Domain-Invariant Feature Distillation for Cross-Domain Sentiment Classification

Mengting Hu[1*] **Yike Wu**[1*] **Shiwan Zhao**[2†]
Honglei Guo[2] **Renhong Cheng**[1] **Zhong Su**[2]
[1] Nankai University [2] IBM Research - China
mthu@mail.nankai.edu.cn, wuyike@dbis.nankai.edu.cn, {zhaosw, guohl}@cn.ibm.com
chengrh@nankai.edu.cn, suzhong@cn.ibm.com

Abstract

Cross-domain sentiment classification has drawn much attention in recent years. Most existing approaches focus on learning domain-invariant representations in both the source and target domains, while few of them pay attention to the domain-specific information. Despite the non-transferability of the domain-specific information, simultaneously learning domain-dependent representations can facilitate the learning of domain-invariant representations. In this paper, we focus on aspect-level cross-domain sentiment classification, and propose to distill the domain-invariant sentiment features with the help of an orthogonal domain-dependent task, i.e. aspect detection, which is built on the aspects varying widely in different domains. We conduct extensive experiments on three public datasets and the experimental results demonstrate the effectiveness of our method.

1 Introduction

Sentiment classification based on deep learning methods has developed rapidly in recent years. While achieving outstanding performance, these methods always need large-scale datasets with sentiment polarity labels to train a robust sentiment classifier. However, in most cases, large-scale labeled datasets are not available in practice and manual annotation costs much. One of the solutions to this problem is *cross-domain sentiment classification*, which aims to exploit the rich labeled data in one domain, i.e. *source domain*, to help the sentiment analysis task in another domain lacking for or even without labeled data, i.e. *target domain*. The rationality of this solution is that the source domain and target domain share some domain-invariant knowledge that can be transferred across domains.

Source Domain: The fried rice is amazing here.
Target Domain: Surprisingly, Britney Spears is amazing.

Figure 1: Example sentences from the source domain (restaurant) and the target domain (twitter) respectively. The sentiment expressions marked by solid lines are domain-invariant, while the aspect terms marked by dashed lines are domain-specific.

Previous works on cross-domain sentiment classification mainly focus on learning the domain-invariant representations in both source and target domains, either based on manual feature selection (Blitzer et al., 2006; Pan et al., 2010) or automatic representation learning (Glorot et al., 2011; Chen et al., 2012; Ganin and Lempitsky, 2015; Li et al., 2017). The sentiment classifier, which makes decisions based on the domain-invariant features and receives the supervisory signals from the source domain, can be also applied to the target domain. We can draw an empirical conclusion: the better domain-invariant features the method obtains, the better performance it gains. However, few studies explore the usage of the domain-specific information, which is also helpful to the cross-domain sentiment classification. Peng et al. (2018) propose to extract the domain-invariant and domain-dependent features of the target domain data and train two classifiers accordingly, but they require a few sentiment polarity labels in the target domain, which limits the practical application of the method.

In this paper, we exploit the domain-specific information by adding an orthogonal domain-dependent task to "distill" the domain-invariant features for cross-domain sentiment classification. The proposed method *domain-invariant feature distillation* (DIFD) does not need any sentiment

[*] Work performed while interning at IBM Research - China.

[†] Corresponding author.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 5559–5568,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

polarity labels in the target domain, which is more consistent with the practical settings. Specifically, we focus on the aspect-level cross-domain sentiment classification, and train a shared sentiment classifier and two respective aspect detectors in the source and target domains. We argue that aspect detection is an orthogonal domain-dependent task with respect to the sentiment classification. As shown in Figure 1, given an input sentence, the sentiment classifier predicts its sentiment polarity based on the opinion words shared by different domains, while the aspect detector identifies the aspect terms which vary significantly across domains. The information on which the two tasks depend is mutually exclusive in the sentence, i.e. orthogonal. Therefore, by training these two tasks simultaneously, the aspect detectors will try to strip the domain-specific features from the input sentence and make the domain-invariant features purer, which is helpful to the cross-domain sentiment classification.

Moreover, we design two effective modules to boost the distillation process. One is the word-level context allocation mechanism. It modulates the importance of the words in the input sentence according to the property of different tasks. The other is the domain classifier. It tries to correctly judge which domain the domain-invariant feature comes from, while the other modules in the proposed method try to "fool" it, and the whole framework is trained in an adversarial way.

To summarize, the main contributions of our paper are as follows:

- We distill the domain-invariant sentiment features to improve the cross-domain sentiment classification by simultaneously training an aspect detection task that striping the domain-specific aspect features from the input sentence.

- We boost the separation process of the domain-invariant and domain-specific features by two effective modules which are the context allocation mechanism and domain classifier respectively.

