Fine-Grained Domain Adaptation for Aspect Category Level Sentiment Analysis

Mengting Hu, Hang Gao, Yike Wu, Zhong Su, and Shiwan Zhao

***Abstract*—Aspect category level sentiment analysis aims to identify the sentiment polarities towards the aspect categories discussed in a sentence. It usually suffers from a lack of labeled data. A popular solution is to transfer knowledge from a labeled source domain to an unlabeled target domain by unsupervised domain adaptation. However, most domain adaptation methods in sentiment analysis are coarse-grained, considering the source or target domain as a whole during the adaptation. We argue that these single-source single-target methods are inefficient since they ignore the difference between different aspect categories. In this article, we propose a fine-grained domain adaptation method to address the aspect category level sentiment analysis task by considering the adaptation between subdomains. Specifically, the source/target domain is divided into multiple subdomains according to the hierarchical structure of the aspect categories. We then design a multi-source multi-target transfer network to achieve fine-grained transfer. Extensive experimental results demonstrate the effectiveness of our fine-grained domain adaptation method on aspect category level sentiment analysis.**

***Index Terms*—Multi-source multi-target domain adaptation, sentiment analysis, aspect category.**

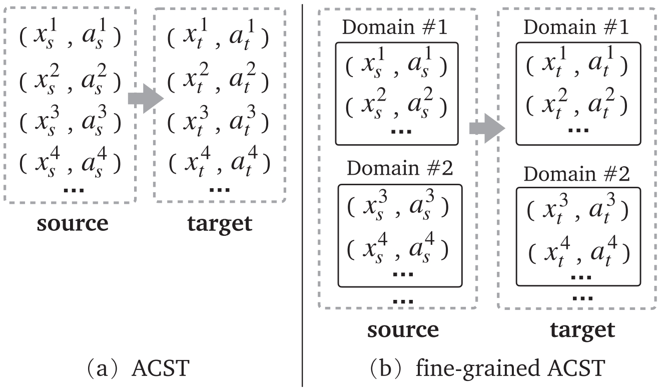


Fig. 1. (a) Existing ACST is coarse-grained, considering the source or target domain as a whole during the adaptation. (b) In fine-grained ACST, both source and target instances are grouped into multiple domains (i.e., subdomains) by the hierarchical structure of their aspect categories, respectively. Each (x, a) is an instance where x denotes a sentence and a denotes an aspect category. For concise, the sentiment polarity y of each instance in source domain(s) is omitted.

I. Introduction

Aspect category level sentiment analysis [1], [2] is a subtask in sentiment analysis. Given a predefined aspect category, it aims to recognize its sentiment polarity in a sentence. For instance, *"the room is not clean and the pizza is very tasty"* describes two aspect categories *room* and *food* with negative and positive sentiment polarities, respectively. Recently, this task has achieved outstanding performance through deep learning methods [3], [4], [5]. The achievements, to a great extent, depend on large-scale labeled datasets to learn supervised models. However, in a target domain of interest, the labeled dataset is not always available in practice and manual annotation is time-consuming and expensive.

Unsupervised domain adaptation [6] seeks to alleviate the aforementioned problem by transferring knowledge from a labeled source domain to an unlabeled target domain. In this work, we adopt unsupervised domain adaptation for aspect category level sentiment analysis and formulate it as aspect category level sentiment transfer (ACST), as shown in Fig. 1(a), where each (x, a) is an instance. The purpose of ACST is to predict the sentiment polarities of the aspect categories in the target domain by leveraging the supervised information in the source domain.

Existing domain adaptation methods in sentiment analysis [7], [8], [9] are mainly coarse-grained, considering the source or target domain as a whole during the adaptation. We argue that these single-source single-target methods are inefficient since they ignore the distinction between aspect categories. The transferable knowledge of different source aspects to different target aspects is varied according to our observations. First, opinion words are usually aspect-specific. For example, opinion words like *"delicious"* and *"tasty"* are popular in aspects *food_snack* and *food_breakfast*. However, these words are seldom seen in *laptop_quality*. Second, the similarity between aspects is diverse. For instance, aspect *food_snack* shares more common opinion expressions with *food_breakfast* than with *laptop_quality*. Third, the sentiment polarity of an opinion word depends on its aspect [10]. For example, *"fast"* is usually positive in describing *laptop_cpu*, while tends to be negative in *laptop_battery* as in a description *"this battery runs out fast"*.

Manuscript received 13 January 2022; revised 13 November 2022; accepted 26 November 2022. Date of publication 12 December 2023; date of current version 29 November 2023. This work was supported in part by the key program of National Science Fund of Tianjin, China under Grant 21JCZDJC00130 and in part by the Basic Scientific Research Fund, China under Grant 63221028. Recommended for acceptance by E. Cambria. *(Corresponding author: Yike Wu.)*

Mengting Hu is with the College of Software, Nankai University, Tianjin 300350, China (e-mail: mthu@nankai.edu.cn).

Hang Gao is with the Institute of Public Safety Research, Tsinghua University, Beijing 100084, China (e-mail: gaohang@mail.tsinghua.edu.cn).

Yike Wu is with the School of Journalism and Communication, Nankai University, Tianjin 300350, China (e-mail: wuyike@nankai.edu.cn).

Zhong Su is with the Alibaba Research, Beijing 100102, China (e-mail: suzhong.sz@alibaba-inc.com).

Shiwan Zhao is with the Natural language processing group, IBM China Research China Beijing, Beijing 100101, China (e-mail: zhaosw@gmail.com).

Digital Object Identifier 10.1109/TAFFC.2022.3228695

Therefore, in this article, we propose a fine-grained domain adaptation method to address the ACST task by considering the adaptation between subdomains, in which similar aspects should be boosted to enhance positive transfer, while distinct aspects should be suppressed to reduce negative transfer. Specifically, as shown in Fig. 1(b), we group the instances by the top level of the aspect category and obtain multiple domains[1] in the source and target datasets, respectively. For instance, the top level of the aspect categories *food_meat_steak* and *food_seafood_fish* are the same, i.e., *food*, so they are grouped into the domain *food*. By grouping instances with the same top level aspects into domains, the original ACST is converted to a multi-source multi-target fine-grained domain adaptation problem.

To address the above problem, we propose a multi-source multi-target transfer network (MMTN) to achieve fine-grained transfer. Concretely, we train a sentiment classifier for each source domain and their weighted combination is utilized for inference on each target domain. A weight computation component is designed to learn the optimal weights. To simplify the weight computation between source and target domains, each domain is represented with a *domain embedding*, which is a parameter vector representing the feature distribution of each domain. The time-consuming weight computation between domains is simplified as the measurement between their corresponding embeddings.

To make the domain embedding accurately reflect the feature distribution of the domain, we further constrain the domain embedding by features of all the instances belonging to this domain. The aspect embedding is also constrained in a similar way. It is worth noting that the domain and aspect embeddings are also used in the attention mechanism of our approach to extract the features. Therefore, the domain/aspect embedding and the features form an interactive cycle to update each other.

In summary, our contributions are threefold:

- We propose a multi-source multi-target transfer network (MMTN) by considering the adaptation between subdomains.
- A weight computation module and constrained domain/aspect embeddings are designed to improve fine-grained knowledge transfer.
- Extensive experimental results demonstrate the effectiveness of our fine-grained domain adaptation method on aspect category level sentiment analysis.

The rest of the paper is organized as follows. Section II introduces a survey of related work on aspect level sentiment analysis, unsupervised domain adaptation, and aspect level sentiment transfer. Section III describes the proposed methodology for fine-grained ACST. Section IV demonstrates experimental settings and evaluation results. Finally, Section V concludes the paper and outlines the future directions.

II. Related Works

A. Aspect Level Sentiment Analysis

Aspect level sentiment analysis aims to detect the sentiment at the aspect level. The aspect refers to *aspect term* or *aspect category* [11]. The difference is that the aspect term is a span existing in the sentence while the aspect category is an abstractive class. Analyzing sentiment at the aspect level has gained many achievements based on neural network techniques. To name a few, the attention-based model [3] and memory-based method [12] are classical works. More recently, the structured information between aspects and opinion expressions has been utilized to obtain further improvements. Graph neural networks contribute to learning such structured information graph neural networks [13], [14], [15], [16]. The neurosymbolic system [17], [18], [19] and meta-learning [20], [21] are both promising research directions. Besides improving the network architectures, other works introduce affective commonsense knowledge to learn better embeddings [22]. In this article, we focus on transferring aspect category level sentiment between different domains. We specifically consider the aspect category, which has hierarchical taxonomy for fine-grained ACST.

