From Learning-to-Match to Learning-to-Discriminate: Global Prototype Learning for Few-shot Relation Classification

Fangchao Liu[1,4,*], **Xinyan Xiao**[3], **Lingyong Yan**[1,4], **Hongyu Lin**[1], **Xianpei Han**[1,2,†], **Dai Dai**[3], **Hua Wu**[3], **Le Sun**[1,2]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]Baidu Inc., Beijing, China
[4]University of Chinese Academy of Sciences, Beijing, China
{fangchao2017,lingyong2014,hongyu,xianpei,sunle}@iscas.ac.cn

Abstract

Few-shot relation classification has attracted great attention recently, and is regarded as an effective way to tackle the long-tail problem in relation classification. Most previous works on few-shot relation classification are based on learning-to-match paradigms, which focus on learning an effective universal matcher between the query and *one* target class prototype based on inner-class support sets. However, the learning-to-match paradigm focuses on capturing the similarity knowledge between query and class prototype, while fails to consider discriminative information between different candidate classes. Such information is critical especially when target classes are highly confusing and domain shifting exists between training and testing phases. In this paper, we propose the *Global Transformed Prototypical Networks(GTPN)*, which learns to build a few-shot model to directly discriminate between the query and *all* target classes with both inner-class local information and inter-class global information. Such learning-to-discriminate paradigm can make the model concentrate more on the discriminative knowledge between all candidate classes, and therefore leads to better classification performance. We conducted experiments on standard FewRel benchmarks. Experimental results show that GTPN achieves very competitive performance on few-shot relation classification and reached the best performance on the official leaderboard of FewRel 2.0 [1].

Introduction

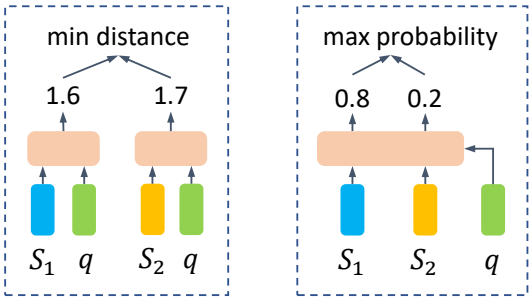
Few-shot relation classification aims to build relation extractors with only a few instances. Different from supervised learning that requires large scale training data of target classes, few-shot learning-based approaches can effectively build relation extractors with only a few examples (i.e., the support set) of each target class. This property makes it very appealing in the real application, where the training data of new target class can be scarce.

Previous few-shot relation classification approaches can be summarized into a learning-to-match paradigm. Specifically, as shown in Figure 1 (a), these methods try to learn a universal matcher between the query and *each* target relation type, and measure their similarity separately to infer the type of queries. Along this line, Gao (2019a) propose the hybrid attention to attach different importance for each relation feature, and match the query to each relation type with the reweighted features. Ye (2019) performs a local matching and aggregation between the query instance and each relation type. Besides, Gao (2019b) proposes an instance-pair matcher for scoring the similarity between query and each relation instance. Generally, the goal of these methods is to effectively obtain a query-class matcher with the limited given instances in the support set.

However, because such learning-to-match based approaches focus on capturing the similarity information between query and classes, they are unable to consider discriminative knowledge between different candidate classes. This can significantly undermine the model performance when target classes are highly confusing, and domain shifting exists between training and testing phases. For example, a query

[*]Part of the work was done during an internship at Baidu.
[†]Corresponding authors.
[1]https://thunlp.github.io/2/fewrel2_da.html

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.



(a) Learning-to-Match **(b)** Learning-to-Discriminate

Figure 1: Illustration of the motivation for our Global Transformed Prototypical Networks (GTPN). (a) is the previous learn-to-match paradigm method, that matches the query instance q to *each* relation support set S_1 and S_2, and measures the similarity seperately. (b) is the learn-to-discriminate paradigm of GTPN, that directly discriminate the query instance q on *all* relation support sets, and give the normalized scores on each relation type for further learning of the prototypes.

with relation mention word "contain" may correspond to either a "Component-Whole" relation or a "Member-Collection" relation. Therefore, under the learning-to-match paradigm, this query will achieve high matching scores with both two relation types, which may lead to confusion and misclassification when we determine the instance relation type. Furthermore, the existence of domain shifting can result in the inaccuracy of the similarity measurement on the out-of-domain relation. To tackle these problems, a model needs to take discriminative information into consideration, and focus more on how to distinguish between confusing target relations.