- Experimental results demonstrate the effectiveness of the proposed method, and we further verify the rationality of the context allocation mechanism by visualization.

2 Related Work

Cross-domain sentiment analysis: Many domain adaptation methods have been proposed for sentiment analysis. SCL (Blitzer et al., 2006) learns correspondences among features from different domains. SFA (Pan et al., 2010) aims at reducing the gap between domains by constructing a bipartite graph to model the co-occurrence relationship between domain-specific words and domain-independent words. SDA (Glorot et al., 2011) learns to extract a meaningful representation for each review in an unsupervised fashion. mSDA (Chen et al., 2012) is an efficient method to marginalize noise and learn features. Gradient Reversal Layer (GRL) (Ganin and Lempitsky, 2015; Ganin et al., 2016; Li et al., 2017) is employed to learn domain-invariant representations by fooling the domain classifier. The replacement of gradient reversal with alternating minimization (Shu et al., 2018) stabilizes domain adversarial training, and we employ this method as the adversarial training.

Aspect-level sentiment domain adaptation: To the best of our knowledge, there are two works about aspect-related cross-domain sentiment classification. Li et al. (2019) propose a method to employ abundant aspect-category data to assist the scarce aspect-term level sentiment prediction. Zhang et al. (2019) propose IATN to address that aspects have different effects in different domains. Their method predicts sentiment polarity for the whole sentence rather than a specific aspect.

Our method concentrates on aspect-term level sentiment domain adaptation by separating the domain-specific aspect features. Bousmalis et al. (2016) and Liu et al. (2017) separate features into two subspaces by introducing constraints on the learned features. The difference is that our method is more fine-grained and utilizes the explicit aspect knowledge.

Auxiliary task for sentiment domain adaptation: Auxiliary task has been employed to improve cross-domain sentiment analysis. Yu and Jiang (2016) use two pivot-prediction auxiliary tasks to help induce a sentence embedding, which works well across domains for sentiment classification. Yu and Jiang (2017) propose to jointly learn domain-independent sentence embeddings by auxiliary tasks to predict sentiment scores of domain-independent words. Chen et al. (2018) design auxiliary domain discriminators for better transferring knowledge between do-

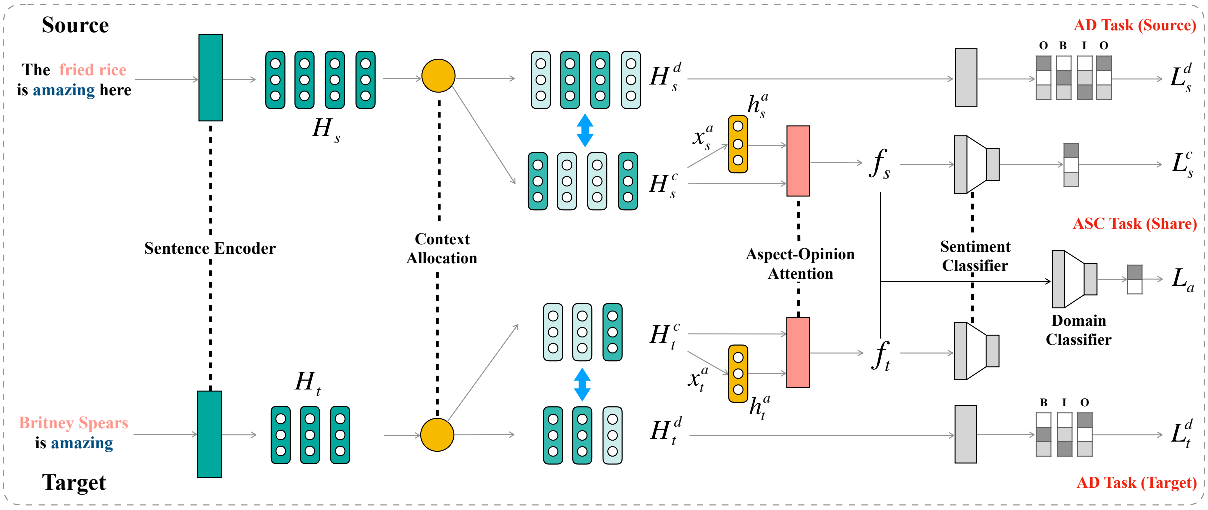

Figure 2: Network Architecture. The dashed line indicates that the parameters are shared by source and target domains. The context from the sentence encoder is divided into task-oriented contexts for ASC and AD tasks.

mains. These auxiliary tasks focus on directly enhancing domain-invariant features, while ours strips domain-specific features to distill domain-invariant features.