B. Unsupervised Domain Adaptation

The achievements of deep learning mainly benefit from large-scale labeled data for supervised learning. However, due to *domain shift* [23], the performance of a model learned in the labeled *source domain* tends to heavily degrade in the unlabeled *target domain*. Unsupervised domain adaptation has been studied to mitigate this domain shift. Its main idea is to enhance the domain-invariant learning between source and target domains, which is beneficial for knowledge transfer. Promising directions include distance-based methods [24], [25], adversarial learning methods [6], [26], mixup [27] and clustering-based regularizations [28]. Most methods address the single-source single-target scenario with one source domain and one target domain.

A few works [29], [30], [31] concentrate on the multi-source single-target domain adaptation problem. The labeled data from multiple source domains are usually exploited to improve the performance of the target domain. However, these approaches are not the optimal choice for our fine-grained ACST, which involves multiple source and multiple target domains. The reasons are two-fold. First, combining all target domains as a whole ignores the distribution discrepancy among them, while such information may contribute to domain adaptation. Second, transferring knowledge from multiple sources to each target domain separately is inefficient, since we need to train a model for each target domain. This issue becomes more severe as the number of target domains grows. Therefore, we propose MMTN to address multi-source multi-target domain adaptation. All domains, either from source or target data, are bridged with domain embeddings. This helps to simplify the complicated problem. One promising work also focuses on multi-source multi-target domain adaptation [32], which addresses semantic segmentation of satellite images. Yet, its shallow data augmentor

[1] In the context of fine-grained ACST, we use the terms domain and subdomain interchangeably.

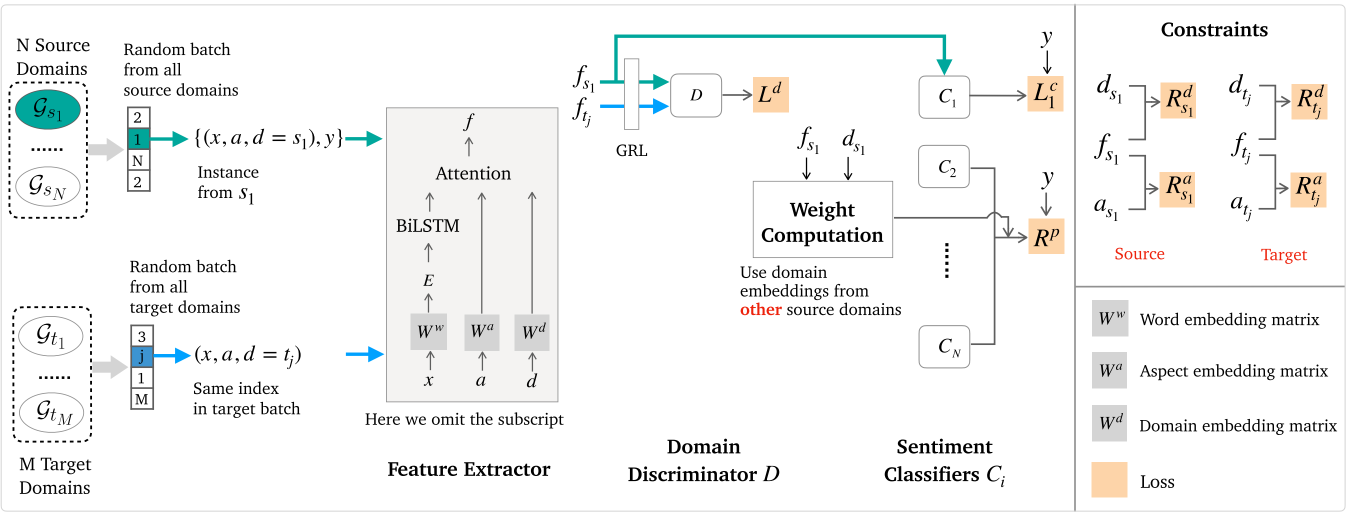


Fig. 2. An overview of MMTN architecture. Our framework is composed of 1) basic architecture (including a shared feature extractor $\boldsymbol{f} = F(x, a, d)$, N sentiment classifiers: $\{C_i\}_{i=1}^N$, one domain discriminator D); 2) constraints; 3) weight computation. Best viewed in color. It is worth noting that, given an instance $(x, a, d = s_1)$ during training, weight computation is trained by its domain embedding with all *other* source domains embeddings.

for stylizing each domain is hard to be directly applied to text data.

C. Aspect Level Sentiment Transfer

Many domain adaptation methods for sentiment classification are proposed. Some methods [33], [34], [35] aim at reducing domain shift by using pivot features as a bridge. Others extract common representation via unsupervised learning [36], [37]. Particularly, Guo et al. [38] propose a mixture-of-experts approach for unsupervised domain adaptation from multiple sources. Liu et al. [39] address multi-source single-target domain adaptation by learning domain representations with self-attention.

However, these methods mainly transfer sentiment at the sentence level. A few works transfer sentiment at the aspect level. Wang et al. [7] design a recursive neural structural correspondence network to address the cross-domain aspect and opinion co-extraction, which extracts the aspect terms and opinion words simultaneously. Liu et al. [8] build a coarse-to-fine network to transfer sentiment information from aspect category level to aspect term level. Hu et al. [9] propose to transfer sentiment at the aspect term level by introducing an auxiliary task to distill the domain-invariant sentiment features. Different from these coarse-grained domain adaptation work, we concentrate on the sentiment transfer at the aspect category level and between subdomains.

III. Methodology

A. Overview

Suppose there are N different source domains $\{\mathcal{G}_{s_i}\}_{i=1}^N$ and M different target domains $\{\mathcal{G}_{t_j}\}_{j=1}^M$, each source domain $\mathcal{G}_{s_i}$ contains labeled instances $\{(x, a, d = s_i), y\}$, and each target domain $\mathcal{G}_{t_j}$ contains unlabeled instances $\{(x, a, d = t_j)\}$, where x is a sentence, a is one of the aspects in x, d is the domain and y is the sentiment polarity label. Fig. 2 shows the overview of the network architecture.

We randomly sample a source batch and a target batch of the same size. The instances of the source batch come from all source domains, as do the target batch. In the two batches, two instances with the same index form a pair. Assume the pair of instances are $\{(x, a, d = s_i), y\}$ and $(x, a, d = t_j)$. For demonstration purposes, we let $s_i = s_1$, and its corresponding target instance comes from t_j as shown in Fig. 2. In the following parts, the computation of this pair is described as an example. The two instances are first fed into the feature extractor to obtain the sentiment features $\boldsymbol{f}_{s_1}$ and $\boldsymbol{f}_{t_j}$. Then $\boldsymbol{f}_{s_1}$ is passed to its corresponding downstream sentiment classifier $C_1(\boldsymbol{f}_{s_1})$. Two features are also input to the domain discriminator D to facilitate the learning of transferable knowledge by training adversarially.

In the upper right of Fig. 2, we constrain the domain/aspect embedding with sentiment features. Specifically, a domain embedding is affected by all the features belonging to this domain. Thus domain embedding can reflect the sentiment feature distribution, and so does the aspect embedding. Benefiting from the constraints, domain embeddings can be leveraged to measure the similarity between domains efficiently. The weight computation module is designed based on domain embeddings to allocate weights for multiple sentiment classifiers.

B. Basic Architecture

Feature Extractor. As shown in Fig. 2, given an input sentence $x = \{w_1, w_2, \ldots, w_n\}$, it is first mapped into an embedding sequence $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\}$ by looking up a word embedding matrix. Then we encode it with BiLSTM into a context sequence $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$. The aspect a are mapped into aspect embedding $\boldsymbol{a}$ by looking up an aspect embedding matrix. The domain d is mapped into domain embedding $\boldsymbol{d}$ by looking up a domain embedding matrix. Since a sentence might contain multiple aspects, $\boldsymbol{a}$ and $\boldsymbol{d}$ are used together in the attention mechanism to compute an accurate sentiment feature

$$\begin{aligned}\boldsymbol{\rho} &= \boldsymbol{w}^{\mathrm{T}}\tanh(W[H;[\boldsymbol{a};\boldsymbol{d}]\otimes n]+\boldsymbol{b}),\\ \boldsymbol{f} &= \mathrm{softmax}(\boldsymbol{\rho})H,\end{aligned} \quad (1)$$

where $[;]$ denotes the vector concatenation and $\otimes n$ means repeating vector n times. W and $\boldsymbol{w}$ are parameters, $\boldsymbol{b}$ is bias.

Sentiment Classifiers. Designing a single classifier by considering all source domains as a whole ignores the difference between diverse source domains. Thus we design a sentiment classifier for each source domain. The weighted combination of N classifiers is utilized for inference on each target domain. For the instance belonging to source domain $\mathcal{G}_{s_i}$, it is fed into the i-th sentiment classifier C_i, which is specific for source domain $\mathcal{G}_{s_i}$. The classification loss is as follows:

$$\mathcal{L}^c = \frac{1}{|\mathcal{B}_s|}\sum_{\mathcal{B}_s} -\boldsymbol{y}\log(C_i(\boldsymbol{f}_{s_i})), \quad (2)$$

where $\boldsymbol{y}$ is the one-hot format of the sentiment polarity label y from a training instance. $|\mathcal{B}_s|$ indicates the number of instances in the source batch.