To this end, this paper proposes to resolve few-shot relation classification in a learning-to-discriminate paradigm. The main idea behind, as shown in Figure 1 (b), is to learn a meta classification model which can directly generate a relation classifier based on the query and small support sets of *all* candidate relations, rather than a meta matcher between the query and *one* relation. Motivated by this, we design the *Global Transformed Prototypical Networks* (GTPN), which takes the query and the support sets of all target relations as input, and output the probabilities of the input query correspond to each relation simultaneously. The architecture of GTPN is shown in Figure 2. Specifically, GTPN first encodes the query and all relation instances in the support sets into the same embedding space. Then we conducted a multi-view global transformation on each relation instance to extract features of different facets, based on the knowledge from both intra-relation and inter-relation instances. After that, all representations of instances of the same relation are summarized to form the class prototypes. Finally, these prototypes, as well as the query representation, are simultaneously sent into a classifier to determine the relation type of the input query. The main advantage of GTPN, compared with previous approaches, is that all candidate relation types are considered jointly rather than independently during relation classification, which makes the model able to leverage discriminative information between classes to more preciously distinguish between confusing relation pairs.

To verify the effectiveness of GTPN, we conduct thorough experiments on FewRel 1.0 (Han et al., 2018) and FewRel 2.0 (Gao et al., 2019b), two standard benchmarks for few-shot relation classification. Experimental results demonstrate that GTPN can achieve effective and robust performance on few-shot relation classification, even domain shifting exists. Furthermore, GTPN reaches the best performance on the FewRel 2.0 official leaderboard. These all demonstrate the effectiveness of GTPN and the proposed learning-to-discriminate paradigm.

Generally, the main contributions of this paper can be summarized as:

- We propose to resolve few-shot relation classification in a learning-to-discriminate paradigm, which is able to leverage discriminative knowledge for better distinguishing between confusing type pairs, compared with previous learning-to-match paradigm.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

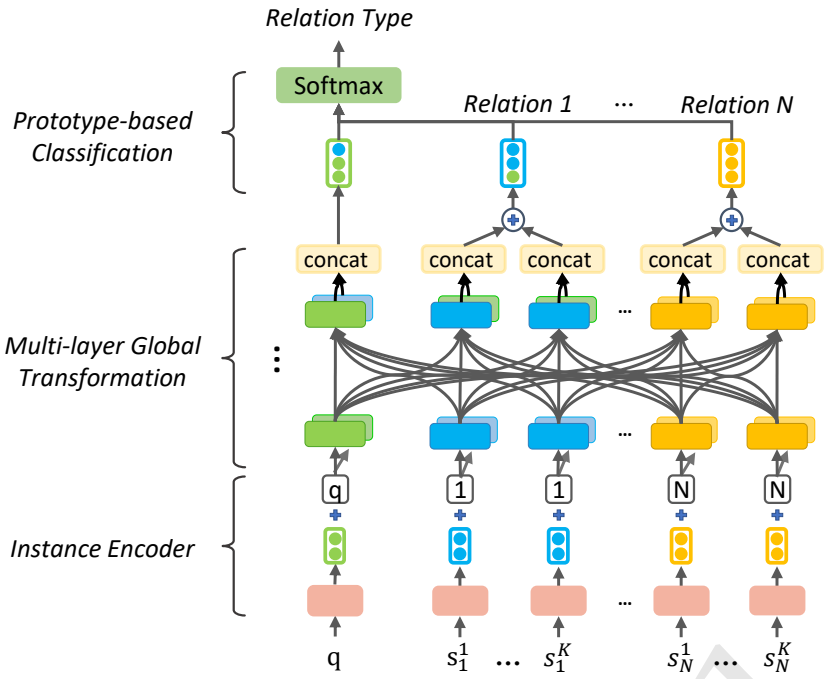


Figure 2: Framework of GTPN. q is the query instance, S_i^j means the j^{th} support instance of relation i.

- Based on the paradigm, we propose the Global Transformed Prototypical Networks (GTPN), an effective neural network-based architecture with global transformation for few-shot relation classification.
- The proposed GTPN achieved the new state-of-the-art performance on the official leaderboard of FewRel 2.0, which demonstrates the effectiveness of the proposed paradigm and architecture.

