Feature Adaptation of Pre-Trained Language Models across Languages and Domains with Robust Self-Training

Hai Ye[1] **Qingyu Tan**[*1,2] **Ruidan He**[2] **Juntao Li**[3]
Hwee Tou Ng[1] **Lidong Bing**[2]
[1]Department of Computer Science, National University of Singapore
[2]DAMO Academy, Alibaba Group
[3]School of Computer Science and Technology, Soochow University
{yeh,nght}@comp.nus.edu.sg
{qingyu.tan,ruidan.he,l.bing}@alibaba-inc.com
ljt@suda.edu.cn

Abstract

Adapting pre-trained language models (PrLMs) (e.g., BERT) to new domains has gained much attention recently. Instead of fine-tuning PrLMs as done in most previous work, we investigate how to adapt the features of PrLMs to new domains without fine-tuning. We explore unsupervised domain adaptation (UDA) in this paper. With the features from PrLMs, we adapt the models trained with labeled data from the source domain to the unlabeled target domain. Self-training is widely used for UDA, and it predicts pseudo labels on the target domain data for training. However, the predicted pseudo labels inevitably include noise, which will negatively affect training a robust model. To improve the robustness of self-training, in this paper we present class-aware feature self-distillation (CFd) to learn discriminative features from PrLMs, in which PrLM features are self-distilled into a feature adaptation module and the features from the same class are more tightly clustered. We further extend CFd to a cross-language setting, in which language discrepancy is studied. Experiments on two monolingual and multilingual Amazon review datasets show that CFd can consistently improve the performance of self-training in cross-domain and cross-language settings.

1 Introduction

Pre-trained language models (PrLMs) such as BERT (Devlin et al., 2019) and its variants (Liu et al., 2019c; Yang et al., 2019) have shown significant success for various downstream NLP tasks. However, these deep neural networks are sensitive to different cross-domain distributions (Quionero-Candela et al., 2009) and their effectiveness will be much weakened in such a scenario. How to adapt PrLMs to new domains is important. Unlike the most recent work that fine-tunes PrLMs on the unlabeled data from the new domains (Han and Eisenstein, 2019; Gururangan et al., 2020), we are interested in how to adapt the PrLM features without fine-tuning. To investigate this, we specifically study unsupervised domain adaptation (UDA) of PrLMs, in which we adapt the models trained with source labeled data to the unlabeled target domain based on the features from PrLMs.

Self-training has been proven to be effective in UDA (Saito et al., 2017), which uses the model trained with source labeled data to predict pseudo labels on the unlabeled target set for model training. Unlike the methods of adversarial learning (Ganin et al., 2016; Chen et al., 2018) and Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) that learn domain-invariant features for domain alignment, self-training aims to learn discriminative features over the target domain, since simply matching domain distributions cannot make accurate predictions on the target after adaptation (Lee et al., 2019; Saito et al., 2017). To learn discriminative features for the target, self-training needs to retain a model's high-confidence predictions on the target domain which are considered correct for training. Methods like ensemble learning (Zou et al., 2019; Ge et al., 2020; Saito et al., 2017) which adopt multiple models to jointly make decisions on pseudo-label selections have been introduced to achieve this goal. Though these methods can substantially reduce wrong predictions on the target, there will still be noisy labels in the pseudo-label set, with negative effects on training a robust model, since deep neural networks with their high capacity can easily fit to corrupted labels (Arpit et al., 2017).

In our work, to improve the robustness of self-training, we propose to jointly learn discriminative features from the PrLM on the target domain to alleviate the negative effects caused by

[*] Qingyu Tan is under the Joint PhD Program between Alibaba and National University of Singapore.

noisy labels. We introduce class-aware feature self-distillation (CFd) to achieve this goal (§4.2). The features from PrLMs have been proven to be highly discriminative for downstream tasks, so we propose to distill this kind of features to a feature adaptation module (FAM) to make FAM capable of extracting discriminative features (§4.2.1). Inspired by recent work on representation learning (van den Oord et al., 2018; Hjelm et al., 2019), we introduce mutual information (MI) maximization for feature self-distillation (Fd). We maximize the MI between the features from the PrLM and the FAM to make the two kinds of features more dependent. Since Fd can only distill features from the PrLM, it ignores the cluster information of data points which can also improve feature discriminativeness (Chapelle and Zien, 2005; Lee et al., 2019). Hence, for the features output by FAM, if the corresponding data points belong to the same class, we further minimize their feature distance to make the cluster more cohesive, so that different classes will be more separable. To retain high-confidence predictions, we re-rank the predicted candidates and balance the numbers of samples in different classes (§4.1).

We use XLM-R (Conneau et al., 2019) as the PrLM which is trained on over 100 languages. We also extend our method to cross-language, as well as cross-language and cross-domain settings using XLM-R, since it has already mapped different languages into a common feature space. We experiment with two monolingual and multilingual Amazon review datasets for sentiment classification: MonoAmazon for cross-domain and MultiAmazon for cross-language experiments. We demonstrate that self-training can be consistently improved by CFd in all settings (§5.3). Further empirical results indicate that the improvements come from learning lower errors of ideal joint hypothesis (§4.3,5.4).

2 Related Work

Adaptation of PrLMs. Recently, significant improvements on multiple NLP tasks have been enabled by pre-trained language models (PrLMs) (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019c; Howard and Ruder, 2018; Peters et al., 2018). To enhance their performance on new domains, much work has been done to adapt PrLMs. Two main adaptation settings have been studied. The first is the same as what we study in this work: the PrLM provides the features based on which domain adaptation is conducted (Han and Eisenstein, 2019; Cao et al., 2019; Logeswaran et al., 2019; Ma et al., 2019; Li et al., 2020). In the second setting, the corpus for pre-training a language model has large domain discrepancy with the target domain, so in this scenario, we need the target unlabeled data to fine-tune the PrLM after which we train a task-specific model (Gururangan et al., 2020). For example, Lee et al. (2020) and Alsentzer et al. (2019) transfer PrLMs into biomedical and clinical domains. Instead of fine-tuning PrLMs with unlabeled data from the new domain as in most previous work (Rietzler et al., 2019; Han and Eisenstein, 2019; Gururangan et al., 2020), we are interested in the feature-based approach (Devlin et al., 2019; Peters et al., 2019) to adapt PrLMs, which does not fine-tune PrLMs. The feature-based approach is much faster, easier, and more memory-efficient for training than the fine-tuning-based method, since it does not have to update the parameters of the PrLMs which are usually massive especially the newly released GPT-3 (Brown et al., 2020).

Domain Adaptation. To perform domain adaptation, previous work mainly focuses on how to minimize the domain discrepancy and how to learn discriminative features on the target domain (Ben-David et al., 2010). Kernelized methods, e.g., MMD (Gretton et al., 2012; Long et al., 2015), and adversarial learning (Ganin et al., 2016; Chen et al., 2018) are commonly used to learn domain-invariant features. To learn discriminative features for DA, self-training is widely explored (Saito et al., 2017; Ge et al., 2020; Zou et al., 2019, 2018; He et al., 2018). To retain high-confidence predictions for self-training, ensemble methods like tri-training (Saito et al., 2017), mutual learning (Ge et al., 2020) and dual information maximization (Ye et al., 2019) have been introduced. However, the pseudo-label set will still have noisy labels which will negatively affect model training (Arpit et al., 2017; Zhang et al., 2017). Other methods on learning discriminative features include feature reconstruction (Ghifary et al., 2016), semi-supervised learning (Laine and Aila, 2017), and virtual adversarial training (Lee et al., 2019). Based on cluster assumption (Chapelle and Zien, 2005) and the relationship between decision boundary and feature representations, Lee et al. (2019) explore class information to learn discriminative features. Class information is also studied in distant supervision learning for relation extraction (Ye et al., 2017). In NLP, early work explores domain-invariant and

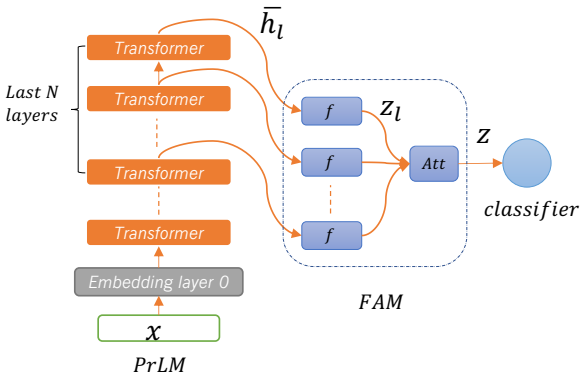

Figure 1: Illustration of our model architecture which includes a pre-trained language model, a feature adaptation module, and a classifier.

domain-specific words to reduce domain discrepancy (Blitzer et al., 2007; Pan et al., 2010; He et al., 2011).