Domain Discriminator. A domain discriminator is trained to accurately distinguish an instance from the source or target domain. The gradient reversal layer (GRL) [6] flips the gradients from the domain discriminator, which can make the previous layers hold an opposite training objective. Thus feature extractor tries to "fool" the domain discriminator D, making it unable to tell the origin of the feature. This facilitates the learning of transferable knowledge from source to target domains. The loss of the domain discriminator is defined as

$$\mathcal{L}^d = \frac{1}{|\mathcal{B}_s|}\sum_{\substack{\boldsymbol{f}_{s_i}\in\mathcal{B}_s\\ \boldsymbol{f}_{t_j}\in\mathcal{B}_t}} [-\boldsymbol{y}_{s_i}\log(D(\boldsymbol{f}_{s_i})) - \boldsymbol{y}_{t_j}\log(D(\boldsymbol{f}_{t_j}))], \quad (3)$$

where label $\boldsymbol{y}_{s_i}$ and $\boldsymbol{y}_{t_j}$ are (1,0) and (0,1) respectively.

C. Constraints

To ensure that the domain/aspect embedding can reflect its sentiment feature distribution approximately, we constrain the domain/aspect embedding with sentiment features. Then the domains/aspects from either source or target are bridged indirectly via the feature space.

Domain Embedding. As shown in Fig. 3, the feature is mapped to the domain embedding space through a linear transformation and then constrained to approach its own domain embedding by minimizing the L2 norm. We define the constraint loss for domain embedding by

$$R^d = \frac{1}{|\mathcal{B}_s|}\sum_{\mathcal{B}_s}||T(\boldsymbol{f}_{s_i}) - \boldsymbol{d}_{s_i}||_2 + \frac{1}{|\mathcal{B}_t|}\sum_{\mathcal{B}_t}||T(\boldsymbol{f}_{t_j}) - \boldsymbol{d}_{t_j}||_2, \quad (4)$$

where $T()$ is the linear transformation, $\|\cdot\|_2$ is the L2 norm.

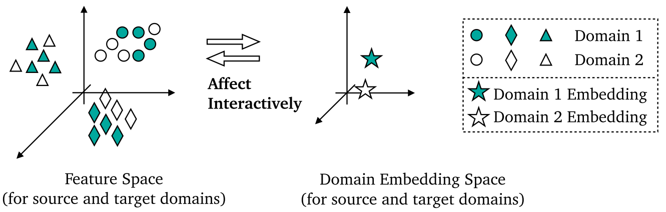

Affect Interactively
Feature Space (for source and target domains)
Domain Embedding Space (for source and target domains)
● ◆ ▲ Domain 1
○ ◇ △ Domain 2
★ Domain 1 Embedding
☆ Domain 2 Embedding

Fig. 3. Illustration of the interaction between sentiment features and domain embeddings. Small circles, diamonds, and triangles represent features from different classes, i.e., positive, neutral, and negative polarities. All features belonging to the same domain are employed to constrain its domain embedding. Two spaces affect each other interactively.

Algorithm 1. Optimizing

```
Input: $N$ labeled source domains $\{\mathcal{G}_{s_i}\}_{i=1}^{N}$;
       $M$ unlabeled target domains $\{\mathcal{G}_{t_j}\}_{j=1}^{M}$;
repeat
    for source batch $\mathcal{B}_s = \{X_s, Y_s\}$ from $\{\mathcal{G}_{s_i}\}_{i=1}^{N}$ do
        Sample target batch $\mathcal{B}_t = \{X_t\}$ from
          $\{\mathcal{G}_{t_j}\}_{j=1}^{M}$; $|\mathcal{B}_s| = |\mathcal{B}_t|$.
        Extract features $F_s$ and $F_t$.
        Compute $\mathcal{L}^c$ by Eq.2
        Compute $\mathcal{L}^d$ with $F_s$ and $F_t$ by Eq.3
        Compute $R^d$ by Eq.4
        Compute $R^a$ by Eq.5
        Compute $R^p$ with source instances by Eq.9
        Combine all losses with Eq.12 and update the
          parameters;
until performance on the developing data set does not
  improve in 5 epochs.
```

Aspect Embedding. The aspect embedding is also constrained similarly by

$$R^a = \frac{1}{|\mathcal{B}_s|}\sum_{\mathcal{B}_s}||T_a(\boldsymbol{f}_{s_i}) - \boldsymbol{a}_{s_i}||_2 + \frac{1}{|\mathcal{B}_t|}\sum_{\mathcal{B}_t}||T_a(\boldsymbol{f}_{t_j}) - \boldsymbol{a}_{t_j}||_2, \quad (5)$$

where $T_a()$ maps features into the aspect embedding space.

In this way, the sentiment feature is extracted based on its aspect and domain embeddings and at the same time, the feature also constrains the aspect and domain embeddings. This forms an interactive cycle. It is worth noting that the sentiment features fall into different classes (i.e., polarities). Therefore, different classes of a domain/aspect gradually move away from each other due to the training of the sentiment classifier. However, constraints, including R^d and R^a, reduce the distances between features and domain/aspect embeddings, which contradicts the training of sentiment classifiers. Therefore, the gradients from constraints R^d and R^a do not propagate to the feature extractor in our implementation.

D. Weight Computation

The theoretical guarantee of our method comes from the distribution weighted combination rule [29], [40], which indicates that the target distribution can be represented by a weighted combination of multiple source distributions. In the fine-grained ACST, we learn the optimally weighted combination of N source classifiers for each target domain. The rationality is that if a source domain is more similar to the target domain, the sentiment classifier trained by this source domain has higher confidence to infer the target one correctly. Thus its prediction gains higher weight. Concretely, the weight is computed by similarity measurement in the domain embedding space. We combine two granularities, i.e., *global similarity* between domains and *local similarity* between the instance and domain.

Since we do not have labeled data in the target domains, in the training phase, we only leverage source domains to learn the weight computation module and generalize to target domains in the testing phase.

1) Training Phase: Given N source domains, we consider each of them as the *"target"* domain, and use the rest $N-1$ source domains to compute the weight combination for it. Our main idea is that, for a source instance, the weighted combination by sentiment classifiers from other source domains, should be consistent with its ground-truth label. Specifically, given an instance from source domain $\mathcal{G}_{s_i}$, the extracted feature is $\boldsymbol{f}_{s_i}$ and its domain embedding is $\boldsymbol{d}_{s_i}$. We denote N source domain embeddings as $E_s = \{\boldsymbol{d}_{s_1}, \boldsymbol{d}_{s_2}, \ldots, \boldsymbol{d}_{s_N}\}$ and denote $E_{s'}$ by removing $\boldsymbol{d}_{s_i}$ from E_s. We feed $\boldsymbol{f}_{s_i}$ and $\boldsymbol{d}_{s_i}$ into the weight computation module to calculate weight with $E_{s'}$

$$\boldsymbol{g} = \mathrm{Sim}(\boldsymbol{d}_{s_i}, E_{s'}), \qquad \boldsymbol{l} = \mathrm{Sim}(T(\boldsymbol{f}_{s_i}), E_{s'}), \tag{6}$$

where $\boldsymbol{g} \in \mathbb{R}^{N-1}$ and $\boldsymbol{l} \in \mathbb{R}^{N-1}$ are the global similarity and local similarity respectively, and Sim computes the cosine similarity. We adopt a fusion gate to learn the contribution of two granularities

$$\boldsymbol{G} = \sigma(W_g[\boldsymbol{d}_{s_i}; T(\boldsymbol{f}_{s_i})] + b_g) \otimes (N-1), \tag{7}$$

$$\boldsymbol{\beta} = \mathrm{softmax}(\boldsymbol{G}\boldsymbol{g} + (1-\boldsymbol{G})\boldsymbol{l}), \tag{8}$$

where W_g and b_g are parameter and bias. W_g maps the vector $[\boldsymbol{d}_{s_i}; T(\boldsymbol{f}_{s_i})]$ into a scalar. $\otimes(N-1)$ means repeating $N-1$ times. $\boldsymbol{G}$ indicates the importance of global similarity.