Background

This section describes the definition and the notations of few-shot relation classification task. Besides, we will briefly illustrate previous learning-to-match based approaches.

Relation Classification. Let $\mathcal{X} = [x_0, ..., x_{n-1}]$ be the sequence which contains n words, $e_1 = [i, j]$ and $e_2 = [k, l]$ indicate the entity pair spans, where $0 \leq i \leq j, j < k \leq l$ and $l \leq n-1$, a relation instance is defined as $(\mathcal{X}, e_1, e_2)$. For example, (*"Steve Jobs was the co-founder of Apple Inc."*, *"Steve Jobs"*, *"Apple Inc."*) is a relation instance. The aim of relation classification is to learn a mapping function: $f : r \rightarrow c$, where c is the relation class. For example, we want mapping (*"Steve Jobs was the co-founder of Apple Inc."*, *"Steve Jobs"*, *"Apple Inc."*) to its relation class *"Founder-of"*.

Few-Shot Relation Classification. Few-shot relation classification task contains a support set S and a query instance q, where $S = \{S_1, .., S_N\}$ are support sets for N relation classes and each

$$S_i = \{s_i^1, ..., s_i^K\}$$

contains K support relation instances for relation i. Then the query instance q is to be classified based on the support relation instances in S. Such a few-shot setting is commonly named as an **N-way K-shot** relation classification.

Learning-to-match based Methods. Learning-to-match based methods regard few-shot relation classification as a matching problem(Vinyals et al., 2016; Snell et al., 2017). Generally, it encodes each relation class into a prototype, which is learned from its support set S_i:

$$\mathcal{P}_{S_i} = f(S_i) \in \mathcal{R}^{d_k},$$

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

where d_k is the dimension of the metric space. Then a matcher is learned to measure the similarity between each prototype and the query. For classification, it first encodes the query instance to the same metric space:

$$\mathcal{P}_q = g(q) \in \mathcal{R}^{d_k}.$$

Then the matcher computes the distances between the query instance and each prototype of different relation classes, and the relation class of the query instance is inferred by the nearest distance:

$$y = \arg\min_i \mathcal{D}(\mathcal{P}_{S_i}, \mathcal{P}_q), \tag{1}$$

where y is the label of the nearest relation class prototype, $\mathcal{D}(.)$ is the distance function to measure the similarity between support set and query instance.

The main drawback of such learning-to-match approaches is that the prototype only contains the information from the support instances of its own class. This results in the absence of discriminative knowledge between different candidate classes, which can significantly undermine the performance on confusing relation type pairs. Besides, the inaccurate measurement of similarities on out-of-domain relations can also lead to a negative impact on system performance.

Global Transformed Prototypical Networks

In this section, we introduce our Global Transformed Prototypical Networks (GTPN) for few-shot relation classification. Different from previous work, GTPN follows a learning-to-discriminate paradigm, which directly takes the query and support sets of all classes as input. Figure 2 shows the framework of our method. First, GTPN encodes the query and instances in support sets into a hidden vector space via an instance encoder, as well as a relation marker, which represents whether two support instances belong to the same relation support set. These embeddings are then all synchronously sent to multi-view global transformation layers to learn the discriminative information based on both intra-relation and inter-relation knowledge. Then, we summarize the representations of all instances of the same relation to generate the prototypes of each class. Finally, the query representation and all prototypes are sent into a classifier to predict the relation type. In the following, we will describe each step of GTPN in detail.

Instance Encoder

The instance encoder module encodes each instance x into the same embedding space:

$$\mathbf{h} = \mathcal{H}(x), \ \ \mathbf{h} \in \mathcal{R}^{d_k}, \tag{2}$$

where d_k is the dimension of the embedding space, $\mathbf{h}$ is the embedding of each instance, which will be used for further prototype learning. Recent years, several instance encoders have been proposed, including Convolutional Neural Networks (CNN) (Zeng et al., 2014; Han et al., 2018); Recurrent Neural Networks (RNN) (Hochreiter and Schmidhuber, 1997; Zhou et al., 2016); and the recent Transformer architecture (Vaswani et al., 2017). Pre-trained language models like BERT (Devlin et al., 2019) also provide promising encoders for relation instances (Gao et al., 2019b; Baldini Soares et al., 2019). In this work, we adopt the BERT-based instance encoder similar to (Baldini Soares et al., 2019), which wraps entities in the instance with special markers [ENTITY] and [/ENTITY], and concatenate the representations of the first marker [ENTITY] of each entity as the instance embedding.