3 Methodology

3.1 Formulation and Overview

Suppose the source domain contains labeled data $\mathcal{D}_s = \{(x_s^k, a_s^k), y_s^k\}_{k=1}^{N_s}$, and the target domain contains unlabeled data $\mathcal{D}_t = \{(x_t^k, a_t^k)\}_{k=1}^{N_t}$, where x is a sentence, a is one of the aspects in x, and y is the sentiment polarity label of a. The proposed method handles two kinds of tasks. One is the main task Aspect-level Sentiment Classification (**ASC**). It learns a mapping $\mathcal{F}$: $\{x\} \rightarrow \{\boldsymbol{f}\} \rightarrow \{y\}$ shared by source and target domains, where $\boldsymbol{f}$ is the domain-invariant feature of x. The other is the orthogonal domain-dependent task Aspect Detection (**AD**). It learns a mapping $\mathcal{G}_s$: $\{x_s\} \rightarrow \{\boldsymbol{z_s}\} \rightarrow \{a_s\}$ in the source domain, and the other one $\mathcal{G}_t$: $\{x_t\} \rightarrow \{\boldsymbol{z_t}\} \rightarrow \{a_t\}$ in the target domain, where $\boldsymbol{z}_s$ and $\boldsymbol{z}_t$ are both domain-dependent features of x_s and x_t respectively. The domain-invariant and domain-dependent features are orthogonal, i.e. $\boldsymbol{f} \perp \boldsymbol{z_s}$ and $\boldsymbol{f} \perp \boldsymbol{z_t}$. We facilitate the distillation of $\boldsymbol{f}$ by simultaneously learning $\mathcal{G}_s$ and $\mathcal{G}_t$ which try to strip $\boldsymbol{z_s}$ and $\boldsymbol{z_t}$ from x, and the purer $\boldsymbol{f}$ leads to the better $\mathcal{F}$ for the cross-domain sentiment classification.

Figure 2 illustrates the architecture overview of our method. Given an input sentence either from the source or target domain, we first feed it into the sentence encoder to obtain its distributed representation. Then, the context allocation mechanism divides the distributed representation into two orthogonal parts: domain-invariant and domain-dependent features. Finally, the two orthogonal features are fed into their corresponding downstream tasks. Specifically, we input the domain-invariant feature into the sentiment classifier to predict the sentiment polarity, and input the domain-dependent feature into the aspect detector of the specific domain to identify the aspect terms. In addition, we add a domain classifier to the architecture. It tries to correctly judge which domain the domain-invariant feature comes from. The whole framework is trained in an adversarial way. Next, we will introduce the components of our method in detail.

3.2 Sentence Encoder

Given an input sentence $x = \{w_1, w_2, ..., w_n\}$, we first map it into an embedding sequence $\hat{E} = \{\boldsymbol{e_1}, \boldsymbol{e_2}, ..., \boldsymbol{e_n}\} \in \mathbb{R}^{n \times d_e}$. Then we inject the positional information of each token in x into $\hat{E}$ to obtain the final embedded representation E, following the Position Encoding (PE) method in the work (Vaswani et al., 2017):

$$\begin{aligned} PE(pos, 2i) &= sin(pos/10000^{2i/d_e}) \\ PE(pos, 2i+1) &= cos(pos/10000^{2i/d_e}) \\ E &= \hat{E} + PE \end{aligned} \tag{1}$$

where pos is the word position in the sentence and i is the i-th dimension of d_e. We consider that the injected positional information can facilitate the aspect-level sentiment classification, based on the observation that sentiment words tend to be close to its related aspect terms (Tang et al., 2016; Chen et al., 2017).

Next we employ a Bi-directional LSTM (BiLSTM) (Graves et al., 2013) to encode E into the contextualized sequence representation $H = [\boldsymbol{h_1}, \boldsymbol{h_2}, ...\boldsymbol{h_n}] \in \mathbb{R}^{n \times 2d_h}$, which preserves the contextual information of each token in the input sentence.

We unify the embedding layer and BiLSTM as the sentence encoder in which different tasks or domains all share the same weights. The advantages of sharing the weights are two-fold: first, different tasks in the same domain can benefit from each other in a multi-task manner; second, distilling the domain-invariant feature from a common transformation is more simple.

