Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

RESEARCH PAPER

Three Heads Better than One: Pure Entity, Relation Label and Adversarial Training for Cross-domain Few-shot Relation Extraction

Wenlong Fang, Chunping Ouyang†, Qiang Lin, Yue Yuan

School of Computer, University of South China, Hengyang, Hunan, 421001, China

Keywords: Few-shot learning; Relation extraction

Citation:
Submitted: June 4, 2022; Revised: October 9, 2022; Accepted: November 11, 2022

ABSTRACT

In this paper, we study cross-domain relation extraction. Since new data mapping to feature spaces always differs from the previously seen data due to a domain shift, few-shot relation extraction often perform poorly. To solve the problems caused by cross-domain, we propose a method for combining the pure entity, relation labels and adversarial (PERLA). We first use entities and complete sentences for separate encoding to obtain context-independent entity features. Then, we combine relation labels which are useful for relation extraction to mitigate context noise. We combine adversarial to reduce the noise caused by cross-domain. We conducted experiments on the publicly available cross-domain relation extraction dataset Fewrel 2.0[1][①], and the results show that our approach improves accuracy and has better transferability for better adaptation to cross-domain tasks.

1. INTRODUCTION

Few-shot relation extraction aims to learn a classifier to identify unknown relations with a small number of labels. Currently, few-shot relation extraction can already achieve good results [1, 2]. Still, the current few-shot are assumed to train and test datasets from the same domain. In the real world, due to different application scenarios, the data may come from different domains, and there may even be differences in the domain of the training set. Table 1 is an example of cross-domain relation extraction. The current main method

[†] Corresponding author: Chunping Ouyang (E-mail: ouyangcp@126.com; ORCID: 0000-0002-2154-0079).

[①] https://github.com/thunlp/FewRel

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

for few-shot relation extraction to deal with cross-domain relation extraction is meta-learning [3, 4, 5]. Several studies use meta-learning applied to relation extraction.

Although the Prototypical networks [6] achieve excellent results in most of the tasks of few- shot relation extraction, when the Prototypical networks encounter cross-domain, the results still perform unsatisfactorily, and the performance drops significantly. In few-shot relation extraction, a sentence with noise will lead to a significant deviation in prototype construction, especially the smaller the sample number, the more serious the deviation, as shown in Figure 1. This noise is mainly caused by context and cross-domain issues. Context noise causes entities of the same relation to map far away in the feature space. Due to different domains, the syntax and semantics of sentences are very different, resulting in the sentence mapping to feature space being very different, resulting in noise. It is pointed out by Peng et al. [7] that pure entity information and context information has a significant impact on the performance of relation extraction in few-shot relation extraction.

Table 1. An example comes from FewRel 2.0, the words in red represent entity.

	Train: wiki		Test: pubmed	
support	owned by	The Cowley plant is now where BMW's Mini is assembled, known At Plant Oxford.	Ingredient of	Cellulose gel membranes have been prepared by a pre-gelation method employing cellulose solutions.
	publisher	Rur Minase began publishing the series in Houbunsha's Manga Time magazine in 2012.	classified as	these tumors are the most common non- epithelial neoplasms epithelial neoplasms of gastric wall.
query	Which?	Rostokino is a station on the Moscow Central of the Moscow Metro that opened in 2016.	Which?	We report the effects of intravesical oxybutynin chloride with oxybutynin (modified intravesical oxybutynin)

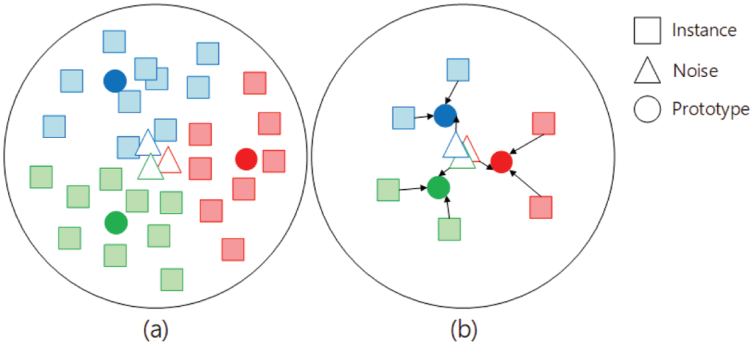


Figure 1. Different colors represent different relations, Figure a represent the result of traditional large sample relation extraction, the prototypes of different relations are very far. Figure b represent the result of few-shot relation extraction, the prototypes of different relations are very close because of the noise.

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

To mitigate the above problems and improve the transferability of the model, we have made some contributions. Firstly, to mitigate the effect of context noise on entities, we combine pure entities and context based on multiple transformer encoders for separately extracting pure entities features and context features. The pure entity features are extracted from the head and tail entities, and the context features are extracted from complete sentences. Combining pure entity and context obtains both context features and entity features that are not affected by context and reduce context noise. In addition, we combine relation labels as additional feature information to better extract features of the same class, reduce the noise brought by the context, and pull the instances of the same class closer to construct a better prototype. We compute similarity for each support based on the relation labels information. Then the instances are brought closer to the instances of the same class based on the similarity score. To further mitigate the impact due to cross-domain, we further optimize the model by introducing adversarial to improve the model's performance. Adversarial can shorten the distance of source domain and target domain data mapping to feature space, thus alleviating the significant difference in feature space caused by cross-domain. At last, we introduce Fine-tuning to improve the cross-domain adaptability of our model.