Relation Marker

Apart from the instance encoder, it is necessary to introduce relation markers to encode whether two instances belong to the same relation support set. To this end, we propose to use two kinds of relation markers for GTPN, including:

- **Randomly initialized relation encoding.** For each instance embedding $\mathbf{h}_i, 1 \leq i \leq N * K$ with relation type r_i, we randomly initialize the relation encoding embedding $\mathbf{h}_{r_i} \in \mathcal{R}^{d_k}$ for it, and add it with the relation encoding embedding: $\mathbf{h}_i = \mathbf{h}_i + \mathbf{h}_{r_i}$ as the new hidden states. For the query instance, we regard it as the instance of relation r_{N+1}: $\mathbf{h}_q = \mathbf{h}_q + \mathbf{h}_{r_{N+1}}$.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

- **One-hot relation encoding.** we use an one-hot vector to indicate each relation class, e.g., $[1, 0, ..., 0]$ for the first relation class and $[0, 1, 0, ..., 0]$ for the second relation class. For the query instance, we use the last one-hot vector to indicate the relation class.

The relation markers are directly concatenated to the embeddings from previous instance encoder to form the instance representations. Finally, we represent the output after instance encoder as:

$$\mathcal{X} = [\mathbf{h}_q, \mathbf{h}_{s_1^1}, ..., \mathbf{h}_{s_1^K}, ..., \mathbf{h}_{s_N^1}, ..., \mathbf{h}_{s_N^K}]. \tag{3}$$

Multi-view Global Transformation

After obtaining the representations of all support instances, we will conduct global transformation among them to learn the knowledge from both intra-relation and inter-relation support sets. This module is based on a multi-layer transformer model(Vaswani et al., 2017). Specifically, given the embeddings of all instances, each instance is firstly mapped into multiple views of semantics by a multi-view projection:

$$\mathbf{v}_j^i = \mathbf{W}_i \cdot \mathbf{h}_j + \mathbf{b}_i, \tag{4}$$

where $\mathbf{v}_j^i$ is the i^{th} view of the j^{th} instance, $\mathbf{W}_i$ is the mapping matrix and $\mathbf{b}_i$ is the bias for view i. Then the model will transform the representation of all instances of all relations from each view via an attention-based mechanism:

$$\begin{aligned} \mathbf{v}_j^i &= \sum_{m=0}^{N*K} \alpha_{jm} \cdot \mathbf{v}_m^i \\ \alpha_{jm} &= \text{Softmax}(\frac{\mathbf{v}_j^i \cdot \mathbf{v}_m^{i\ T}}{\sqrt{d_k}}), \end{aligned} \tag{5}$$

where α_{jm} is the normalized attention score between the j^{th} and m^{th} instances, the attention score is divided by dimension d_k to rescale the inner product of two vectors. After that, the output representation of each instance of this transformation layer is obtained by concatenating the representations in its all views. This output representation is then fed into next global transformation layer. We denote the output representation of instance S_i^j as $\mathbf{o}_i^j$. Furthermore, we use $\mathbf{h}_q$ to represent the final output representation of the given query.

Prototype-based Classification

After obtaining the global transformed instance representations, we calculate the prototype of each class by averaging the representations of all instances in its support set:

$$\mathbf{p}_i = \frac{1}{K} \sum_{k=1}^{K} \mathbf{o}_i^k, \tag{6}$$

where $\mathbf{p}_i$ is the prototype of relation i. Then we calculate the probability of the query being an instance of relation y_i by sending the query representation $\mathbf{h}_q$ and all prototypes into a softmax-based classifier:

$$\mathcal{P}(y_i|S_i, q; \theta) = \frac{\exp(-\mathcal{D}(\mathbf{p}_i, \mathbf{h}_q))}{\sum_{j=1}^{N} \exp(-\mathcal{D}(\mathbf{p}_j, \mathbf{h}_q))}. \tag{7}$$

Here $\mathcal{D}$ is a score function between the prototype and the query. In this paper, we simply choose $\mathcal{D}$ to be the Euclidean distance function. But it can be easily replaced with other parametrized functions such as MLP. Finally, we choose the relation y_i with maximum probability as the relation class of the given query:

$$y = \arg\max_i \mathcal{P}(y_i|S_i, q, \theta)\,. \tag{8}$$

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

Hyperparameter	Value
Batch Size	1
Layer Number	1, 2, 3
View Number	1, 2, 4, 8
Learning Rate	10^{-5}
Weight Decay	10^{-6}
Optimization strategy	Adam
Learning Rate Decay	0.6
Learning Rate Decay Step	1000
Maximun Sequence Length	256

Table 1: Hyperparameter settings.