3.3 Context Allocation (CA)

In an input sentence, some words have a strong bias towards domain-specific information, such as the aspect terms, e.g. *"pizza"* in Restaurant domain, while others focus on the domain-invariant knowledge, such as the opinion words, e.g. *"amazing"*. Meanwhile, the ASC task and AD task exactly require orthogonal information as discussed before. Therefore, we argue that different words contribute differently according to the property of the downstream task. To facilitate the distillation of the domain invariant features, we propose a Context Allocation (CA) mechanism to allocate different weights on the same word in different downstream tasks. The values of the weights depend on how the information contained in the word matches the need of the specific task. Concretely, at each time step i, the module divides the contextualized representation h_i of word w_i into the sentiment-dominant context h_i^c and aspect-dominant context h_i^d as follows:

$$\boldsymbol{h_i^c} = \beta_i^c \boldsymbol{h_i}, \tag{2}$$

$$\boldsymbol{h_i^d} = \beta_i^d \boldsymbol{h_i}, \tag{3}$$

$$\beta_i^c + \beta_i^d = 1, \quad i \in \{1, 2, ..., n\}. \tag{4}$$

The two-dimensional vector $\beta_i = (\beta_i^c, \beta_i^d)$ is normalized considering that the domain-specific information and domain-invariant knowledge are mutually exclusive. It reflects the importance of w_i on the ASC task and AD task respectively, and is calculated on h_i as follows:

$$\beta_i = \mathrm{softmax}(W_b \tanh(W_a \boldsymbol{h_i}^{\mathrm{T}})), \tag{5}$$

where $W_a \in \mathbb{R}^{2d_h \times 2d_h}$ and $W_b \in \mathbb{R}^{2 \times 2d_h}$. The whole division process at all time steps can be formulated in the following form:

$$\begin{bmatrix} H_c \\ H_d \end{bmatrix} = \boldsymbol{\beta} \cdot \begin{bmatrix} H \\ H \end{bmatrix}, \tag{6}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_n]$. The sentiment-dominant context $H^c \in \mathbb{R}^{n \times 2d_h}$ and aspect-dominant context $H^d \in \mathbb{R}^{n \times 2d_h}$ of the input sentence are then fed into the ASC task and AD task for downstream processing respectively.

3.4 Aspect-level Sentiment Classification (ASC) Task

Aspect-Opinion Attention In the ASC task, we design an attention mechanism to model the relationship between the position of the aspect terms and their corresponding opinion words. For a specific aspect term, the domain-invariant feature based on the aspect-opinion attention contains more information of its corresponding opinion words, which is beneficial to the final aspect-level sentiment classification. Specifically, we first calculate the position representation of a specific aspect term with its position x^a and the sentiment-dominant context H^c:

$$\boldsymbol{h^a} = H^c x^a \tag{7}$$

where $x^a = \{0_1, ..., 1_{i+1}, ..., 1_{i+m}, ...0_n\}$ represents the word positions of an aspect subsequence in the input sentence x with non-zero values and m is the length of the aspect. Then the representation $\boldsymbol{h^a} \in \mathbb{R}^{2d_h}$ is further utilized to calculate the sentiment-dominant feature $\boldsymbol{f}$, which is domain-invariant and should be aligned across source and target domains.

$$\begin{aligned} \gamma_i &= \tanh(\boldsymbol{h_i^c} \cdot W_p \cdot \boldsymbol{h^a} + b_p) \\ \boldsymbol{f} &= \sum_{i=1}^{n} \gamma_i \boldsymbol{h_i^c} \end{aligned} \tag{8}$$

where γ_i reflects how much the word w_i corresponds with the opinion on the aspect term, and $W_p \in \mathbb{R}^{2d_h \times 2d_h}$ and $b_p \in \mathbb{R}^1$ are weight matrix and bias respectively.

Sentiment Classification Loss The sentiment-dominant features $\boldsymbol{f_s}$ and $\boldsymbol{f_t}$ generated from the source and target domains respectively share the same sentiment classifier. Note that the source domain data has sentiment polarity label, while the target domain is unlabeled. Thus we train the sentiment classifier only with the labeled data in the source domain, while utilizing it for inference in both source and target domains. The training objective of the sentiment classifier is to minimize the following loss on the source domain dataset, which is marked as L_s^c:

$$L_s^c = -\frac{1}{N_s}\sum^{N_s} \mathrm{log}P(y|\boldsymbol{f_s}) \tag{9}$$

where y is the ground-truth sentiment polarity label. For simplicity, we omit the enumerated number of the instance in the loss equation.