The contributions of this paper are summarized as follows:

- We combine pure entities and relation labels, which can extract effective information from entities and contexts and reduce the noise caused by context.
- We combine adversarial. We use source domain and target domain data for adversarial learning to shorten the feature space distance between the source and target domains and reduce the noise caused by cross-domain.
- We also introduce fine-tuning to mitigate the effects caused by cross-domain. The experiments based on the public dataset Fewrel2.0 demonstrate our model's effectiveness.

2. RELATED WORK

Nowadays, few-shot relation extraction is receiving more and more attention, and the proposal of Fewrel2.0[1] has brought a good dataset and experimental framework for few-shot relation extraction. Recently, meta-learning has been shown to be very effective in few-shot learning and can usually be divided into the following categories.

Optimization-based model, some [8, 9, 10, 11] proposes a gradient optimization-based meta-learning, which hopes to quickly generalize to new tasks through the learned knowledge. Model Agnostic Meta Learning (MAML) [12], learns an initial parameter and then fine-tunes few-shot relation extraction based on this parameter. Bayesian Model Agnostic Meta-learning [13, 14] proposes a new framework that can complement meta-learning based on gradient optimization.

The metric-based model, Siamese network [15] is applied in few-shot relation extraction, using convolutional neural networks coding and then calculate the similarity on the input. Matching network [4] proposes to count the similarity between samples using cosine distance, and classify those with the closest similarity into same category. Multi-level matching and aggregation network (MLMAN) [16] based on

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

matching network improvements. Prototype network [6] proposes to construct a prototype by taking the instances of the same class in the support set and the query set is classified based on the Euclidean distance to the prototype of each class. Hybrid attention-based prototypical network (HATT) [17], designs a hybrid attention-based prototype network that highlights key information at both instance level and feature level, effective in reducing noise caused by context. FSL Graph Neural Networks (GNN) [18] proposed the use of graph neural networks for relation extraction.

Introducing an external information-enhanced prototype network approach, based on bert [1], each instance is compared based on similarity, and each query set is compared to all support sets, using the bert sequence classification model. And [19] introduced a large number of pre-trained language models to improve accuracy. [14] uses external relation graphs to investigate the relations between different relations. Wang et al. [20] apply added relative position information and syntactic relation information to enhance the prototype network. DaFec [21] utilizes the unlabeled data of the target domain by clustering and Pseudo label, using Pseudo label added to in the training dataset to enhance the feature extraction for the target domain. Some people use fine-tuning in few-shot learning, which base on the pre-training model [22, 23, 24]. MSEPN, [25] Proposed Ensemble models and fine-tuning enhancements to cross-domain relation extraction mitigate high variance and discrepancies caused by differences in domains.

3. PROBLEM FORMULATION

Few-shot relation extraction usually has two respectively datasets as training and test sets. In this paper, we study cross-domain few-shot relation extraction, the training set is $D_{meta-train}$ from Wikipedia, and the test set is $D_{meta-test}$ from medical-biological domain, the label spaces of training and test sets do not intersect each other. Each dataset contains multiple instances (s, p, r), where s in the instance represents the complete sentence, p represents the location of the head and tail entities, and r represents the relation extraction label of this instance. In the few-shot setting, $D_{meta-test}$ is divided into two parts: $D_{test-support}$ and $D_{test-query}$. If $D_{test-support}$ contains N classes with K instances each, this task is called N-way-K-shot. $D_{test-query}$ contains test instances, each of which has a label that means which of the N classes it belongs to. Usually K is small, leading to poor results when predicting, so the model uses $D_{test-support}$ to predict $D_{test-query}$.

We also have data D_{UT} from the test set without labels, and we rename $D_{meta-train}$ as D_S. Our goal is that with D_S and D_{UT} we can learn meta-knowledge to migrate to the test domain for relation extraction based on few support set instances with labels and achieve better results.

4. THE PROPOSED APPROACH

In this section, we will go into the details of our model implementation. The framework of our model is shown in figure 2, and is divided into two main parts. The first part uses pure entities as input encoders, eliminating the influence of context on entities to creating a better prototype. The second part is the introduction of relation labels as external data to bring different instances of the same class closer together in vector space, pushing away instances of different classes and reducing the noise caused by context.

