Domain Adaptation with Category Attention Network for Deep Sentiment Analysis

Dongbo Xi
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
University of Chinese Academy of Sciences, Beijing 100049, China
xidongbo17s@ict.ac.cn

Fuzhen Zhuang*
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
University of Chinese Academy of Sciences, Beijing 100049, China
zhuangfuzhen@ict.ac.cn

Ganbin Zhou
Search Product Center, WeChat Search Application Department, Tencent, China
ganbinzhou@tencent.com

Xiaohu Cheng
Search Product Center, WeChat Search Application Department, Tencent, China
xiaohucheng@tencent.com

Fen Lin
Search Product Center, WeChat Search Application Department, Tencent, China
felicialin@tencent.com

Qing He
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
University of Chinese Academy of Sciences, Beijing 100049, China
heqing@ict.ac.cn

ABSTRACT

Domain adaptation tasks such as cross-domain sentiment classification aim to utilize existing labeled data in the source domain and unlabeled or few labeled data in the target domain to improve the performance in the target domain via reducing the shift between the data distributions. Existing cross-domain sentiment classification methods need to distinguish *pivots*, i.e., the domain-shared sentiment words, and *non-pivots*, i.e., the domain-specific sentiment words, for excellent adaptation performance. In this paper, we first design a Category Attention Network (CAN), and then propose a model named **CAN-CNN** to integrate CAN and a Convolutional Neural Network (CNN). On the one hand, the model regards *pivots* and *non-pivots* as unified category attribute words and can automatically capture them to improve the domain adaptation performance; on the other hand, the model makes an attempt at interpretability to learn the transferred category attribute words. Specifically, the optimization objective of our model has three different components: 1) the supervised classification loss; 2) the distributions loss of category feature weights; 3) the domain invariance loss. Finally, the proposed model is evaluated on three public sentiment analysis datasets and the results demonstrate that **CAN-CNN** can outperform other various baseline methods.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Applied computing** → *Document analysis*.

KEYWORDS

Domain Adaptation, Category Attention Network, Sentiment Analysis, Interpretability

ACM Reference Format:
Dongbo Xi, Fuzhen Zhuang, Ganbin Zhou, Xiaohu Cheng, Fen Lin, and Qing He. 2020. Domain Adaptation with Category Attention Network for Deep Sentiment Analysis. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380088

1 INTRODUCTION

Domain adaptation model can take the advantage of existing large scale labeled data in the source domain and adapt to unlabeled data with similar distribution in the target domain. However, in sentiment classification, the expression of users' emotions usually varies across domains [21]. For example, the word *delicious* is used to express positive sentiment in the *Foods* domain, whereas *heartwarming* is used in the *Movies* domain. Due to the domain discrepancy, a sentiment classifier trained in a source domain may not work well when directly applied to a target domain.

To address the problem, traditional methods need to distinguish the domain-shared pivots and domain-specific non-pivots. For example, Blitzer et al. [2, 3] proposed a Structural Correspondence Learning (SCL) method which utilizes multiple pivot prediction

[*] Corresponding author: Fuzhen Zhuang.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380088

tasks to infer the correlation between pivots and non-pivots. However, these methods need to deal with pivots and non-pivots separately, and they cannot be unified. Recently, deep neural models are explored to reduce the shift in data distributions for domain adaptation. Some studies utilized Maximum Mean Discrepancy (MMD) measure as a regularization to reduce the distribution mismatch between the source and target domains [11, 24, 30]. However, these deep models lack interpretability to directly learn the transferred category attribute words (i.e., pivots and non-pivots).

Along this line, we propose a **C**ategory **A**ttention **N**etwork and **C**onvolutional **N**eural **N**etwork based model (**CAN-CNN**). On the one hand, the proposed model regards pivots and non-pivots as unified category attribute words to learn, which no longer needs to design different networks. On the other hand, the category attribute words in target domain can be automatically learned to interpret what to transfer. Specifically, the CAN module is the core of CAN-CNN and it contains a Category Memory Module (CMM), a Dynamic Matching (DM) process and a Category Attention (CA) layer. We firstly construct a CMM which contains some category attribute words. Then, a DM process can dynamically match category attribute words from the CMM for each sample. Moreover, the CA acts on each category attribute word and each sample to pay more attention to category features in the sample. Finally, we apply CAN to the source domain where CMM is specially constructed, and the target domain where CMM is randomly initialized to interpret what can be transferred from the source domain to the target domain. Also, for obtaining significant performance improvements and improving the interpretability, the optimization objective of our model has three different components as described in Section 3.3. By optimizing the objective of our model, the proposed CAN can focus on category features and ignore non-category features in a sample for better performance in target domain, and the domain-aware CMM and the CA in the CAN can give inspiration on the interpretability in domain adaptation. Comprehensive experimental results on real-world datasets demonstrate that the proposed **CAN-CNN** can outperform other various baselines, and the further analysis of the experimental results demonstrates that the domain-aware CMM in CAN can be seen as a kind of knowledge transfer between source and target domain which can interpret what can be transferred from the source domain to the target domain.