3 Preliminary

In this section, we introduce the problem definition and the model architecture based on which we build our domain adaptation algorithm presented in the next section.

3.1 Unsupervised Domain Adaptation

In order to improve the feature adaptability of pre-trained transformers cross domains, we study unsupervised domain adaptation of pre-trained language models where we train models with labeled data and unlabeled data from the source and target domain respectively. We use the features from PrLMs to perform domain adaptation. Labeled data from the source domain are defined as $S = \{X_s, Y_s\}$, in which every sample $\mathbf{x}_s \in X_s$ has a label $\mathbf{y}_s \in Y_s$. The unlabeled data from the target domain are $T = \{X_t\}$. In this work, we comprehensively study domain adaptation in cross-domain and cross-language settings, based on the features from the multi-lingual PrLM where we adopt XLM-R (Conneau et al., 2019) for evaluation. By using XLM-R, different languages can be mapped into a common feature space. In this work, we evaluate our method on the task of sentiment classification using two datasets.

3.2 Model Architecture

As presented in Figure 1, our model consists of a pre-trained language model (PrLM), a feature adaptation module (FAM), and a classifier.

3.2.1 Pre-trained Language Model

Following BERT (Devlin et al., 2019), most PrLMs consist of an embedding layer and several transformer layers. Suppose a PrLM has $L+1$ layers, layer 0 is the embedding layer, and layer L is the last layer. Given an input sentence $\mathbf{x} = [w_1, w_2, \cdots, w_{|\mathbf{x}|}]$, the embedding layer of the PrLM will encode $\mathbf{x}$ as:

$$\mathbf{h}_0 = \text{Embedding}(\mathbf{x}) \quad (1)$$

where $\mathbf{h}_0 = [\mathbf{h}_0^1, \mathbf{h}_0^2, \cdots, \mathbf{h}_0^{|\mathbf{x}|}]$. After obtaining the embeddings of the input sentence, we compute the features of the sentence from the transformer blocks of PrLM. In layer l, we compute the transformer feature as:

$$\mathbf{h}_l = \text{Transformer}_l(\mathbf{h}_{l-1}) \quad (2)$$

where $\mathbf{h}_l = [\mathbf{h}_l^1, \mathbf{h}_l^2, \cdots, \mathbf{h}_l^{|\mathbf{x}|}]$ and $l \in \{1, 2, \cdots, L\}$. Using all the $|\mathbf{x}|$ features will incur much memory space. After experiments, we take the average of $\mathbf{h}_l$ as:

$$\bar{\mathbf{h}}_l = \frac{1}{|\mathbf{x}|} \sum_{i=1}^{|\mathbf{x}|} \mathbf{h}_l^i \quad (3)$$

and $\bar{\mathbf{h}}_l$ will be fed into the FAM.

3.2.2 Feature Adaptation Module

To transfer the knowledge from the source to the target domain, the features from PrLMs should be more transferable. Previous work points out that the PrLM features from the intermediate layers are more transferable than the upper-layer features, and the upper-layer features are more discriminative for classification (Hao et al., 2019; Peters et al., 2018; Liu et al., 2019b). By making a trade-off between speed and model performance, we combine the last N-layer features from the PrLM for domain adaptation, which is called the multi-layer representation of the PrLM.

Our FAM consists of a feed-forward neural network (followed by a $\tanh$ activation function) and an attention mechanism. We map $\bar{\mathbf{h}}_l$ from layer l into $\mathbf{z}_l$ with the feed-forward neural network:

$$\mathbf{z}_l = f(\bar{\mathbf{h}}_l) \quad (4)$$

Multi-layer Representation. Since feature effectiveness differs from layer to layer, we use an attention mechanism (Luong et al., 2015) to learn to weight the features from the last N layers. We get

the multi-layer representation $\mathbf{z}$ of the PrLM as:

$$\mathbf{z} = E(\mathbf{x};\theta) = \sum_{i=L-N+1}^{L} \alpha_i \mathbf{z}_i$$
$$\alpha_i = \frac{e^{\tanh(\mathbf{W}_{att}\mathbf{z}_i)}}{\sum_{j=L-N+1}^{L} e^{\tanh(\mathbf{W}_{att}\mathbf{z}_j)}} \quad (5)$$

in which $\mathbf{W}_{att}$ is a matrix of trainable parameters. Inspired by Berthelot et al. (2019), we want the model to focus more on the higher-weighted layers, so we further calculate the attention weight as:

$$\alpha_i = \frac{\alpha_i^{1/\tau}}{\sum_{j=L-N+1}^{L} \alpha_j^{1/\tau}} \quad (6)$$

where θ is a set of learnable parameters that includes the parameters from the feed-forward neural network and the attention mechanism.

3.2.3 Classifier

After obtaining the multi-layer representation $\mathbf{z}$, we train a classifier with the source domain labeled set S. We define the loss function for the task-specific classifier as:

$$\mathcal{L}_{pred}^{S} = \frac{1}{|S|} \sum_{\langle \mathbf{x},\mathbf{y} \rangle \in S} l\big(g(E(\mathbf{x};\theta);\phi), \mathbf{y}\big) \quad (7)$$

where g is a classifier that takes in the features out of E, and g is parameterized by ϕ. l is the loss function which is cross-entropy loss in our work.

4 Class-aware Feature Self-distillation for Domain Adaptation

In this section, we introduce our method for domain adaptation. Our domain adaptation loss function takes the form of:

$$\mathcal{L} = \mathcal{L}_{pred}^{S} + \mathcal{L}_{pred}^{T'} + \mathcal{L}_{CFd} \quad (8)$$

in which $\mathcal{L}_{pred}^{S}$ is for learning a task-specific classifier with the source labeled set S (Eq. 7), $\mathcal{L}_{pred}^{T'}$ is the self-training loss trained with the pseudo-label set T' (§4.1), and $\mathcal{L}_{CFd}$ is to enhance the robustness of self-training by learning discriminative features from the PrLM (§4.2), which is the main algorithm for domain adaptation in this work.

4.1 Self-training for Adaptation

We build our adaptation model based on self-training, which predicts pseudo labels on unlabeled target data. The predicted pseudo labels will be used for model training. In the training process, we predict pseudo labels on all the target samples in T. To retain high-confidence predictions from T, we introduce a simple but effective method called *rank-diversify* to build the pseudo-label set T' , which is a subset of T:

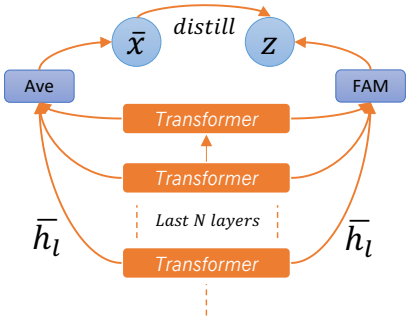

Figure 2: Illustration of feature self-distillation. We take the sum of the last N-layer features for distillation.