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

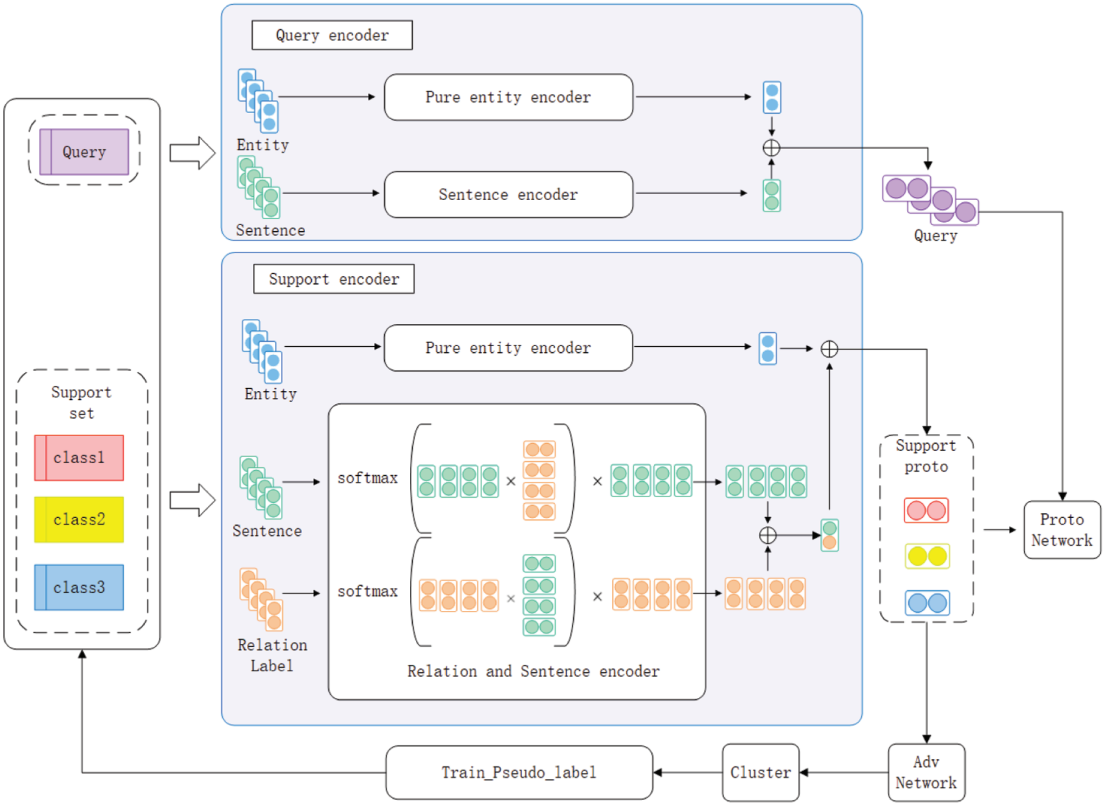


Figure 2. The overall framework of PERLA.

4.1 Pure Entity and Relation Labels

The encoding of the head entity, the tail entity, and the complete sentences are all done separately, with the aim of obtaining a feature matrix in which the three do not affect each other. The entity matrix is obtained by the dot product of the two entity encoding. The complete sentences matrix is obtained by calculating similarity through sentence encoding and relation labels encoding.

4.1.1 Pure Entity

According to [7], both entity information and complete sentences information are important for relation extraction, where there may be an over-reliance on entity information leading to poor relation extraction results. Ultimately it is the entity information that influences the context information leading to the neglect of the context information. Usually only a few words in the context are useful for relation extraction and most words introduce a lot of noise. Context information affects entity information leading to the loss of entity information and thus the construction of prototypes. In order to remove the effect of context on entities we decided to separate out the entities in the sentence and use only entities for feature extraction.

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

We use transformer to perform feature extraction on the entities, given a sentence $S = \{W_1, W_2, \ldots, W_n\}$ and the position of the entity, W_i denotes the i-th word in the sentence, we extract entity words by their position in the sentence. We use glove to generate the word embedding of head entities E_h and tail entities E_t. If the words constitute the entity is less than 12, we will fill in 0, if the words constitute the entity is more than 12 we will intercept the first 12 words. A sentence corresponds to a head entity and a tail entity, so the number of head and tail entities is the same. After that, we can let the each entity embedding has the same length. We use Transformer to extract the features of the entities from the word embedding vectors: $V_{head} = Transformer(E_h)$, $V_{tail} = Transformer(E_t)$.

After obtaining the head and tail entity matrices, we have to perform a dot product on the two matrices to obtain the final entity matrix: $S_E = V_{head} \cdot V_{tail}$.

Once we have obtained the entity matrix we also need to derive the corresponding entity prototype from the entity matrix, k means the number of entities in each class:

$$p_{ei} = \frac{1}{k}\sum_{n=1}^{k} S_E \tag{1}$$

4.1.2 Relation Labels and Sentences

We directly use the sentence as the input of the encoder, hoping to get the sentence features of the entity-context interaction and use them for relation extraction. But there are only a few words in the context that can contribute to relation extraction, and many of the remaining words are noisy. The source and target domains map to the vector space with great differences. These differences make the prototype difficult to construct. In order to reduce the noise of context, distinguish the features of complete sentences between different classes and construct better prototypes. We introduce relation labels to encode context information, use relation labels to shorten the euclidean distance between different instances of the same relation in vector space, push the euclidean distance between different classes. We filter out the dimensions that contribute more to the prototype by relation labels, so as to build a better prototype.