The predictions by all other sentiment classifiers are denoted as $Y_{s'} = \{C_j(\boldsymbol{f}_{s_i})\}_{j=1}^{N}$ and $j \neq i$. The weighted prediction for this instance should be consistent with its sentiment label $\boldsymbol{y}$ and the loss is defined as

$$R^p = \frac{1}{|\mathcal{B}_s|} \sum_{\mathcal{B}_s} ||\boldsymbol{\beta} Y_{s'} - \boldsymbol{y}||_2. \tag{9}$$

2) Testing Phase: Fig. 4 shows the weight computation in the testing phase. The testing sets of M target domains are evaluated by the trained model. Given an instance $(x, a, d = t_j)$ from the target domain $\mathcal{G}_{t_j}$, its feature is first extracted, which is then fed to all source sentiment classifiers to obtain prediction results. To compute weights for these predictions, its feature $\boldsymbol{f}_{t_j}$ and domain embedding $\boldsymbol{d}_{t_j}$ are employed with all source domain embeddings E_s as below. The rationality is that we constrain all source and target domain embeddings in the same way. Therefore, the testing phase is consistent with the training

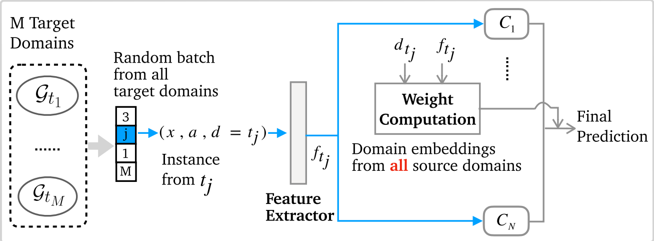


Fig. 4. The weight computation in the testing phase.

TABLE I
DATASET STATISTICS. #POS, #NEU, #NEG REPRESENT THE NUMBER OF INSTANCES WITH POSITIVE, NEUTRAL, AND NEGATIVE POLARITIES

Dataset			#Pos	#Neu	#Neg	#D	#A
YelpAspect	R	Train	45858	45676	15766	5	64
		Test	5162	4921	1716		
	H	Train	38822	35403	19720	11	35
		Test	4200	3881	2319		
	B	Train	45762	42576	16021	7	43
		Test	5056	4793	1823		
SemEval	R14	Train	1759	416	711	5	5
		Test	1039	175	326		
	R15	Train	995	43	350	6	13
		Test	657	55	399		

#Pos, #Neu, #Neg represent the number of instances with positive, neutral, and negative polarities. #D and #A denote the number of domains and aspects respectively.

$$\boldsymbol{g} = \mathrm{Sim}(\boldsymbol{d}_{t_j}, E_s), \qquad \boldsymbol{l} = \mathrm{Sim}(T(\boldsymbol{f}_{t_j}), E_s), \tag{10}$$

$$\boldsymbol{G} = \sigma(W_g[\boldsymbol{d}_{t_j}; T(\boldsymbol{f}_{t_j})] + b_g) \otimes N. \tag{11}$$

Then with (8), we obtain the weights $\boldsymbol{\beta}$. With predictions $Y_s = \{C_i(\boldsymbol{f}_{t_j})\}_{i=1}^{N}$ from all sentiment classifiers, the weighted prediction $\boldsymbol{\beta} Y_s$ is the final output.

E. Loss

We combine all losses into an overall objective function

$$\mathcal{L} = \mathcal{L}^c + \mu \mathcal{L}^d + \lambda(R^a + R^d + R^p), \tag{12}$$

where μ and λ balance the effect of different losses.

IV. EXPERIMENTS

A. Datasets

The ACST experiments are conducted with the following two public datasets.

YelpAspect. This dataset[2] [8] consists of three English sub-datasets: Restaurant (R), Hotel (H), and BeautySpa (B). The statistics of them are displayed in Table I. In this dataset, aspects are hierarchical, such as *food_breakfast* or *food_snack*. The top level of an aspect is set as the domain. The original ACST is converted to fine-grained ACST. Moreover, for balancing the

[2] https://github.com/hsqmlzno1/MGAN

scale of different domains, the domains that have less than 1000 instances are filtered out. It is worth noting that this dataset is originally annotated by *Opinion Parser* [41], which might be noisy. Thus we also choose SemEval for evaluation.

SemEval. We conduct experiments on the more widely used English benchmarks from SemEval 2014 task 4 [42] Restaurant 14 (denoted as R14) and SemEval 2015 task 12 [43] Restaurant 15 (denoted as R15). In R14, the aspect category is a single word, which is directly set as the domain. In R15, the top level of an aspect is set as the domain.

B. Implementation Details

We construct six ACST tasks: R → H, H → R, R → B, B → R, H → B, B → H on YelpAspect, two ACST tasks: R14→R15, R15→R14 on SemEval. The arrow → represents the transfer direction from the source to the target data. Grouped by aspects, each task is transformed into a fine-grained version: $\{\mathcal{G}_{s_i}\}_{i=1}^{N} \rightarrow \{\mathcal{G}_{t_j}\}_{j=1}^{M}$. The training set is composed of two parts: the labeled source training set and the unlabeled target training set. The source test set is adopted for validation. The target test set is used for evaluation.

To make a fair evaluation, all models and baselines are implemented by Pytorch with the same hyperparameters. The word embeddings are initialized with 100-dimension Glove vectors [44] and fine-tuned during the training. All other parameters are initialized by sampling from a normal distribution of $\mathcal{N}(0, 0.02)$. The hidden size of BiLSTM is set to 50×2. The dimension of feature, domain embedding and aspect embedding are 100, 5, and 20 respectively. Moreover, μ and λ are set to 0.5 and 0.01. The model is optimized by the Adadelta method [45] with a learning rate of 1.0. The batch size is 128 and 8 on YelpAspect and SemEval, respectively. An early stop strategy is exploited, which stops the training if the accuracy on the validation set does not improve in 5 epochs. And the best model is chosen for evaluation. To learn the parameters efficiently, the N sentiment classifiers are packed as a three-dimensional tensor $[N, 100, 3]$, where 100 is the feature dimension and 3 is the three sentiment polarities. The key to the implementation is using index_select() in Pytorch. We could choose a specific classifier for each instance within a batch. Then, our method is trained efficiently with batch learning. All the reported results for baselines and our approach are the average of 8 runs. The seeds are [5, 10, 15, 20, 25, 30, 35, 40].

C. Compared Methods

Single-Source Single-Target Domain Adaptation (SSDA) Baselines We do not group the aspects in the source and target datasets.

- *IATN* [46]: It learns interactive attention between the sentence and aspect to transfer sentiment knowledge crossing domains.
- *MMD* (Maximum Mean Discrepancy) [47]: It empirically mitigates the domain divergence by directly minimizing the distance between feature representations of source and target datasets [24]. Thus the divergence between the source and target distributions can be reduced.
- *CORAL* (CORrelation ALignment) [25]: Similar to MMD, it minimizes the domain shift by aligning the second-order statistics of source and target distributions.
- *SRDC* [28]: This method uncovers the intrinsic target discrimination via discriminative clustering of target data. The clustering solutions are constrained by structural source regularization.
- *FixBi* [27]: It is a fixed ratio-based mixup to augment multiple intermediate domains between the source and target domain. In our experiments, the sequences of word embedding are mixed up.

Multi-source Single-target Domain Adaptation (MSDA) baselines We group the aspects in the source dataset only.

- *MDAN* [30]: It designs one classifier and multiple domain discriminators to address the MSDA setting. This is the hard version that combines all losses with max-min selection.
- *MDAN(s)* [30]: It is the soft version of MDAN that combines all losses as an overall objective.
- *MS-MMD*: Different from MMD, it reduces the MMD distance between combined target domains and each source one respectively.
- *MS-CORAL*: It is similar to MS-MMD but replaces MMD with CORAL.

Model Variants for Separate Target Domains

- *MMTN(sep)*: This variant has the same architecture as MMTN. The only difference is that it trains the target domains separately. In other words, MMTN(sep) indicates transferring knowledge from multiple source domains to each target domain individually.

D. Experimental Analysis

The classification accuracy and macro-f1 of various methods are reported in Table II, Table III and Table IV. The best scores on each metric are marked in bold. To validate our approach, we analyze the results from the following perspectives.

1) Compare With Baselines: As displayed in Table II, the first two parts show the results of baselines. Compared with SSDA methods, MMTN outperforms MMD, IATN, CORAL by +3.39%, +2.97%, +2.62% respectively on the average of all evaluation metrics. Particularly, compared with a strong baseline CORAL, MMTN achieves significant improvements by +3.44% accuracy and +3.80% macro-f1 on H → B. This verifies that our approach can improve the domain adaptation performance of ACST by fine-grained transfer. In addition, it is observed that the FixBi performs slightly worse than other SSDA methods. A possible reason is that the sentiment polarity is closely corresponding with the aspect category. When mixing up the word embeddings of two sentences from the source and target domains, it might introduce noises for attention mechanisms to learn less-accurate sentiment representation.