Rank. We calculate the entropy loss for every sample in T, specifically:

$$g(\mathbf{z}) = Softmax\big(g(\mathbf{z})\big)$$
$$\mathcal{L}_e(\mathbf{z}) = -\sum g(\mathbf{z})^T \log g(\mathbf{z}) \quad (9)$$

in which $\mathbf{z}$ is the multi-layer feature and g is the classifier in Eq. 7. A lower entropy loss indicates a higher confidence of the model for the pseudo label. Then we use the entropy loss to re-rank T. However, after re-ranking, some classes may have too many samples in the top K candidates, which will bias model training, so we also need to diversify the pseudo labels in the top K list.

Diversify. We classify the samples into different classes with pseudo labels, and re-rank them with entropy loss in ascending order in every class. Samples are selected following the order from every class in turn until K samples are selected.

With the retained pseudo-label set T', we have the loss function for training as:

$$\mathcal{L}_{pred}^{T'} = \alpha \frac{1}{|T'|} \sum_{\langle \mathbf{x},\mathbf{y} \rangle \in T'} l\big(g(E(\mathbf{x};\theta);\phi), \mathbf{y}\big) \quad (10)$$

in which α is a hyper-parameter which will increase gradually in the training process.

4.2 Robust Self-training by Discriminative Feature Learning

To alleviate the negative effects caused by the noisy labels in the pseudo-label set T', we propose to learn discriminative features from the PrLM.

4.2.1 Feature Self-distillation

To maintain the discriminative power of PrLM features, we propose to self-distill the PrLM features into the newly added feature adaptation module (FAM). Similar to traditional knowledge distil-

lation (Hinton et al., 2015), feature distillation in our work is to make the FAM (*student*) also capable of generating discriminative features for adaptation as the PrLM (*teacher*) does. Since the source domain already has the labeled data, there is no need for self-distillation on the source domain, and we apply feature self-distillation (Fd) to the target domain. Inspired by recent work on representation learning (van den Oord et al., 2018; Hjelm et al., 2019; Tian et al., 2020), we propose to use mutual information (MI) maximization for Fd.

MI for Feature Self-distillation. MI measures how different two random variables are. Maximizing the MI between them can reduce their difference. By maximizing the MI between the features from PrLM and FAM, we can make the two features more similar. We are interested in distilling the PrLM features into the multi-layer representation $\mathbf{z}$. We can distill the feature $\bar{\mathbf{h}}_l$ from any layer l into $\mathbf{z}$. However, only distilling one-layer feature of the PrLM may neglect the information from other layers, so we use the sum of the last N-layer features for distillation[1]:

$$\bar{\mathbf{x}} = \sum_{i=L-N+1}^{L} \bar{\mathbf{h}}_i \tag{11}$$

The distillation process is illustrated in Figure 2. Then we maximize the MI $I(\mathbf{z}, \bar{\mathbf{x}})$. We need to find its lower bound for maximization, since it is hard to directly estimate mutual information. Following van den Oord et al. (2018), we also use Noise Contrastive Estimation (NCE) to infer the lower bound as:

$$I(\mathbf{z}, \bar{\mathbf{x}}) \geq \mathcal{J}_{NCE}^{feat}(\mathbf{z}, \bar{\mathbf{x}}) \tag{12}$$

To estimate the NCE loss, we need a negative sample set in which the PrLM features are randomly sampled for the current $\mathbf{z}$. Given a negative sample set $\bar{X}^{neg} = \{\bar{\mathbf{x}}_i^{neg}\}_{i=1}^{|\bar{X}^{neg}|}$, we estimate $\mathcal{J}_{NCE}^{feat}$ as:

$$\mathcal{J}_{NCE}^{feat} = f(\mathbf{z}, \bar{\mathbf{x}}) - \frac{1}{|\bar{X}^{neg}|} \sum_{\bar{\mathbf{x}}_i^{neg} \in \bar{X}^{neg}} f(\mathbf{z}, \bar{\mathbf{x}}_i^{neg}) \tag{13}$$

$f(\mathbf{z}, \bar{\mathbf{x}}^*)$ is the similarity function, defined as:

$$f(\mathbf{z}, \bar{\mathbf{x}}^*) = \cos\left(\inf(\mathbf{z}), \bar{\mathbf{x}}^*\right) \tag{14}$$

where $\bar{\mathbf{x}}^* \in \{\bar{\mathbf{x}}\} \cup \bar{X}^{neg}$; $\inf(\cdot)$ is a trainable feed-forward neural network followed by the $\tanh$ activation, which is to resize the dimension of $\mathbf{z}$ to be equal to $\bar{\mathbf{x}}^*$. To obtain the negative sample set, we select one negative $\bar{\mathbf{x}}$ by randomly shuffling the batch of features which the negative $\bar{\mathbf{x}}$ is in, and this process is repeated $|\bar{X}^{neg}|$ times.

[1]Based on Eq.14, taking the sum or average of the last N-layer features will have the same effect.

4.2.2 Class Information

Feature distillation can only maintain the discriminative power of PrLM features but ignores the class information present in class labels. To explore the class information, when performing feature self-distillation, we further introduce an intra-class loss to minimize the feature distance under the same class. By giving the pseudo-label set T' and the source labeled set S, we group the multi-layer features out of the FAM into different classes. For every class c, we calculate the center feature as $\mathbf{z}_c$. We define the intra-class loss as follows:

$$\mathcal{L}_{intra\_class} = \sum_{c \in C} \sum_{\mathbf{z}_i \in S_c \cup T'_c} \|\mathbf{z}_i - \mathbf{z}_c\|_2 \tag{15}$$

where C is the set of classes. The center feature $\mathbf{z}_c$ for class $c \in C$ is calculated as:

$$\mathbf{z}_c = \frac{1}{|S_c \cup T'_c|} \sum_{\mathbf{z}_j \in S_c \cup T'_c} \mathbf{z}_j \tag{16}$$

Before training for an epoch, the center features will be calculated and fixed during training. After one epoch of training, the center features will be updated. After the above analysis, our final CFd loss becomes:

$$\mathcal{L}_{CFd} = \mathcal{L}_{Fd}^{T} + \mathcal{L}_{C}^{S,T'} = -\sum_{\mathbf{x} \in T} \mathcal{J}_{NCE}^{feat}\big(E(\mathbf{x}; \theta), \bar{\mathbf{x}}\big) + \lambda \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in S \cup T'} \mathcal{L}_{intra\_class} \tag{17}$$

where λ is a hyper-parameter which controls the contribution of $\mathcal{L}_{C}^{S,T'}$.

4.3 Analysis

We provide a theoretical understanding for why CFd can enhance self-training based on the domain adaptation theory from Ben-David et al. (2010).

Theorem 1. (Ben-David et al., 2010) *Let $\mathcal{H}$ be the hypothesis space. With the generalization error δ_s and δ_t of a classifier $G \in \mathcal{H}$ on the source S and target T, we have:*

$$\delta_t(G) \leq \delta_s(G) + d_{\mathcal{H}\Delta\mathcal{H}}(S, T) + \epsilon \tag{18}$$

in which $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the domain discrepancy and is defined as:

$$d_{\mathcal{H}\Delta\mathcal{H}}(S, T) = \sup_{h, h' \in \mathcal{H}} \big|\mathbb{E}_{\mathbf{x} \in S}[h(\mathbf{x}) \neq h'(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \in T}[h(\mathbf{x}) \neq h'(\mathbf{x})]\big| \tag{19}$$