Model Learning

Similar to previous work (Han et al., 2018; Gao et al., 2019b), all components in GTPN are trained in an end-to-end manner. Specifically, given a training task with N support relations (S_n, y_n) and one query instance (q, y_q), GTPN is learned by minimizing the following loss function:

$$\mathcal{J}(\theta) = -\sum_{n=1}^{N} \boldsymbol{I}(y_n) \log P(y_n|S_n, q, \theta), \tag{9}$$

where y_n is the relation of support set S_n and $\boldsymbol{I}(.)$ is an indicator function:

$$\boldsymbol{I}(y_n) = \begin{cases} 1, & y_n = y_q \\ 0, & y_n \neq y_q \end{cases}, \tag{10}$$

which indicates whether the relation corresponds to the golden relation of the given query.

Experiments

Experimental Settings

Datasets. We conducted experiments on two standard few-shot relation classification benchmarks: FewRel 1.0 (Han et al., 2018) dataset and FewRel 2.0 Domain Adaptation(Gao et al., 2019b) dataset [2]. FewRel 1.0 is constructed based on the articles from Wikipedia. FewRel 2.0 Domain Adaptation task further extends FewRel 1.0 by introducing an additional test data from PubMed footnotehttps://www.ncbi.nlm.nih.gov/pubmed/, which is a large database of biomedical literature, and is significantly different from Wikipedia train set. Totally, FewRel 1.0 consists of 100 relation classes and 700 instances for each relation class, and the standard 64/16/20 of train/validate/test relation class splits are adopted in our experiments. While FewRel 2.0 shares the same training data with FewRel 2.0, but a different test set including XXX relations in medical domain.

Baselines. We compared GTPN with the following published baselines:

- **Bert-Pair** (Gao et al., 2019b): A few-shot model that utilize the BERT sequence-pair model to measure similarity between two instances.
- **DaFeC** (Cong et al., 2020): An inductive unsupervised domain adaptation framework for few-shot relation classification.
- **Proto-Adv(Bert)** (Gao et al., 2019b): The adversarial trained prototypical networks with Bert as the encoder.
- **Proto-Bert** (Gao et al., 2019b): The vanilla prototypical networks using Bert as the encoder and the [CLS] token to represent the instance.

[2] https://github.com/thunlp/FewRel

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

Model	FewRel 2.0				
	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot	Avg.
Proto-CNN	35.1	49.4	23.0	35.2	35.7
Proto-Bert	40.1	51.5	26.5	36.9	38.8
Proto-Adv(Bert)	41.9	54.7	27.4	37.4	40.4
Proto-Adv(CNN)	42.2	58.7	28.9	44.4	43.6
DaFeC	61.2	77.0	47.6	64.8	62.7
BERT-PAIR	67.4	78.6	54.9	66.9	66.9
CP	79.7	84.9	68.1	79.8	78.1
MTB*	74.7	87.9	62.5	81.1	76.6
GTPN	80.0	**92.6**	69.25	**86.9**	**82.2**

Table 2: Accuracies(%) on FewRel 2.0 test sets. * is reported in Peng *et al.*, [2020].

Model	FewRel 1.0				
	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot	Avg.
Proto-Adv(CNN)	70.3	84.6	56.3	74.7	71.5
Proto-Adv(Bert)	73.4	82.3	61.5	72.6	72.4
Proto-CNN	74.5	88.4	62.4	80.5	76.4
HATT	–	90.1	–	83.1	–
Proto-Bert	80.7	89.6	71.5	82.9	81.2
MLMAN	83.0	92.7	73.6	87.3	84.1
BERT-PAIR	88.3	93.2	80.6	87.0	87.3
Bert-EM	89.8	93.6	83.4	88.6	88.9
REGRAB	90.3	94.3	84.1	89.9	89.6
CP	**95.1**	**97.1**	**91.2**	**94.7**	**94.5**
MTB	93.9	**97.1**	89.2	94.3	93.6
GTPN	89.4	97.0	84.4	93.8	91.2

Table 3: Accuracies(%) on FewRel 1.0 test sets. MTB and CP pretrained the matcher on Wikipedia, which is also the source of FewRel 1.0. While Pony introduced additional world knowledge to improve the performance. So there results are not directly comparable with GTPN.