Moreover, MMTN consistently outperforms four MSDA baselines on the average metric. It is worth noting that our method achieves significant improvement compared with the strongest baseline MDAN. In Table II, the p-values between

TABLE II
EVALUATION RESULTS OF BASELINES AND OUR MODEL ON YELPASPECT IN TERMS OF ACCURACY(%) AND MACRO-F1(%)

Models	R→H		H→R		R→B		B→R		H→B		B→H		Avg
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	
MMD	71.13	67.30	75.82	71.79	74.46	69.56	76.60	72.62	75.42	72.35	74.74	71.91	72.808
IATN	74.36	72.16	74.84	70.56	76.42	71.97	76.06	71.51	75.63	71.92	72.72	70.61	73.230
CORAL	74.74	72.27	74.22	70.37	75.35	71.29	77.20	72.93	75.57	71.83	74.61	72.56	73.578
FixBi	68.20	66.23	71.98	66.51	71.11	66.57	72.59	68.33	71.41	67.60	68.22	65.47	68.691
SRDC	75.22	73.66	72.96	67.76	74.97	70.16	75.45	70.10	75.57	71.00	74.65	72.40	72.829
MS-MMD	72.86	71.20	74.42	69.93	75.39	71.06	74.28	69.34	75.02	71.45	72.34	70.35	72.303
MDAN(s)	74.52	72.75	75.57	71.24	76.92	73.22	76.09	72.07	77.15	73.17	74.08	72.39	74.097
MS-CORAL	75.00	73.36	75.89	71.84	77.13	73.63	76.27	71.97	76.51	72.53	74.93	73.10	74.347
MDAN	75.37	74.07	76.13	71.51	76.41	72.87	76.51	72.18	76.76	73.32	75.38	73.78	74.524
MMTN	**76.15**	**74.84**	**78.09**	**73.85**	**78.31**	**75.03**	**78.03**	**73.27**	**79.01**	**75.63**	**76.94**	**75.20**	**76.196**
(p-value)	.2681	.8482	.0003	.0060	.0003	7.6e-5	.0804	.1964	1.1e-5	.0005	.0348	.0240	1.5e-6

We list the p-values of the T-test between MMTN and MDAN.

TABLE III
EVALUATION RESULTS OF BASELINES AND OUR MODEL ON SEMEVAL IN TERMS OF ACCURACY(%) AND MACRO-F1(%)

Models	R14→R15		R15→R14		Avg
	Acc	F1	Acc	F1	
IATN	72.85	55.10	74.62	49.34	62.98
MMD	71.96	54.12	74.71	49.68	62.62
CORAL	65.68	51.57	74.64	49.67	60.39
FixBi	74.88	55.40	75.51	47.76	63.39
SRDC	78.57	55.58	74.35	47.51	64.01
MS-MMD	73.09	52.67	73.25	48.81	61.96
MDAN(s)	66.34	50.50	74.28	**50.93**	60.51
MS-CORAL	72.10	51.01	70.91	47.43	60.36
MDAN	76.42	53.03	73.34	48.58	62.84
MMTN	**79.75**	**58.06**	**76.82**	50.65	**66.32**
(p-value)	5.06e-5	.0019	.0189	.0046	0.0001

We list the p-values of the T-test between MMTN and MDAN.

TABLE IV
EVALUATION RESULTS OF VARIANTS OF OUR MODEL IN TERMS OF ACCURACY(%) AND MACRO-F1(%)

Models	B→R		R14→R15	
	Acc	F1	Acc	F1
MMTN (full model)	78.03	73.27	79.75	58.06
MEAN	76.16	71.38	75.25	55.58
w/o R^d	76.35	71.87	79.48	54.25
w/o R^p	77.32	73.16	79.75	55.49
w/o R^a	77.55	72.80	76.69	55.38
w/o global similarity	76.74	72.08	78.31	54.11
w/o local similarity	76.73	71.97	79.12	56.40

MEAN: remove R^a, R^d, R^p, and also the weight computation component from MMTN, and predict with the average of multiple sentiment classifiers.

MMTN and MDAN are almost all smaller than 0.05 (p-values<0.05 indicates the improvements are significant). MSDA baselines regard the multiple target domains as a whole, which ignores the relationship and distinction among them. The experimental results further validate that our approach successfully achieves appropriate transfer for each target domain.

The experimental results on SemEval are shown in Table III. It can be seen that MMTN achieves the best performance on almost all evaluation metrics. Compared with a strong baseline MDAN, MMTN significantly outperforms it (p-values<0.05). These results also prove the effectiveness of the proposed method by transferring sentiment at a fine-grained level.

2) Ablation Study: The results of variants are reported in Table IV. It is first observed that MMTN consistently outperforms MEAN on all evaluation metrics. This proves that the weight computation module is effective in learning weight for each target instance. Then we found that removing individual loss R^d, R^a and R^p causes a slight performance drop, respectively. The results of w/o R^a and w/o R^d show the effectiveness of constraints. And the results from w/o R^p validate the effectiveness of the weight computation component. We can also see that removing R^d causes more performance decreases compared with R^a and R^p. This shows that, in our approach, constraining domain embeddings with the feature distributions plays a key role. The constraints bridge all domains and facilitate the domain similarity measurement in the weight computation component. Finally, we observe that only using global similarity or local similarity results in slight performance declines. This shows the usefulness of the combination of both levels of similarities.

3) Evaluation on Separated Target Domains: Since there are multiple subdomains in the target domain of fine-grained ACST, we further compare the evaluation results on separated target domains. As reported in Fig. 5, three methods are compared, including MDAN, MMTN(sep) and MMTN. First, it is observed that MMTN outperforms MDAN in almost all target domains. MMTN achieves significant improvement by +13.04% on *location* domain of R → H and +4.66% on *food* domain of R → B. In addition, it is worth noting that a few domains are overlapping between source and target, such as the *experience* and *food* between R and H. We notice that MMTN outperforms MDAN not only on overlapping domains but also on unseen domains. This also verifies the effectiveness of fine-grained transfer on separated target domains. Second, MMTN(sep) indicates training MMTN with each target domain individually. By comparing MMTN and MMTN(sep), MMTN still outperforms MMTN(sep) in most separate target domains.

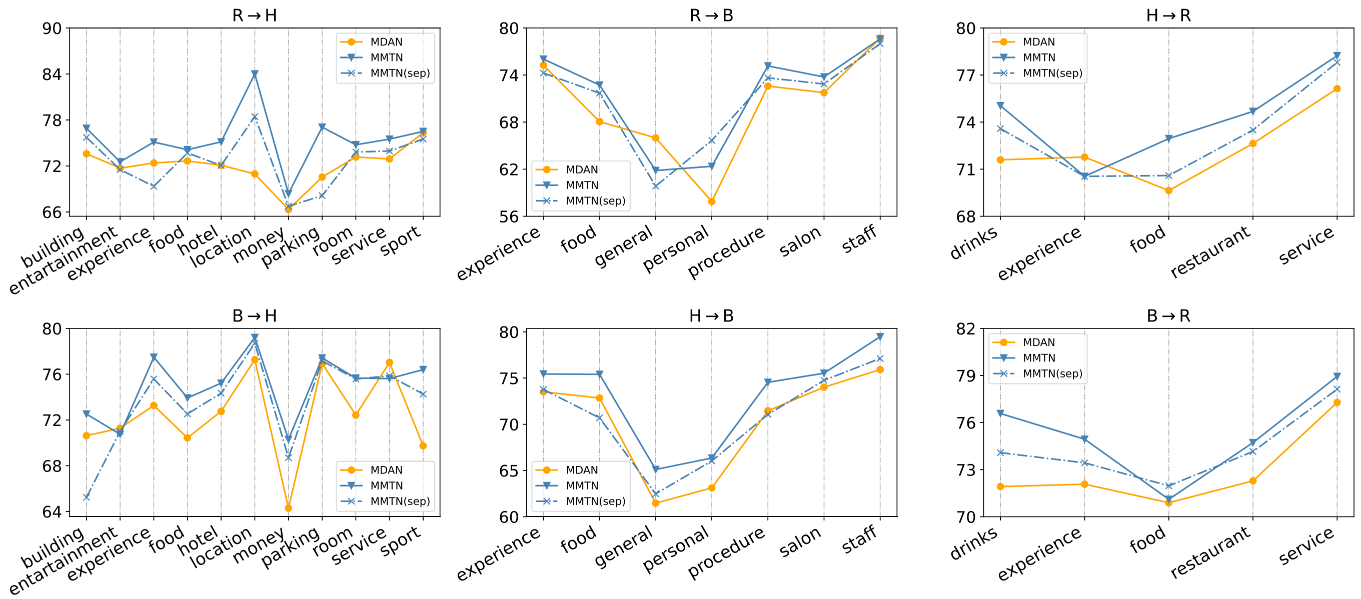


Fig. 5. Evaluation results of separated domains in the target testing set for six transfer pairs on YelpAspect. Each subfigure compares the results of MDAN, MMTN(sep), and MMTN in terms of macro-f1 (%).

