Inductive Unsupervised Domain Adaptation for Few-Shot Classification via Clustering

Xin Cong[1,2], Bowen Yu[1,2], Tingwen Liu[1,2] (✉), Shiyao Cui[1,2], Hengzhu Tang[1,2], and Bin Wang[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{congxin,yubowen,liutingwen,cuishiyao,tanghengzhu}@iie.ac.cn
[3] Xiaomi AI Lab, Xiaomi Inc., Beijing, China
{wangbin11}@xiaomi.com

Abstract. Few-shot classification tends to struggle when it needs to adapt to diverse domains. Due to the non-overlapping label space between domains, the performance of conventional domain adaptation is limited. Previous work tackles the problem in a transductive manner, by assuming access to the full set of test data, which is too restrictive for many real-world applications. In this paper, we set out to tackle this issue by introducing a inductive framework, DaFeC, to improve **D**omain **a**daptation performance for **Fe**w-shot classification via **C**lustering. We first build a representation extractor to derive features for unlabeled data from the target domain (no test data is necessary) and then group them with a cluster miner. The generated pseudo-labeled data and the labeled source-domain data are used as supervision to update the parameters of the few-shot classifier. In order to derive high-quality pseudo labels, we propose a Clustering Promotion Mechanism, to learn better features for the target domain via Similarity Entropy Minimization and Adversarial Distribution Alignment, which are combined with a Cosine Annealing Strategy. Experiments are performed on the FewRel 2.0 dataset. Our approach outperforms previous work with absolute gains (in classification accuracy) of 4.95%, 9.55%, 3.99% and 11.62%, respectively, under four few-shot settings.

Keywords: Few-shot classification · Domain adaptation · Clustering

1 Introduction

Few-shot classification aims to learn a classifier to recognize unseen classes with few labeled examples. While significant progress has been made [20, 5, 16, 13, 17], most previous works are under the assumption that the samples of unseen classes should be drawn from the same domain as the training data that was used to train the model. However, in the real world, the application can be used in unusual environments and novel datasets, which means that these samples are likely from different domains. Even a slight departure from a model's training

	Source domain: Wikipedia		Target domain: PubMed	
Support	*member_of*	*Newton* served as the president of *the Royal Society*.	*may_treat*	*Ribavirin* remains essential to *Chronic Hepatitis C* treatment.
	instance_of	The *Romanian Social Party* is a left *political party* in Romania.	*classified_as*	All references about a *viral infection* called *ebola haemorrhagic fever* were reviewed.
Query	*Which?*	*Euler* was a member of *the Royal Swedish Academy of Sciences*.	*Which?*	The *dental cysts*, especially *radicular cysts*, are compared.

Table 1. An example comes from FewRel 2.0, a few-shot dataset for relation classification with domain adaptation. Different colors indicate different entities, red for head entity, and blue for tail entity. Relation classification aims to determine the relation between two given entities based on their context.

domain can cause it to make spurious predictions and significantly hurt its performance. Table 1 illustrates a typical example of domain shift in the relation classification task which aims to classify the semantic relation between entities in a sentence. The training data is collected from Wikipedia, but the actual data encountered at test time comes from PubMed, a biomedical literature corpus. The new relations in the target domain such as *may_treat* are different from those in the source domain. Due to distinct domain characteristics like morphology and syntax, the performance of existing few-shot models drops drastically in such a situation.

Unsupervised domain adaptation algorithms (UDA) aims at addressing the domain shift problem between a labeled source dataset and an unlabeled target dataset [7, 18, 22]. Conventional UDA methods typically assume that the target domain shares the same label space with the source domain so that the knowledge can be transferred from across domains via these same labels. However, in the few-shot settings, the source domain and target domain do not have any overlap in categories. This unique setting renders most existing UDA methods inapplicable. Previous work [4] solves this problem by making use of test data (or query set in the few shot scenario) from the target domain in the transductive manner. However, in some real-world scenarios, it is completely unrealistic to forecast test data in advance. In this paper, we work on a more realistic setting: inductive unsupervised domain adaptation for few-shot classification. It is obvious that although we do not know the ground truth classes of the target domain, some of the unlabeled target-domain data may belong to the same classes. According to the cluster hypothesis [1], the features of unlabeled data with the same latent label may cluster together in the representation space. Mining these latent cluster structures can provide auxiliary information about the target domain, which could be beneficial to improve the adaption ability of few-shot models. Based on such motivation, we design a novel framework named **DaFeC** (Unsupervised

Domain **a**daption for **Fe**w-shot classification via **C**lustering), which effectively train the few-shot classifier with clustering-generated pseudo labels. The first step of DaFeC is the training of a representation extractor. Based on the features of unlabeled target-domain data derived from the extractor, a cluster miner is applied to group these unlabeled instances and the subsequent cluster assignments are deemed as pseudo labels. Finally, a few-shot classifier is trained based on both target-domain data with pseudo labels and source-domain training data to enable the classifier to adapt to the target domain.