MonoAmazon	E→BK	BT→BK	M→BK	BK→E	BT→E	M→E	BK→BT	E→BT	M→BT	BK→M	E→M	BT→M	Ave.
DAS	67.12	66.53	70.31	58.73	66.14	55.78	51.30	60.76	50.66	55.98	59.06	60.50	60.24
xlmr-tuning	$70.03_{0.2}$	$69.94_{0.9}$	$70.71_{0.8}$	$61.27_{0.5}$	$68.49_{0.4}$	$63.52_{1.0}$	$66.27_{1.3}$	$69.81_{1.2}$	$68.32_{0.6}$	$61.69_{2.5}$	$59.22_{1.1}$	$61.75_{1.9}$	65.92
xlmr-1	64.70	64.26	68.64	53.21	66.39	55.67	57.88	70.10	55.20	61.05	63.92	65.60	63.52
xlmr-10	$70.58_{0.3}$	$69.96_{0.6}$	$71.10_{0.5}$	$59.80_{0.3}$	$70.88_{0.3}$	$64.64_{0.7}$	$63.93_{0.9}$	$72.48_{0.5}$	$65.06_{0.9}$	$65.79_{0.4}$	$67.78_{0.4}$	$63.49_{1.0}$	67.12
KL	$70.91_{0.7}$	$71.12_{0.3}$	$72.10_{0.3}$	$65.61_{0.1}$	$70.30_{0.5}$	$66.85_{0.4}$	$67.69_{0.7}$	$72.68_{0.2}$	$70.36_{0.3}$	$67.66_{0.7}$	$66.46_{1.1}$	$68.56_{1.1}$	69.19
MMD	$71.91_{0.7}$	$73.58_{0.6}$	$70.48_{0.8}$	$69.37_{0.6}$	$71.27_{0.5}$	$65.92_{0.9}$	$\mathbf{71.71}_{0.5}$	$72.81_{0.5}$	$69.30_{0.5}$	$69.24_{0.5}$	$65.87_{1.0}$	$69.14_{1.0}$	70.05
Adv	$71.28_{0.5}$	$69.53_{1.0}$	$72.39_{0.2}$	$61.20_{0.6}$	$69.98_{0.4}$	$66.47_{0.2}$	$63.91_{1.3}$	$72.84_{0.3}$	$70.47_{0.1}$	$66.53_{0.7}$	$67.65_{0.4}$	$64.47_{1.6}$	68.06
p	$70.90_{0.4}$	$71.38_{0.8}$	$72.18_{0.9}$	$64.00_{1.2}$	$70.41_{0.5}$	$67.01_{0.3}$	$67.48_{0.4}$	$71.67_{0.5}$	$\mathbf{70.71}_{0.3}$	$67.16_{0.6}$	$67.92_{1.1}$	$69.77_{0.2}$	69.21
p+CFd	$\mathbf{75.25}_{0.5}$	$\mathbf{74.70}_{0.5}$	$\mathbf{75.08}_{0.6}$	$\mathbf{70.19}_{0.2}$	$\mathbf{72.00}_{0.3}$	$\mathbf{68.96}_{0.3}$	$71.63_{0.4}$	$\mathbf{73.73}_{0.5}$	$70.05_{0.4}$	$\mathbf{70.86}_{0.3}$	$\mathbf{69.80}_{0.7}$	$\mathbf{70.46}_{0.4}$	**71.89**

Table 1: The cross-domain classification accuracy (%) results on MonoAmazon. Models are evaluated by 5 random runs except xlmr-tuning which is run for 3 times. We report the mean and standard deviation results. Best task performance is boldfaced. Results of DAS are taken from He et al. (2018).

DATA	train (S)	valid (S)	test (T)	unlabeled (T)	$\|C\|$
MonoAmazon	5,000	1,000	6,000	6,000	3
MultiAmazon	2,000	2,000	2,000	8,000	2

Table 2: The data splits for the experiments. $|C|$ is the number of classes. $(\cdot)$ denotes the domain which the data comes from.

and ϵ is the error of the ideal joint hypothesis which is defined as:

$$\epsilon = \delta_s(h^*) + \delta_t(h^*) \tag{20}$$

where $h^* = \arg\min_{h\in\mathcal{H}} \delta_s(h) + \delta_t(h)$.

From Ineq. 18, the performance of domain adaptation is bounded by the generalization error on the source domain, domain discrepancy, and the error of the ideal joint hypothesis (joint error). Self-training aims to learn a low joint error by learning discriminative features on the target domain, so that the adaptation performance can be improved (Saito et al., 2017). Our proposed CFd enhances the robustness of self-training by self-distilling the PrLM features and exploring the class information. In this way, the joint error can be further reduced compared to self-training (Fig. 3). Besides, by optimizing the intra-class loss, $d_{\mathcal{H}\Delta\mathcal{H}}$ in Ineq. 18 can be reduced since under the same class, the feature distance of samples from both the source and target domain is minimized (Fig. 4).

5 Experiments

5.1 Datasets

We use two Amazon review datasets for evaluation. One is monolingual and the other is multilingual.

MonoAmazon. This dataset consists of English reviews from He et al. (2018) and has four domains: Book (BK), Electronics (E), Beauty (BT), and Music (M). Each domain has 2,000 positive, 2,000 negative, and 2,000 neutral reviews.

MultiAmazon. This is a multilingual review dataset (Prettenhofer and Stein, 2010) in English, German, French, and Japanese. For every language, there are three domains: Book, Dvd, and Music. Each domain has 2,000 reviews for training and 2,000 for test, with 1,000 positive and 1,000 negative reviews in each set. 6,000 additional reviews form the unlabeled set for each domain. The source domains are only selected from the English corpus.

Table 2 shows the data split. To construct the unlabeled set for the target domain, we use reviews from the test set as the unlabeled data in MonoAmazon following He et al. (2018). For MultiAmazon, reviews from the training set and original unlabeled set both from the target domain are combined.

We also evaluate our model on the benchmark dataset of (Blitzer et al., 2007). The results are presented in Appendix B.

5.2 Experimental Setup

Model Settings. To enable cross-language transfer, we use XLM-R[2] (Conneau et al., 2019) which has 25 layers as the pre-trained language model. The dimension of its token embeddings is 1024 which is mapped into 256 by the FAM. Based on one transfer result, the last 10-layer features are used in FAM. λ for intra-class loss is set as 1 and 2 for MonoAmazon and MultiAmazon respectively. We set the size of negative sample set as 10 and we perform Fd training only in the target domain. τ for attention mechanism in Eq. 6 is set as 0.3. In the training process, we gradually increase the number of retained pseudo labels for self-training, in which we increase the number by 100 for MonoAmazon and 300 for MultiAmazon every epoch. α for $\mathcal{L}^{T'}_{pred}$ is the linear and quadratic function of epoch for MonoAmazon and MultiAmazon respectively. More details of the experimental settings are in Appendix A.