We introduce the relation labels as external data for support. The sentences are encoded by transformer to obtain the each support sentences features S^i_k of relation i. The relation labels words are encoded by transformer to obtain the relation labels features R^i. We calculate the similarity between the R^i and S^i_k, and then get the final features $\hat{r}^i$ of relation i.

$$\alpha^r = softmax(sum(R^i (S^i_k)^T)) \tag{2}$$

$$\hat{r}^i = \sum_{n=1}^{k} \alpha^r_n \left[R^i\right]_n \tag{3}$$

where, i stands for relation i and k stands for the k-th instance. $[\cdot]_n$ means the n-th row of the vector, and sum() represents a summation over all elements of each row. We assign weights to relations based on differences in the similarity of instances and relations, and obtain the relation features by weighting the sum.

Similarly we calculate the similarity of the sentence features and relations of the support set to obtain the support set sentence features. That will lead to selecting instance that have similar words as relation words:

$$\alpha^S = softmax(sum(S_k^i (R^i)^T)) \tag{4}$$

$$\hat{S}^i = \sum_{n=1}^{k} \alpha_n^S \left[S_k^i \right]_n \tag{5}$$

We generate prototypes of each class from the sentence features and relation features of the support set: $p_{si} = \hat{S}^i + \hat{r}^i$.

Our final prototype consisting of support entity prototype concat support sentence prototype: $p_i = \{p_{ei}; p_{si}\}$. Similarly our query finally consists of the query entity q_e and the concat of the query sentence: $q_i = \{q_e; q_s\}$. The sentence and relation tags are multiplied by each other to obtain the corresponding score weights, and the weight matrices of the sentence and relation tags are directly added to obtain the final sentence code.

4.2 Prototypical Network

The prototype network is a traditional and effective few-shot classification network. The principle of prototype networking is simple but effective. Averaging all samples of the same class, this average is the prototype for this class. There are as many prototypes as there are categories. Calculate the Euclidean distance between the sentence to be classified and each prototype, and the category of the prototype closest to the sentence is the category of the sentence.

After getting the pure entity and complete sentence vector of the support set, these vectors are divided according to class to do the prototype of this class we use the prototype network to train our model, using the support set as training and the query set as testing, each query sentence q has a corresponding label i, meaning that the query sentence belongs to class i. The formula for each prototype and the category calculation formula for each query are as follows.

$$P(y = i \mid q) = \frac{exp(-d(v(q), p_i))}{\sum_{j=1}^{N} exp(-d(v(q), p_j))} \tag{6}$$

The loss function of the prototype network is calculated as follows:

$$minL_{ProtoSD} = -E[logP_{q \in Q}(y \mid q)] \tag{7}$$

By minimizing the loss function, each query set can continuously approach the prototype corresponding to it, allowing the model to better extract the desired features for distinguishing between different classes of relations.

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

4.3 Adversarial Learning and Pseudo Label

In order to reduce the impact caused by the difference domains when mapping the vectors of the training and test sets into the same space, and to improve the transferability of our model. We introduce an adversarial learning network to bridge the distance between different domains. At the same time we also generate pseudo labels and add the data with pseudo labels to the training set for the next training, so that more information about the features of the test set domain can be learned.

4.3.1 Adversarial Learning

In order to improve the transferability of our model, we used adversarial training [1, 21]. Hoping to improve the transferability of our model by closing the euclidean distance between different domains through adversarial training. Basing on bringing the invariant domains of the two domains closer together and the variable domains farther apart. Adversarial learning mines similar features between the two domains. Shorten the Euclidean distance of vectors in different fields by similar features. This uses a discriminator D and a feature extractor E. The discriminator is to distinguish whether the features extracted by the feature extractor come from the source or target domain, where the discriminator consists of a two-layer perceptron network, and we use the data with labels from the training set and the data without labels from the test set for training. The loss of discriminator is defined as follows:

$$\min_{\theta} L_{Dis} = -E_{s\sim D_S}[logD_{\theta}(E(s))] - E_{s\sim D_{UT}}[log(1-D_{\theta}(E(s)))] \tag{8}$$

The role of the feature extractor is to generate the features of the confusion discriminator. Feature extractor is the sentence encoder in the model, it is a transformer which has the same structure of sentences encoder. The loss of feature extractor is defined as follows:

$$\min_{\varphi} L_{Enc} = -E_{s\sim D_S}[log(1-D(E_{\varphi}(s)))] - E_{s\sim D_{UT}}[log(D(E_{\varphi}(s)))] \tag{9}$$

4.3.2 Pseudo Label

We hope to learn the test set data better, and we perform pseudo-label generation by clustering algorithm. In order to learn the sentence features of the target domain, we use clustering to pseudo-label the unlabeled data. And these pseudo-labeled data are retrained to learn the sentence features of the target domain. But the clustering algorithm requires high quality of the upstream data input, and the clusters generated based on the more differentiated data are more distinct and credible. The upstream data input is the source domain and target domain sentences features. If only using few-shot classifier can not distinguish the unlabeled test domain data well, resulting in poor results of the pseudo-label generation data.