Intuitively, the quality of pseudo labels significantly influences the performance of the few-shot classifier. Theoretically, if input features are well discriminative, the cluster miner can group the instances easily and assign them pseudo labels with high-confidence. Therefore, to generate high-confidence pseudo labels, we further propose a **C**lustering **P**romotion **M**echanism (**CPM**) to assist in training the representation extractor to produce cluster-distributed features for unlabeled target-domain data. CPM contains three modules: First, to encourage features with the same latent class to get closer, we design a Similarity Entropy Minimization (CPM-S) objective. It calculates the Euclidean distance between each instance and others in the target domain and then minimizes the entropy of instance-wise distance vector to drive similar instances closer. Second, in our preliminary study, we observe that although the source and target domain have different label space, they may still share some similar but not identical labels. For instance (Table 1), the class *classified_as* from the target domain is semantically similar to *instance_of* from the source domain with slightly difference. Inspired by this phenomenon, we design an Adversarial Distribution Alignment method (CPM-A). It introduces a domain discriminator to play an adversarial minimax game with the representation extractor to align the distribution of similar classes cross domains. Third, we propose a Cosine Annealing Strategy (CPM-C) to support learning with CPM-S and CPM-A for achieving the optimal domain adaptation performance.

To summarize, our contributions are as following:

- For the first time, we present an inductive unsupervised domain adaptation framework, DaFeC, for few-shot classification. To the best of our knowledge, there is no similar work in few-shot classification.
- We propose a Clustering Promotion Mechanism to help the representation extractor produce cluster-distributed features for the generation of high-confidence pseudo labels.
- Our presented DaFeC is model-agnostic, which means that it can be incorporated into other models.
- Our approach achieves new state-of-the-art performance on FewRel 2.0, the currently largest unsupervised domain adaptation dataset for few-shot classification, delivering 3.99-11.62% absolute gains over previous work[4].

[4] The source code and data of this paper are available now and they can be obtained from https://github.com/congxin95/DaFeC.

2 Related Work

Few-shot classification aims to develop models and algorithms which are able to recognize novel classes based on few labeled instances. Recently, meta-learning has been shown to be highly effective in few-shot learning, which can be generally classified into three categories: (1) Model-based methods [15, 12] design a special module such as memory to exploit meta information to make models generalize to new tasks rapidly with only a few instances. (2) Optimization-based methods [5, 13] aim at learning a good initialized parameters which can achieve good performance through a few update steps. (3) Metric-based methods [20, 16, 17] attempt to learn a good metric function which embeds data with the same classes into adjacent distance space. Although many existing few-shot methods have achieved promising results, the performance of these methods is significantly degraded when the test data are drawn from different domains from training data, which is a quite common case in the real world.

Domain adaptation methods aim at exploiting labeled data in the source domain to perform a prediction task in the target domain. Because annotating sufficient labeled data is time-consuming and labor-intensive, unsupervised domain adaptation (no need for labeled data of the target domain), has been extensively studied recently [7, 19, 22]. However, all of these methods assume that the categories of the target domain are shared with the source domain, which is too restrictive to generalize to the novel classes in the few-shot classification scenario. To address this issue, [4] leverages reinforcement learning to select source data similar to the target test data to train few-shot classifiers. Nevertheless, this method works in the transductive manner, while in some real-world applications, we cannot know the test data when training. By contrast, our approach works in a more realistic inductive fashion that models cannot get information about test instances in the training phase. The only thing we can use is the unlabeled target-domain data, which can be different from the test data.

3 Task Formulation

In the few-shot classification, formally, we have two datasets: $\mathcal{D}_{meta-train}$ and $\mathcal{D}_{meta-test}$. These datasets contains a set of instances (x, y) but $\mathcal{D}_{meta-train}$ and $\mathcal{D}_{meta-test}$ have their own label space that are disjoint with each other. In the few-shot settings, $\mathcal{D}_{meta-test}$ is split into two parts: $\mathcal{D}_{support}$ and $\mathcal{D}_{query}$. If the support set contains K instances for each of N classes, this few-shot problem is called N-way-K-shot. Usually, K is really small, resulting in the poor performance when predicting $\mathcal{D}_{query}$. Therefore, models should use $\mathcal{D}_{support}$ to predict $\mathcal{D}_{query}$ labels utilizing $\mathcal{D}_{meta-train}$.

For unsupervised domain adaptation in few-shot classification, $\mathcal{D}_{meta-train}$ and $\mathcal{D}_{meta-test}$ are sampled from different domains. $\mathcal{D}_{meta-train}$ from the source domain and $\mathcal{D}_{meta-test}$ from the target domain. We rename $\mathcal{D}_{meta-train}$ as $\mathcal{D}_S$. To overcome domain discrepancy, an unlabeled target-domain dataset $\mathcal{D}_{UT} = \{x_1, x_2, \ldots, x_{UT}\}$ is provided. Our goal is to develop a model that acquires knowledge from the $\mathcal{D}_S$ and $\mathcal{D}_{UT}$, so that we can make predictions over $\mathcal{D}_{meta-test}$.