2 RELATED WORK

Traditional methods need to distinguish the pivots and non-pivots. For example, Structural Correspondence Learning (SCL) method [2, 3] induce a low-dimensional feature representation shared across domains based on the co-occurrence between pivots and non-pivots, and it needs to utilize multiple pivot prediction tasks to infer the correlation between pivots and non-pivots. Similarly, the Spectral Feature Alignment (SFA) method [27] aims to align the non-pivots from different domains into unified clusters, with the help of pivots as a bridge. Besides, the Hierarchical Attention Transfer Network (HATN) [21] can automatically capturing pivots and non-pivots simultaneously, but it needs to utilize a P-net and a NP-net to deal with pivots and non-pivots separately, and they cannot be unified.

Recently, deep learning methods form another line of work to automatically produce superior feature representations for cross-domain sentiment classification. Glorot et al. [12] utilized Stacked Denoising Autoencoder (SDA) to extract a meaningful representation in an unsupervised fashion and then trained a classifier with this high-level feature representation. Other leading works usually utilize Maximum Mean Discrepancy (MMD) measure as a regularization to reduce the distribution mismatch between the source and target domains [11, 23, 24, 30]. Other methods that have gained popularity recently are utilizing adversarial training or Generative Adversarial Nets (GAN) [13]. These core approaches include Domain Adaptation with Adversarial Training (DAAT) [9], Domain-Adversarial Neural Network (DANN) [10], Domain Separation Networks (DSN) [5], Selective Adversarial Networks (SAN) [6] and some other approaches [18, 22, 29]. However, these studies usually lack interpretability to directly learn the transferred category attribute words, and their poor interpretability in domain adaptation causes trouble for user understanding and makes the model performance untrustworthy when applied to real applications.

3 METHODOLOGY

In this section, we first formulate the problem, then present the details of the proposed model **CAN-CNN** as shown in Figure 1(a).

3.1 Problem Formulation

Given a source domain $\mathcal{D}_s$ and a target domain $\mathcal{D}_t$. Suppose the source and target domain both have a classification task whose number of categories is C. Let $\boldsymbol{X}_s = \{(\boldsymbol{x}_1^s, y_1^s), \cdots, (\boldsymbol{x}_{N_s}^s, y_{N_s}^s)\}$ is a set of N_s labeled samples in the source domain, where $\boldsymbol{x}_i^s \in \mathbb{R}^L$ is the i-th sample with length L in the source domain, $y_i^s \in \{1, 2, \cdots, C\}$ is the label of i-th sample $\boldsymbol{x}_i^s$. Similarly, $\boldsymbol{X}_t = \{\boldsymbol{x}_1^t, \cdots, \boldsymbol{x}_{N_t}^t\}$ is a set of N_t unlabeled samples in the target domain, where $\boldsymbol{x}_j^t \in \mathbb{R}^L$ is the j-th target domain sample with length L. The task is to utilize labeled training data $\boldsymbol{X}_s^{train}$ in the source domain and unlabeled training data $\boldsymbol{X}_t^{train}$ in the target domain to train a model, which has a good classification performance on the target data $\boldsymbol{X}_t^{test}$. Here, for simplicity, we will discuss only binary classification ($C = 2$) task here. But our model can be adapted to multi-class tasks.

3.2 Category Attention Network (CAN)

Category features (e.g. words with category attributes) play an important indicative role in classification tasks. Taking sentiment analysis as an example, some positive sentiment words (e.g., *funny*, *pleasing*) occur frequently in positive text, while some negative sentiment words (e.g., *boring*, *disgusting*) are more common in negative text. Therefore, we design the **CAN**, as shown in Figure 1(b), to focus on category features and ignore no-category features. As mentioned in Section 1, the CAN module is the core of CAN-CNN and it contains a Category Memory Module (CMM), a Dynamic Matching (DM) process and a Category Attention (CA) layer.