Domain Adversarial Loss The domain classifier maps the sentiment-dominant feature $\boldsymbol{f}$ into a two-dimensional normalized value $\boldsymbol{y} = (y_s, y_t)$, which indicates the probability that $\boldsymbol{f}$ comes from the source and target domains respectively. The ground-truth domain label is $\boldsymbol{g_s} = (1, 0)$ for instances in the source domain, and $\boldsymbol{g_t} = (0, 1)$ in the target domain. The training objective of the domain classifier is to minimize the following loss on both source and target domain datasets, which is marked as $L_a^{\theta_D}$:

$$L_a^{\theta_D} = -\frac{1}{N_s}\sum^{N_s} \boldsymbol{g_s}\mathrm{log}\boldsymbol{y} - \frac{1}{N_t}\sum^{N_t} \boldsymbol{g_t}\mathrm{log}\boldsymbol{y}. \tag{10}$$

The part in our architecture which joins the generating process of $\boldsymbol{f}$ (including Sentence Encoder, Context Allocation and Aspect-Opinion Allocation in Figure 2) can be regarded as a domain-invariant feature extractor, which works with the domain classifier in an adversarial way. To further accelerate the distillation process of the domain-invariant features, we also introduce an adversarial loss of the domain classifier for the feature extractor. Specifically, we calculate the loss in Equation 10 with the flipped domain labels inspired by the work (Shu et al., 2018):

$$L_a^{\theta_F} = -\frac{1}{N_s}\sum^{N_s} \boldsymbol{g_t}\mathrm{log}\boldsymbol{y} - \frac{1}{N_t}\sum^{N_t} \boldsymbol{g_s}\mathrm{log}\boldsymbol{y}. \tag{11}$$

3.5 Aspect Detection (AD) Task

We model the AD task as a sequence labeling problem, and each word in the sentence is marked as a tag in $\{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$, which means the word is at the beginning (B) or the inside (I) of an aspect term or other word (O). In this way, we can detect all the aspect terms of an input sentence in one forward pass. Specifically, we first linearly transform the aspect-dominant hidden state $\boldsymbol{h^d}$ into a three-dimensional vector. Then we calculate the aspect detection loss of the source domain as follows:

$$L_s^d = -\frac{1}{N_s}\sum^{N_s}\frac{1}{n}\sum^{n}\lambda_l \mathrm{log}P(y^d|\boldsymbol{h^d}) \tag{12}$$

where y^d is the ground-truth aspect label, n is the sentence length and λ_l is the weight of different labels. The weight λ_l aims to solve the class imbalance problem because the words labeled by O usually make up the majority of one sentence. It is dynamically calculated in the training phase according to the ratio of the words with a specific label in each batch. Henceforth we denote the loss of the AD task in the target domain as L_t^d.

Algorithm 1 Adversarial Training

```
Input: labeled $\mathcal{D}_s$ and unlabeled $\mathcal{D}_t$
repeat
    Train All parameters except Domain Classifier with $L$;
    Train Domain Classifier with $\lambda^a L_a^{\theta_D}$;
until performance on the validation set does not improve in 10 epochs.
```

3.6 Training

We combine each component loss into an overall object function:

$$L = L_s^c + \lambda^a L_a^{\theta_F} + \lambda^d(L_s^d + L_t^d) \tag{13}$$

where λ^a and λ^d balance the effect of the domain classifier and the auxiliary task (i.e. aspect detection). L and $\lambda^a L_a^{\theta_D}$ are alternatively optimized. The aspect-level sentiment analysis in the unlabeled target domain is predicted by the ASC task.

4 Experiments

4.1 Datasets

To make an extensive evaluation, we employ three different datasets: Restaurants (R) and Laptops (L) from SemEval 2014 task 4 (Pontiki et al., 2014), and Twitters (T) from the work (Dong et al., 2014). The statistics of these three datasets are shown in Table 1. Specifically, we collect the aspect-term level sentences and corresponding labels from these datasets. Comparing aspect terms

Dataset		#Pos	#Neg	#Neu	Total
Restaurants (R)	Train	2164	805	633	3502
	Test	728	196	196	1120
Laptops (L)	Train	987	866	460	2313
	Test	341	128	169	638
Twitters (T)	Train	1561	1560	3127	6248
	Test	173	173	346	692

Table 1: Datasets statistics. #Pos, #Neg, #Neu represent the number of instances with positive, negative and neutral polarities.

in these three datasets, we find more than 98% aspect terms are different between Restaurants and Laptops domains, and there exists no same aspect between Restaurants and Twitters, also only 0.09% same aspects between Laptops and Twitters. This indicates that the aspect terms vary violently in different domains.

4.2 Experimental Settings

To evaluate our proposed method, we construct six aspect-level sentiment transfer tasks: R→L, L→R, R→T, T→R, L→T, T→L. The arrow indicates the transfer direction from the source domain to the target domain. For each transfer pair $\mathcal{D}_s \rightarrow \mathcal{D}_t$, the training set is composed of two parts: one is the labeled training set in $\mathcal{D}_s$, and the other is all unlabeled data which only contain the aspect term information in $\mathcal{D}_t$. The test set in $\mathcal{D}_s$ is employed as the validation set. The reported results are evaluated on all the data of $\mathcal{D}_t$.