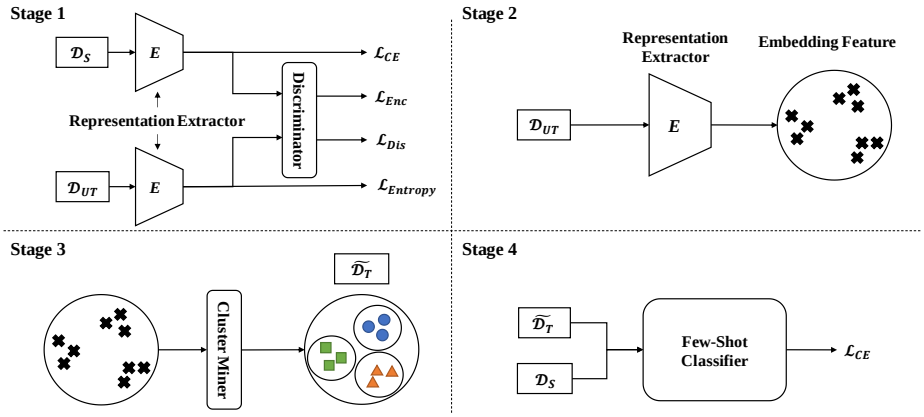


Fig. 1. The overview of our DaFeC framework. In the first stage, we train a representation extractor based on our clustering promotion mechanism and then use it to extract features for unlabeled target-domain data. Next, all unlabeled target-domain instances are grouped with a cluster miner to generate pseudo-labels. In the last stage, the few-shot classifier is trained jointly with the target-domain pseudo-labeled data and the source-domain training data.

4 Methodology

Figure 1 gives an over illustration of our framework, which operates in four stages as follows:

- **Stage 1** Training the representation extractor with clustering promotion mechanism.
- **Stage 2** Extracting the features of unlabeled target-domain data.
- **Stage 3** Using the cluster miner to produce pseudo-labels for unlabeled target-domain data.
- **Stage 4** Training the few-shot classifier based on source-domain data and target-domain data with pseudo labels.

4.1 DaFeC Framework

Representation Extractor The Representation Extractor $\boldsymbol{E}$ is used to extract features $\mathbf{x}$ for each input instance x. Such operations is denoted as $\mathbf{x} = \boldsymbol{E}(x)$. For the subsequent clustering, we use prototypical networks training method [16] and our proposed CPM (Details would be presented in Section 4.2) to train our representation extractor. Following [16], a Prototypical Vector representation for each class in $\mathcal{D}_S$ is generated, by averaging all the examples' representations of that label:

$$\mathbf{c}_i = \frac{1}{K} \sum_{j=1}^{K} \mathbf{x}_i^j, \quad i = 1, 2, \ldots, N, \tag{1}$$

where $\mathbf{c}_i$ refers to the prototype for class y_i and $\mathbf{x}_i^j$ represents the embedding feature of the j-th instance of class y_i. Then the probability of each class for the query instance x can be computed as:

$$P(y = y_i|x) = \frac{\exp(-d(\boldsymbol{E}(x), \mathbf{c}_i))}{\sum_{j=1}^{N} \exp(-d(\boldsymbol{E}(x), \mathbf{c}_j))}, \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidean distance. In the training phase, we expect to minimize the following objective function:

$$\min_{\theta} \mathcal{L}_{CE} = -\mathbb{E}_{x \in \mathcal{D}_S}[\log P(y|x)], \tag{3}$$

When updating the representation extractor with Equation 3, the Euclidean distance between each instance and the prototypical vector of its class could be reduced. As a result, instances with the same class get closer to their class centroid and away from other classes. After training the representation extractor, we use it to embed all unlabeled target-domain data into embedding features for the next clustering stage.

Cluster Miner Given the encoded features of all the target-domain instances in $\mathcal{D}_{UT}$ produced by the trained representation extractor, a cluster miner is deployed to group them into pre-defined $\tilde{N}$ distinct clusters.

Clustering has been widely studied and many approaches have been developed for a variety of circumstances. In our work, we focus on a standard clustering algorithm, k-means [11]. Same as the representation extractor, k-means get the centroid of each cluster by averaging all instances of that and use Euclidean distance to calculate the distance of every instance to their cluster centroid. Therefore, the cluster results grouped by k-Means could reveal the cluster structure generated by the representation extractor better. The subsequent cluster assignments are used as pseudo labels to guide the transformation of unlabeled $\mathcal{D}_{UT}$ to a pseudo-labeled dataset $\tilde{\mathcal{D}}_{\mathcal{T}}$, which is then merged with the source-domain training set $\mathcal{D}_S$ into a new training set $\tilde{\mathcal{D}}_{meta-train} = \{\mathcal{D}_S, \tilde{\mathcal{D}}_{\mathcal{T}}\}$.

Few-shot Classifier The few-shot classifier is trained on $\tilde{\mathcal{D}}_{meta-train}$. Because $\tilde{\mathcal{D}}_{meta-train}$ contains pseudo-labeled target-domain data, the performance of the classifier on the target domain could be improved. Our proposed DaFeC is a generally applicable and model-agnostic framework, which means it is compatible with any existing few-shot classifier. Following previous work [8], we use Proto-CNN, Proto-BERT, BERT-PAIR for the classifier backbone to demonstrate the model-agnostic property of our framework. All settings of these models are the same as the original paper.

