The Journal of Supercomputing (2023) 79:13724–13743
https://doi.org/10.1007/s11227-023-05191-6

Moka-ADA: adversarial domain adaptation with model-oriented knowledge adaptation for cross-domain sentiment analysis

Maoyuan Zhang[1,2,3] · **Xiang Li**[1,2,3] · **Fei Wu**[1,2,3]

Accepted: 12 March 2023 / Published online: 29 March 2023

Abstract

Cross-domain sentiment analysis (CDSA) aims to overcome domain discrepancy to judge the sentiment polarity of the target domain lacking labeled data. Recent research has focused on using domain adaptation approaches to address such domain migration problems. Among them, adversarial learning performs domain distribution alignment via domain confusion to transfer domain-invariant knowledge. However, this method that transforms feature representations to be domain-invariant tends to align only the marginal distribution, and may inevitably distort the original feature representations containing discriminative knowledge, thus making the conditional distribution inconsistent. To alleviate this problem, we propose adversarial domain adaptation with model-oriented knowledge adaptation (Moka-ADA) for the CDSA task. We adopt the adversarial discriminative domain adaptation (ADDA) framework to learn domain-invariant knowledge for marginal distribution alignment, based on which knowledge adaptation is conducted between the source and target models for conditional distribution alignment. Specifically, we design a dual structure with similarity constraints on intermediate feature representations and final classification probabilities, so that the target model in training learns discriminative knowledge from the trained source model. Experimental results on a publicly available sentiment analysis dataset show that our method achieves new state-of-the-art performance.

Keywords Cross-domain sentiment analysis · Domain adaptation · Adversarial learning · Knowledge distillation

✉ Xiang Li
xli@mails.ccnu.edu.cn

Extended author information available on the last page of the article

1 Introduction

Sentiment analysis aims to judge the sentiment polarity of the given textual data. Recently, with the development of deep networks and pre-trained language model, the performance of sentiment analysis has been greatly improved. Whereas, most existing works heavily rely on a large amount of labeled training data to train separate sentiment classifiers for each domain, which are both time-consuming and labor-intensive to obtain [1]. Thus, it is very necessary to leverage the labeled data-rich domain (source domain) to help sentiment analysis on the labeled data-poor domain (target domain). Therefore, the cross-domain sentiment analysis (CDSA) task becomes a worthy research direction.

The major challenge of CDSA is domain discrepancy between the source and target domains. Domain adaptation is a widely studied field of research that can be effectively used to tackle this problem [2], which can be grouped into three major categories. First, pseudo-labeling techniques [3, 4], use a model trained on the source labeled data to produce pseudo-labels for unlabeled target data and then train a model for the target domain in a supervised manner. Second, pivot-based methods [5, 6], aim to select domain-invariant features and use them as a basis for cross-domain mapping. Third, adversarial training approaches [7, 8], aim to learn a domain-independent mapping for input samples by adding an adversarial cost during model training, that minimizes the distance between the source and target domain distributions.

Adversarial domain adaptation performs adversarial training to confuse the distribution between two domains by maximizing domain difference while minimizing classification error. The representative work includes adversarial discriminative domain adaptation (ADDA) [8], which incorporates discriminative modeling, untied weight sharing, and GAN-based loss. Specifically, the source encoder and classifier are first trained with labeled source data and the source encoder weights are copied to the target encoder. Then, the target encoder and discriminator are alternately optimized in a two-player adversarial game similar to GANs [9]. In terms of its purpose, the discriminator learns to distinguish the source and target domains, while the target encoder learns to fool the discriminator by acquiring domain-invariant knowledge.

Although adversarial training approaches such as ADDA can largely reduce the domain discrepancy, they are flawed when matching the feature distribution of the source domain to that of the target domain, and their discriminability of features may not be guaranteed. As shown in Fig. 1b, they mainly tend to align only the marginal distribution between the two domains to bridge the domain gap. However, this may not be efficient enough, since there is still a conditional distribution inconsistency as shown in Fig. 1c. The reason is that the original feature representations containing discriminative knowledge are distorted, leading to an enlarged error of the ideal joint hypothesis. Based on the domain adaptation theory [10, 11], the error of the ideal joint hypothesis is an explicit quantification of the adaptability between the two domains. When the adaptability is poor,

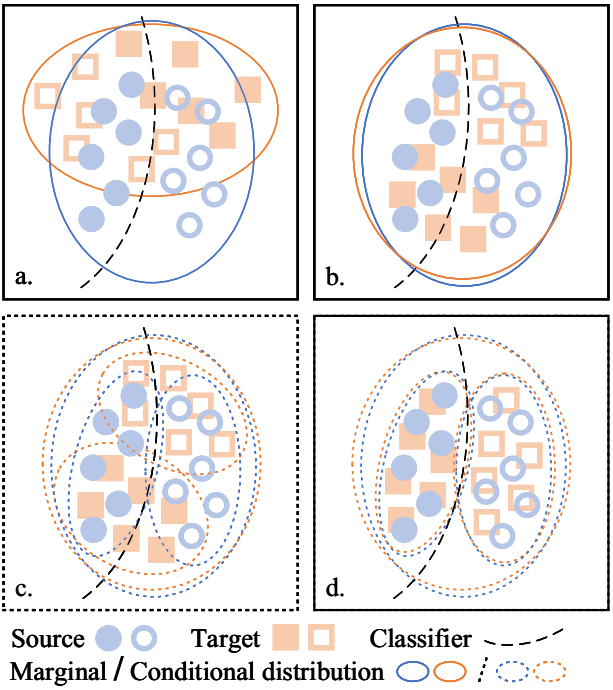

Fig. 1 An illustration of domain adaptation. **a** A classifier trained on the source domain does not apply well to the target domain before domain adaptation. **b** Aligning the marginal distribution via adversarial learning. **c** The inconsistency of the conditional distribution in (**b**) may lead to still high classification error of the target domain. **d** The marginal and conditional distributions are aligned simultaneously by our method

we can hardly expect to learn a classifier with low target error by minimizing the source error as well as the distance between the two domain distributions.

To resolve the above problem, we propose adversarial domain adaptation with model-oriented knowledge adaptation (Moka-ADA) for the CDSA task, which aims to simultaneously align the marginal and conditional distributions as shown in Fig. 1d. In this work, we adopt ADDA as a base adversarial training framework to learn domain-invariant knowledge for marginal distribution alignment. Meanwhile, to learn discriminative knowledge to align conditional distribution, we first consider measuring and minimizing the distance of intermediate feature representations by maximum mean difference (MMD) [12] to reduce domain discrepancy. Wang et al. demonstrate that minimizing MMD leads to an increase in intra-class distance, while the relationship between intra-class and inter-class distances is one decreasing and the other increasing [13]. Thus, we further perform knowledge distillation (KD) [14] at the final classification probabilities to facilitate knowledge transfer, which helps to increase the inter-class distance and thus decrease the intra-class distance. Therefore, we propose the complete model-oriented knowledge adaptation (Moka) module, including intermediate feature representations similarity constraint

(ISC) and final classification probabilities similarity constraint (FSC), which aims to help the target model in training to learn discriminative knowledge from the trained source model, so that the effectiveness of adversarial domain adaptation (ADA) can be improved. In particular, the ablation study indicates that this possibly prevents a mode collapse phenomenon in adversarial training.