The word embeddings are initialized with 100-dimension Glove vectors (Pennington et al., 2014) and fine-tuned during the training. The model hidden size d_h is set to be 64. The model is optimized by the SGD method with the learning rate of 0.01. The batch size is 32. We employ ReLU as the activation function.

We adopt an early stop strategy during training if the performance on the validation set does not improve in 10 epochs, and the best model is chosen for evaluation.

4.3 Compared Methods

We compare with extensive baselines to validate the effectiveness of the proposed method. Some variants of our approach are also compared for analyzing the impacts of individual components.

Transfer Baseline: The aspect-level cross-domain sentiment classification has been rarely explored. We choose the state-of-the-art method IATN (Zhang et al., 2019) which has the most similar settings with our method as the transfer baseline. It proposes to incorporate the information of both sentences and aspect terms in the cross-domain sentiment classification.

Non-Transfer Baselines: The non-transfer baselines are all representative methods in recent years for the aspect-level sentiment classification in a single domain. We train the models on the training set of the source domain, and directly test them in the target domain without domain adaptation.

- **AT-LSTM** (Wang et al., 2016): It utilizes the attention mechanism to generate an aspect-specific sentence representation.
- **ATAE-LSTM** (Wang et al., 2016): It also employs attention. The difference with AT-LSTM is that the aspect embedding is as input to LSTM.
- **MemNet** (Tang et al., 2016): It employs a memory network with multi-hops attentions and predicts the sentiment based on the top-most context representations.
- **IAN** (Ma et al., 2017): It adopts interactive attention mechanism to learn the representations of the context and the aspect respectively.
- **RAM** (Chen et al., 2017): It employs multiple attentions with a GRU cell to non-linearly combine the aggregation of word features in each layer.
- **GACE** (Xue and Li, 2018): It is based on the convolutional neural network with gating mechanisms.

Variants of Our Method:

- **ASC+AT** (ASC with adversarial training): A single task that handles the ASC task with adversarial training.
- **DIFD:** The proposed method in this work.
- **DIFD(S)**: It contains components of the source domain from DIFD and is trained only by the source domain data.
- **DIFD-CA**: DIFD without context allocation.
- **DIFD-AT**: DIFD without adversarial training.
- **DIFD-AT+MMD**: Replace the adversarial training with Maximum Mean Discrepancy (MMD) (Tzeng et al., 2014).
- **DIFD-AT+CORAL**: Replace the adversarial training with CORAL (Sun et al., 2016).

Models	R→L		L→R		R→T		T→R		L→T		T→L	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ATAE-LSTM	56.56	47.71	63.21	46.01	34.97	33.82	50.78	44.06	40.43	39.82	42.39	40.73
MemNet	57.17	47.79	65.97	49.28	34.47	32.37	50.36	42.68	44.94	44.37	36.36	34.96
RAM	58.32	43.98	54.68	26.81	33.18	29.96	48.18	43.23	44.87	44.06	42.73	42.51
GACE	61.74	50.39	66.60	49.07	32.87	29.34	45.62	37.52	44.93	45.18	47.98	42.39
AT-LSTM	60.62	45.38	66.75	46.99	32.98	28.98	50.64	43.64	38.47	36.78	47.34	42.87
IAN	60.39	50.69	66.50	47.97	35.09	33.02	51.14	44.46	44.16	44.34	43.88	40.16
IATN	62.05	46.37	67.13	51.64	34.14	30.55	40.13	37.79	44.80	44.69	46.12	42.91
DIFD(S)	63.81	57.74	68.30	53.66	36.89	33.74	55.63	44.10	45.73	46.17	44.53	43.76
DIFD	**64.86**	**60.51**	**68.53**	**57.31**	**40.13**	**38.85**	**57.60**	**46.59**	**47.32**	**47.31**	**48.97**	**47.56**

Table 2: Evaluation results of baselines in terms of accuracy (%) and macro-f1 (%).

Models	R→L		L→R		R→T		T→R		L→T		T→L	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ASC+AT	62.86	56.47	67.66	49.84	35.86	33.43	52.33	44.41	44.15	44.31	42.46	37.94
DIFD-CA	64.18	57.59	68.17	53.42	34.27	31.52	44.39	41.79	44.14	44.25	43.54	43.60
DIFD-AT	63.47	60.09	68.15	57.27	40.76	38.66	**59.76**	45.08	45.73	45.91	47.68	40.20
DIFD-AT+MMD	64.25	58.79	63.13	56.49	39.44	38.69	46.93	44.44	**47.81**	**47.91**	42.87	38.69
DIFD-AT+CORAL	63.77	58.61	68.30	53.46	**44.65**	**42.83**	57.54	46.41	46.96	46.96	38.80	35.97
DIFD	**64.86**	**60.51**	**68.53**	**57.31**	40.13	38.85	57.60	**46.59**	47.32	47.31	**48.97**	**47.56**

Table 3: Evaluation results of variants of our model in terms of accuracy(%) and macro-f1(%). The minus sign (-) means to remove the module, and the addition (+) means to add the module.