Processing the data upstream of clustering has been shown to be effective in few-shot cross-domain relation extraction [25]. Therefore, we use a clustering mechanism based on similar entropy minimization, and since we want to classify unlabeled data, similar entropy minimization can better distinguish between different classes of unlabeled data, pulling away dissimilar features and bringing similar features closer

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

together by entropy minimization for the purpose of generating differentiated high-confidence data. We obtain the distance vector V(X) by calculating the distance between each unlabeled target domain D_{UT} instances:

$$[V(x_i)]_j = x_i - x_{j2}^{\,2} \qquad x_i, x_j \in D_{UT}, \quad i \neq j \tag{10}$$

X_i means the i-th instance, $||\cdot||_2$ means L2 regularization, and $[\cdot]_j$ means the j-th element of the vector X. In order to keep the dissimilar instance features separated and the similar instance features close as possible, we minimize the entropy of each V(X) distance vector matrix.

$$\min_{\theta} L_{Entropy} = -E_{x \sim D_{UT}} [H(softmax(v(x)/\tau))] \tag{11}$$

The meaning of H(·) is the Shannon entropy on the softmax distribution and τ is the temperature coefficient used to adjust the percentage of similar instances of the target data.

By minimizing the similarity entropy each unlabeled instance of the target domain will be pushed to its similar samples and away from the dissimilar ones, so that the clustering will result in more separated data and less noise.

4.4 Fine-tuning

Due to the different feature distributions in the training and test sets, the trained feature extractor tends to be closer to the training set, and to improve the transferability of the model, we propose to fine-tune the parameters of our model using the support set $S = \{\{x_i, y_i\}^{n1}{}_{i=1}, \ldots, \{x_i, y_i\}^{nK}{}_{i=1}\}$, n_k represents the number of samples available in category k. We encode the xi from S and feed into the feedforward classifier as follows:

$$p_i = softmax(M \cdot f_{\tilde{\theta}}(x_i) + b) \tag{12}$$

Both M and b are parameters of the feedforward classification layer, which we train by means of the cross-entropy loss function:

$$min \sum_i CrossEntropy(y_i, p_i) \tag{13}$$

Through cross-entropy, we can optimize the parameters of our model. By carefully and appropriately adjusting the parameters, we can obtain a model that is more adapted to the target domain and alleviate the problems caused by cross-domain.

5. EXPERIMENTS AND ANALYSIS

In this section, we present the experiments of our method and the details of the experiments. First, we compare our model with other models to reflect the superiority of our model, and then we further investigate the effects of different parts of our model.

Table 2. Results on Fewrel2.0.*Results reported by Gao et al. (2019c), - one shot tasks is not suitable for our fine-tuning method. PERL means without combining Adversarial, PERLA is our model, FT means Fine-tuning.

FewRel2.0	5 way			10 way		
	1 shot	5 shot	10 shot	1 shot	5 shot	10 shot
Siamese	39.66	47.72	53.08	27.47	33.58	38.84
GNN	35.95	46.57	52.20	22.73	29.74	-
Proto-CNN	40.16	52.62	58.69	28.39	39.38	44.98
Proto_hatt	40.78	56.81	63.72	29.26	43.18	50.36
Bert-pair	56.25*	67.44*	-	43.64*	53.17*	-
Ensemble	45.44	63.43	69.85	33.34	50.30	57.31
Ensemble_FT	-	72.40	78.00	-	65.31	71.60
DaFec-Proto-CNN	48.58	65.80	70.62	35.53	52.71	59.94
DaFec-Proto-Bert	46.39	56.32	64.57	32.09	40.53	50.28
Ours	52.14	72.70	77.00	39.26	61.13	66.50
ours+Pl	54.24	73.28	77.67	40.08	61.85	67.07
Ours+Pl+FT	-	76.20	80.26	-	66.54	67.43

5.1 Datasets and Experimental Setup

The dataset we use is the Fewrel2.0[1], which is widely used in few-shot relation extraction, and it is the most used dataset for cross-domain few-shot relation extraction at present. This dataset provides data from Wikipedia (Wiki) and biomedical (pubmed). Our training set uses both the Wikipedia data with labels and the pubmed data without labels. the Wikipedia data consists of 64 classes with 700 instances each, for a total of 44,800 instances, and the pubmed data without labels for a total of 2,500 instances. In addition, the data of SemEval-2010 task 8 and pubmed with labels are also available in Fewrel2.0. Our validation and testing is done with the pubmed labeled data, which includes 10 classes, each with 100 instances, for a total of 1000 instances. Each instance includes the complete sentence, the head entity, the tail entity, the head entity position and the tail entity position. We validated our model in different scenarios: 5way1shot, 5way5shot, 5way10shot, 10way1shot, 10way5shot, 10way10shot, and evaluated with accuracy.