The main contributions are summarized as follows:

- We propose a new method, Moka-ADA, to learn domain-invariant and discriminative knowledge to ensure that the marginal and conditional distributions are aligned simultaneously.
- We design a model-oriented knowledge adaptation module containing dual structure with similarity constraints, which enables the target model in training to learn discriminative knowledge from the trained source model.
- We adopt knowledge distillation to facilitate the transfer of discriminative knowledge, which helps to increase inter-class distance and thus reduce intra-class distance, and enhance the stability of adversarial domain adaptation.
- We conduct extensive experiments on the Amazon reviews benchmark datasets with an average accuracy of 94.25%, improving the state-of-the-art performance of the CDSA task by 1.11%.

2 Related work

2.1 Cross-domain sentiment analysis

The CDSA task investigates the problem of cross-domain sentiment transfer. There are many approaches that have been proposed, such as word embedding-based techniques [15, 16], pivot and non-pivot-based methods [17, 18], and domain adaptation-based approaches [19, 20]. Recently, as pre-trained language models have evolved, they have brought tremendous performance improvements in numerous natural language processing tasks including CDSA. Du et al. pose domain adversarial training in the context of pre-trained language model BERT [21]. Karouzos et al. have highlighted the merits of using language modeling as an auxiliary task during fine-tuning [22]. Zhou et al. pre-trains a sentiment-aware language model (SentiX) via domain-invariant sentiment knowledge from large-scale review datasets [23]. In this work, we utilize the pre-trained language model to extract feature representations containing semantic information and then apply them to domain adaptation methods.

2.2 Domain adaptation

Domain adaptation aims to acquire transferable information by reducing domain discrepancy, which is widely used in various cross-domain tasks. Traditionally, the main direction has been to minimize some measure of distance between the source and target feature distributions. Deep Domain Confusion (DDC) [24] introduces an

adaptation layer to minimize maximum mean discrepancy in addition to classification loss on the source data. Deep Adaptation Network (DAN) [25] applies multiple kernels to multiple layers based on previous work. Recently, adversarial training approaches to minimize domain discrepancy have received much attention. Domain Adversarial Neural Network (DANN) [7] proposes a domain binary classification with a gradient reversal layer to train in the presence of domain confusion. Adversarial Discriminative Domain Adaptation (ADDA) [8] trains two feature extractors for the source and target domains respectively, and produces embeddings fooling the discriminator. However, during adversarial training, there is a distortion of the original feature representations containing discriminative knowledge, which will lead to an enlarged error of the ideal joint hypothesis in domain adaptation theory. Based on existing studies, we adopt ADDA as a base adversarial training framework and attempt to improve it further by designing a model-oriented knowledge adaptation module.

2.3 Knowledge distillation

Knowledge distillation (KD) transfers knowledge from a trained teacher model to a student model in training [14]. Originally, KD is a model compression technique that transfers knowledge from a cumbersome model to a tiny model that is more suitable for deployment [27]. But Furlanello et al. found that given the student and teacher models of the same size, it is possible to make the student model outperform the teacher model [28]. Wang et al. point out that hard label is sensitive to incorrectly predicted samples, which may mislead the modeling process of label-induced loss [29]. Zhang et al. utilize softer final classification probabilities for the teacher model as the learning objective for the student model, while adjusting an appropriate distillation temperature to mitigate the negative transfer phenomenon [30]. In our model-oriented knowledge adaptation module, the student and teacher models have the same network structure, and the aligned KD objectives include intermediate feature representations and final classification probabilities, thereby facilitating knowledge transfer.

3 Methodology

3.1 Problem definition and notations

The CDSA task aims to generalize a robust classifier trained on labeled source data to judge the sentiment polarity of unlabeled target data. Let $\mathbb{D}_S$ and $\mathbb{D}_T$ represent the source and target sample distributions, respectively, y_s^d and y_t^d is the corresponding domain label. In the source domain, $\boldsymbol{X}_S = \left\{(\boldsymbol{x}_s^i, y_s^i)\right\}_{i=1}^{n_s}$ are n_s labeled source domain samples, where $\boldsymbol{x}_s$ means a sentence and y_s is the corresponding polarity label, $(\boldsymbol{x}_s, y_s) \sim \mathbb{D}_S$. In the target domain, there is a set of unlabeled samples $\boldsymbol{X}_T = \left\{(\boldsymbol{x}_t^i)\right\}_{i=1}^{n_t}$, where n_t is the number of unlabeled target domain samples, $\boldsymbol{x}_t \sim \mathbb{D}_T$

.

As shown in Fig. 2, the underlying network of our model consists of three components, including two feature extractors E_s and E_t that extract feature representations $\boldsymbol{h}$, a classifier C_s that maps the feature representations $\boldsymbol{h}$ to the classification logits $\boldsymbol{p}$, and a domain discriminator C_d that maps the feature representations $\boldsymbol{h}$ to the domain probabilities $\boldsymbol{q}$.

3.2 Model-oriented knowledge adaptation

To make the target encoder in training learn discriminative knowledge from the trained source encoder, we design a model-oriented knowledge adaptation module , including intermediate feature representations similarity constraint (ISC) and final classification probabilities similarity constraint (FSC).

3.2.1 Intermediate similarity constraints (ISC) based on the reproducing kernel hilbert space

The source and target encoders map the source data to a common feature space to obtain the feature representations, which are then transformed to the reproducing kernel Hilbert space (RKHS) by using kernel functions, for increasing their matching probability in the high-dimensional space. Still, there is no known pairwise correspondence between them, so pairwise testing is not possible. Thus, we can formulate the problem as a two-sample test, and consider measuring the distance by the maximum mean difference (MMD). By minimizing MMD to reduce the distance between intermediate feature representations, the knowledge of the source model is transferred to the target model, resulting in better feature representations and improved generalization ability of the model.