4.4 Experimental Analysis

We report the classification accuracy and macro-f1 of various methods in Table 2 and Table 3, and the best scores on each metric are marked in bold. To validate the effectiveness of our method, we analyze the results from the following perspectives.

Compare with the baselines: We display the comparison results with baselines in Table 2. Comparing with the transfer baseline IATN, we observe that DIFD significantly outperforms IATN on all metrics by +5.51% accuracy and +7.36% macro-f1 on average. This shows that the distillation of domain-invariant features really facilitates the transfer of sentiment information across domains. In addition, for a fair comparison with the non-transfer methods which only exploit the source domain data, we also train our DIFD model without the target domain data and denote this variant as DIFD(S). We observe that DIFD(S) outperforms all the non-transfer baselines on most metrics. It is worth noting that, compared to a strong baseline IAN, DIFD(S) achieves significant improvement by +4.49% accuracy on T→R and +7.05% macro-f1 on R→L. This verifies that the orthogonal task is helpful in striping the domain-specific features from the source domain and effective for accelerating the domain adaptation.

Compare the variants of our method: The results of the variants of our method are reported in Table 3. We first observe that DIFD outperforms ASC+AT on all metrics significantly. This validates that the orthogonality really helps to distill the domain-invariant features and improve the performance of the cross-domain sentiment classification.

Then we can see that DIFD-CA performs much worse than DIFD, which reveals that the context allocation mechanism plays an important role in our method. We further visualize the allocation scores in Figure 3 and the result also indicates that the reasonability of the CA module. The gray tokens and red tokens have a bias towards ASC task and AD task respectively. The allocation scores are consistent with the bias of words: red tokens get larger scores for the aspect detection task, while gray tokens get larger scores for opinion expressions. This shows that our model generates task-oriented contexts successfully.

Finally, DIFD also achieves improvement over DIFD-AT on most metrics. This indicates that

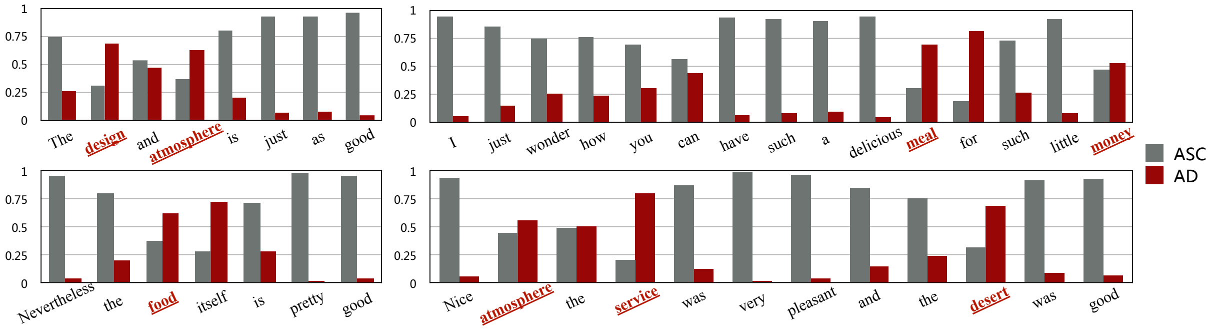

Figure 3: Visualization of context allocation weights of DIFD for different tokens within a sequence.

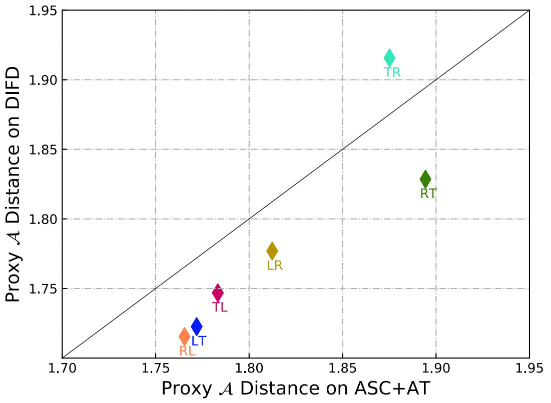

Figure 4: Proxy $\mathcal{A}$ Distance between domain-invariant features for the 6 different pairs.