4.2 Clustering Promotion Mechanism

Generally, the few-shot classifier learns the information of the target domain by optimizing with pseudo labels created by the cluster miner. While this seems

reasonable, the inevitable label noise caused by the clustering procedure is ignored. Such noisy pseudo labels substantially hinder the model's capability to further improve the classification performance on the target domain. It is generally known that, as a typical machine learning algorithm, clustering depends heavily on the input representations, thus learning discriminative representations is fundamental to the high-confidence pseudo label generation. In order to generate features with more discriminativeness, our framework further incorporates a novel **C**lustering **P**romotion **M**echanism (CPM) into the training process. CPM is built on three components: Similarity Entropy Minimization, Adversarial Distribution Alignment, and Cosine Annealing Strategy. We describe the details of all components below.

Similarity Entropy Minimization Obviously, only if the features of similar instances are close together, the cluster miner can assign them the same pseudo label. In order to promote this similarity without supervision, we introduce the Similarity Entropy Minimization (CPM-S) method.

We first compute the instance-wise distance vector $\mathbf{v}(\mathbf{x})$ for each target instance of $\mathcal{D}_{UT}$ as follows:

$$[\mathbf{v}(\mathbf{x}_i)]_j = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \quad \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_{UT}, \quad i \neq j, \tag{4}$$

where $[\cdot]_j$ means the j-th element of a vector, $\|\cdot\|_2$ means the l_2 norm and $\mathbf{x}_i, \mathbf{x}_j$ are both from $\mathcal{D}_{UT}$. To mine the latent cluster structure, we minimize the entropy of the normalized instance-wise distance vector $\mathbf{v}(\mathbf{x})$ for each target instance

$$\min_{\theta} \mathcal{L}_{Entropy} = \mathbb{E}_{x \sim \mathcal{D}_{UT}}[H(\text{softmax}(\mathbf{v}(\mathbf{x})/\tau))], \tag{5}$$

where $H(\cdot)$ refers to the Shannon entropy over the softmax distribution, $\tau \in \mathbb{R}^+$ is a temperature scaling parameter of the softmax distribution to control the percentage of instances we expect the target data to be similar to. Too small τ sharpens the distribution as one-hot, resulting in several pair-wise clusters while setting too large τ can smooth the distribution to be uniform, making instances get close to dissimilar ones.

Different from the conventional class-level entropy minimization [10] which calculate the entropy over the output logits of the classifier, our similarity entropy minimization over the instance-wise distance vector. Through entropy minimization, the distribution of the instance-wise distance vector will be pushed away from the uniform distribution, which means that each instance is pushed to approach its similar ones and move away from other dissimilar samples. As a result, instances are encouraged to cluster together.

Adversarial Distribution Alignment Naturally, training the representation extractor with Equation 3, the features of source-domain instances have been properly distributed into several distinct clusters. Although in our few-shot scenario, the target domain does not share the same label space with the source

domain, we still observe that they may have some similar classes that can be leveraged to promote target-domain instances to cluster together. Recent efforts [7, 19] have shown that adversarial training can align distributions of two domains, especially per-class distribution alignment. Inspired by this, we introduce the Adversarial Distribution Alignment (CPM-A) method to promote the clustering of target-domain instances by mining similar classes across domains.

First, a domain discriminator $\boldsymbol{D}$ is built to accept features encoded by the representation extractor $\boldsymbol{E}$ and classify whether a data point is drawn from the source or the target domain. Thus, $\boldsymbol{D}$ is optimized according to a standard supervised loss where the labels indicate the origin domain, defined below:

$$\min_{\phi} \mathcal{L}_{Dis} = -\mathbb{E}_{x\sim\mathcal{D}_S}[\log \boldsymbol{D}_\phi(\boldsymbol{E}(x))] - \mathbb{E}_{x\sim\mathcal{D}_{UT}}[\log(1-\boldsymbol{D}_\phi(\boldsymbol{E}(x)))] \quad (6)$$

Second, the representation extractor $\boldsymbol{E}$ playing as the generator is optimized to maximize $\mathcal{L}_{Dis}$, which updates $\boldsymbol{E}$ to generate features to confuse discriminator. This process can be reformulated as follow:

$$\min_{\theta} \mathcal{L}_{Enc} = -\mathbb{E}_{x\sim\mathcal{D}_S}[\log(1-\boldsymbol{D}(\boldsymbol{E}_\theta(x)))] - \mathbb{E}_{x\sim\mathcal{D}_{UT}}[\log \boldsymbol{D}(\boldsymbol{E}_\theta(x))] \quad (7)$$

Theoretically, by iterative optimization of Equation 6 and Equation 7, the domain discriminator $\boldsymbol{D}$ and the representation extractor $\boldsymbol{E}$ are alternated to reach the global optimality that $\boldsymbol{D}$ cannot distinguish between the features of source-domain and target-domain examples produced by $\boldsymbol{E}$. Based on Equation 3, the representation extractor is amended to encode instances of the source domain into cluster-distributed features. After the adversarial training phrase, the target-domain instances with similar ones in the source domain can be aligned with them as clusters.