Given the source data $\boldsymbol{x}_s \sim \mathbb{D}_S$, we can obtain the feature representations $\boldsymbol{h}_s = E_s(\boldsymbol{x}_s)$ and $\hat{\boldsymbol{h}}_t = E_t(\boldsymbol{x}_s)$. Let $\boldsymbol{H}_S = \{(\boldsymbol{h}_s^i)\}_{i=1}^n \sim \mathbb{H}_S$, $\boldsymbol{H}_T = \{(\hat{\boldsymbol{h}}_t^i)\}_{i=1}^n \sim \mathbb{H}_T$, where $\mathbb{H}_S$ and $\mathbb{H}_T$ are the respective feature distribution and n is the set cardinality. Thus, the distance between $\mathbb{H}_S$ and $\mathbb{H}_T$ can be defined below:

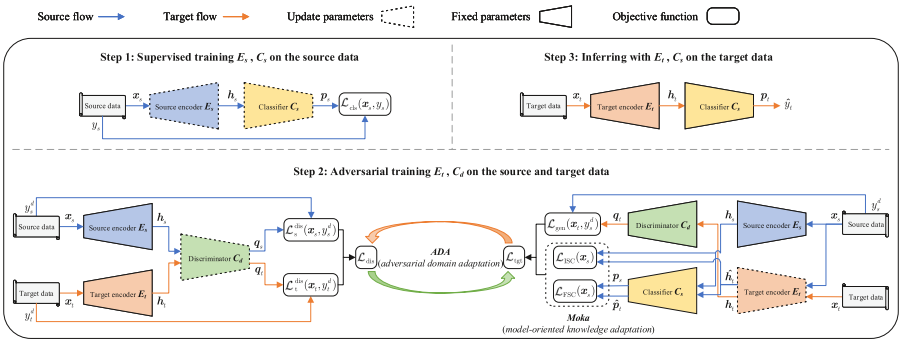

Fig. 2 The overall framework of our proposed method, where E_s and E_t are the feature extractors, C_s is the classifier, and C_d is the domain discriminator; $\boldsymbol{h}$ denotes the feature representations, $\boldsymbol{p}$ denotes the classification logits, and $\boldsymbol{q}$ denotes the domain probabilities

$$
\begin{aligned}
&\mathrm{MMD}[\mathcal{F}, \boldsymbol{h}_s, \hat{\boldsymbol{h}}_t] \\
&= \sup_{\substack{f \in \mathcal{F} \\ \|f\|_{\mathcal{H}} \leq 1}} \left(\mathbb{E}_{\boldsymbol{h}_s \sim \mathbb{H}_S} f(\boldsymbol{h}_s) - \mathbb{E}_{\hat{\boldsymbol{h}}_t \sim \mathbb{H}_T} f(\hat{\boldsymbol{h}}_t) \right) \\
&= \sup_{\substack{f \in \mathcal{F} \\ \|f\|_{\mathcal{H}} \leq 1}} \left(\mathbb{E}_{\boldsymbol{h}_s \sim \mathbb{H}_S} \langle \phi(\boldsymbol{h}_s), f \rangle_{\mathcal{H}} - \mathbb{E}_{\hat{\boldsymbol{h}}_t \sim \mathbb{H}_T} \langle \phi(\hat{\boldsymbol{h}}_t), f \rangle_{\mathcal{H}} \right) \\
&= \sup_{\substack{f \in \mathcal{F} \\ \|f\|_{\mathcal{H}} \leq 1}} \left\langle \mathbb{E}_{\boldsymbol{h}_s \sim \mathbb{H}_S} \phi(\boldsymbol{h}_s) - \mathbb{E}_{\hat{\boldsymbol{h}}_t \sim \mathbb{H}_T} \phi(\hat{\boldsymbol{h}}_t), f \right\rangle_{\mathcal{H}} \\
&= \left\| \mathbb{E}_{\boldsymbol{h}_s \sim \mathbb{H}_S} \phi(\boldsymbol{h}_s) - \mathbb{E}_{\hat{\boldsymbol{h}}_t \sim \mathbb{H}_T} \phi(\hat{\boldsymbol{h}}_t) \right\|_{\mathcal{H}},
\end{aligned}
\tag{1}
$$

where $\mathcal{H}$ is a RKHS, function class $\mathcal{F} = \{f : \|f\| \leq 1\}$, and infinite dimensional feature map $\phi(\cdot) : \mathcal{X} \to \mathcal{H}$. In addition, the feature map $\phi(\cdot)$ corresponds to a positive semi-definite kernel k so that $k(\boldsymbol{u}, \boldsymbol{v}) = \langle \phi(\boldsymbol{u}), \phi(\boldsymbol{v}) \rangle_{\mathcal{H}}$, thus Eq. (1) can be rewritten in terms of k. Therefore, the objective function of similarity constraints in the **"intermediate"** can be written as:

$$
\begin{aligned}
&\min_{E_t} \mathcal{L}_{\mathrm{ISC}}(\boldsymbol{x}_s) \\
&\quad = \mathrm{MMD}^2[\mathcal{F}, \boldsymbol{h}_s, \hat{\boldsymbol{h}}_t] \\
&\quad = \left\| \mathbb{E}_{\boldsymbol{h}_s \sim \mathbb{H}_S} \phi(\boldsymbol{h}_s) - \mathbb{E}_{\hat{\boldsymbol{h}}_t \sim \mathbb{H}_T} \phi(\hat{\boldsymbol{h}}_t) \right\|_{\mathcal{H}}^2 \\
&\quad = \mathbb{E}_{\boldsymbol{h}_s, \boldsymbol{h}'_s \sim \mathbb{H}_S, \mathbb{H}_S} k(\boldsymbol{h}_s, \boldsymbol{h}'_s) \\
&\quad - 2\mathbb{E}_{\boldsymbol{h}_s, \hat{\boldsymbol{h}}_t \sim \mathbb{H}_S, \mathbb{H}_T} k(\boldsymbol{h}_s, \hat{\boldsymbol{h}}_t) \\
&\qquad + \mathbb{E}_{\hat{\boldsymbol{h}}_t, \hat{\boldsymbol{h}}'_t \sim \mathbb{H}_T, \mathbb{H}_T} k(\hat{\boldsymbol{h}}_t, \hat{\boldsymbol{h}}'_t),
\end{aligned}
\tag{2}
$$

where $\boldsymbol{h}'_s$ is an independent copy of $\boldsymbol{h}_s$ with the same distribution, and $\hat{\boldsymbol{h}}'_t$ is an independent copy of $\hat{\boldsymbol{h}}_t$. As for the kernel function k, we choose to use a linear combination of multiple Gaussian kernels over a range of standard deviations, such as $k(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=1}^{m} \exp\left\{ -\frac{1}{2\delta_i} \|\boldsymbol{u} - \boldsymbol{v}\|_2^2 \right\}$, where m is the number of kernel functions and δ_i denotes the standard deviation of the i-th Gaussian kernel.