3.2.1 Category Memory Module (CMM). The CMM in source domain contains some category attribute words drawn from the labeled data. Different categories should have different category attribute words and each category's attribute words should occur

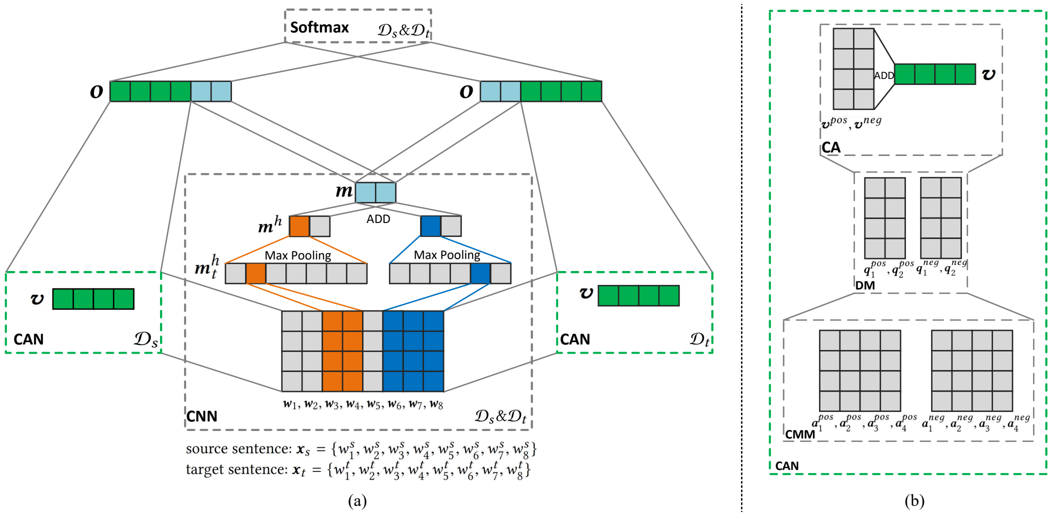
Softmax
$\mathcal{D}_s\&\mathcal{D}_t$
$\boldsymbol{o}$
$\boldsymbol{o}$
$\boldsymbol{m}$
ADD
$\boldsymbol{m}^h$
$\boldsymbol{m}_t^h$
Max Pooling
Max Pooling
$\boldsymbol{v}$
CAN
$\mathcal{D}_s$
$\boldsymbol{v}$
CAN
$\mathcal{D}_t$
$\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4, \boldsymbol{w}_5, \boldsymbol{w}_6, \boldsymbol{w}_7, \boldsymbol{w}_8$
CNN
$\mathcal{D}_s\&\mathcal{D}_t$
source sentence: $\boldsymbol{x}_s = \{w_1^s, w_2^s, w_3^s, w_4^s, w_5^s, w_6^s, w_7^s, w_8^s\}$
target sentence: $\boldsymbol{x}_t = \{w_1^t, w_2^t, w_3^t, w_4^t, w_5^t, w_6^t, w_7^t, w_8^t\}$
(a)
ADD
$\boldsymbol{v}$
$\boldsymbol{v}^{pos}, \boldsymbol{v}^{neg}$
CA
q_1^{pos}, q_2^{pos} q_1^{neg}, q_2^{neg}
DM
$a_1^{pos}, a_2^{pos}, a_3^{pos}, a_4^{pos}$ $a_1^{neg}, a_2^{neg}, a_3^{neg}, a_4^{neg}$
CMM
CAN
(b)

Figure 1: (a) The CAN and CNN based domain adaptation model CAN-CNN . (b) The CAN module details for binary classification sentiment analysis task.

frequently in this category but rarely seen in other categories. Therefore, for the labeled source domain data, we extract the positive category attribute words as follows:

$$a_1^{pos} = \arg\max_i \left(\log t_i^{pos} - \log t_i^{neg}\right), \tag{1}$$

where t_i^{pos} and t_i^{neg} are the numbers of i-th word in the vocabulary appears in the positive samples and negative samples, respectively. According to Equation (1), we can get the positive top-1 attribute word a_1^{pos}. Besides, we can also get the negative top-1 attribute word a_1^{neg} by replacing arg max with arg min in Equation (1). The CMM can be constructed via selecting top-M attribute words for each category. Therefore, the CMM should contain $M \times 2$ category attribute words which are M positive words $\{a_1^{pos}, \cdots, a_M^{pos}\}$ and M negative words $\{a_1^{neg}, \cdots, a_M^{neg}\}$.

Because our proposed CMM needs to utilize labeled data to extracted category attribute words, it can't be directly utilized on the unlabeled target domain data. Therefore, we need to apply a domain-aware CMM. The domain-aware CMM contains pivots and non-pivots within a unified framework. In our model, the CMM in the target domain should be randomly initialized word vectors and can be learned automatically during the model training by minimizing the Distributions Loss of Category Feature Weights in Section 3.3.2. Finally, we hope the learned CMM in the target domain can carry category information which can be seen as a kind of knowledge transfer between the source and target domains and can improve the interpretability.