Our encoder is based on transformer, where the hidden size is set to 60 and the word embedding uses glove. The complete sentence word embedding dimension is 60, containing sentence dimension 50 and both entity position dimensions of 5. The maximum sentence length is 80.The relation word embedding dimension of the sentence is 50, with a maximum length of 80. The dimension of entity word embedding is 50 and the maximum length is 12. Our model is trained and fine-tuned using stochastic gradient descent with a learning rate of 0.1 and a number of pseudo-label classifications of 10. We fine-tune it with 60 epochs and update the weights using cross-entropy.

5.2 Baseline

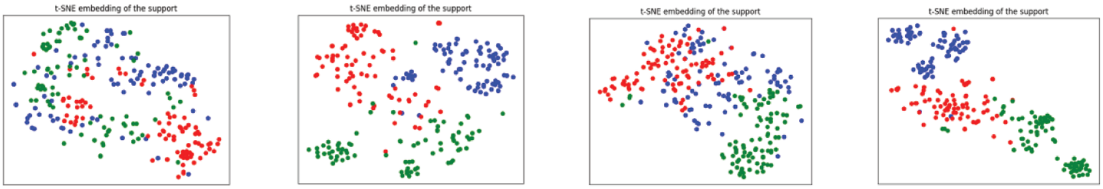
Siamese networks [15] is based on two samples mapped to the same space then use Euclidean distance to calculate the distance between two samples, graph neural network (GNN) [18], use both support set and query set as nodes of graph neural network, then use graph neural network for classification. Prototypical network (Proto) [6] averaging the support set to calculate a category of prototypes, and the query set to calculate the distance to each prototype, classified by the label of the prototype closest to the query set. Hybrid attention-based prototypical networks (HATT) [17], a modified version of the hybrid attention-based prototype network, which adds instance-level attention and feature-level attention in the hope of learning better and less noisy features. MSEPN [25], Prototype network built by ensembling various encoder and computational similarity methods and introducing fine-tuning. DaFec [21] prototype network combined with clustering, also a prototype network based improvement.

5.3 Comparison Experiments with Existing Methods

In this section we show the results of our model as shown in Table 2, the results of our method compared to other baseline methods. From the results we can observe that (1) compared to the current state-of-the-art methods Bert-pair, Ensemble-FT, and DaFec-Proto-CNN our model has better results, which effectively prove the effectiveness and rationality of our model under different scenario settings. (2) Compared with Bert-pair, DaFec-Proto-Bert, our model is 9.20% and 19.80% higher in the 5way5shot scenario, respectively. This shows that our model can achieve better results even without a large pre-trained language model. (3) The comparison between PERLA+FT and PERLA demonstrates that Fine-tune can improve adaptability to cross-domain relations extraction.

5.4 Impact of PERLA

In this section, in order to verify the advanced and transferable nature of our model we visualized the features based on the data from the test set. We used three relations from test: 'inheritance type of', 'is normal tissue origin of disease', and 'occurs in'. Each relation chooses 80 sentences to display feature distribution. The figure 3 shows the t-SNE plots of instance embedding, different colors represent different categories, (a) only use pure entity to encode, (b) only use sentence, (c) use adversarial, (d) use our model PERLA. Features are more discrete in figure(a, b, c), the features are more concentrated in figure(d). It can be clearly seen that our model can make instances of the same class closer together and instances of different classes further apart, which helps us to construct better prototypes. This effectively proves the advancedness of our model. PERLA reduce the noise brought by the context, and alleviate the problems brought by the cross-domain.



(a) Pure entity (b) Only sentence (c) Adversarial (d) PERLA

Figure 3. t-SNE plots of instance embeddings, different colors represent different relations.

5.5 Impact of Pure Entities

In this section, to verify the impact of pure entities and complete sentences we use different inputs to encode with the same parameters at 5way and 10way respectively. The results are shown in Figure 4, where our approach outperforms the results using only entities and only complete sentences, providing strong evidence that our proposed approach can learn more useful information in few-shot scenarios. The model may be badly affected by context on entities, which affects the vector feature space of the whole sentence. We introduce pure entities to reduce the noise of context on entities, which improves the generalisation of the prototype. If only pure entities were used without context, the context features would be lost, and thus the prototype would only have entity features that would easily cause over-fitting and rote memorization of entities, so we need a combination of pure entities and context with both entity and context features as the basis for prototype construction.