3.2.2 Final similarity constraints (FSC) based on the knowledge distillation

The trained classifier will receive the feature representations and map them to the classification logits for judgment. The traditional training directly takes one-hot encoded labels as the target, which is prone to result in overfitting during repeated training epochs. To alleviate this problem, we utilize knowledge distillation (KD) to control the degree of knowledge transfer by producing a softer probability distribution. Unlike the

hard label, which focuses only on the label value of maximum probability, the soft label describes the probability distribution by multiple probability values, which can better handle noise and uncertainty. Moreover, it contains information about the correlation between different classes, which can help to increase the inter-class distance and thus reduce the intra-class distance.

Given the acquired feature representations $\boldsymbol{h}_s$ and $\hat{\boldsymbol{h}}_t$, the trained classifier C_s will map them to the classification logits $\boldsymbol{p}_s = C_s(\boldsymbol{h}_s)$ and $\hat{\boldsymbol{p}}_t = C_s(\hat{\boldsymbol{h}}_t)$, respectively. As with KD, we obtain the softer classification probabilities $\boldsymbol{P} = \sigma(\boldsymbol{p}_s/T)$ and $\boldsymbol{Q} = \sigma(\hat{\boldsymbol{p}}_t/T)$, where $\sigma(\cdot)$ is the softmax function and T is temperature value that controls the degree of knowledge transfer. Therefore, the objective function of similarity constraints in the **"final"** can be conducted by using the Kullback–Leibler divergence between $\boldsymbol{P}$ and $\boldsymbol{Q}$:

$$\begin{aligned} &\min_{E_t} \mathcal{L}_{\mathrm{FSC}}(\boldsymbol{x}_s) \\ &\quad = T^2 \cdot \mathrm{KL}(\boldsymbol{P} \| \boldsymbol{Q}) \\ &\quad = T^2 \cdot \mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{D}_S} \sum_{k=1}^{K} P_k \log \frac{P_k}{Q_k}, \end{aligned} \tag{3}$$

where $\boldsymbol{P} \triangleq [P_1, \cdots, P_K] \in \mathbb{R}^{1\times K}$, $\sum_{k=1}^{K} P_k = 1$ and $\boldsymbol{Q} \triangleq [Q_1, \cdots, Q_K] \in \mathbb{R}^{1\times K}$, $\sum_{k=1}^{K} Q_k = 1$, P_k and Q_k is the probability of the k-th class, and K is the number of classes.

In summary, the inputs to the source and target encoders are the same, and the target encoder imitates the source encoder in terms of **"intermediate"** and **"final"**, thereby transferring discriminative knowledge for conditional distribution alignment.

3.3 Adversarial domain adaptation with model-oriented knowledge adaptation

In order to compensate for the deficiencies of adversarial domain adaptation in discriminative knowledge via model-oriented knowledge adaptation, we propose the Moka-ADA, which guarantees that both domain-invariant knowledge and discriminative knowledge are fully learned. Figure 2 illustrates the overall framework of our proposed model, which consists of three steps. **Step 1**: Supervised training the source encoder E_s and classifier C_s on the source data. **Step 2**: Adversarial training the target encoder E_t and discriminator C_d to align the source and target domain distributions. **Step 3**: Inferring with the trained target encoder E_t and classifier C_s on the target data.

In **Step 1**, we aim to train a well-performing source model using labeled data from the source domain, which serves as a "teacher" for subsequent training of the target model. The source error can be minimized through supervised training of the source encoder E_s and classifier C_s on $(\boldsymbol{x}_s, y_s)$ by using the Cross-Entropy loss:

$$\begin{aligned} &\min_{E_s, C_s} \mathcal{L}_{\mathrm{cls}}(\boldsymbol{x}_s, y_s) \\ &\quad = \mathbb{E}_{(\boldsymbol{x}_s, y_s) \sim \mathbb{D}_S} - \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log \sigma(\boldsymbol{p}_s), \end{aligned} \tag{4}$$

where $\boldsymbol{p}_s = C_s(\boldsymbol{h}_s)$, $\boldsymbol{h}_s = E_s(\boldsymbol{x}_s)$, $\sigma(\cdot)$ is the softmax function, and K is the number of classes.

Then, the source encoder parameters are frozen, which fixes the source domain feature distribution. Thus, we obtained the reference distribution for adversarial training, which is analogous to the real image distribution in the GANs setting [9]. Prior to adversarial training, we first initialize the target encoder weights with the source encoder weights, as this practice can improve the convergence properties.

In **Step 2**, the discriminator C_d aims to infer the domain probabilities $\boldsymbol{q}_s$ or $\boldsymbol{q}_t$ of an sample, i.e., coming from the source or target domain. Thus, the discriminator C_d is optimized on $(\boldsymbol{x}_s, y_s^d = 0)$ and $(\boldsymbol{x}_t, y_t^d = 1)$:

$$\begin{aligned} &\min_{C_d} \mathcal{L}_{\mathrm{s}}^{\mathrm{dis}}(\boldsymbol{x}_s, y_s^d) \\ &\quad = \mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{D}_S} - [y_s^d \log \boldsymbol{q}_s + (1 - y_s^d) \log(1 - \boldsymbol{q}_s)] \\ &\quad = \mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{D}_S} - \log(1 - \boldsymbol{q}_s), \end{aligned} \tag{5}$$

where $\boldsymbol{q}_s = C_d(\boldsymbol{h}_s)$, $\boldsymbol{h}_s = E_s(\boldsymbol{x}_s)$, and

$$\begin{aligned} &\min_{C_d} \mathcal{L}_{\mathrm{t}}^{\mathrm{dis}}(\boldsymbol{x}_t, y_t^d) \\ &\quad = \mathbb{E}_{\boldsymbol{x}_t \sim \mathbb{D}_T} - [y_t^d \log \boldsymbol{q}_t + (1 - y_t^d) \log(1 - \boldsymbol{q}_t)] \\ &\quad = \mathbb{E}_{\boldsymbol{x}_t \sim \mathbb{D}_T} - \log \boldsymbol{q}_t, \end{aligned} \tag{6}$$

where $\boldsymbol{q}_t = C_d(\boldsymbol{h}_t)$, $\boldsymbol{h}_t = E_t(\boldsymbol{x}_t)$.