3.2.2 Dynamic Matching (DM). In the word embedding space, the DM process utilizes cosine similarity to dynamically match the top-K most similar category attribute words from the CMM for each sample during each gradient-descent step as follows:

$$s_m^c = \max_{l \in \{1,\cdots,L\}} cos(\boldsymbol{w}_l, \boldsymbol{a}_m^c), \tag{2}$$

$$q_1^c = \arg\max_{m \in \{1,\cdots,M\}} s_m^c, \tag{3}$$

where $\boldsymbol{w}_l \in \mathbb{R}^d$ is the embedding vector of l-th word in sentence $\boldsymbol{x}$, $\boldsymbol{a}_m^c \in \mathbb{R}^d$ is the embedding vector of m-th category attribute word in category c ($c \in \{pos, neg\}$), and d is the dimension of the embedding vector. According to Equation (3), we can get the top-1 matched category attribute word q_1^c for category c. The DM process dynamically selects top-K matched category attribute words from M category attribute words for each category ($K \ll M$). Therefore, we finally get $K \times 2$ matched category attribute words, i.e., K positive words $\{q_1^{pos}, \cdots, q_K^{pos}\}$ and K negative words $\{q_1^{neg}, \cdots, q_K^{neg}\}$. These matched category attribute words can indicate the sentiment polarity of the sentence $\boldsymbol{x}$.

3.2.3 Category Attention (CA). Attention mechanism has been proved to be effective in machine translation [1], where the attention makes the model focus on useful words for the current task. Hence, we utilize CA to act on each matched category attribute words q_k^c and the sentence $\boldsymbol{x}$ to pay more attention to the category features in the sentence. We compute the weight vector $\boldsymbol{s}_k^c \in \mathbb{R}^L$ between category attribute word $\boldsymbol{q}_k^c$ and sentence $\boldsymbol{x}$ as:

$$s_{k,l}^c = \frac{\exp\left(\boldsymbol{q}_k^{c\top} \tanh(\boldsymbol{W}_a \boldsymbol{w}_l + \boldsymbol{b})\right)}{\sum_{l=1}^L \exp\left(\boldsymbol{q}_k^{c\top} \tanh(\boldsymbol{W}_a \boldsymbol{w}_l + \boldsymbol{b})\right)}, \tag{4}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b} \in \mathbb{R}^d$ are the parameters of weight and bias of the category attention, respectively. The degree of correspondence between $\boldsymbol{q}_k^c$ and $\boldsymbol{w}_l$ can be assessed by the Equation (4). The resulting vector of scores $\boldsymbol{s}_k^c$ yields category feature weights for each word in the sentence. The output sentence-level category feature vector $\boldsymbol{v}^c$ for category c is that first a sum over the sentence inputs $\boldsymbol{x}$ weighted by the scores vector $\boldsymbol{s}_k^c$ and then a sum over K weighted vectors $\boldsymbol{v}_k^c$: $\boldsymbol{v}^c = \sum_{k=1}^{K} \sum_{l=1}^{L} s_{k,l}^c \boldsymbol{w}_l$.

Finally, We sum the sentence-level category features $\{\boldsymbol{v}^{pos}, \boldsymbol{v}^{neg}\}$ to get the overall feature representation: $\boldsymbol{v} = \boldsymbol{v}^{pos} + \boldsymbol{v}^{neg}$.

3.3 CAN and CNN based Domain Adaptation

Convolutional Neural Network (CNN) has been proved to be efficient on text classification [17], and recently has been applied to transfer text classification [8, 9]. We apply the same CNN architecture as the TextCNN [17] (as shown in Figure 1(a)) which is an embedding layer, followed by a convolution layer with T filters of different width h, and then a summation (or concatenation) of global max pooled features to generate the final features $\boldsymbol{m}$.

The CAN learned features $\boldsymbol{v} \in \mathbb{R}^d$ and the CNN learned features $\boldsymbol{m} \in \mathbb{R}^T$ are concatenated, and then we apply a fully-connected layer with softmax activation to output the final prediction:

$$\begin{aligned} \boldsymbol{o} &= [\boldsymbol{v}; \boldsymbol{m}], && (5) \\ \hat{\boldsymbol{y}} &= \operatorname{softmax}\left(\boldsymbol{W}_s^{\top} \boldsymbol{o}\right), && (6) \end{aligned}$$

where $[\cdot;\cdot]$ denotes the concatenation of two column vectors, $\boldsymbol{W}_s^{\top} \in \mathbb{R}^{C \times (d+T)}$ are the weights of fully-connected layer, and $\hat{\boldsymbol{y}} \in \mathbb{R}^C$ is a distribution which indicates different category probability.