[2] https://github.com/pytorch/fairseq/tree/master/examples/xlmr

MultiAmazon	German			Ave.	French			Ave.	Japanese			Ave.
	Book	Dvd	Music		Book	Dvd	Music		Book	Dvd	Music	
Cross-language												
xlmr-tuning	$91.03_{0.3}$	$88.02_{0.6}$	$90.13_{0.2}$	89.73	$92.12_{0.5}$	$91.17_{0.3}$	$89.58_{0.8}$	90.96	$87.52_{0.5}$	$87.12_{0.4}$	$88.52_{0.7}$	87.72
xlmr-1	73.69	69.86	87.34	76.96	91.26	91.13	88.37	90.25	70.96	71.20	87.07	76.41
xlmr-10	$93.15_{0.8}$	$89.59_{1.2}$	$92.26_{0.6}$	91.67	$93.79_{0.4}$	$93.28_{0.4}$	$92.23_{0.6}$	93.10	$87.13_{1.1}$	$88.63_{0.1}$	$88.05_{0.5}$	87.94
KL	$\mathbf{93.99}_{0.4}$	$91.12_{0.4}$	$\mathbf{93.89}_{0.2}$	93.00	$93.91_{0.1}$	$93.31_{0.3}$	$92.39_{0.2}$	93.20	$88.60_{0.1}$	$88.82_{0.2}$	$88.12_{0.2}$	88.51
MMD	$93.97_{0.1}$	$90.77_{0.8}$	$93.53_{0.4}$	92.76	$93.48_{0.2}$	$93.21_{0.2}$	$92.67_{0.2}$	93.12	$89.17_{0.1}$	$\mathbf{89.22}_{0.1}$	$88.54_{0.4}$	88.98
Adv	$93.27_{0.4}$	$89.78_{0.6}$	$92.53_{0.6}$	91.86	$93.70_{0.4}$	$93.03_{0.4}$	$92.28_{0.3}$	93.00	$88.22_{0.8}$	$88.68_{0.1}$	$88.34_{0.2}$	88.41
p	$92.99_{1.0}$	$89.33_{0.6}$	$93.82_{0.3}$	92.05	$93.81_{0.1}$	$93.00_{0.2}$	$92.50_{0.2}$	93.10	$88.68_{0.3}$	$88.86_{0.1}$	$88.39_{0.1}$	88.64
p+CFd	$93.95_{0.2}$	$\mathbf{91.69}_{0.3}$	$\mathbf{93.89}_{0.2}$	**93.18**	$\mathbf{94.25}_{0.2}$	$\mathbf{93.79}_{0.1}$	$\mathbf{93.39}_{0.1}$	**93.81**	$\mathbf{89.41}_{0.2}$	$88.68_{0.1}$	$\mathbf{89.54}_{0.3}$	**89.21**
Cross-language and Cross-domain												
CLDFA	83.95	83.14	79.02	82.04	83.37	82.56	83.31	83.08	77.36	80.52	76.46	78.11
MAN-MoE	82.40	78.80	77.15	79.45	81.10	84.25	80.90	82.08	62.78	69.10	72.60	68.16
xlmr-tuning	90.84	88.48	89.75	89.69	90.29	90.54	89.65	90.16	85.90	86.02	87.85	86.59
xlmr-1	74.10	77.16	66.52	72.59	87.95	88.00	88.15	88.03	76.46	75.20	65.93	72.53
xlmr-10	91.00	85.95	92.48	89.81	90.17	90.29	92.66	91.04	85.67	85.69	87.89	86.41
KL	93.24	90.39	93.00	92.21	91.98	92.53	92.81	92.44	86.65	88.21	88.61	87.82
MMD	93.44	90.50	92.58	92.17	92.70	92.53	93.07	92.77	87.75	88.25	88.73	88.24
Adv	92.76	88.77	92.80	91.44	91.58	91.70	92.64	91.97	86.88	88.11	88.03	87.67
p	93.11	88.43	92.84	91.46	92.09	92.41	92.52	92.34	87.10	88.22	88.57	87.96
p+CFd	**94.29**	**90.73**	**93.62**	**92.88**	**93.10**	**92.81**	**93.62**	**93.18**	**88.93**	**89.00**	**89.41**	**89.11**

Table 3: The classification accuracy (%) results on MultiAmazon. Models are evaluated by 5 random runs except xlmr-tuning which is run for 3 times. Results of CLDFA and MAN-MoE are taken from Xu and Yang (2017) and Chen et al. (2019) respectively. More detailed transfer results are included in Appendix D.

Baselines. Since we are interested in adapting features of PrLMs without tuning, we mainly set up the baselines that use the features from XLM-R by freezing XLM-R. Trained on the source domain, **xlmr-1** directly tests on the target without domain adaptation and it only uses the last-layer features of XLM-R. **xlmr-10** is the same as xlmr-1, except that it uses the multi-layer representation of XLM-R with last 10-layer features. **KL** (Zhuang et al., 2015) uses the balanced Kullback-Leibler divergence loss to decrease the domain discrepancy for domain adaptation. **MMD** adopts the Maximum Mean Discrepancy loss (Gretton et al., 2012) in which Gaussian Kernel is implemented. **Adv** (Ganin et al., 2016; Chen et al., 2018) adversarially trains a domain classifier to learn domain-invariant features by reversing the gradients from the domain classifier following Ganin et al. (2016). **p** is our self-training method introduced in §4.1. **p+CFd** is our full model that uses CFd to enhance the robustness of self-training. **DAS** (He et al., 2018) uses semi-supervised learning. **CLDFA** (Xu and Yang, 2017) is a cross-lingual baseline which uses cross-lingual resources. **MAN-MoE** (Chen et al., 2019) studies multi-lingual transfer which has multiple languages in the source domain. MoE learns to focus on more transferable source domains for adaptation. xlmr-10, KL, MMD, Adv, p, and p+CFd are all based on the multi-layer representations with last 10-layer features. For KL, MMD, and Adv, to minimize domain discrepancy, we use an unlabeled set of the same size in the source domain as the target domain.

xlmr-tuning[3] first fine-tunes XLM-R with source labeled data using the representation from the final layer [CLS] and being fed to the classifier (Devlin et al., 2019), then tests on the target. By setting up this baseline, we want to see how well the feature-based approach works.

More detailed baseline settings can be found in Appendix A.3.

5.3 Main Results

We conduct experiments in cross-domain (CD), cross-language (CL), and both cross-language and cross-domain (CLCD) settings. Results of CD are evaluated on MonoAmazon (Table 1) and results of CL and CLCD are on MultiAmazon (Table 3). For CL, English is set as the source language. The domains in the source and target languages are the same, i.e., When German&book is the target, the source will be English&book. For CLCD, the sources are also only from English. For example, when the target is German&book, the source language is English and the source domain is dvd or music, in which two sources are set up: English&dvd and English&music, and the two adaptation results are averaged for German&book.

[3]Because of the limited computing resources, we cannot fine-tune XLM-R with unlabeled target data using LM loss.

METHOD	E→BK	BT→BK	M→BK	BK→E	BT→E	M→E	BK→BT	E→BT	M→BT	BK→M	E→M	BT→M	Ave.
xlmr-10	70.41	67.80	70.83	56.47	70.65	64.74	61.30	71.57	65.38	63.33	67.69	64.47	65.16
p	70.90	71.38	72.18	64.00	70.41	67.01	67.48	71.67	70.71	67.16	67.92	69.77	69.21
p+CFd	**75.25**	**74.70**	**75.08**	**70.19**	**72.00**	**68.96**	**71.63**	73.73	70.05	**70.86**	**69.80**	**70.46**	**71.89**
p+C (w/o Fd)	73.16	73.59	74.80	68.72	71.11	68.15	69.80	**74.02**	**71.03**	66.78	69.22	68.93	70.78
p+Fd (w/o C)	71.61	71.10	72.39	67.14	71.23	67.38	69.41	73.04	70.80	68.84	68.14	68.97	70.00
CFd (w/o p)	70.08	72.37	71.30	66.72	70.57	64.21	68.32	72.38	69.27	68.23	66.12	68.37	69.00
Fd (w/o p+C)	68.16	69.55	70.18	66.59	71.02	63.92	69.18	72.10	67.77	69.73	65.71	66.13	68.34

Table 4: The classification accuracy (%) results of p+CFd and its ablations on MonoAmazon.

We have the following findings from Table 1 and 3 based on the overall average scores. **xlmr-10 vs. xlmr-tuning:** xlmr-10 is slightly better than xlmr-tuning which demonstrates the effectiveness of the feature-based approach. **xlmr-1 vs. xlmr-10:** xlmr-10 is much better than xlmr-1 which means our multi-layer representation of XLM-R is much more transferable than the last-layer feature. **xlmr-10 vs. p:** p is consistently better than xlmr-10 which shows our self-training method is effective. **p vs. p+CFd:** After using CFd, p can be consistently improved and p+CFd achieves the best performance among all the methods, which shows the effectiveness of CFd.

5.4 Further Analysis

Ablation Study. We conduct the ablation experiments to see the contributions of feature self-distillation (Fd) and class information (C), which are evaluated on MonoAmazon based on last 10-layer features. By ablating p+CFd, we have four baselines of p+C (w/o Fd), p+Fd (w/o C), CFd (w/o p) and Fd (w/o p+C). From the results in Table 4, p+Fd and p+C perform worse than p+CFd but still better than p, so feature self-distillation and class information both contribute to the improvements of p. Also, by removing the effects of p, CFd and Fd substantially outperform xlmr-10, which means CFd and Fd are both effective for domain adaptation, independent of the self-training method.