According to Eqs. (5) and (6), we can obtain the final objective function of the discriminator C_d:

$$\begin{aligned} &\min_{C_d} \mathcal{L}_{\mathrm{dis}}(\boldsymbol{x}_s, \boldsymbol{x}_t, y_s^d, y_t^d) \\ &\quad = \min_{C_d} \left[\frac{\mathcal{L}_{\mathrm{s}}^{\mathrm{dis}}(\boldsymbol{x}_s, y_s^d) + \mathcal{L}_{\mathrm{t}}^{\mathrm{dis}}(\boldsymbol{x}_t, y_t^d)}{2} \right] \\ &\quad = \frac{\mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{D}_S} - \log(1 - \boldsymbol{q}_s) + \mathbb{E}_{\boldsymbol{x}_t \sim \mathbb{D}_T} - \log \boldsymbol{q}_t}{2}. \end{aligned} \tag{7}$$

To adversarially train the target encoder E_t, it is encouraged to fool the discriminator C_d by reversing the domain label. Thus, the target encoder E_t is optimized on $(\boldsymbol{x}_t, y_s^d = 0)$:

$$\begin{aligned} &\min_{E_t} \mathcal{L}_{\mathrm{gen}}(\boldsymbol{x}_t, y_s^d) \\ &\quad = \mathbb{E}_{\boldsymbol{x}_t \sim \mathbb{D}_T} - [y_s^d \log \boldsymbol{q}_t + (1 - y_s^d) \log(1 - \boldsymbol{q}_t)] \\ &\quad = \mathbb{E}_{\boldsymbol{x}_t \sim \mathbb{D}_T} - \log(1 - \boldsymbol{q}_t), \end{aligned} \tag{8}$$

where $\boldsymbol{q}_t = C_d(\boldsymbol{h}_t)$, $\boldsymbol{h}_t = E_t(\boldsymbol{x}_t)$.

Based on Eq. (2) and Eq. (3) in Sect. 3.2 and Eq. (8), the final objective function for training the target encoder E_t can be defined as:

$$\begin{aligned}&\min_{E_t} \mathcal{L}_{\text{tgt}}(\boldsymbol{x}_s, \boldsymbol{x}_t, y_s^d)\\&= \min_{E_t}[\mathcal{L}_{\text{gen}}(\boldsymbol{x}_t, y_s^d) + \mathcal{L}_{\text{ISC}}(\boldsymbol{x}_s) + \mathcal{L}_{\text{FSC}}(\boldsymbol{x}_s)].\end{aligned} \tag{9}$$

Through Eq. (7) and Eq. (9), the discriminator C_d and target encoder E_t are alternately optimized in a two-player adversarial game similar to GANs [9], as in the ADDA framework [8].

In **Step 3**, we can finally use the trained target encoder E_t and classifier C_s to make inferences on the target data used for testing, whose sentiment polarity label can be predicted as below:

$$\hat{y}_t = \arg\max \boldsymbol{p}_t, \tag{10}$$

where $\boldsymbol{p}_t = C_s(\boldsymbol{h}_t), \boldsymbol{h}_t = E_t(\boldsymbol{x}_t)$.

The overall iterative training procedure of Moka-ADA is summarized in Algorithm 1.

Algorithm 1 Training procedure of Moka-ADA

Require: $\boldsymbol{X}_S$, $\boldsymbol{X}_T$, E_s, E_t, C_s, C_d

Step 1: Supervised training E_s, C_s on $\boldsymbol{X}_S$
for $i \in [1, pre\_epochs]$ **do**
  for minibatch $\boldsymbol{B}_S \in \boldsymbol{X}_S$ **do**
    compute $\mathcal{L}_{\text{cls}}$ based on Eq. (4)
    update E_s, C_s to minimize $\mathcal{L}_{\text{cls}}$
  end for
end for
Step 2: Adversarial training E_t, C_d on $\boldsymbol{X}_S$, $\boldsymbol{X}_T$
fix E_s, C_s, initialize E_t with E_s
for $j \in [1, adapt\_epochs]$ **do**
  for minibatch $\boldsymbol{B}_S \in \boldsymbol{X}_S, \boldsymbol{B}_T \in \boldsymbol{X}_T$ **do**
    compute $\mathcal{L}_{\text{dis}}$ based on Eq. (7)
    update C_d to minimize $\mathcal{L}_{\text{dis}}$
    compute $\mathcal{L}_{\text{tgt}}$ based on Eq. (9)
    update E_t to minimize $\mathcal{L}_{\text{tgt}}$
  end for
end for
Step 3: Inferring with E_t, C_s on $\boldsymbol{X}_T$
for minibatch $\boldsymbol{B}_T \in \boldsymbol{X}_T$ **do**
  compute $\hat{y}_t$ based on Eq. (10)
end for

3.4 Theoretical analysis

We provide a theoretical understanding of why our method can enhance adversarial domain adaptation based on the domain adaptation theory from Ben-David et al. [10, 11], a key outcome of which is the following theorem:

Theorem 1. *Let $\mathcal{H}$ be the hypothesis space, ϵ_S and ϵ_T be the generalization error on the source domain $\mathbb{D}_S$ and the target domain $\mathbb{D}_T$, respectively. Then for any $h \in \mathcal{H}$, there is*

$$\epsilon_T(h) \leq \epsilon_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{D}_S, \mathbb{D}_T\right) + \lambda, \tag{11}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ is the $\mathcal{H}\Delta\mathcal{H}$-divergence [31] to measure the domain discrepancy between $\mathbb{D}_S$ and $\mathbb{D}_T$, defined as:

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}} \triangleq \sup_{h,h' \in \mathcal{H}} |&\mathbb{E}_{\mathbf{x}_s \sim \mathbb{D}_S}\left[h(\mathbf{x}_s) \neq h'(\mathbf{x}_s)\right] \\ &-\mathbb{E}_{\mathbf{x}_t \sim \mathbb{D}_T}\left[h(\mathbf{x}_t) \neq h'(\mathbf{x}_t)\right]|, \end{aligned} \tag{12}$$

where h and h' are two sets of hypotheses in $\mathcal{H}$, and λ is the error of the ideal joint hypothesis h^*, where *h^* is defined as $h^* = \arg\min_{h \in \mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$, such that*

$$\lambda = \epsilon_S(h^*) + \epsilon_T(h^*). \tag{13}$$

From Eq. (11), the generalization error on the target domain $\epsilon_T(h)$ is upper bounded by a combination of the generalization error on the source domain $\epsilon_S(h)$, the domain discrepancy $d_{\mathcal{H}\Delta\mathcal{H}}$, and the error of the ideal joint hypothesis λ. First, it is easy to minimize $\epsilon_S(h)$ by supervised training with labeled source data. Then, $d_{\mathcal{H}\Delta\mathcal{H}}$ can be reduced by aligning the marginal distribution via adversarial domain adaptation. Moreover, the dual structure with similarity constraints can yield lower λ and further reduce $d_{\mathcal{H}\Delta\mathcal{H}}$ by acquiring discriminative knowledge for conditional distribution alignment.