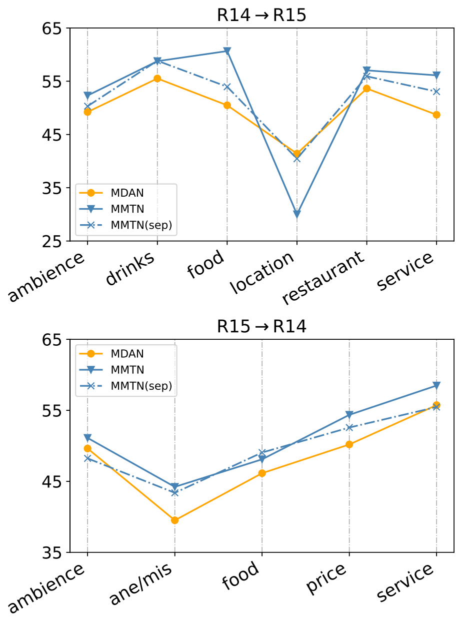


Fig. 6. Evaluation results of separated domains in the target testing set for two transfer pairs on SemEval. Each subfigure compares the results of MDAN, MMTN(sep), and MMTN in terms of macro-f1 (%). *ane/mis* is the *anecdotes/miscellaneous* for short.

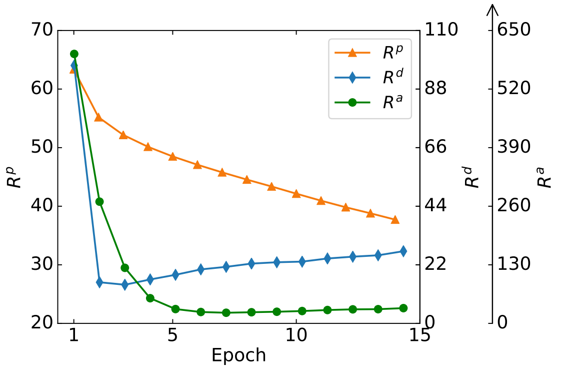


Fig. 7. The loss curves of R^a, R^d, and R^p during the training of MMTN on R → B.

Based on this observation, we guess that training with multiple target domains would bring more knowledge to each target one. This is also intuitive, where multiple target domains could enhance the domain-invariant sentiment knowledge with source ones.

As shown in Fig. 6, we display the evaluation results of the separate target domains on the SemEval dataset. We can see that the proposed method outperforms MDAN and MMTN(sep) on almost all separate target domains.

4) Constraint Losses: The loss curves of R^a, R^d and R^p are visualized in Fig. 7. Note that we use separated y-axes for them. Specifically, the training loss of the weight computation module, i.e., R^p, follows the decreasing trend. R^p ensures the consistency between weights of multiple predictions and the ground-truth label. The curve illustrates the learnability of the weight computation component. Therefore, this contributes to calculating more accurate weights for target domains through their domain embeddings.

The constraints on aspect embeddings R^a and domain embeddings R^d decrease dramatically at the beginning but gradually increase after several epochs. This is because they are both constrained by sentiment features. All features, that belong to the same domain, fall into different sentiment polarities, including positive, neutral and negative. As the training of sentiment classifiers, the features of diverse classes gradually move away from each other. Thus, the L2 distance between domain embedding and features, i.e., R^d, slightly increases, and so does R^a. It is also worth noting that R^a and R^d pull the features together,

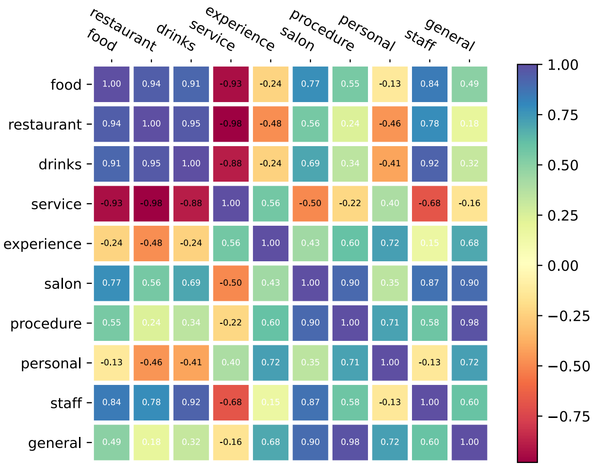


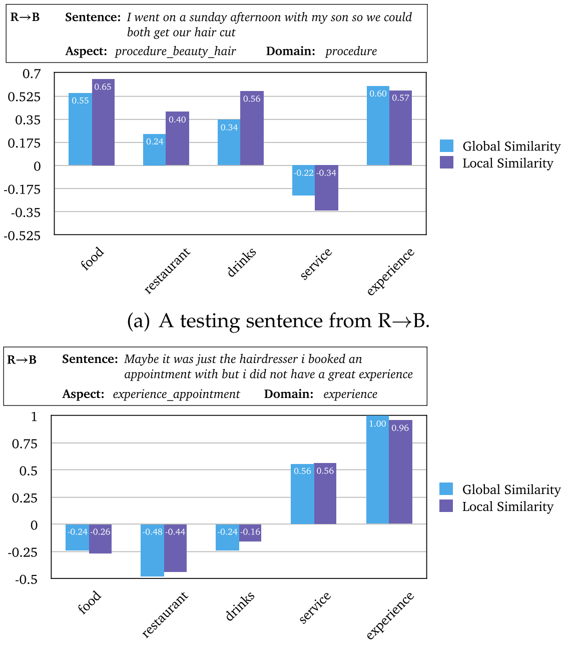
Fig. 8. Cosine similarities of all domains from the transfer task R → B. It is worth noting that R and B contain 5 and 7 domains, respectively. The number of domains in task R → B is 10. The reason is that *food* and *experience* are two overlapping domains in this task.

which might reduce the separability of the features. Therefore, the gradients from R^a and R^d do not propagate to the features, but only influence aspect embeddings and domain embeddings (see Section 3.3).

5) Domain Embedding: To validate the domain embeddings learned by MMTN, we visualize the cosine similarities of transfer task R → B in Fig. 8. Every single block represents the cosine similarity between two domain embeddings. It can be seen that various domain pairs have different similarities. We first found that some pairs gain a larger score, such as *food* and *drinks*, *food* and *restaurant*, *drinks* and *restaurant*. The cosine similarities of the above three pairs are all larger than 0.9. These domains are closely related because customers usually comment about the food and drinks in the restaurant. Thus they share more common opinion features, which contributes to the similarity of their domain embeddings. For another example, the cosine score between *service* and *experience* is 0.56, which is the highest score among all the pairs between *service* and the other domains. This is because the service potentially affects the experience of customers. In addition, *procedure* is also similar to *experience*. The reason is that the procedure of production is closely connected with the experience of customers. Then we can see some divergent domains do receive smaller scores, such as *food* and *service*, *drinks* and *service*, *salon* and *service*. The cosine similarities of the above three pairs are all smaller than -0.4. These results are in line with expectations.

However, there exist domain pairs that have inaccurate similarities. For example, the cosine score between *service* and *staff* is -0.68. In our intuition, their similarity should be large since the service is provided by the staff. Yet the score does not indicate this potential relationship. Thus, an overall conclusion is drawn that to a certain extent, domain embeddings can represent the feature distribution of a domain, and the cosine score calculated by domain embeddings reflects the similarity between domains.

6) Case Study: To further explore the similarity scores learned by MMTN, we visualize the global similarity and local similarity between the target instance and all source domains.


R→B **Sentence:** *I went on a sunday afternoon with my son so we could both get our hair cut*
Aspect: *procedure_beauty_hair* **Domain:** *procedure*

(a) A testing sentence from R→B.

R→B **Sentence:** *Maybe it was just the hairdresser i booked an appointment with but i did not have a great experience*
Aspect: *experience_appointment* **Domain:** *experience*

(b) A testing sentence from R→B.

Fig. 9. Similarities visualization, where global similarity is computed with domain embeddings, and local similarity is calculated by feature and domain embeddings (see Section 3.4.1).

The case studies are depicted in Fig. 9. The first sentence (Fig. 9(a)) is a testing instance from target domain *procedure* of R → B. There are five source domains. It can be observed that the global similarity reflects the relationships between domains, where *procedure* is most similar to the source domain *experience*. We can also see that the local similarity revises the relationships between a specific instance with source domains. The target sentence expresses *neutral* polarity for the aspect *procedure_beauty_hair*. Its local similarities with some source domains are larger than the global ones considering its specific contextual information might be similar to the source domains. For example, *"I went on a Sunday afternoon"* might also be utilized to describe *restaurant* and *drinks* simultaneously. The combination of global and local similarities contributes to learning the weight computation component. The second case is shown in Fig. 9(b), which displays a testing instance from target domain *experience* of R → B. This target domain is also existing in the source domains. As presented in Fig. 9(b), the global similarity between *experience* and *service* is 0.56. The reason is that service would influence the experience of customers. We further see that the local scores are similar to the global ones, which provide references to make the prediction.