Joint errors. Here we study why CFd can enhance self-training and provide empirical results to demonstrate the theoretical understanding in §4.3. By testing on MonoAmazon based on last 10-layer features, Figure 3 presents the joint error results. For example, to find h^* in Eq. 20 for baseline p, following Liu et al. (2019a), we train a classifier using the combined source and target labeled data based on the fixed FAM trained by p. We note that p+Fd and p+C can achieve lower joint errors compared to p, and p+CFd has the best performance, which is consistent with our analysis in §4.3.

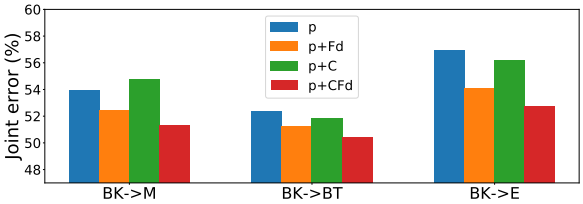

Figure 3: The errors of ideal joint hypothesis tested on MonoAmazon.

METHOD	M→BK	BK→E	M→E	BK→BT	M→BT	BK→M	Ave.
Super	77.54	74.56	74.08	75.66	75.48	72.88	75.03
Fd	76.34	72.44	72.44	74.35	73.15	74.12	73.81

Table 5: The classification accuracy (%) results of in-domain test evaluated on MonoAmazon.

Effects of Feature Self-distillation. We conduct an in-domain test to verify that Fd learns discriminative features from the PrLM. We build a sentiment classification model with in-domain data based on the last 10-layer features. From the same domain in MonoAmazon, we select 4,000 labeled pairs for training, 1,000 for validation, and 1,000 for test. We first pre-train the FAM by Fd using the entire 6,000 raw texts, then we freeze FAM and train a classifier with the training data with features out of FAM. We compare the results with the baseline that directly trains the FAM and classifier with training set (Super). From the results in Table 5, the performances of Fd are very close to Super, showing that the features out of FAM after Fd training are discriminative.

Effects of Class Information. Table 6 presents the average intra-class loss in the training process. By exploring class information, the intra-class loss can be dramatically minimized and accordingly the transfer performances are improved.

$\mathcal{A}$-distance. As an indicator of domain discrep-

BK→M	p	p+C	p+Fd	p+CFd
$\mathcal{L}_{intra\_class}$	966.38	**11.00**	327.72	**12.17**
Acc. (%)	66.59	**68.88**	70.16	**70.95**

Table 6: Effects of class information tested on MonoAmazon with last 10-layer features.

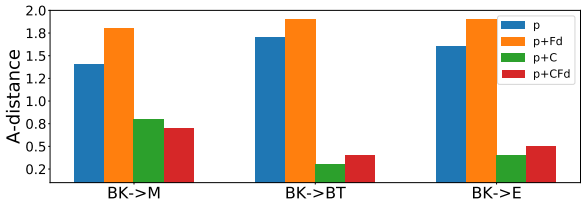

Figure 4: The $\mathcal{A}$-distance tested on MonoAmazon.

METHOD	One layer	last-10		last-20	
		AVE	ATT	AVE	ATT
BK→M	69.51	69.20	**70.07**	66.62	69.31
BK→BT	69.27	67.62	**69.34**	64.16	69.02
BK→E	66.35	64.62	66.71	62.90	**67.08**

Table 7: Study of our attention mechanism based on Fd baseline and tested on MonoAmazon.

ancy, following Saito et al. (2017), we calculate the $\mathcal{A}$-distance based on the last 10-layer features out of FAM trained by method of p or others, and train a classifier to classify the source and target domain data. $d_{\mathcal{A}}$ is equal to $2(1-\delta)$ and δ is the domain classification error. From Figure 4, p+C and p+CFd have much smaller $\mathcal{A}$-distance, which means that the intra-class loss reduces the domain discrepancy. p+Fd has larger $\mathcal{A}$-distance, probably because Fd learns domain-specific information from the target so the domain distance becomes larger.

Effects of Attention Mechanism. We further show whether combining the intermediate-layer features can enhance adaptation. In Table 7, one layer means only using one-layer features for transfer and the results are obtained by using the feature from the most transferable layer. We introduce the attention mechanism to combine the last N-layer features. We demonstrate that using last 10-layer features with attention can achieve better performances. AVE that averages the last N-layer features cannot improve the performance, since it lacks the ability to focus more on effective features.

We also study how the size of negative sample set affects feature distillation and the effects of sharpen on attention mechanism. The analysis is included in Appendix C.

6 Conclusion

In this paper, we study how to adapt the features from the pre-trained language models without tuning. We specifically study unsupervised domain adaptation of PrLMs, where we transfer the models trained in labeled source domain to the unlabeled target domain based on PrLM features. We build our adaptation method based on self-training. To enhance the robustness of self-training, we present the method of class-aware feature self-distillation to learn discriminative features. Experiments on sentiment analysis in cross-language and cross-domain settings demonstrate the effectiveness of our method.

Acknowledgments

This work was supported by Alibaba Group through the Alibaba Innovative Research (AIR) Program.

References

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of ICML*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*.

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Proceedings of NeurIPS*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. 2019. Unsupervised domain adaptation on reading comprehension. *CoRR*, abs/1911.06137.

Olivier Chapelle and Alexander Zien. 2005. Semi-supervised classification by low density separation. In *Proceedings of AISTATS*.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of ACL*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *TACL*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*.

Yixiao Ge, Dapeng Chen, and Hongsheng Li. 2020. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *Proceedings of ICLR*.

Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proceedings of ECCV*.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of EMNLP-IJCNLP*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of BERT. In *Proceedings of EMNLP-IJCNLP*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive semi-supervised learning for cross-domain sentiment classification. In *Proceedings of EMNLP*.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of ACL*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proceedings of ICLR*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of ACL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *Proceedings of ICLR*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. 2019. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of ICCV*.

Juntao Li, Ruidan He, Hai Ye, Hwee Tou Ng, Lidong Bing, and Rui Yan. 2020. Unsupervised domain adaptation of a pretrained cross-lingual language model. In *Proceedings of IJCAI-PRICAI*.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. 2019a. Transferable adversarial training: A general approach to adapting deep classifiers. In *Proceedings of ICML*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019b. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*.

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

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of ACL*.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *Proceedings of ICML*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of WWW*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. *Proceedings of the 4th Workshop on Representation Learning for NLP*.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of ACL*.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. *Dataset Shift in Machine Learning*. The MIT Press.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2019. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. *CoRR*, abs/1908.11860.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of ICML*.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive representation distillation. In *Proceedings of ICLR*.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *Proceedings of ACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*.

Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of ACL*.

Hai Ye, Wenjie Li, and Lu Wang. 2019. Jointly learning semantic parser and natural language generator via dual information maximization. In *Proceedings of ACL*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *Proceedings of ICLR*.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. 2015. Supervised representation learning: Transfer learning with deep autoencoders. In *Proceedings of IJCAI*.

Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of ECCV*.

Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *Proceedings of ICCV*.

A Experimental Settings

A.1 Datasets

We obtain the datasets from He et al. (2018) which can be downloaded online[4]. Then we follow He et al. (2018) to pre-process the datasets which only involves splitting the data into training, validation, and test sets.

A.2 Model Configuration

For MonoAmazon, the learning rate is 0.0001, and the batch size is 50 for classifier training and MI learning. We run 35 times for each baseline except xlmr-1 and xlmr-10 which are run 20 times and the batch size is 100. In epoch 0, we set to retain the top 950 high-confidence predictions for self-training and we increase the number of retained data by 100 every epoch. λ for Fd training is 1.

For MultiAmazon and Benchmark, the learning rate is 0.0005. The batch size for classifier learning is 50 and for MI training is 200. The training epoch is 20. λ for MultiAmazon and Benchmark is 2. In epoch 0, we set to retain the top 1000 high-confidence predictions for self-training. We increase by 150 retained samples every epoch for Benchmark, and by 300 for MultiAmazon.

α for $\mathcal{L}_{pred}^{T'}$ is the linear function of epoch for MonoAmazon, and the quadratic function for MultiAmazon and Benchmark. Adam (Kingma and Ba, 2015) is used for model training. In the training process, if the validation performance does not improve after 10 consecutive epochs, the learning rate will be halved.