The source domain and target domain data share all network structure and parameters except the domain-aware CMM. The loss function $\mathcal{L}(\theta)$ of our model is split into three components each of which plays a different role:

$$\mathcal{L}(\theta) = \mathcal{L}_C(\theta) + \alpha \mathcal{L}_D(\theta) + \beta \mathcal{L}_I(\theta), \tag{7}$$

where θ is the parameters set, the tunable hyper-parameters α and β control the different strength of the components. Training is finished through stochastic gradient descent over shuffled mini-batches with the Adam [19] update rule.

3.3.1 *Supervised Classification Loss.*

The labeled data $\boldsymbol{X}_s$ in the source domain $\mathcal{D}_s$ are available for our classification task. We need to minimize the *cross entropy* of predicted distribution and the actual distribution:

$$\mathcal{L}_C(\theta) = -\frac{1}{N_s} \sum_{\mathcal{D}_s} \sum_{c=1}^{C} y_{i,c}^s \log(\hat{y}_{i,c}^s | \boldsymbol{x}_i^s, \theta), \tag{8}$$

where N_s is the number of source domain samples, C is the number of categories, $\boldsymbol{y}_i^s \in \mathbb{R}^C$ and $\hat{\boldsymbol{y}}_i^s \in \mathbb{R}^C$ are the one-hot label and the predicted distribution of source domain sample $\boldsymbol{x}_i^s$, respectively, and θ is the parameters set.

3.3.2 *Distributions Loss of Category Feature Weights.*

The different category attribute words should pay different attention to different category features. For example, positive category attribute words should pay more attention to positive words (e.g., *happy*, *pleased*, $\cdots$) in a sentence while pay less attention to the negative words (e.g., *boring*, *unfunny*, $\cdots$). Therefore, distribution $\boldsymbol{s}_k^c$ of category feature weights should be discrepant for different categories. We need to minimize negative Kullback-Leibler (KL) divergence [20] between positive and negative categories. Here, the constraint should be applied to both source and target domains:

$$\begin{aligned} \mathcal{L}_D(\theta) = &-\frac{1}{N_s} \sum_{\mathcal{D}_s} \left(\mathrm{KL}\left(\boldsymbol{s}^{pos}, \boldsymbol{s}^{neg}\right) + \mathrm{KL}\left(\boldsymbol{s}^{neg}, \boldsymbol{s}^{pos}\right)\right) \\ &-\frac{1}{N_t} \sum_{\mathcal{D}_t} \left(\mathrm{KL}\left(\boldsymbol{s}^{pos}, \boldsymbol{s}^{neg}\right) + \mathrm{KL}\left(\boldsymbol{s}^{neg}, \boldsymbol{s}^{pos}\right)\right), \end{aligned} \tag{9}$$

where $\boldsymbol{s}^c = \sum_{k=1}^{K} \boldsymbol{s}_k^c$ and $\boldsymbol{s}_k^c = \{s_{k,1}^c, \cdots, s_{k,L}^c\}$ are the distribution of category feature weights for category c and k-th matched category attribute word, respectively. By minimizing the Equation (9), the attention distributions of different category attribute words are pulled apart. This constraint can force the target domain to yield domain-aware CMM which can be seen as a kind of knowledge transfer to interpret what can be transferred from the source domain to the target domain. Utilizing the domain-aware CMM, the model can increase the performance in the target domain.

Table 1: The extracted top-5 category attribute words for positive class and negative class by CMM on three different domain datasets CR, AFF and MR.

CR		AFF		MR	
pos.	neg.	pos.	neg.	pos.	neg.
love	useless	Delicious	worst	engrossing	badly
happy	flaw	beat	terrible	realistic	unfunny
pleased	press	Perfect	worse	riveting	tiresome
amazing	recognize	Excellent	horrible	gem	disguise
great	downside	fast	disgusting	heartwarming	boring

3.3.3 *Domain Invariance Loss.*

Due to the distribution discrepancy between source and target domains, directly training a model utilizing only the source data usually leads to overfitting to the source data distribution [30]. It also reduces the performance when it predicts the label in the target domain. Our model expects to reduce the distribution mismatch between source and target domains in the latent space to learn domain invariant features and provide a good domain adaptation performance.To minimize this discrepancy, we adopt the Maximum Mean Discrepancy (MMD) [4] as the metric. MMD is a widely used regularization method in image domain adaptation [11, 23, 30]. The MMD is performed on the source and target domains:

$$\mathcal{L}_I(\theta) = \| \frac{1}{N_s} \sum_{\mathcal{D}_s} \phi(\boldsymbol{x}_s) - \frac{1}{N_t} \sum_{\mathcal{D}_t} \phi(\boldsymbol{x}_t) \|, \tag{10}$$

this distance is computed with respect to a particular representation, $\phi(\cdot)$, and this representation operates on source data point $\boldsymbol{x}_s$ and target data point $\boldsymbol{x}_t$. In our case, it is an empirical approximation to this distance. We apply the MMD to the penultimate layer of our model as performed in [30]. By minimizing the classification loss $\mathcal{L}_C(\theta)$ and the domain invariance loss $\mathcal{L}_I(\theta)$ in Equation (7) simultaneously, the model can remain invariant to domain shift and yield a strong classification representation.