Cosine Annealing Strategy Unfortunately, CPM-S and CPM-A cannot work well by a simple multi-task learning strategy. In the early training phase, the representation extractor has not learned well, so the produced features are crude and inaccurate. In this time, using CPM-S to promote clustering may make instances get close to dissimilar ones, resulting in undesirable clustering results. With the training procedure going on, we can gradually increase the training weight of CPM-S to mine the latent cluster structure. CPM-A helps to align distributions of source and target domain, but over-alignment could have a detrimental effect on target-domain class separation. Since there still exist some target-domain classes completely different from classes in the source domain, the excessive alignment would mislead these classes to have inappropriate distributions for clustering, which hurts the quality of pseudo labels. Previous work [9] indicates that deep models would memorize easy instances first, and gradually adapt to hard instances as training epochs become large. Thus in the training process, our representation extractor would first align distributions of similar classes between the two domains and then those of dissimilar classes. Therefore, we can decrease the training weight of CPM-A gradually to allow model focus on the alignment of similar cross-domain classes and avoid unwanted over-alignment.

These observations motivate us to develop a Cosine Annealing Strategy (CPM-C) to adjust CPM-S and CPM-A weights in the training process. Specifically, the overall loss function $\mathcal{L}$ is designed as the combination of Equation 3, Equation 7, and Equation 4 as follows:

$$\min_{\theta} \mathcal{L} = \mathcal{L}_{CE} + (1-\lambda)\mathcal{L}_{Enc} + \lambda\mathcal{L}_{Entropy}, \tag{8}$$

where λ is the weighting parameter of $\mathcal{L}_{Entropy}$, which is designed to increase with the training epoch in the form of,

$$\lambda = \begin{cases} -\dfrac{\cos(\pi t/T)+1}{2}, & t \leq T \\ 1, & \text{otherwise} \end{cases}, \tag{9}$$

where t is the current training epoch and T denotes a pre-defined epoch annealing hyperparameter. In the early stage of the training procedure, $\mathcal{L}_{Enc}$ has a larger weight than $\mathcal{L}_{Entropy}$, which makes the representation extractor tend to learn transferable knowledge between domains to improve the ability of encoding target-domain instances. With the training procedure going on, the weight of $\mathcal{L}_{Entropy}$ will increase continually, so CPM-S can be encouraged to promote clustering for the target domain since the representation extractor has learned enough knowledge. Compared to linear annealing, cosine function can pay more attention to CPM-A in the beginning and increase the weight of CPM-S more quickly. We experimentally found that cosine annealing outperforms linear annealing with absolute gains (in classification accuracy) of 2%-3% under different settings.

4.3 Overall Workflow

In this section, we introduce the overall working procedure of our framework DaFeC. Algorithm 1 gives the scratch.

Due to the size imbalance between $\mathcal{D}_S$ and $\mathcal{D}_{UT}$, we use the episodic paradigm proposed by [20] to train the representation extractor. In each iteration, N classes are sampled from $\mathcal{D}_S$ randomly and each class will also randomly select K instances as support instances. In this way, we can obtain the temporary support set S. And we choose other M instances from the same N classes to construct the temporary query set Q. Then the parameters of the representation extractor are optimized with Equation 8. We use S and Q to calculate $\mathcal{L}_{CE}$. For the unlabeled target dataset $\mathcal{D}_{UT}$, we randomly sample $N \times K$ instances to construct the temporary unlabeled set U. S and U are encoded by the representation extractor into low-dimensional features. Then the discriminator takes these features as input and compute $\mathcal{L}_{Dis}$ and updates its parameters with Equation 6. After that, we update the model weights of representation extractor following Equation 8. Once the representation extractor converges, we use it to encode all the instances in $\mathcal{D}_{UT}$ into embedding features $\{\mathbf{x}_{UT}\}$. Next, we apply the k-means algorithm to mine latent cluster structure and assign them pseudo labels

Algorithm 1 DaFeC

Input: Labeled Source-domain Dataset $\mathcal{D}_S$ and Unlabeled Target-domain Dataset $\mathcal{D}_{UT}$
Output: Few-shot Classifier **C**

while not convergence **do**
  Sample S and Q from $\mathcal{D}_S$
  Sample U from $\mathcal{D}_{UT}$
  Update representation extractor $\boldsymbol{E}$ with Equation 3
  Update discriminator $\boldsymbol{D}$ with Equation 6
  Update representation extractor $\boldsymbol{E}$ with Equation 7 and Equation 5
end while
Encode $\mathcal{D}_{UT}$ into feature representations $\{\mathbf{x}_{UT}\}$ using $\boldsymbol{E}$.
Run k-means on $\{\mathbf{x}_{UT}\}$ to generate clusters $\mathcal{C}_{UT}$
Assign each cluster in $\mathcal{C}_{UT}$ a pseudo label to construct pseudo-labeled dataset $\tilde{\mathcal{D}}_{\mathcal{T}}$
Merge $\tilde{\mathcal{D}}_{\mathcal{T}}$ and $\mathcal{D}_S$ into $\tilde{\mathcal{D}}_{meta-train}$
Train few-shot classifier **C** based on $\tilde{\mathcal{D}}_{meta-train}$
return C

to construct pseudo-labeled target-domain dataset $\tilde{\mathcal{D}}_{\mathcal{T}}$, which is merged with the source-domain training set $\mathcal{D}_S$ into a new dataset $\tilde{\mathcal{D}}_{meta-train}$. Finally, the few-shot classifier is trained based on $\tilde{\mathcal{D}}_{meta-train}$.