4 Experiments

4.1 Datasets

We evaluate our method on the Amazon reviews benchmark datasets collected by Blitzer et al. [32], which is publicly available and widely used for the CDSA task. It includes reviews from four product domains: Books (B), DVDs (D), Electronics (E), and Kitchen appliances (K). Each domain contains 2000 labeled samples, of which 1000 are negative and 1000 are positive. Following the previous works [22, 33], we construct 12 cross-domain tasks of source-target domain pairs. For each domain pair, 1600 labeled source samples and the same number of unlabeled target samples are used for training, and the remaining 400 labeled source samples for validation. Then, we perform a test with all the labeled target samples. Table 1 lists the relevant statistics.

4.2 Implementation details

We adopt SentiX as the context feature extractor, which is a sentiment-aware pre-trained language model proposed by Zhou et al. [23]. For all experiments, we limit the maximum sequence length is 256, while the batch size is set to 32. The optimizer is Adam with learning rate 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. During the supervised training, we train for 5 epochs and use the validation dataset to choose an appropriate epoch to save the model. For adversarial training, we train for 1 to 5 epochs to report the average results and empirically set some hyperparameters with a gradient norm of 1.0, a clip value of 0.01, and a knowledge distillation temperature of 20 for more stable adversarial training.

4.3 Compared methods

We consider the following methods for comparison, including **PERL** [34], **DAAT** [21], **p+CFd** [35], **UDALM** [22], **DA-SDS** [33], and **AdSPT** [36]. We present the best results reported in the original paper of these approaches. Besides, we adopt the SentiX model as a baseline and design several variants of our model:

- **Baseline**: The sentiment-aware pre-trained language model SentiX.
- **ISC-ADA**: A variant of the proposed model, which only imposes similarity constraints on intermediate feature representations.
- **FSC-ADA**: A variant of the proposed model, which only imposes similarity constraints on final classification probabilities.
- **Moka-ADA**: The full model introduced in Sect. 3.3.

4.4 Experimental results

In Table 2, we report the accuracy results of the compared methods on 12 cross-domain tasks. Compared with most other works, the baseline achieves better performance, which is mainly attributed to its learning of sentiment knowledge through pre-training with large-scale review datasets. Notably, our Moka-ADA can improve the average accuracy by 1.57% compared to the baseline and has an improvement of 6.75%, 4.13%, 3.62%, 2.51%, 2.77% and 1.11% compared to other methods, respectively.

Table 1 Statistics of the Amazon reviews benchmark datasets

Domain	Class	Positive	Negative	Train	Validation	Test
Books	2	1000	1000	1600	400	2000
DVDs	2	1000	1000	1600	400	2000
Electronics	2	1000	1000	1600	400	2000
Kitchen	2	1000	1000	1600	400	2000

As shown in Fig. 3, it can be observed that our methods outperform the baseline in almost all domain pairs, which proves that either ISC-ADA or FSC-ADA can effectively conduct similarity constraints to enhance adversarial domain adaptation. Compared to ISC-ADA and FSC-ADA, the full Moka-ADA performed better on 7 of the 12 domain pair tasks, and has mostly relatively smaller standard deviations, indicating greater robustness.

4.5 Visualization of features

To more intuitively assess the effect of model-oriented knowledge adaptation on the feature distribution, we further visualize the feature representations of the source and target data for the B → D task. The visualization of the feature representations is performed using the t-SNE algorithm to transform the 768-dimensional feature space into a two-dimensional space. In Fig. 4, the visualization results of Baseline, ISC-ADA, FSC-ADA, and Moka-ADA are presented separately.

In Fig. 4a, we observe that samples of different polarities in the source domain are well separated, while for the target domain, some samples of different polarities are mixed together with unclear decision boundaries. In Fig. 4b, the situation has improved and samples of the same polarity across domains tend to be consistent, indicating that ISC-ADA reduces the distance between feature representations across domains and thereby reduces discrepancy in domain distributions. In Fig. 4c, although samples of the same polarity across domains are less aligned, FSC-ADA increases the inter-class distance and reduces the intra-class distance, making the decision boundaries more clear. In Fig. 4d, the Moka-ADA not only makes samples of the same polarity across domains become compact and aligned, but also has better decision boundaries.

4.6 Ablation studies

To analyze the effect of our method on adversarial training, we conduct ablation experiments and the results are shown in Tables 3 and 4, where the Only-ADA represents adversarial training without model-oriented knowledge adaptation. By comparison, it is easy to observe that our methods are effective and robust, while the Only-ADA experiences a dramatic decrease with increasing training epochs.

For further study, we perform feature visualization of the Only-ADA for the K → B task as shown in Fig. 5. In the first subplot, all samples belong to four clusters, which indicates that adversarial training brings domain awareness to the model. Nonetheless, in the remaining subplots, it appears that samples of different polarities in the target domain gradually mix into the same cluster, which is a mode collapse phenomenon in adversarial training. In contrast, our models have better stability and flexibility of adversarial training, which effectively prevents the mode collapse phenomenon.

Table 2 Accuracy results on 12 domain pairs from the Amazon reviews benchmark datasets. (The best performance is indicated in bold.)

Task	PERL (2020)	DAAT (2020)	p+CFd (2020)	UDALM (2021)	DA-SDS (2022)	AdSPT (2022)	Baseline	ISC-ADA	FSC-ADA	Moka-ADA
B → D	87.80	89.70	87.65	90.97	91.28	92.00	91.30	92.88 ± 0.14	92.83 ± 0.31	**93.11** ± 0.12
B → E	87.20	89.57	91.30	91.69	90.22	93.75	93.25	94.54 ± 0.18	**94.79** ± 0.26	94.64 ± 0.20
B → K	90.20	90.75	92.45	93.21	90.66	93.10	96.20	96.34 ± 0.21	96.49 ± 0.07	**96.50** ± 0.15
D → B	85.60	90.86	91.50	91.00	91.18	92.15	91.15	**93.31** ± 0.11	92.99 ± 0.29	93.04 ± 0.31
D → E	89.30	89.30	91.55	92.30	92.08	94.00	93.55	**94.91** ± 0.14	94.74 ± 0.12	94.75 ± 0.22
D → K	90.40	90.50	92.45	93.66	91.15	93.25	96.00	96.31 ± 0.08	**96.47** ± 0.08	96.35 ± 0.08
E → B	83.90	88.91	88.65	90.61	91.45	92.70	90.40	92.60 ± 0.20	92.72 ± 0.18	**92.81** ± 0.07
E → D	84.80	90.13	88.20	88.83	90.88	**93.15**	91.20	92.73 ± 0.39	92.77 ± 0.16	93.02 ± 0.26
E → K	91.20	93.18	93.60	94.43	93.28	94.75	**96.20**	96.13 ± 0.17	96.08 ± 0.17	96.12 ± 0.19
K → B	83.00	87.98	89.75	90.29	92.23	92.35	89.55	92.92 ± 0.12	92.92 ± 0.23	**93.16** ± 0.20
K → D	85.60	88.81	87.80	89.54	91.40	**92.55**	89.85	92.37 ± 0.16	92.24 ± 0.19	92.41 ± 0.13
K → E	91.20	91.72	92.60	94.34	91.98	93.95	93.55	95.01 ± 0.14	94.89 ± 0.11	**95.08** ± 0.07
Average	87.50	90.12	90.63	91.74	91.48	93.14	92.68	94.17 ± 0.02	94.16 ± 0.06	**94.25** ± 0.04