- **Proto-Adv(CNN)** (Gao et al., 2019b): The adversarial trained prototypical networks with CNN as the encoder.
- **Proto-CNN** (Gao et al., 2019b): The vanilla prototypical networks using CNN as the encoder.
- **REGRAB** (Qu et al., 2020): A bayesian meta-learning method that utilize additional global relation graph knowledge.
- **Bert-EM** (Baldini Soares et al., 2019): A simple Bert-based prototypical networks that wrap entity with special markers.
- **MTB** (Baldini Soares et al., 2019): The same base model with **Bert-EM**, and uses large-scaled wiki data to pre-train the model, which is of the source with FewRel 1.0.
- **MLMAN** (Ye and Ling, 2019): The multi-level matching and aggregation network that involve intr-class information and support-query interactions.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, August 13 - 15, 2021.

- **HATT** (Gao et al., 2019a): The hybrid attention-based prototypical networks that also includes local context to avoid noisy data.

- **CP** (Peng et al., 2020): This model utilizes the same model of Bert-EM and proposes a different contrastive-based pretraining on Wikipedia.

Evaluation Metrics. We follow the settings in FewRel (Han et al., 2018) and adopt the official evaluation scripts [3] to evaluate the accuracy of all models on tasks including 5-way 1-shot, 5-way 5-shot, 10-way 1-shot, and 10-way 5-shot tasks.

Hyperparameter Settings and Infrastructure specifications. In the experimental period, we fix all the hyper-parameters listed in Table 1 during training stage except for the layer number and view number. We conducted grid search on validation set of FewRel 1.0 to find the best layer number and view number, which will be detailedly analyzed in the following. We train and evaluate our model using one Titan RTX GPU with about 24GB memory. Each training period costs about 2 hours for about 15000 steps of batches. The trainable parameters of GTPN are about 4M, excluding the BERT parameters.

Overall Results

Table 2 shows the overall results on FewRel 2.0 Domain Adaptation task and Table 3 shows the result on FewRel 1.0. From these tables, we can see that comparing with previous work:

1. **GTPN achieves the very competitive performance on few-shot relation classification.** From Table 2, we can see that GTPN achieved the best performance on the FewRel 2.0 domain adaptation task. Besides, we can also see that in Table 3, GTPN achieved a very competitive performance on the Fewrel 1.0. Note that the top 2 systems (i.e. MTB and CP) on FewRel 1.0 introduced additional in-domain knowledge into the task, and therefore is not directly comparable with GTPN. Specifically, MTB and CP pretrained the matcher on Wikipedia, which is also the source of FewRel 1.0. While the second-best model introduced additional world knowledge to improve the performance. On the contrast, GTPN achieves very strong performance without introducing any additional knowledge, which demonstrates the effectiveness of GTPN.

2. **GTPN is robust when domain shifting exists.** In FewRel 2.0 domain adaptation task, GTPN outperforms all other baseline models with a large margin, and also reaches the Top 1 in leaderboard of FewRel 2.0 domain adaptation benchmark, which demonstrates that GTPN is very robustness in few-shot relation classification task even without any knowledge of the target domain. We believe that this is because the proposed learning-to-discriminate paradigm can transfer more knowledge between different domains than previous learning-to-match paradigm. Besides, the learning-to-discriminate paradigm also more corresponds to the nature of few-shot relation classification task. These all results in the performance improvements when domain shifting exists.

3. **GTPN is even more effective on 5-shot learning paradigm.** Comparing with the improvement over the 1-shot settings, GTPN achieves more improvements on 5-shot settings. We believe that this is because a little bit more support instances could provide more sufficient discriminative information between different relations, and therefore results in better learning-to-discriminate performance.

Detailed Analysis

In this section, we conducted detailed analysis on the behavior of GTPN. Since the test sets of both FewRel 1.0 and FewRel 2.0 are unavailable to the public, we choose FewRel 1.0 validation set as the development set to select model, and use FewRel 2.0 validation set as the test set to evaluate the final performance for each model.