5 Experiments

5.1 Dataset and Metric

We conduct experiments on the recently widely used benchmark FewRel 2.0 dataset introduced in [8], which is the currently largest unsupervised domain adaptation dataset for few-shot classification. It consists of 44,800 labeled instances (64 classes and 700 instances per class) from Wikipedia (source domain) as the training set and 2500 labeled instances (25 classes and 100 instances per class) from Pubmed (target domain) as the test set. It also provides SemEval-2010 task 8 as the validation set (17 classes and 8,851 instances) and unlabeled PubMed data (2500 instances) for unsupervised domain adaptation. This dataset focus on the relation classification task. Each labeled example is a single sentence, annotated with a head entity, a tail entity, and their relation. The goal is to predict the correct relation between the head and tail.

We investigate our experiments in four few-shot scenarios: 5-way-1-shot, 5-way-5-shot, 10-way-1-shot, 10-way-5-shot and report the mean and standard deviation of test accuracy according to the official evaluation scripts [5].

5.2 Implementation Details

Following [21], we implement the representation extractor $\boldsymbol{E}$ based on a convolutional neural network to encode sentences for relation classification. The window

[5] https://thunlp.github.io/2/fewrel2_da.html

Model	5-Way-1-Shot	5-Way-5-Shot	10-Way-1-Shot	10-Way-5-Shot
Proto-CNN	35.09 ± 0.10	49.37 ± 0.10	22.98 ± 0.05	35.22 ± 0.06
Proto-BERT	40.12 ± 0.19	51.50 ± 0.29	26.45 ± 0.10	36.93 ± 0.01
BERT-PAIR	56.25 ± 0.40	67.44 ± 0.54	43.64 ± 0.46	53.17 ± 0.09
Proto-CNN-ADV	42.21 ± 0.09	58.71 ± 0.06	28.91 ± 0.10	44.35 ± 0.09
Proto-BERT-ADV	41.90 ± 0.44	54.74 ± 0.22	27.36 ± 0.50	37.40 ± 0.36
DaFeC+Proto-CNN	48.58 ± 0.65	65.80 ± 0.44	35.53 ± 0.67	52.71 ± 0.54
DaFeC+Proto-BERT	46.39 ± 0.68	56.32 ± 0.84	32.09 ± 0.98	40.53 ± 0.75
DaFeC+BERT-PAIR	**61.20 ± 0.91**	**76.99 ± 0.82**	**47.63 ± 1.01**	**64.79 ± 0.77**

Table 2. Accuracies (%) of different models on the FewRel 2.0 test set with domain adaptation. Bold marks the highest number among all models. All the results of baseline models are quoted directly from [8]. "DaFeC+" denotes our proposed method.

size of CNN is set to 3, and the number of filters is 230. The discriminator $\boldsymbol{D}$ is implemented as a two-layer feed forward neural network. We use the 50 dimension Glove embeddings [14] to initialize word embeddings. Following [21], we also concatenated the input word embeddings with 5-dimensional position embeddings. The model is trained using stochastic gradient descent with the learning rate of 0.1. $\tilde{N}$ (the number of clusters), τ (the temperature scaling parameter) and T (the epoch annealing parameter) are set as 10, 2 and 6000, respectively. All the hyper-parameters are tuned on the validation set. We run all experiments using PyTorch 1.1.0 on the Nvidia Tesla V100 GPU.

5.3 Baselines

We compare our model against the following 5 models proposed in [8]:

- **Proto-CNN** is a prototypical network using CNN[21] as the encoder.
- **Proto-BERT** is also a prototypical network but it uses BERT [3] as its encoder.
- **Proto-CNN-ADV** is straightforward to combine traditional domain adaptation technique, adversarial training, with few-shot model, Proto-CNN.
- **Proto-BERT-ADV** like Proto-CNN-ADV, simply utilizes adversarial training technique to augment Proto-BERT
- **BERT-PAIR** pairs each query instance with all the supporting instances, and send the paired sequence to the BERT sequence classification model, which is the state-of-the-art on the FewRel 2.0 dataset.