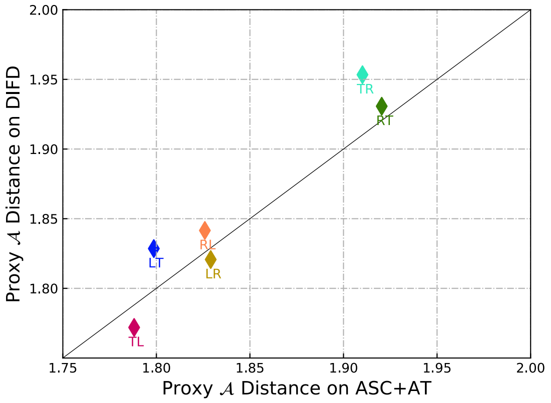

Figure 5: Proxy $\mathcal{A}$ Distance between domain-specific features for the 6 different pairs.

adversarial training with the domain classifier promotes the distillation process of the domain-invariant features. To further validate the effectiveness of adversarial training, we also try to directly minimize the divergence between domain-invariant features from source and target domains based on MMD and CORAL. Comparing with DIFD-AT+MMD and DIFD-AT+CORAL, DIFD is more robust considering that DIFD outperforms the two methods in most experimental settings.

4.5 Transfer Distance Analysis

In this section, we analyze the similarity of features between domains. We exploit the $\mathcal{A}$-distance (Ben-David et al., 2007) to measure the similarity between two probability distributions. The proxy $\mathcal{A}$-distance is 2(1-2ϵ), where ϵ is the generalization error of a classifier (a linear SVM) trained on the binary classification problem to distinguish inputs between the two domains. We focus on the methods ASC+AT and DIFD, and first compare the similarity of domain-invariant features $\boldsymbol{f_s}$ and $\boldsymbol{f_t}$. Figure 4 reports the results for each pair of domains. The proxy $\mathcal{A}$-distance on DIFD is generally smaller than its corresponding value on ASC+AT. This indicates that DIFD can learn purer domain-invariant features than ASC+AT. Secondly, we compare the domain-specific features learned by ASC+AT and DIFD, which are represented by the average hidden state of BiLSTM in ASC+AT and the average aspect-context H^d in DIFD respectively. Figure 5 reports the results for each pair of domains. The proxy $\mathcal{A}$-distance on DIFD is generally larger than its corresponding value on ASC+AT, which demonstrates that DIFD can strip more domain-specific information by the aspect detection task than ASC+AT.

There are exceptions in both Figures, i.e., TR in Figure 4, TL and LR in Figure 5. A possible explanation is that the balance between ASC and AD losses causes some domain-specific information to remain in the domain-invariant space, and vice versa.

5 Conclusion

In this work, we study the problem of aspect-level cross-domain sentiment analysis and propose a domain-invariant feature distillation method that simultaneously learns domain-invariant and domain-specific features. With the help of the orthogonal domain-dependent task (i.e., aspect detection), the aspect sentiment classification task can learn better domain-invariant features and improve transfer performance. Experimental results clearly verify the effectiveness of our method.

6 Acknowledgement

This work is supported by National Science and Technology Major Project, China (Grant No. 2018YFB0204304).

References

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *(NIPS)*, pages 137–144.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *(EMNLP)*, pages 120–128.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *(NIPS)*, pages 343–351.

Cen Chen, Yinfei Yang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. 2018. Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In *(NAACL)*, pages 602–607.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. *(ICML)*.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *(EMNLP)*, pages 452–461.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *(ACL)*, volume 2, pages 49–54.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. *(ICML)*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *(ICML)*, pages 513–520.

Alex Graves, Abdel Rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *(ICASSP)*, pages 6645–6649.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, Xin Li, and Qiang Yang. 2019. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. *(AAAI)*.

Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *(IJCAI)*, pages 2237–2243.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *(IJCAI)*.

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

Minlong Peng, Qi Zhang, Yu-gang Jiang, and Xuanjing Huang. 2018. Cross-domain sentiment classification with target domain specific information. In *(ACL)*, pages 2505–2513.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *(EMNLP)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *SemEval*, pages 27–35.

Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. 2018. A dirt-t approach to unsupervised domain adaptation. *(ICLR)*.

Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *(AAAI)*.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *(EMNLP)*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *(NIPS)*, pages 5998–6008.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level sentiment classification. In *(EMNLP)*, pages 606–615.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. *(ACL)*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *(EMNLP)*, pages 236–246.

Jianfei Yu and Jing Jiang. 2017. Leveraging auxiliary tasks for document-level cross-domain sentiment classification. ACL.

Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *(AAAI)*.