In summary, the global and local similarities, to some extent, reflect the semantic relationships among domains and features. While sometimes the similarities are not as the expectation of human common sense. For example, the global similarity between *service* and *procedure* is negative, while the score of *food* and *procedure* is positive. A possible reason is that the extracted features might be noisy and less accurate, making the global and local similarities also less precise.

Sentence	Aspect Category	Domain	Prediction	Ground Truth
(a) There is no way it justifies the accolades it receives , the attitude of the staff or the wait for a table .	service#general	service	Positive	Negative
(b) The menu looked great , and the waiter was very nice , but when the food came , it was average .	food#quality	food	Positive	Neutral
(c) Food was Luke warm	food#quality	food	Positive	Negative
(d) The space is limited so be prepared to wait up to 45 minutes - 1 hour , but be richly rewarded when you savor the delicious indo-chinese food .	ambience#general	ambience	Positive	Neutral
(e) Be prepared to wait , because the place is pretty tiny .	ambience#general	ambience	Positive	Negative
(f) Also , they do not take credit card so come with cash !	restaurant# miscellaneous	restaurant	Negative	Neutral

Fig. 10. Typical error instances chosen from the testing target domains of transfer task R14→R15.

7) Error Analysis: As displayed in Fig. 10, typical error instances are listed. They can mainly be concluded into four error types. First, the multi-aspect sentence tends to be difficult to analyze. Specifically, there may exist multiple aspects in a sentence. Take sentence (b) as an example, the sentiment polarity towards *menu* and *staff* are positive, which is noise for *food#quality*. Then for *food#quality*, the sentiment polarity is wrongly predicted as *positive*, which is actually *neutral*. An analogous problem also happens in sentence (d). The opinion *"richly rewarded"* describes positive sentiment for *"Chinese food"*. However, this is noise for *ambience#general*, making the sentiment polarity wrongly predicted as *positive*. Second, the opinion is implicit, which is difficult for ACST. In sentence (a), there is no explicit opinion, but potentially expresses dissatisfaction toward service. Sentence (f) is alike. Such a situation is difficult even for closed-domain sentiment analysis, which is also a big challenge for the affective computing area. Third, an unknown word may cause an error. In sentence (c), *"Luke"* is an unknown word, indicating that this word does not exist in the source domains. This makes correct prediction challenging. Fourth, ambiguous words result in different sentiment polarities. Take sentence (e) as an example, *"pretty tiny"* is *negative* in the *ambience* domain. Yet *"pretty"* is both an adjective and adverb. It might be *positive* when being used as an adjective, such as *"pretty decoration"*, *"pretty environment"*. This leads to errors.

8) Model Scale: Since the proposed model design a sentiment classifier for each source domain, it might increase the model scale. Here we compare the number of parameters of various approaches. As shown in Fig. 11, it can be seen that FixBi has the largest number of parameters, almost twice compared with other models. The reason is that FixBi designs separate neural networks for source and target domains, respectively. Two networks are interactively learned via mixing up representations. In addition, it can be observed that the proposed MMTN has only a slightly larger number of parameters compared to other methods. Thus, under the experiments of this work, MMTN is parameter-friendly. Yet MMTN still has a potential drawback that the model scale will become larger as the number of domains grows. This problem also exists in MDAN and MDAN(s), which design multiple domain discriminators for source domains.

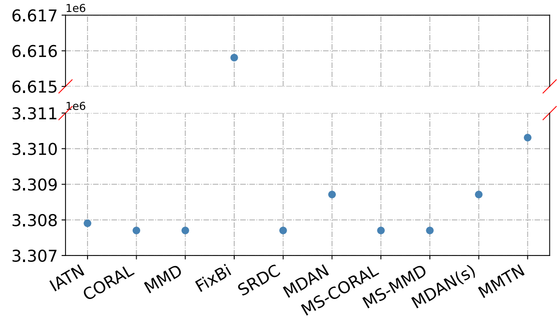


Fig. 11. Number of parameters in different models.

E. Experiments on General Multi-Domain Dataset

The above experiments are conducted on two public datasets about aspect category level sentiment. To further evaluate the proposed approach, we choose a more general multi-domain dataset [48]. The details are introduced below.

- *DRANZIERA* [48] It is a multi-domain opinion mining dataset, which contains 20 domains and each domain consists of 50,000 instances. The domains include *[Amazon_Instant_Vide, Automotive, Baby, Beauty, Books, Clothing_Accessories, Electronics, Health, Home_Kitchen, Movies_TV, Music, Office_Products, Patio, Pet_Supplies, Shoes, Software, Sports_Outdoors, Tools_Home_Improvement, Toys_Games, Video_Games]*. This dataset is designed for sentence-level sentiment classification, with two sentiment polarities, i.e., positive and negative. In the experiments, we choose the first half of domains as the source data, denoted as S, and the second half of domains as the target data, denoted as T.

The name of a domain is regarded as the aspect category and the first level. For example, *Amazon* is regarded as the grouped domain name of *Amazon_Instant_Vide*. In this way, the proposed method and baselines could be used in this dataset without extra network modification. We construct two ACST tasks S → T and T → S on DRANZIERA. The batch size is set to 32. Other hyper-parameters are the same as the previous experiments. We randomly sample 2,000 samples from each domain and

TABLE V
EVALUATION RESULTS OF BASELINES AND OUR MODEL ON DRANZIERA IN TERMS OF ACCURACY(%) AND MACRO-F1(%)

Models	S→T		T→S		Avg
	Acc	F1	Acc	F1	
IATN	84.07	83.23	84.62	84.17	84.02
MMD	84.47	83.80	84.94	84.52	84.43
CORAL	84.82	84.28	84.74	84.35	84.54
FixBi	85.12	84.84	82.89	82.56	83.85
SRDC	85.02	84.58	84.89	84.47	84.74
MDAN	62.21	53.03	63.99	55.88	58.78
MS-MMD	84.94	84.36	84.15	83.73	84.30
MS-CORAL	85.48	84.93	84.68	84.29	84.85
MDAN(s)	85.31	84.73	84.88	84.46	84.85
MMTN	**86.26**	**85.71**	**85.87**	**85.44**	**85.82**
(p-value)	0.0038	0.0053	8.1e-5	2.8e-5	4.3e-5

We list the p-values of the T-test between MMTN and MDAN(s).

TABLE VI
EVALUATION RESULTS OF BASELINES AND OUR MODEL ON DRANZIERA, WHEN SAMPLING 5,000 SAMPLES FROM EACH DOMAIN, IN TERMS OF ACCURACY(%) AND MACRO-F1(%)

Models	S→T		T→S		Avg
	Acc	F1	Acc	F1	
IATN	87.47	87.08	87.63	87.01	87.30
MDAN	84.63	84.06	83.45	83.20	83.84
MDAN(s)	87.65	87.28	87.48	87.01	87.36
MMTN	**87.82**	**87.42**	**87.70**	**87.20**	**87.54**

split the training and testing set at the ratio of 0.8 : 0.2. The experimental results are displayed in Table V. All the reported results for baselines and our method are the average of 8 runs. It can be observed that MMTN achieves the best performances. Comparing a strong baseline MDAN(s), the improvements of MMTN are significant.

In addition, when sampling 5,000 samples from each domain, the results are reported in Table VI. The improvements of MMTN are not significant. Comparing the differences between 2,000 and 5,000 samples per domain, the first observation is that the proposed method MMTN performs better in the resource-limited scenario. In addition, we guess that when the training samples are enough, such as 5,000 in each domain, the sentiment distribution between S and T tends to have little differences, where S and T are both combined with 10 domains, respectively. Thus different domain adaptation methods, e.g., IATN, MDAN, MDAN(s) and MMTN, achieve similar performances.

V. CONCLUSION

In this article, we address aspect category level sentiment analysis from the view of unsupervised domain adaptation. Most domain adaptation methods in sentiment analysis are coarse-grained, which ignore the relationships between different aspect categories. We propose a fine-grained domain adaptation method by considering the adaptation between subdomains. Specifically, the source/target domain is divided into multiple subdomains according to the aspect category. We further propose MMTN to achieve fine-grained transfer. Extensive experiments are conducted on three public datasets and the results demonstrate the effectiveness of our approach.

In light of the above analyses and observations, potential future directions may include representing domains with more accurate domain embeddings, deeply studying the overlapping and non-overlapping domains between the source and target datasets, introducing external knowledge during domain adaptation, etc. Moreover, studying multi-source multi-target domain adaptation in more general scenarios is also an interesting direction, which has been under-investigated.