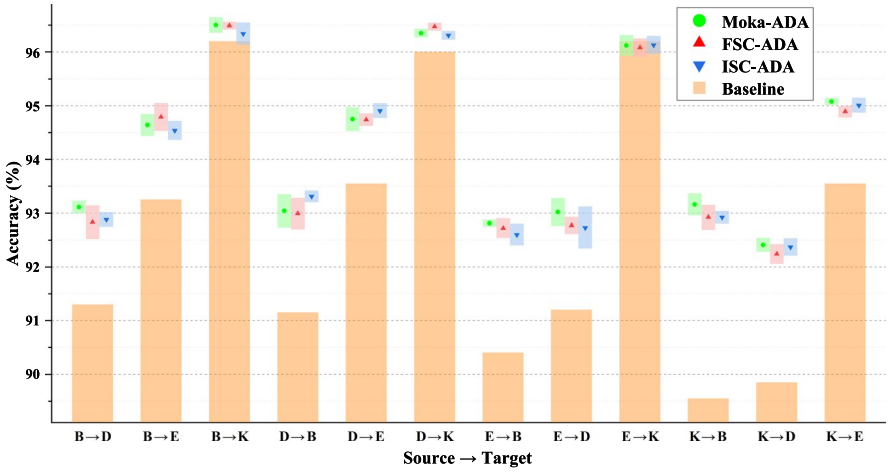

Fig. 3 Accuracy results of our methods compared to the baseline

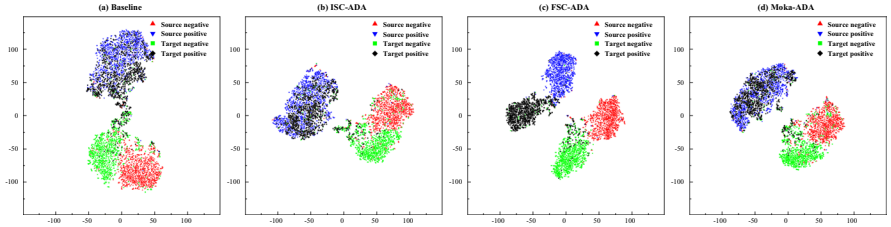

Fig. 4 Feature visualization for the B → D task using the t-SNE algorithm

5 Conclusion and future work

In this study, we propose a novel method, Moka-ADA, for cross-domain sentiment analysis. It aims to learn domain-invariant and discriminative knowledge to ensure that the marginal and conditional distributions are aligned simultaneously. The model-oriented knowledge adaptation module we designed can effectively facilitate knowledge transfer. Extensive experiments show that our Moka-ADA outperforms the state-of-the-art result on the Amazon reviews benchmark datasets. Theoretical analysis and ablation studies verify the reasonableness and effectiveness of our method.

In future, we would like to adapt our method to more realistic and challenging scenarios, such as multi-source domain [37] and sparsely labeled source domain [38], and further explore applications for other cross-domain tasks in the direction of natural language processing and computer vision.

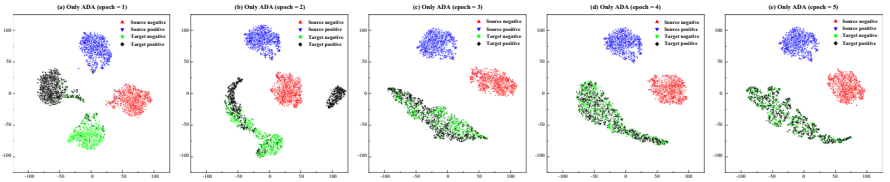

Fig. 5 Feature visualization of the Only-ADA at different adversarial training epochs for the K → B task

Table 3 Experimental results of the Only-ADA

Epochs	1	2	3	4	5
B → D	92.30	92.30	91.30	50.35	50.00
B → E	94.45	91.95	94.15	50.00	50.00
B → K	96.00	96.15	94.85	50.00	50.00
D → B	70.35	50.05	50.00	50.00	50.00
D → E	93.85	92.40	50.05	50.00	50.05
D → K	96.00	95.90	95.70	95.40	95.70
E → B	92.70	50.05	50.00	50.00	50.00
E → D	92.40	92.25	49.70	50.20	50.00
E → K	96.00	94.75	90.05	95.30	95.30
K → B	**92.65**	**69.80**	**49.50**	**49.95**	**50.00**
K → D	91.35	91.90	49.95	52.80	49.95
K → E	93.10	69.75	50.00	50.00	50.00
Average	91.76	82.27	67.94	57.83	57.58

Table 4 Experimental results of the Moka-ADA

Epochs	1	2	3	4	5
B → D	93.05	93.00	93.15	93.05	93.30
B → E	94.35	94.60	94.60	94.90	94.75
B → K	96.75	96.40	96.45	96.50	96.40
D → B	93.15	93.20	93.35	92.55	92.95
D → E	94.45	94.80	94.90	94.60	95.00
D → K	96.35	96.40	96.45	96.30	96.25
E → B	92.90	92.80	92.75	92.85	92.75
E → D	92.65	92.95	93.00	93.35	93.15
E → K	96.30	96.35	96.05	95.95	95.95
K → B	93.45	93.25	93.10	92.90	93.10
K → D	92.55	92.45	92.50	92.30	92.25
K → E	95.00	95.15	95.05	95.05	95.15
Average	94.25	94.28	94.28	94.19	94.25

Appendix A supplemental experimental results

See Tables 5 and 6.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Maoyuan Zhang, Xiang Li and Fei Wu. The first draft of the manuscript was written by Xiang Li and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work is supported by the Fundamental Research Funds of the National Language Committee (Grant No. YB135-40).