It is worth noting that our approach is a general framework and the optimization objective acts on the CAN and the features extractor CNN. So the CNN can be replaced by any other efficient feature extractors (e.g., LSTM [14], Transformer [31]).

Table 2: Accuracy based on 10-fold cross validation on different target domain datasets transferred from different source domain datasets, comparing our CAN-CNN model against various baselines.

	Model	MR→CR	AFF→CR	CR→AFF	MR→AFF	CR→MR	AFF→MR
Direct Transfer	fastText-rand	0.6250	0.6700	0.6738	0.6878	0.5727	0.5824
	fastText-non-static	0.6674	0.7462	0.7232	0.7476	0.6535	0.6878
	CNN-char	0.5588	0.6658	0.7139	0.6608	0.5603	0.5923
	CNN-rand	0.6052	0.6985	0.7117	0.6731	0.5884	0.5996
	CNN-non-static	0.6884	0.7578	0.7517	0.7624	0.6679	0.6918
Domain Adaptation	SDA	0.6058	0.6624	0.6742	0.6920	0.6238	0.6324
	mSDA	0.5954	0.6426	0.6803	0.7051	0.6206	0.6387
	SDA-fine-tune	0.6224	0.6939	0.6893	0.7143	0.6305	0.6422
	DAAT	0.6981	0.7302	0.7211	0.7433	0.6235	0.6521
	CAN-CNN-share	0.7149	0.7493	0.7654	0.7808	0.6548	0.6960
	CAN-CNN	**0.7302**	**0.7647**	**0.7882**	**0.7929**	**0.6796**	**0.7098**

Table 3: Target domain accuracy of ablation study.

Model	MR→CR	CR→AFF	AFF→MR
CAN-CNN ($\alpha = 0, \beta = 0$)	0.7164	0.7661	0.7008
CAN-CNN ($\alpha = 0$)	0.7281	0.7700	0.7044
CAN-CNN ($\beta = 0$)	0.7148	0.7867	0.6989
CAN-CNN	**0.7302**	**0.7882**	**0.7098**

4 EXPERIMENTS

In this section, we perform experiments to evaluate the proposed model against various baselines on three real-world datasets.

4.1 Datasets

We utilize three public sentiment analysis datasets in the experimental study.

1) **CR**[1] [15] is a customer review sentiment analysis dataset of various products (e.g., cameras, DVD player, MP3 etc.) from amazon.

2) **AFF**[2] [25] is a sentiment dataset consisting of reviews of fine foods from amazon. We chose a random subset of it as done in [8].

3) **MR**[3] [28] is a movie review sentiment analysis dataset.

As there is no standard train/test split for the three datasets, we adopt 10-fold cross validation. The extracted top-5 category attribute words for positive class and negative class by CMM as Equation (1) on three source domain datasets CR, AFF and MR are shown in Table 1. And the results further show that different domains should have different category attribute words even on the same category, so we can not directly apply the source domain CMM to the target domain, we need to apply a domain-aware CMM.

4.2 Implementation Details

We can construct 6 domain adaptation tasks based on the three different datasets: MR→CR, AFF→CR, CR→AFF, MR→AFF, CR→MR, AFF→MR,

For fair comparison, for all datasets we use: filter width (h) of 3, 4, 5 with 100 (T) feature maps each and embedding dimension of 300 (d) which are same as the setting in the TextCNN. Besides, we use: mini-batch size of 128 and learning rate of 0.001. These hyper-parameters are the same for all datasets.

In the CAN module, for all datasets we use: $M = 50$ category attribute words for each category in CMM, $K = 5$ dynamically matched category attribute words for each category in DM, $\alpha = 0.05$ and $\beta = 0.01$ as the strength of the different components in the loss function. These values are chosen via a grid search on the CR validation set which splits 10% from training set.

In all experiments, we adopt **Accuracy** as our evaluation metric based on 10-fold cross validation.

4.3 Baselines

We compare the proposed method **CAN-CNN** with two kinds of baselines. The one is direct transfer baselines following the existing work [9]: we train a model (e.g., fastText, CNN) on the source domain and transfer it directly to the target domain. The other one is deep domain adaptation baselines which do not need to capture pivots and non-pivots separately.