5.4 Results

Table 2 reports the results of our methods (DaFeC+Proto-CNN, DaFeC+Proto-BERT and DaFeC+BERT-PAIR) against other baseline methods. From the results, we can observe that: (1) Over the previous state-of-the-art method BERT-PAIR, DaFeC+BERT-PAIR achieves substantial improvements of 4.95%, 9.55%,

Model	5-Way-1-Shot	5-Way-5-Shot	10-Way-1-Shot	10-Way-5-Shot
DaFeC+BERT-PAIR	61.00	76.83	46.00	65.27
- Pseudo-labeled Data	56.27	68.47	41.20	55.30
- CPM-S	59.07	72.03	43.13	59.43
- CPM-A	58.13	71.43	43.10	60.56
- CPM-C	57.50	69.40	42.63	57.43

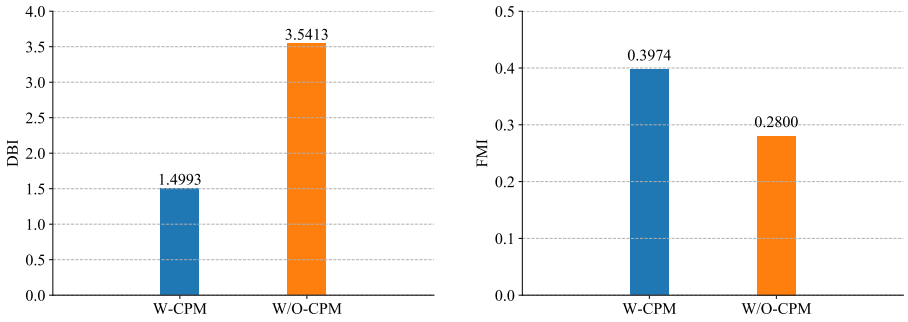
Table 3. An ablation study of our proposed framework on the FewRel 2.0 dataset.

3.99% and 11.62% on four few-shot settings respectively, which confirms the effectiveness and rationality of our proposed training framework. (2) Besides DaFeC+BERT-PAIR, both DaFeC+Proto-CNN and DaFeC+Proto-BERT also exceed Proto-CNN and Proto-BERT significantly. The accuracy of DaFeC+Proto-CNN and DaFeC+Proto-BERT increase 7.11% and 3.48% on average compared to Proto-CNN and Proto-BERT. This demonstrates the model-agnostic property of our framework. (3) Our models with DaFeC clearly perform better than the Proto-CNN-ADV and Proto-BERT-ADV, showing that naive applying UDA to few-shot classification is not as effective as our specifically designed framework. (4) The standard deviations of DaFeC+Proto-CNN, DaFeC+Proto-BERT and DaFeC+BERT-PAIR are slightly larger than the original models because of the pseudo-labeled noise. However, these models still outperform the original ones even considering the worst performance.

6 Analyses

6.1 Ablation Study

To study the contribution of each component in our framework, we run an ablation study (see also Table 3). From these ablations, we find that: (1) Removing the pseudo-labeled target-domain data hurts the result by 4.73%, 8.36%, 4.80% and 9.97% in four scenarios, respectively, which indicates that training the network with clustering-generated pseudo labels is vital for domain adaptation. (2) By introducing the similarity entropy minimization method (CPM-S), we can mine the latent cluster structure of unlabeled target-domain instances, which is beneficial to high-quality pseudo label generation. (3) When we remove CPM-A, the score drops by 3.81% on average, which demonstrates the effectiveness of adversarial distribution alignment over similar classes across domains. (4) When we fix the coefficients in Equation 8 rather than use the cosine annealing strategy to adjust them, the performance declines extremely and is not on par even with only using CPM-S and CPM-A, which powerfully proves that CPM-S and CPM-A cannot work properly without CPM-C.



(a) Davies-Boulbin Index (b) Fowlkes and Mallows Index

Fig. 2. Validation analysis of CPM. The Davies-Boulbin Index (DBI) is used to represent the tightness in a cluster (lower is better) and the Fowlkes and Mallows Index (FMI) is employed to indicate the clustering accuracy (higher is better).

6.2 Effectiveness of CPM

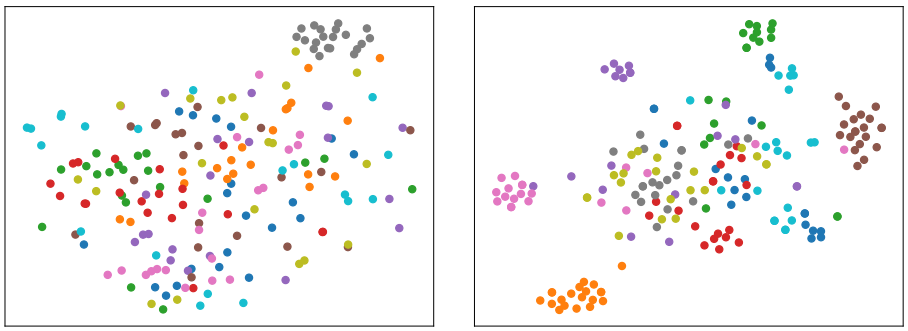
The effectiveness of the learned discriminative feature representations through CPM can be investigated quantitatively and qualitatively.

In order to measure the cohesiveness of intra-clusters and the separation of inter-clusters, we calculate Davies-Boulbin Index (DBI) [2] on the unlabeled target-domain data. We first train two representation extractor: (1) one trained with CPM, named *W-CPM*, (2) the other trained without CPM, named *W/O-CPM*, and use them to encode the unlabeled target-domain instances from the training set respectively. Then the DBI value is calculated based on the cluster results of these features, and the lower, the better. From Figure 2(a), we observe that W-CPM yields a considerably lower DBI value (1.4993) compared with *W/O-CPM* (3.5413).