For all the datasets, the size of negative sample set is set as 10. τ for attention mechanism is set as 0.3, tuned from $\{0.1, 0.3, 0.5, 0.8, 1.0\}$.

[4] https://github.com/ruidan/DAS

parameter	MonoAmazon	MultiAmazon	Benchmark
learning rate	0.0001	0.0005	0.0005
λ	1	2	2
α	linear function of epoch	quadratic function of epoch	quadratic function of epoch
Max epoch	35	20	20
Size of negative sample set	10	10	10

Table 8: Hyper-parameter settings for main experiments.

DATA	train (S)	valid (S)	test (T)	unlabeled (T)	$\|C\|$
Benchmark	1,600	400	400	6,000	2

Table 9: The data split for training, validation, test, and unlabeled set on Benchmark. $|C|$ is the number of classes.

A.3 Settings for Baselines

KL. The KL-divergence loss (Zhuang et al., 2015) is defined as:

$$\text{KL} = D_{KL}(\xi_s||\xi_t) + D_{KL}(\xi_t||\xi_s) \tag{21}$$

where

$$\begin{aligned} \xi'_s &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_s^i \quad \xi_s = softmax(\xi'_s) \\ \xi'_t &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_t^i \quad \xi_t = softmax(\xi'_t) \end{aligned} \tag{22}$$

in which n is the batch size. We set the weight of KL loss as 500, tuned from {100, 500, 1000, 5000}.

MMD. We use the Gaussian kernel to implement the MMD loss (Gretton et al., 2012). The kernel number is 5. The weight for MMD loss is set to 1, tuned from {1, 0.1, 0.5}

Adv. We follow Ganin et al. (2016) to reverse the gradients from the domain classifier. We set the learning rate for Adv to be the same as the baselines, but set the weight for domain classifier as 0.01, tuned from {1, 0.1, 0.01, 0.001}.

xlmr-tuning. The fine-tuning baseline uses the first [CLS] token as the document representation. The learning rate is 1e-5 and the batch size for gradient update is 32. The fine-tuning models generally overfit the training data in 5 epochs.

B Results on Benchmark

Benchmark. This is a benchmark dataset for domain adaptation (Blitzer et al., 2007), whose reviews are also in English. Four domains are included: Book (B), DVDs (D), Electronics (E), and Kitchen (K). Each domain has 1,000 positive and 1,000 negative reviews. Following He et al. (2018), there are 4,000 unlabeled reviews for each domain. Table 9 summarizes the data split when training on Benchmark. The unlabeled set is the combination of the training set and the original unlabeled set. Table 10 shows the results on Benchmark.

C Further Analysis

Size of Negative Sample Set. We study how the size of negative sample set will affect Fd training. The results are shown in Fig. 5. The method used is xlmr-10+Fd. We find that using a size that is too small or too big is not a good strategy for Fd learning. Size of 10 is a good option for Fd learning.

Effects of Sharpen on Attention Mechanism. In Fig. 6, we show the effects of sharpen mechanism in our attention method which demonstrates that when not using sharpen (τ is ∞), the performance will drop and τ set as 0.3 is a good option for our attention method.

D Full Results on MultiAmazon

Table 11 shows the full results on MultiAmazon.

Benchmark	D→B	K→B	E→B	B→D	K→D	E→D	B→K	D→K	E→K	B→E	D→E	K→E	Ave.
AsyTri	73.20	72.50	73.20	80.70	74.90	72.90	82.50	82.50	86.90	79.80	77.00	84.60	78.39
DAS	82.05	80.05	80.00	82.75	81.40	80.15	82.25	81.50	84.85	81.15	81.55	85.80	81.96
xlmr-1	88.50	78.45	82.50	85.25	80.55	81.80	84.50	81.15	88.45	81.25	79.35	90.05	83.48
xlmr-10	$91.30_{1.0}$	$87.95_{1.0}$	$87.95_{0.3}$	$\mathbf{87.90}_{0.5}$	$87.05_{0.6}$	$86.85_{0.4}$	$90.45_{1.0}$	$87.55_{1.5}$	$92.30_{0.7}$	$88.90_{0.5}$	$89.05_{1.7}$	$91.60_{0.3}$	89.07
KL	$\mathbf{91.50}_{0.8}$	$88.95_{0.6}$	$88.05_{0.5}$	$87.20_{0.6}$	$\mathbf{87.85}_{0.5}$	$87.30_{0.6}$	$90.00_{1.0}$	$91.15_{0.3}$	$92.70_{0.4}$	$89.70_{0.6}$	$90.65_{0.2}$	$91.35_{1.0}$	89.70
MMD	$91.75_{0.5}$	$88.65_{1.1}$	$87.55_{0.9}$	$87.05_{0.7}$	$86.45_{0.3}$	$86.50_{0.6}$	$90.05_{0.3}$	$90.70_{0.5}$	$92.30_{0.3}$	$90.15_{0.3}$	$91.50_{0.6}$	$91.65_{0.7}$	89.53
Adv	$91.40_{0.8}$	$88.10_{0.4}$	$88.15_{0.4}$	$87.70_{1.0}$	$87.35_{0.8}$	$86.65_{0.3}$	$90.65_{0.5}$	$87.55_{1.5}$	$92.25_{0.2}$	$89.25_{0.5}$	$89.80_{1.3}$	$91.60_{0.6}$	89.20
p	$91.40_{0.3}$	$89.50_{0.4}$	$88.20_{0.6}$	$87.40_{0.3}$	$87.15_{0.3}$	$87.05_{0.9}$	$90.00_{0.6}$	$87.55_{1.7}$	$92.60_{0.3}$	$88.85_{0.2}$	$89.65_{1.9}$	$91.85_{0.4}$	89.27
p+CFd	$\mathbf{91.50}_{0.4}$	$\mathbf{89.75}_{0.8}$	$\mathbf{88.65}_{0.4}$	$87.65_{0.1}$	$87.80_{0.4}$	$\mathbf{88.20}_{0.4}$	$\mathbf{92.45}_{0.6}$	$\mathbf{92.45}_{0.2}$	$\mathbf{93.60}_{0.5}$	$\mathbf{91.30}_{0.2}$	$\mathbf{91.55}_{0.3}$	$\mathbf{92.60}_{0.5}$	**90.63**

Table 10: The cross-domain classification accuracy (%) results on Benchmark. Models are evaluated by 5 random runs. We report the mean and standard deviation results. The best task performance is boldfaced. Results of DAS and AsyTri are taken from He et al. (2018) and Saito et al. (2017) respectively. **AsyTri** (Saito et al., 2017) is a self-training baseline with tri-training.

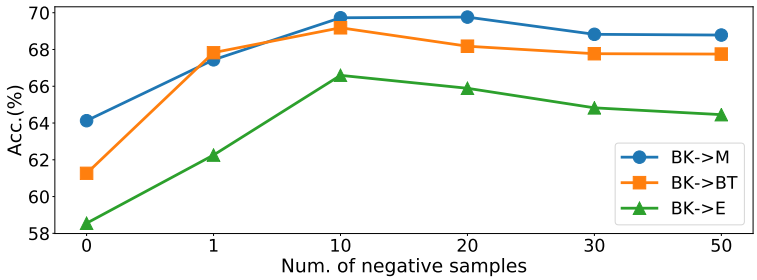

Figure 5: The effects of the negative sample set size for feature self-distillation. Method is xlmr-10+Fd which is evaluated on MonoAmazon.

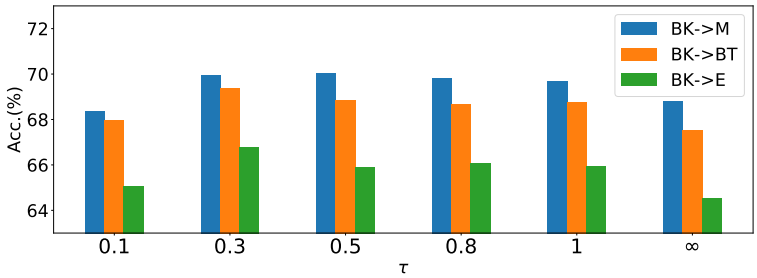

Figure 6: Effects of sharpen on MonoAmazon with method of xlmr-10+Fd.