1) **direct transfer baselines**: We adopt fastText [16] with randomly initialized word vectors **fastText-rand** and pre-trained fine-tuned word vectors **fastText-non-static** as our baselines. We also adopt TextCNN [17] with randomly initialized word vectors **CNN-rand** and pre-trained fine-tuned word vectors **CNN-non-static** as our baselines. Besides, character-level [32] TextCNN **CNN-char** is also adopted in our experiment. We initialize the word embedding using a open-source pre-trained data [26][4].

2) **domain adaptation baselines**: For domain adaptation, we adopt **SDA** [12] and **mSDA** [7] with a SVM classifier as the baselines. Also, we initialize the network architecture and parameters with the learned SDA, and fine-tune (**SDA-fine-tune**) with a softmax classifier on the source domain data. To be fair to compare, we also adopt **DAAT** [9] which is a domain adaptation method with adversarial training without the semi-supervised component.

[1] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[2] http://snap.stanford.edu/data/web-FineFoods.html

[3] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[4] https://code.google.com/p/word2vec/

Table 4: The words in the vocabulary most similar to category attribute words in the target domain CMM before (random initialization) and after (fine-tuning) training, respectively.

	MR→CR		CR→AFF		AFF→MR	
	Before	After	Before	After	Before	After
pos.	indicated	great	buck	nice	dolgin	heartwarming
	sizing	wonderful	semi	delicious	programs	dramatize
	took	amazing	United	good	adults	satiric
	Lithium	vividly	diagnoised	simple	cloying	screamingly
	multiple	repeatly	shift	perfect	reawaken	vividly
neg.	had	unfortunately	addresses	unfortunately	adaptation	but
	rebooting	useless	unclutterd	worse	assert	badly
	4mp	supposed	Shoprite	awful	hope	unfortunately
	glow	flaw	toasted	a	plying	boring
	stamp	awful	weight	terrible	niccol	loveless

4.4 Performance Comparison

The experimental results evaluated by Accuracy based on 10-fold cross validation on different target domain datasets are presented in Table 2. CNN-char performs the worst due to the lack of word-level information. CNN-rand improves the results comparing with CNN-char by utilizing word-level information, and fastText-rand obtains similar improvement. Utilizing pre-trained word vectors can further improve the performance on target domain datasets (fastText-non-static vs fastText-rand, CNN-non-static vs CNN-rand). It makes sense that the pre-trained word vectors which are trained on large-scale dataset utilize out-of-domain knowledge to reduce the overfitting risk and can give a better transfer performance.

For domain adaptation baselines, SDA and mSDA obtain similar performance, and SDA-fine-tune outperforms SDA via fine-tuning the model parameters. Besides, CNN-non-static is the best method among the baselines including the domain adaptation model DAAT. DAAT utilizes adversarial training which is difficult to converge and does not perform good enough on the sentiment analysis task.

And we also let the CMM in target domain be the same as the CMM in source domain (**CAN-CNN**-share) to demonstrate the effectiveness of the domain-aware CMM. We can see that **CAN-CNN**-share performs worse than **CAN-CNN** due to utilizing shared CMM not the domain-aware CMM. This further illustrates that we can not directly apply the source domain CMM to the target domain. On the contrary, **CAN-CNN** obtains the best performance in multiple pairs experiments via utilizing domain-aware CMM. These improvements indicate that the proposed **CAN-CNN** model can better handle the domain adaptation task. It is worth noting that all the experiments are unsupervised in the target domain, and if we utilize a small amount of labeled samples in the target domain, the performance of our proposed model will be further improved.

Furthermore, for understanding the contribution of different components in loss function in Equation (7), we mask different components to demonstrate the effectiveness: 1) Only supervised loss **CAN-CNN** ($\alpha, \beta = 0$); 2) With domain invariance loss **CAN-CNN** ($\alpha = 0$); 3) With distributions loss of category feature weights **CAN-CNN** ($\beta = 0$). As shown in Table 3, we can see that the **CAN-CNN** obtains the best performance in multiple pairs experiments, which indicates each component matters for the performance.

Table 5: The visualization of category attention weights on target domain MR.

		Sentences
pos.	(1)	an extremely funny , ultimately heartbreaking look at life ...
	(2)	beautifully observed , miraculously unsentimental comedy drama .
neg.	(1)	godawful boring slug of a movie .
	(2)	certainly not a good movie , but it wasn't horrible either .