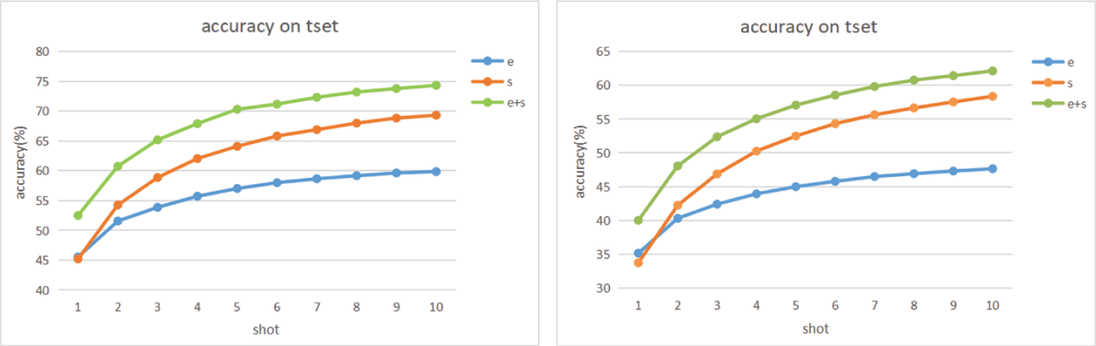


Figure 4. Accuracy of support, left is 5wayNshot, right is 10wayNshot, 'e' means only using pure entity to encode, 's' means using sentence to encode, 'e+s' means using pure entity and sentence without relation labels to encode.

5.6 Impact of Relation Labels

This section is designed to verify the impact caused by the introduction of relation labels. We conducted experiments with 5way and 10way equivalent parameters, respectively, and the experimental results are shown in Figure 5. The results show that the introduction of relation labels can effectively improve the accuracy of the prototype network, proving that the introduction of relation labels effectively brings the distance between different instances of the same class closer, thus constructing a prototype with less noise and better results. In Fewrel 2.0, because the sentences in the target domain and the sentences in the source domain are very different, the sentences in the target domain are longer and more complex, which will bring a lot of noise, we can reduce the noise brought by the cross-domain problem by introducing relation labels to filter out the dimensions that contribute more to the prototype.

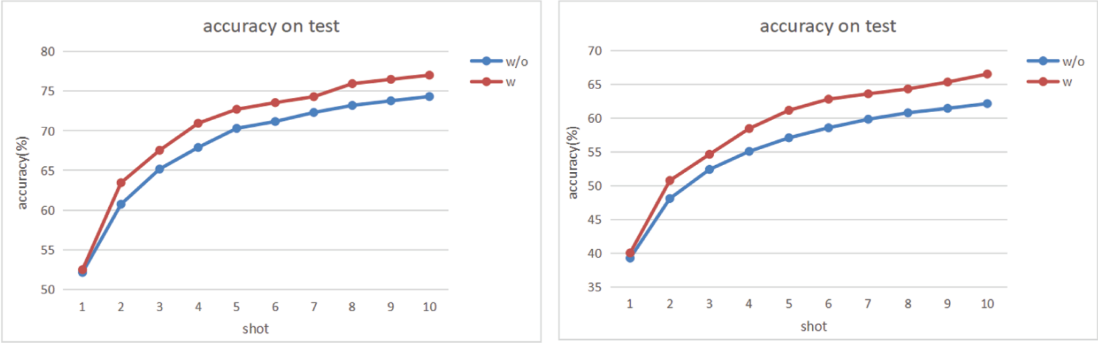


Figure 5. Accuracy of support, left is 5wayNshot, right is 10wayNshot, 'w/o' means with PERLA, 'w' means with PERLA.

6. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of cross-domain few-shot relation extraction. We use transformer to encode entities and sentences separately, and obtain features of pure entities and contexts for relation extraction. We also introduce relation labels, which draws instances of the same class closer and pushes instances of different classes farther through label information. In order to improve the transferability of the model and alleviate problem of different feature space caused by cross-domain, we combine adversarial to close the feature space distance between source domain and target domain. To mitigate the impact from cross-domain, we use fine-tuning. The experimental results on Fewrel2.0 show that our model significantly improves accuracy and portability and achieves state-of-the-art results. In the future, we will further explore the cross-domain issues associated with relation extraction and adopt more advanced approach to mitigate the above issues.

ACKNOWLEDGEMENTS

The State Key Program of National Natural Science of China, Grant/Award Number:61533018; National Natural Science Foundation of China, Grant/Award Number: 61402220; The Philosophy and Social Science Foundation of Hunan Province, Grant/Award Number: 16YBA323; Natural Science Foundation of Hunan Province, Grant/Award Number: 2020JJ4525, 2022JJ30495; Scientific Research Fund of Hunan Provincial Education Department, Grant/Award Number: 18B279, 19A439.

AUTHOR CONTRIBUTIONS

Wl. F (fsfwl11406@gmail.com, 0000-0002-5035-3571) was jointly responsible for the investigation of the study, organization of data, design, analysis, and validation of methods, preparation of resources, and preparation of the original manuscript. Cp. Ouyang (ouyangcp@126.com, 0000-0002-2154-0079) was jointly responsible for the investigation of the study, organization of the data, the conceptualization of the methods, and review and editing of the manuscript, as well as supervision and project management. Q.L (linqiang0219@163.com) contributed to the investigation of the study, the conceptualization of the methods, analysis,validation, and data curation. Y.Y (uscyuanyue@gmail.com) contributed to the review and editing of the manuscript.