[3] https://github.com/thunlp/FewRel

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.

Model	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot
Bert-EM	79.0	88.4	64.4	84.4
Bert-EM+GTPN	**82.8**	**91.4**	**71.0**	**86.0**

(a) Accuracies (%) of GTPN over base encoder on FewRel 2.0. Bert-EM is our reimplement of Soares et al., (2019).

Relation Encoding	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot
One-hot	**82.8**	**91.4**	71.0	86.0
Random	82.2	90.2	**71.4**	**86.2**
None	82.2	91.0	68.6	85.8

(b) Accuracies(%) with different relation encodings of GTPN on FewRel 2.0 validation set.

View Number	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot
1	81.8	90.4	69.2	84.6
2	82.6	90.8	69.8	**86.0**
4	**82.8**	**91.4**	**71.0**	**86.0**
8	82.4	89.6	68.4	85.6

(c) Accuracies(%) with different view numbers of GTPN on FewRel 2.0 validation set.

Layer Number	5-Way 1-Shot	5-Way 5-Shot	10-Way 1-Shot	10-Way 5-Shot
1	**82.8**	**91.4**	**71.0**	**86.0**
2	81.8	89.4	69.4	85.8
3	80.4	88.2	68.2	83.6

(d) Accuracies(%) with different layer numbers of GTPN on FewRel 2.0 validation set.

Effect of GTPN with Different Instance Encoder. This experiment analyzes the effect of GTPN over the base instance encoder. We reproduce Bert-EM as the base encoder for GTPN and use the Prototypical Networks on Bert-EM as comparison. As we can see from Table 4a, with the same instance encoder, GTPN significantly outperforms vanilla Prototypical Networks, which means that the improvement of GTPN does not stem from the power of instance encoder, but from the effective learning-to-discriminate paradigm.

Effect of Relation Marker. In this experiment, we study the effect of different relation markers we proposed of GTPN. For fair comparison, we fix the layer number to 1 and the view number to 4. As shown in Table 4b, we can see that the performance of different kinds of one-hot and random relation markers are quite similar. One-hot relation marker performs slightly better on 5-way settings, while random marker performs slightly better on 10-way settings. However, the difference between them is not large, which means that both of them are effective relation marker for GTPN.

Effect of Transformation Layer Number. In this experiment, we study the effect of different transformation layer of GTPN. For fair comparison, we fixed all other hyper-parameters the same as the main experimental setting and the view number to 4. The layer number in this experiment is varied from 1 to 3. From the results shown on Table 4d, we can see that model with one transformation layers performs relatively best and more layer number leads the performance to fall. This is perhaps because the global transformation in GTPN is fully-connected with all instances in the task. Hence one layer is enough to capture global information from all other instances. Besides, the limited training data size can also undermine the performance of deep models, because it requires more training data to learn effective parameters.

Effect of View Number. In this experiment, we verify the effect of different view numbers. we vary the view number of GTPN from 1, 2, 4 to 8, and fix the layer number to 1. From the result shown on Table 4c, we can see that with the view number increases, the performance of GTP also improved. The model with 2 and 4 view number reach the relatively best performance on FewRel 2.0, and the model with single view performs the worst. We believe that this is because the relation representation may contain several perspectives of semantics. So with multi-view semantics, the model is easier to find the specific semantic and aggregates more information with other instances with the similar features. Besides, the model with 8 views are marginally worse than model with 4 views, which may indicate that 4 views are enough to capture different aspects information.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.

Related Work

Relation Classification. Relation classification has long been an important information extraction task. Conventional methods commonly employ syntax structure-based representations, e.g., dependency tree (Bunescu and Mooney, 2005) and constituent tree(Qian et al., 2008; Nguyen et al., 2009). In recent years, neural network-based methods dominate relation classification. (Zeng et al., 2014) proposed to encode relation instances via convolutional neural networks (CNN). (Zhou et al., 2016) proposed an attention-based BiLSTM (AttBLSTM) for instance encoding and classification. (Wang et al., 2016) proposed a multi-level attention CNNs for capturing multi-level lexical and semantic features. (Peng et al., 2017) tried to incorporate dependency information into neural networks by extending tree LSTM (Tai et al., 2015). (Velikovi et al., 2018) proposed graph attention networks (GATs) to incorporate the dependency information. The main drawback of supervised methods is that they require a large amount of annotated data, which is costly and cannot be easily obtained when adapting to new relation classes.