4.5 Interpretability

As mentioned in Section 3.2.1, our proposed CMM needs to utilize labeled data to extract category attribute words, so it can't be directly utilized on the unlabeled target domain data. Besides, we can observe (see Table 1) that different domains have different category attribute words even on the same category (e.g., "delicious", "fast" for positive class in AFF, while "heartwarming", "vividly" for positive class in MR), so we can not directly apply the source domain CMM to the target domain, and the experiment results of model **CAN-CNN**-share in Table 2 also show the poor performance comparing with **CAN-CNN**. Therefore, in our model, the CMM in the target domain needs to be randomly initialized and can be learned automatically during the model training. Table 4 shows the words in the vocabulary most similar to category attribute words in the target domain CMM before (random initialization) and after (fine-tuning) training, respectively. Before training, the words most similar to category attribute words are some random words due to randomly initialization. After training, the most similar words obviously carry category information although with few noise words. It's because we minimize the Equation (9), the attention distributions of positive and negative category attribute words are pulled apart. This constraint can force the target domain to yield domain-aware CMM which can be seen as a kind of transfer of knowledge between source and target domain, and the domain-aware CMM contains pivots and non-pivots within a unified framework. Therefore, the proposed **CAN-CNN** has the capability to interpret what can be transferred from the source domain to the target domain.

4.6 Case Study

For two positive class samples and two negative class samples from target domain MR, we average the weights calculated by the top-K matched category attribute words (see Equation (4)) as $s^c = \frac{1}{K}\sum_k s_k^c$ and visualize the corresponding category attention weights as shown in Table 5. The depth of color for each word illustrates the distributions of category attention weights. The darker the color is, the higher attention weight it has.

First, for the positive sample (1), we can see that the third word "funny" has the highest attention weight, and the word "heartbreaking" has a second highest attention weight. They both tell positive sentiment features which are corresponding to the actual label. In the positive sample (2), the similar case can also be observed. On the other hand, the negative sentiment features word "unsentimental" has a very small weight, which shows that the matched positive category attribute words can mask the fake sentiment information and pay more attention to the actual sentiment information.

Second, for the negative sample (1), most of the weight is held by the second word "boring" which brings negative sentiment features, and the word "slug" also shares some weight due to its negative sentiment features. Besides, since there is a space missing between "god" and "awful" in the raw data, the matched category attribute words do not notice the negative sentiment information very well. For the negative sample (2), the fourth word "good" brings positive sentiment features, while the tri-gram "not a good" tells negative sentiment features, the matched negative category attribute words can notice this, and pay more attention to the tri-gram. Besides, although the bi-gram "wasn't horrible" tells non-negative sentiment features, the whole sentence sentiment is negative. As we can see, our model is not masked by the bi-gram, but rather pays more attention to the negative sentiment word "horrible".

Such above results demonstrate that our **CAN-CNN** model can effectively find the important parts within a sample according to the matched category attribute words for better performance.

5 CONCLUSION

In this paper, we proposed a **CAN-CNN** model which not only improves the domain adaptation performance, but also makes an attempt at interpretability via designing a Category Attention Network integrated with a Convolutional Neural Network. Specifically, the proposed model regards pivots and non-pivots as unified category attribute words to learn, which no longer needs to design different networks, and the category attribute words in target domain can be learned automatically to interpret what to transfer. The experimental results on three public sentiment analysis datasets from three domains demonstrate the performance improvements of our proposed model compared with various baseline methods.

ACKNOWLEDGMENTS

The research work supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant No. U1836206, U1811461, 61773361, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146. This work is also partly supported by the funding of WeChat cooperation project.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473* (2014).

[2] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*. 440–447.

[3] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*. ACL, 120–128.

[4] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* (2006).

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *NeurIPS*.

[6] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018. Partial transfer learning with selective adversarial networks. In *CVPR*.

[7] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *ICML*.

[8] Xin Dong and Gerard de Melo. 2018. A Helping Hand: Transfer Learning for Deep Sentiment Analysis. In *ACL*. 2524–2534.

[9] Muhammad Imran Firoj Alam, Shafiq Joty. 2018. Domain Adaptation with Adversarial Training and Graph Embeddings. In *ACL*.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2015. Domain-adversarial training of neural networks. *JMLR* 17, 1 (2015), 2096–2030.

[11] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. 2014. Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*.

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*. 2672–2680.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*. 168–177.

[16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *ECACL*.

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv:1408.5882* (2014).

[18] Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Adversarial adaptation of synthetic or stale data. In *ACL*.

[19] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*, Vol. 5.

[20] Solomon Kullback. 1987. Letter to the editor: The Kullback-Leibler distance. (1987).

[21] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018. Hierarchical attention transfer network for cross-domain sentiment classification. In *AAAI*.

[22] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *NeurIPS*.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. *arXiv:1502.02791* (2015).

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2016. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*.

[25] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW*.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, Vol. 26. 3111–3119.

[27] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *WWW*. ACM, 751–760.

[28] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

[29] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*.

[30] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv:1412.3474* (2014).

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. (2017), 5998–6008.

[32] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*. 649–657.