English → German										
S	book	dvd	music	book	dvd	music	book	dvd	music	
T	book	book	book	dvd	dvd	dvd	music	music	music	Ave.
xlmr-tuning	$91.03_{0.3}$	$91.03_{0.5}$	$90.65_{0.3}$	$88.47_{0.4}$	$88.02_{0.6}$	$88.48_{0.3}$	$89.75_{0.4}$	$89.75_{0.7}$	$90.13_{0.2}$	89.70
xlmr-1	73.69	62.08	86.12	68.03	69.86	86.28	66.4	66.63	87.34	74.05
xlmr-10	$93.15_{0.8}$	$93.79_{0.6}$	$88.20_{1.4}$	$87.22_{1.1}$	$89.59_{1.2}$	$84.68_{1.3}$	$92.33_{1.5}$	$92.63_{0.5}$	$92.26_{0.6}$	90.43
KL	$\mathbf{93.99}_{0.4}$	$93.99_{0.1}$	$92.49_{0.2}$	$90.81_{0.4}$	$91.12_{0.4}$	$89.96_{0.3}$	$93.13_{0.1}$	$92.87_{0.4}$	$\mathbf{93.89}_{0.2}$	92.47
MMD	$93.97_{0.1}$	$93.81_{0.4}$	$93.07_{0.1}$	$90.89_{0.3}$	$90.77_{0.8}$	$90.10_{0.2}$	$92.92_{0.1}$	$92.23_{0.5}$	$93.53_{0.4}$	92.37
Adv	$93.27_{0.4}$	$94.11_{0.6}$	$91.41_{0.3}$	$90.39_{1.2}$	$89.78_{0.6}$	$87.14_{0.4}$	$92.99_{0.2}$	$92.61_{0.4}$	$92.53_{0.6}$	91.58
p	$92.99_{1.0}$	$93.89_{0.5}$	$92.33_{0.1}$	$87.83_{1.5}$	$89.33_{0.6}$	$89.03_{0.6}$	$92.97_{0.3}$	$92.70_{0.3}$	$93.82_{0.3}$	91.65
p+CFd	$93.95_{0.2}$	$\mathbf{94.83}_{0.1}$	$\mathbf{93.74}_{0.2}$	$\mathbf{91.03}_{0.1}$	$\mathbf{91.69}_{0.3}$	$\mathbf{90.42}_{0.4}$	$\mathbf{93.59}_{0.3}$	$\mathbf{93.65}_{0.3}$	$\mathbf{93.89}_{0.2}$	**92.98**

English → French										
S	book	dvd	music	book	dvd	music	book	dvd	music	
T	book	book	book	dvd	dvd	dvd	music	music	music	Ave.
xlmr-tuning	$92.12_{0.5}$	$90.70_{0.3}$	$89.88_{0.9}$	$90.70_{0.4}$	$91.17_{0.3}$	$90.38_{0.5}$	$90.17_{0.5}$	$89.13_{0.5}$	$89.58_{0.8}$	90.43
xlmr-1	91.26	89.44	86.46	89.33	91.13	86.67	87.18	89.11	88.37	88.77
xlmr-10	$93.79_{0.4}$	$92.67_{0.7}$	$87.67_{0.8}$	$93.21_{0.2}$	$93.28_{0.4}$	$87.37_{1.8}$	$92.86_{0.4}$	$92.45_{0.5}$	$92.23_{0.6}$	91.73
KL	$93.91_{0.1}$	$\mathbf{93.59}_{0.2}$	$90.37_{0.2}$	$92.96_{0.3}$	$93.31_{0.3}$	$92.09_{0.2}$	$92.51_{0.7}$	$93.11_{0.1}$	$92.39_{0.2}$	92.69
MMD	$93.48_{0.2}$	$93.55_{0.2}$	$91.85_{0.6}$	$92.85_{0.2}$	$93.21_{0.2}$	$92.21_{0.2}$	$93.34_{0.4}$	$92.80_{0.6}$	$92.67_{0.2}$	92.88
Adv	$93.70_{0.4}$	$93.42_{0.3}$	$89.73_{0.7}$	$93.14_{0.5}$	$93.03_{0.4}$	$90.26_{0.6}$	$92.43_{0.6}$	$92.85_{0.3}$	$92.28_{0.3}$	92.32
p	$93.81_{0.1}$	$93.57_{0.2}$	$90.61_{0.5}$	$93.14_{0.3}$	$93.00_{0.2}$	$91.68_{0.3}$	$92.24_{0.7}$	$92.80_{0.3}$	$92.50_{0.2}$	92.59
p+CFd	$\mathbf{94.25}_{0.2}$	$93.40_{0.3}$	$\mathbf{92.80}_{0.2}$	$93.10_{0.4}$	$\mathbf{93.79}_{0.1}$	$\mathbf{92.51}_{0.1}$	$\mathbf{93.33}_{0.6}$	$\mathbf{93.91}_{0.2}$	$\mathbf{93.39}_{0.1}$	**93.39**

English → Japanese										
S	book	dvd	music	book	dvd	music	book	dvd	music	
T	book	book	book	dvd	dvd	dvd	music	music	music	Ave.
xlmr-tuning	$87.52_{0.5}$	$85.90_{0.6}$	$85.90_{0.4}$	$86.13_{0.3}$	$87.12_{0.4}$	$85.90_{0.4}$	$88.18_{0.2}$	$87.52_{0.4}$	$88.52_{0.7}$	86.96
xlmr-1	70.96	68.18	84.73	64.96	71.2	85.43	61.81	70.04	87.07	73.82
xlmr-10	$87.13_{1.1}$	$87.52_{0.6}$	$83.81_{1.6}$	$87.88_{1.0}$	$88.63_{0.1}$	$83.49_{2.3}$	$88.94_{0.3}$	$86.83_{1.4}$	$88.05_{0.5}$	86.92
KL	$88.60_{0.1}$	$87.53_{0.4}$	$85.76_{0.5}$	$\mathbf{88.88}_{0.3}$	$88.82_{0.2}$	$87.53_{0.2}$	$88.80_{0.4}$	$88.41_{0.2}$	$88.12_{0.2}$	88.05
MMD	$89.17_{0.1}$	$88.20_{0.1}$	$87.29_{0.2}$	$88.80_{0.3}$	$\mathbf{89.22}_{0.1}$	$87.69_{0.5}$	$89.23_{0.3}$	$88.23_{0.5}$	$88.54_{0.4}$	88.49
Adv	$88.22_{0.8}$	$87.72_{0.3}$	$86.04_{0.5}$	$88.64_{0.5}$	$88.68_{0.1}$	$87.57_{0.4}$	$88.17_{1.3}$	$87.89_{0.3}$	$88.34_{0.2}$	87.92
p	$88.68_{0.3}$	$87.95_{0.2}$	$86.25_{0.5}$	$88.77_{0.2}$	$88.86_{0.1}$	$87.67_{0.2}$	$88.89_{0.3}$	$88.25_{0.3}$	$88.39_{0.1}$	88.19
p+CFd	$\mathbf{89.41}_{0.2}$	$\mathbf{88.78}_{0.2}$	$\mathbf{89.08}_{0.1}$	$88.77_{0.5}$	$88.68_{0.1}$	$\mathbf{89.22}_{0.3}$	$\mathbf{89.83}_{0.2}$	$\mathbf{88.98}_{0.2}$	$\mathbf{89.54}_{0.3}$	**89.14**

Table 11: Full classification accuracy (%) results on MultiAmazon. Models are evaluated by 5 random runs except xlmr-tuning which is run for 3 times to save time. We report the mean and standard deviation results. The best task performance is boldfaced.