (2020). Domain-adaptive few-shot learning. arXiv preprint arXiv:2003.08626.