Table 5 Experimental results of the ISC-ADA

Epochs	1	2	3	4	5
B → D	92.85	93.10	92.90	92.75	92.80
B → E	94.35	94.60	94.35	94.70	94.70
B → K	96.65	96.45	96.20	96.15	96.25
D → B	93.15	93.45	93.35	93.30	93.30
D → E	94.70	94.95	94.85	95.05	95.00
D → K	96.35	96.25	96.20	96.40	96.35
E → B	92.80	92.80	92.60	92.40	92.40
E → D	92.50	92.20	93.10	92.75	93.10
E → K	96.30	96.20	96.15	96.15	95.85
K → B	92.75	92.95	93.00	93.05	92.85
K → D	92.10	92.35	92.50	92.50	92.40
K → E	95.25	95.00	94.95	94.95	94.90
Average	94.15	94.19	94.18	94.18	94.16

Table 6 Experimental results of the FSC-ADA

Epochs	1	2	3	4	5
B → D	93.30	92.90	92.70	92.45	92.80
B → E	94.40	95.00	94.70	95.05	94.80
B → K	96.50	96.60	96.40	96.50	96.45
D → B	93.20	93.30	93.00	92.90	92.55
D → E	94.55	94.80	94.85	94.80	94.70
D → K	96.35	96.55	96.45	96.50	96.50
E → B	92.50	92.85	92.60	92.95	92.70
E → D	92.75	92.55	92.90	92.70	92.95
E → K	95.80	96.15	96.05	96.15	96.25
K → B	92.95	93.20	92.80	93.05	92.60
K → D	92.15	92.20	92.40	92.45	92.00
K → E	94.85	94.75	94.90	94.90	95.05
Average	94.11	94.24	94.15	94.20	94.11

Availability of data and materials Data supporting the results of this study are available upon request from the corresponding author xli@mails.ccnu.edu.cn. Because they contain information that may compromise the consent of study participants, these data are not publicly available.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical Approval Not applicable.

References

1. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642
2. Wilson G, Cook DJ (2018) Adversarial transfer learning. arXiv preprint arXiv:1812.02849
3. Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189–196
4. Zhou Z-H, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on knowledge and data engineering 17(11):1529–1541
5. Blitzer J, McDonald R, Pereira F (2006) Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128
6. Pan SJ, Ni X, Sun J-T, Yang Q, Chen Z (2010) Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th International Conference on World Wide Web, pp. 751–760
7. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096
8. Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176
9. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems **27**
10. Ben-David S, Blitzer J, Crammer K, Pereira F (2006) Analysis of representations for domain adaptation. Advances in neural information processing systems **19**
11. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. Mach Learn 79(1):151–175
12. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A (2012) A kernel two-sample test. J Mach Learn Res 13(1):723–773
13. Wang W, Li H, Ding Z, Nie F, Chen J, Dong X, Wang Z (2023) Rethinking maximum mean discrepancy for visual domain adaptation. IEEE Trans Neural Netw Learn Syst 34(1):264–277. https://doi.org/10.1109/TNNLS.2021.3093468
14. Hinton G, Vinyals O, Dean J, et al (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.025312**2**(7)
15. Bollegala D, Mu T, Goulermas JY (2015) Cross-domain sentiment classification using sentiment sensitive embeddings. IEEE Trans Knowledge and Data Eng 28(2):398–410
16. Liu J, Zheng S, Xu G, Lin M (2021) Cross-domain sentiment aware word embeddings for review sentiment analysis. Int J Mach Learn Cybernet 12(2):343–354
17. Ziser Y, Reichart R (2018) Pivot based language modeling for improved neural domain adaptation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1241–1251
18. Zheng L, Ying W, Yu Z, Qiang Y (2018) Hierarchical attention transfer network for cross-domain sentiment classification. In: AAAI18
19. Li Z, Zhang Y, Wei Y, Wu Y, Yang Q (2017) End-to-end adversarial memory network for cross-domain sentiment classification. In: IJCAI, pp. 2237–2243

20. Qu X, Zou Z, Cheng Y, Yang Y, Zhou P (2019) Adversarial category alignment network for cross-domain sentiment classification. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2496–2508
21. Du C, Sun H, Wang J, Qi Q, Liao J (2020) Adversarial and domain-aware bert for cross-domain sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4019–4028
22. Karouzos C, Paraskevopoulos G, Potamianos A (2021) Udalm: Unsupervised domain adaptation through language modeling. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2579–2590
23. Zhou J, Tian J, Wang R, Wu Y, Xiao W, He L (2020) Sentix: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 568–579
24. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474
25. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 .PMLR
26. Sun S, Cheng Y, Gan Z, Liu J (2019) Patient knowledge distillation for bert model compression. In: EMNLP/IJCNLP (1)
27. Yim J, Joo D, Bae J, Kim J (2017) A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4133–4141
28. Furlanello T, Lipton Z, Tschannen M, Itti L, Anandkumar A (2018) Born again neural networks. In: International Conference on Machine Learning, pp. 1607–1616. PMLR
29. Wang W, Li B, Wang M, Nie F, Wang Z, Li H (2022) Confidence regularized label propagation based domain adaptation. IEEE Trans Circuits and Syst Video Technol 32(6):3319–3333. https://doi.org/10.1109/TCSVT.2021.3104835
30. Zhang B, Zhang X, Liu Y, Cheng L, Li Z (2021) Matching distributions between model and data: Cross-domain knowledge distillation for unsupervised domain adaptation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 5423–5433
31. Kifer D, Ben-David S, Gehrke J (2004) Detecting change in data streams. In: VLDB, vol. 4, pp. 180–191 . Toronto, Canada
32. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 440–447
33. Fu Y, Liu Y (2022) Domain adaptation with a shrinkable discrepancy strategy for cross-domain sentiment classification. Neurocomputing
34. Ben-David E, Rabinovitz C, Reichart R (2020) Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. Trans Assoc Comput Linguis 8:504–521
35. Ye H, Tan Q, He R, Li J, Ng HT, Bing L (2020) Feature adaptation of pre-trained language models across languages and domains with robust self-training. arXiv preprint arXiv:2009.11538
36. Wu H, Shi X (2022) Adversarial soft prompt tuning for cross-domain sentiment analysis. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2438–2447
37. Fu Y, Liu Y (2022) Contrastive transformer based domain adaptation for multi-source cross-domain sentiment classification. Knowledge-Based Syst 245:108649
38. Wang W, Chen S, Xiang Y, Sun J, Li H, Wang Z, Sun F, Ding Z, Li B (2021) Sparsely-labeled source assisted domain adaptation. Pattern Recognition 112:107803

Authors and Affiliations

Maoyuan Zhang[1,2,3] · Xiang Li[1,2,3] · Fei Wu[1,2,3]

Maoyuan Zhang
zhangmy@mail.ccnu.edu.cn

Fei Wu
wufei_better@163.com

[1] Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning, Central China Normal University, Wuhan 430079, Hubei, China

[2] School of Computer, Central China Normal University, Wuhan 430079, Hubei, China

[3] National Language Resources Monitor and Research Center for Network Media, Central China Normal University, Wuhan 430079, Hubei, China