REFERENCES

[1] Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, Jie Zhou.: Fewrel 2.0: Towards more challenging few-shot relation classification. arXiv preprint arXiv:1910.07124 (2019)

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

[3] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, Timothy Lillicrap.: Meta-learning with memory-augmented neural networks. In International Conference on Machine Learning, pp. 1842–1850. PMLR (2016)

[4] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al.: Matching networks for one shot learning. Advances in Neural Information Processing Systems 29 (2016)

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

[6] Jake Snell, Kevin Swersky, Richard Zemel.: Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems 30 (2017)

[7] Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, Jie Zhou.: Learning from context or names? an empirical study on neural relation extraction. arXiv preprint arXiv:2010.01923 (2020)

[8] Sachin Ravi, Hugo Larochelle.: Optimization as a model for few-shot learning (2016)

[9] Tsendsuren Munkhdalai, Hong Yu.: Meta networks. In International Conference on Machine Learning, pp. 2554–2563. PMLR (2017)

[10] Joaquin Vanschoren.: Meta-learning: A survey. arXiv preprint arXiv:1810.03548 (2018)

[11] Thomas Elsken, Benedikt Staffler, Jan Hendrik Metzen, Frank Hutter.: 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 (2020)

[12] Chelsea Finn, Pieter Abbeel, Sergey Levine.: Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)

[13] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, Sungjin Ahn.: Bayesian model-agnostic meta-learning. Advances in Neural Information Processing Systems 31 (2018)

[14] Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, Jian Tang.: Few-shot relation extraction via bayesian meta-learning on relation graphs. In International Conference on Machine Learning, pp. 7867–7876. PMLR (2020)

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

[15] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al.: Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, volume 2, page 0. Lille (2015)

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

[17] Tianyu Gao, Xu Han, Zhiyuan Liu, Maosong Sun. 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 (2019)

[18] Victor Garcia, Joan Bruna.: Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043 (2017)

[19] Fangchao Liu, Xinyan Xiao, Lingyong Yan, Hongyu Lin, Xianpei Han, Dai Dai, Hua Wu, Le Sun.: From learning-to-match to learning-to-discriminate: Global prototype learning for few-shot relation classification. In China National Conference on Chinese Computational Linguistics, pp. 193–208. Springer (2021)

[20] Yuxia Wang, Karin Verspoor, Timothy Baldwin.: Learning from unlabelled data for clinical semantic textual similarity. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, pp 227–233 (2020)

[21] Xin Cong, Bowen Yu, Tingwen Liu, Shiyao Cui, Hengzhu Tang, Bin Wang. Inductive unsupervised domain adaptation for few-shot classification via clustering. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 624–639. Springer (2020)

[22] ME Peters M Neumann, M Iyyer, M Gardner, C Clark, K Lee, L Zettlemoyer.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving language understanding by generative pre-training (2018)

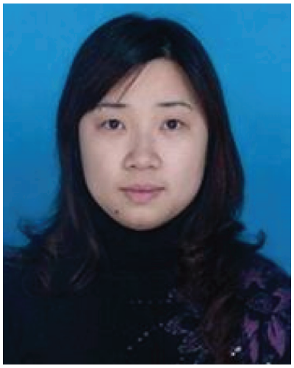
[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[25] Qing Lin, Yongbin Liu, Wen Wen, Zhihua Tao.: Ensemble making few-shot learning stronger. arXiv preprint arXiv:2105.11904 (2021)

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190

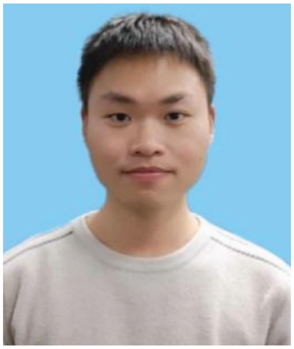
AUTHOR BIOGRAPHY

Wenlong Fang received his B.S. degree in Software Engineering from the School of Computer Science, University of South China, China, in 2020. He is pursuing his M.S. degree with a specialization in software engineering at the University of South China. His research interests include Relation Extraction and Named Entity Recognition.
ORCID: 0000-0002-9643-8829



Chunping Ouyang is a professor and the head of the Department of Software Engineering. Her main research interests are semantic web technology, knowledge graphs, natural language processing, and domain data analysis. She is a visiting scholar of Indiana University, and a young backbone teacher of Hunan Province. She has presided over many national and provincial projects such as the National Natural Science Foundation of China, Hunan Provincial Natural Science Foundation, Hunan Provincial Philosophy and Social Science Foundation, and Hunan Provincial Excellent Youth Project. She has participated in many projects as a critical member of the National Science and Technology Support Program, National Basic Condition Platform, Beijing Science and Technology Plan Key Project, PetroChina Key Science and Technology Plan, etc. She participated in the project "Development and Application Demonstration of Software Integration Platform for Oil and Gas Production System," which won the third prize in Electronic Information Science and Technology at the Chinese Institute of Electronics. She has published more than 30 academic papers in domestic and international journals and conferences as the first author and corresponding author.
ORCID: 0000-0002-2154-0079

Data Intelligence Just Accepted MS.
https://doi.org/10.1162/dint_a_00190



Qiang Lin is received his M.E degree from the University of South China, China, in 2022. Her research interests include Information Extraction and Few-shot learning.