ACKNOWLEDGMENTS

We sincerely thank all the anonymous reviewers for providing valuable feedback.

REFERENCES

[1] P. Zhu, Z. Chen, H. Zheng, and T. Qian, "Aspect aware learning for aspect category sentiment analysis," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 6, pp. 1–21, 2019.

[2] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 2, pp. 845–863, Second Quarter 2020.

[3] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 606–615.

[4] C. Chen, Z. Teng, Z. Wang, and Y. Zhang, "Discrete opinion tree induction for aspect-based sentiment analysis," in *Proc. Conf. Assoc. Comput. Linguistics*, 2022, pp. 2051–2064.

[5] Y. Ling, J. Yu, and R. Xia, "Vision-language pre-training for multimodal aspect-based sentiment analysis," in *Proc. Conf. Assoc. Comput. Linguistics*, 2022, pp. 2149–2159.

[6] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[7] W. Wang and S. J. Pan, "Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction," in *Proc. Conf. Assoc. Comput. Linguistics*, 2018, pp. 2171–2181.

[8] Z. Li, Y. Wei, Y. Zhang, X. Zhang, and X. Li, "Exploiting coarse-to-fine task transfer for aspect-level sentiment classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4253–4260.

[9] M. Hu, Y. Wu, S. Zhao, H. Guo, R. Cheng, and Z. Su, "Domain-invariant feature distillation for cross-domain sentiment classification," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 5559–5568.

[10] F. Wu and Y. Huang, "Sentiment domain adaptation with multiple sources," in *Proc. Conf. Assoc. Comput. Linguistics*, 2016, pp. 301–310.

[11] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," 2022, *arXiv:2203.01054*.

[12] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 452–461.

[13] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, "Dual graph convolutional networks for aspect-based sentiment analysis," in *Proc. Conf. Assoc. Comput. Linguistics*, 2021, pp. 6319–6329.

[14] H. Chen, Z. Zhai, F. Feng, R. Li, and X. Wang, "Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction," in *Proc. Conf. Assoc. Comput. Linguistics*, 2022, pp. 2974–2985.

[15] S. Liang, W. Wei, X.-L. Mao, F. Wang, and Z. He, "BiSyn-GAT: Bi-syntax aware graph attention network for aspect-based sentiment analysis," in *Findings Conf. Assoc. Comput. Linguistics*, 2022, pp. 1835–1848.

[16] Z. Zhang, Z. Zhou, and Y. Wang, "Ssegcn: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2022, pp. 4916–4925.

[17] E. Cambria, Q. Liu, S. Decherchi, F. Xing, and K. Kwok, "SenticNet 7: A commonsense-based neurosymbolic ai framework for explainable sentiment analysis," *Proc. Int. Conf. Lang. Resour. Eval., Conf. Proc.*, 2022, pp. 3829–3839.

[18] B. Zhang et al., “Sentiment interpretable logic tensor network for aspect-term sentiment analysis,” in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 6705–6714.

[19] J. Kocoń et al., “Neuro-symbolic models for sentiment analysis,” in *Proc. Int. Conf. Comput. Sci.*, 2022, pp. 667–681.

[20] M. Hu et al., “Multi-label few-shot learning for aspect category detection,” in *Proc. Conf. Assoc. Comput. Linguistics*, 2021, pp. 6330–6340.

[21] B. Liang et al., “Few-shot aspect category sentiment analysis via meta-learning,” *ACM Trans. Inf. Syst.*, pp. 1–33, 2022.

[22] B. Liang et al., “Embedding refinement framework for targeted aspect-based sentiment analysis,” *IEEE Trans. Affect. Comput.*, to be published, doi: 10.1109/TAFFC.2021.3071388.

[23] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, “Covariate shift by kernel mean matching,” *Dataset Shift Mach. Learn.*, vol. 3, no. 4, 2009, Art. no. 5.

[24] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1–9.

[25] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[26] H. Liu, M. Long, J. Wang, and M. Jordan, “Transferable adversarial training: A general approach to adapting deep classifiers,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4013–4022.

[27] J. Na, H. Jung, H. J. Chang, and W. Hwang, “Fixbi: Bridging domain spaces for unsupervised domain adaptation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1094–1103.

[28] H. Tang, K. Chen, and K. Jia, “Unsupervised domain adaptation via structurally regularized deep clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8725–8735.

[29] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation with multiple sources,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1041–1048.

[30] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, “Adversarial multiple source domain adaptation,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8559–8570.

[31] A. Schoenauer-Sebag, L. Heinrich, M. Schoenauer, M. Sebag, and S. J. Altschuler, “Multi-domain adversarial learning,” in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.

[32] O. Tasar, A. Giros, Y. Tarabalka, P. Alliez, and S. Clerc, “DAugNet: Unsupervised, multisource, multitarget, and life-long domain adaptation for semantic segmentation of satellite images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1067–1081, Feb. 2021.

[33] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” in *Proc. Conf. Assoc. Comput. Linguistics*, 2007, pp. 440–447.

[34] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, “Cross-domain sentiment classification via spectral feature alignment,” in *Proc. Int. Conf. World Wide Web*, 2010, pp. 751–760.

[35] J. Yu and J. Jiang, “Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 236–246.

[36] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: A deep learning approach,” in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 513–520.

[37] M. Chen, Z. Xu, K. Weinberger, and F. Sha, “Marginalized denoising autoencoders for domain adaptation,” in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1627–1634.

[38] J. Guo, D. Shah, and R. Barzilay, “Multi-source domain adaptation with mixture of experts,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4694–4703.

[39] Q. Liu, Y. Zhang, and J. Liu, “Learning domain representation for multi-domain sentiment classification,” in *Proc. Conf. North Amer. Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 541–550.

[40] J. Hoffman, M. Mohri, and N. Zhang, “Algorithms and theory for multiple-source adaptation,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8246–8256.

[41] K. Bauman, B. Liu, and A. Tuzhilin, “Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 717–725.

[42] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, “Semeval-2014 task 4: Aspect based sentiment analysis,” in *Proc. Int. Conf. Semantic Eval.*, 2014, pp. 27–35.

[43] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, “SemEval-2015 task 12: Aspect based sentiment analysis,” in *Proc. Int. Conf. Semantic Eval.*, 2015, pp. 486–495.

[44] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

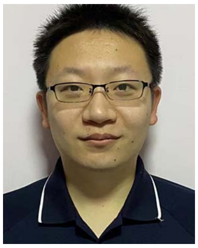
[45] M. D. Zeiler, “Adadelta: An adaptive learning rate method,” 2012, *arXiv:1212.5701*.

[46] K. Zhang, H. Zhang, Q. Liu, H. Zhao, H. Zhu, and E. Chen, “Interactive attention transfer network for cross-domain sentiment classification,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5773–5780.

[47] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, “A kernel method for the two-sample-problem,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

[48] M. Dragoni, A. Tettamanzi, and C. da CostaPereira, “DRANZIERA: An evaluation protocol for multi-domain opinion mining,” in *Proc. Int. Conf. Lang. Resour. Eval., Conf. Proc.*, 2016, pp. 267–272.

Mengting Hu received the BS degree from Tongji University, Shanghai China, in 2015, and the MS and PhD degrees from Nankai University, Tianjin China, in 2018 and 2021, respectively. The PhD degree is jointly educated with IBM CRL. She is currently an assistant professor with the College of Software, Nankai University. Her research interests are sentiment analysis, domain adaptation and few-shot learning.



Hang Gao received the BS degree from the University of Electronic Science and Technology of China, Chengdu China, in 2015, and the MS and PhD degrees from Nankai University, Tianjin China, in 2018 and 2021, respectively. He is currently a postdoctoral scholar with the Institute of Public Safety Research, Tsinghua University. His research interests are computer vision, information security, and social computing.

Yike Wu received the BS and PhD degrees from Nankai University, Tianjin China, in 2017 and 2022, respectively. The PhD degree is also jointly educated with IBM research-China. He is currently a lecturer with the School of Journalism and Communication, Nankai University. His research interests include vision-language understanding and reasoning, natural language processing, and social computing on news.

Zhong Su received the PhD degree in computer science from Tsinghua University, in 2002. He is a senior technical staff member with the IBM Research-China (CRL) and senior manager of the Cognitive Understanding & Analytics department. He has been involved in many projects in CRL including text analytics, NLP, rich media analysis, and information integration. Now, he is a senior technical director with Alibaba Research, China.

Shiwan Zhao received the BS and MS degrees in computer science from Tsinghua University, Beijing China, in 1998 and 2000, respectively. He was a research scientist with IBM Research-China from 2000 to 2020. His research interests include recommender systems, cognitive healthcare, computer vision, and natural language processing.