Few-Shot Learning. To resolve the annotated data bottleneck problem, few-shot learning is a promising approach. Many few-shot algorithms have been proposed in Computer Vision (CV) and Natural Language Processing (NLP), which can be categorized into two main paradigms, metric-learning based methods (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017) and meta-learning based methods (Andrychowicz et al., 2016; Santoro et al., 2016; Ravi and Larochelle, 2017; Finn et al., 2017; Munkhdalai and Yu, 2017; Mishra et al., 2018). Recently, many research interests have been focused on metric-based methods(Oreshkin et al., 2018; Sung et al., 2018; Liu et al., 2019; Gidaris et al., 2019; Zhou et al., 2020; Yang et al., 2021).

Few-Shot Relation Classification. (Han et al., 2018) proposed a few-shot learning task in relation classification, and adopted many few-shot learning methods into relation classification, including prototype-based method (Snell et al., 2017), meta-learning method (Mishra et al., 2018) and graph neural network (Satorras and Estrach, 2018). Since then, many prototype-based methods have been proposed for relation classification. (Baldini Soares et al., 2019) utilize the pre-trained language model BERT (Devlin et al., 2019) for relation encoding and use prototypes to represent different relation classes. (Qu et al., 2020) adds knowledge from graph to guide the meta-gradient for bayesian meta-learning. (Gao et al., 2019a) designed hybrid attention to learn a local matcher between intra-class instances, query and each support instances. (Ye and Ling, 2019) proposed the multi-level matching and aggregation network which updates support and query instances by matching and aggregating evidence on each support set. (Gao et al., 2019b) directly average the support-query scores using the BERT sequence-pair classification model for relation classification. We can see that the methods in few-shot relation classification are mostly in a learning-to-match paradigm, The matcher learned in these model are difficult to handle confusing reltion types, as well the domain shift. Thus, in this paper, we propose the learn-to-discriminate-based GTPN to tackle these limitations.

Conclusions

In this paper, we propose the *Global Transformed Prototypical Networks*, which switches previous learning-to-match paradigm to the learning-to-discriminate paradigm, and therefore can make the model concentrate more on the discriminative knowledge between all candidate relations. GTPN learns to build a few-shot model to directly discriminate between the query and *all* target classes with both inner-class local information and inter-class global information. Experiments on FewRel 1.0 and FewRel 2.0 demonstrate that GTPN achieves very competitive performance on few-shot relation classification, and reached the best performance on the FewRel 2.0 domain adaptation task. Which shows our method can benefit both further study and practice in few-shot relation classification.

Acknowledgements

This work is supported by the National Key R&D Program of China under Grant 2018YFB1005100.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.

References

Marcin Andrychowicz, Misha Denil, Sergio G. Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *NeurIPS*.

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

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *EMNLP*.

Xin Cong, Bowen Yu, Tingwen Liu, Shiyao Cui, Hengzhu Tang, and Bin Wang. 2020. Inductive unsupervised domain adaptation for few-shot classification via clustering. In *ECML-PKDD*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019a. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019b. FewRel 2.0: Towards more challenging few-shot relation classification. In *EMNLP-IJCNLP*.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord. 2019. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October.

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. In *EMNLP*.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop, volume 2*.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. 2019. Learning To Propagate Labels: Transductive Propagation Network For Few-Shot Learning. In *ICLR*.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. In *ICLR*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *ICML*.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *EMNLP*.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *TACL*.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *EMNLP*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *COLING*.

Meng Qu, Tianyu Gao, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *ICML*.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *ICLR*.

Adam Santoro, Sergey Bartunov, Matthew M Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. One-shot learning with memory-augmented neural networks. *ArXiv*, abs/1605.06065.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.

Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Petar Velikovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Li, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NeurIPS*.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *ACL*.

Shuo Yang, Lu Liu, and Min Xu. 2021. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. In *ACL*.

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

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*.

Linjun Zhou, Peng Cui, Xu Jia, Shiqiang Yang, and Qi Tian. 2020. Learning to select base classes for few-shot classification. In *CVPR*.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1012-1023, Hohhot, China, Augest 13 - 15, 2021.