To examine the accuracy of clustering, we first generate pseudo labels for unlabeled target-domain training data using *W-CPM* and *W/O-CPM*, respectively. Then the Fowlkes and Mallows Index (FMI) [6] score can be obtained by comparing the pseudo labels with the ground truth labels [6], and the higher, the better. The results shown in Figure 2(b) suggest that CPM has indeed improved the clustering effect by increasing the FMI score from 0.28 to 0.3974.

In addition, we visualize the features with t-SNE projected onto 2D embedding space. Specifically, we sample 20 instances for each of 10 classes from the target-domain training data with ground truth labels and extract their features using two representation extractors, *W-CPM* and *W/O-CPM*. Figure 3 provides the visualization of the t-SNE-transformed feature representations. We can observe that for the model without trained with CPM, the features actually are

[6] Note that FewRel 2.0 provides ground-truth labels for partial target-domain training data, but researchers are forbidden to use these labels in the training process for unsupervised domain adaption. And we only use them for experimental analysis.



(a) Representations of W/O-CPM (b) Representations of W-CPM

Fig. 3. A t-SNE plot of the computed feature representations of target-domain instances in the FewRel 2.0 training set. Node colors denote classes.The difference between Figure 3(b) and Figure 3(a) is whether the representation extractor is trained with CPM or not.

mixed and the points with the same classes are distributed in different places. Thus, the pseudo labels generated by the cluster miner may have mush noise. While for the model trained with CPM, the representation exhibits discernible clustering in the projected 2D space. Therefore, the cluster results of the cluster miner could have higher quality.

We draw the conclusion that CPM can enhance the discriminativeness of target-domain feature representations and make those instances distribute as clusters. Therefore, when the cluster miner generates pseudo labels for unlabeled target-domain data according to the features encoded by the model trained with CPM, the pseudo labels could have higher confidence, which may provide more useful target-domain information to improve the domain adaption ability of the few-shot classifier.

6.3 Error Analysis

Although our method achieves state-of-the-art results, we still observe some phenomena which could cause failures. From Figure 3(b), we find that five kinds of colored instances, red (*ingredient_of*), grey (*causative_agent_of*), yellow (*classified_as*), cyan (*gene_plays_role_in_process*) and blue (*biological_process-_involves_gene_product*), fail to show the cluster structure. It reveals that even trained with CPM, the representation extractor still cannot produce discriminative features for all the target-domain data. In other words, some instances belonging to different classes may have similar representations. As a result, the pseudo labels produced by the cluster miner inevitably contain noise that some instances from different classes are assigned the same pseudo label, while other instances from the same class could have distinct labels, which limits the few-

shot classifier as its performance on the target domain is determined by the quality of the pseudo labels.

7 Conclusion

In this paper, we study the problem of inductive unsupervised domain adaption in the few-shot classification. We first train a representation extractor with the Clustering Promotion Mechanism. It uses Similarity Entropy Minimization to promote clustering and Adversarial Distribution Alignment to align similar class distribution across domains. Two methods are combined by the proposed Cosine Annealing Strategy. The representation extractor is used to encode unlabeled target-domain data into features, which are passed to a k-means cluster miner to generate pseudo labels. Finally, we utilize pseudo-labeled target-domain data and labeled source-domain data to train the few-shot classifier. Experimental results demonstrate that our approach achieves new state-of-the-art on FewRel 2.0 dataset. In the future, we will work on how to reduce the noise of pseudo labels to improve the domain adaption performance.

Acknowledgements

We would like to thank reviewers for their insightful comments. This work is supported in part by the National Key Research and Development Program of China (grant No. 2016YFB0801003), the Strategic Priority Research Program of Chinese Academy of Sciences (grant No. XDC02040400) and grant No. BMKY2019B04-1.

References

1. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence (2), 224–227 (1979)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
4. Dong, N., Xing, E.P.: Domain adaption in one-shot learning. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018 (2018)
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017 (2017)
6. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. Journal of the American statistical association **78**(383), 553–569 (1983)
7. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 59:1–59:35 (2015)

8. Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J.: FewRel 2.0: Towards more challenging few-shot relation classification. In: EMNLP (2019)
9. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in neural information processing systems. pp. 8527–8537 (2018)
10. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) NeurIPS (2016)
11. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)
12. Munkhdalai, T., Yu, H.: Meta networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017 (2017)
13. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. CoRR **abs/1803.02999** (2018), http://arxiv.org/abs/1803.02999
14. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
15. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International conference on machine learning. pp. 1842–1850 (2016)
16. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. pp. 4077–4087 (2017)
17. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018 (2018)
18. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 (2017)
19. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2962–2971 (2017)
20. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016)
21. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING (2014)
22. Zou, H., Zhou, Y., Yang, J., Liu, H., Das, H.P., Spanos, C.J.: Consensus adversarial domain adaptation. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019 (2019)