EATN: An Efficient Adaptive Transfer Network for Aspect-level Sentiment Analysis

Kai Zhang, Qi Liu, *Member, IEEE,* Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, *Member, IEEE,* Enhong Chen, *Senior Member, IEEE*

Abstract—Aspect-level sentiment analysis is a granular emotional classification task that refers to identifying sentiment polarities towards aspects in a sentence. Although previous research has reached a great achievement, this task remains very challenging. First, previous approaches only focus on one specific domain, which lacks the capability of transferring to other domains. Moreover, the majority of prior studies ignore the direct relationship between aspects and the corresponding sentiment words. To this end, in this paper, we propose a novel model named Efficient Adaptive Transfer Network (EATN) for aspect-level sentiment analysis which emphasizes the need to incorporate the correlation among multiple domains. The proposed EATN provides a Domain Adaptation Module (DAM) to learn common features from the sufficiently labelled source domain and guide the target domain's classification performance. Specifically, DAM comprises two special tasks, with one sentiment classification task aiming to learn sentiment knowledge and the other domain classification task focusing on learning domain-invariant features. Here, we adopt a multiple-kernel selection method to further reduce the feature discrepancy among domains. Besides that, EATN contains a novel aspect-oriented multi-head attention to capture the direct associations between the aspects and the contextual sentiment words, which is beneficial to learn the aspect-aware semantic knowledge. Extensive experiments on six public datasets with two granularities, compared with current state-of-the-art methods, demonstrate the effectiveness and universality of our method.

Index Terms—Aspect-level Sentiment Analysis, Domain Adaptation, Transfer Learning, Multi-head Attention Mechanism

1 Introduction

Aspect-level sentiment analysis is a branch of the sentiment classification, which aims to identify the sentiment polarity (i.e., positive, negative or neutral) of one specific aspect in a sentence. It is worth noting that an aspect represents a specific entity occurs in the sentence, which describes a finer granularity category of the sentence. For example, as shown in Fig. 1(a), the user expresses negatively and positively towards two aspects *"customer service"* and *"food"*, respectively. As this task is crucial in many real-life applications, such as dialogue system [1], [2], question-answering [3] and online commerce [4], a great amount of research attention has been attracted from both the academia and the industry.

In the literature, there are numerous efforts for this challenging problem, especially for sentiment relation mining between the aspect (e.g., *customer service*) and its context sentence (e.g.,*"The $, as a matter of fact, is not as good as I think, but I still like their food"*). The conventional way is to build an aspect-level sentiment classifier by supervised training. A fruitful stream of previous works [5], [6], [7] have designed several models to improve the performance for aspect-level sentiment analysis, typically by creating several features based on the intrinsic grammar structures of the sentences. These studies mainly focus on manual feature selection, which cannot extract deep implicit features well. Recently, some deep neural network (DNN) models [8], [9], [10] are proposed to automatically learn high-dimensional semantic relationship representations for the aspects and their contextual words. Besides, to model the sequential feature of aspect-level sentiment, researchers have also devised some Recurrent Neural Networks (RNNs) [11], [12], [13], [14]. After that, RNN-based methods, equipped with attention mechanism and memory module, aims to pay more attention to the important words in the contextual sentence. Those methods greatly improved classification accuracy and became a mainstream approach.

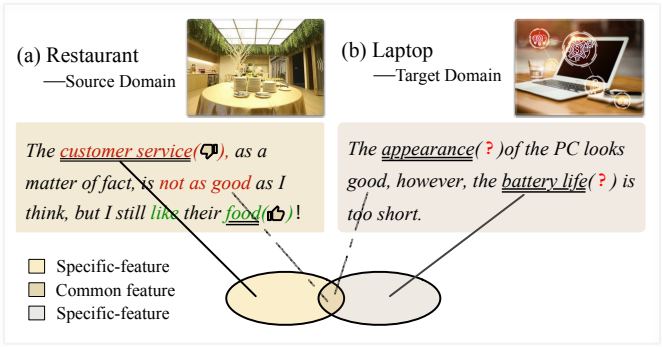


Fig. 1: **When transferring information from the source domain to the target domain directly, most existing approaches will suffer the domain discrepancy problem (e.g., different aspects), which will cause the performance drop.**

- *Kai Zhang is with the Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science, University of Science and Technology of China, Hefei, Anhui 230027, China. E-mail: kkzhang0808@mail.ustc.edu.cn*
- *Qi Liu and Enhong Chen are with the Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China. E-mail: {qiliuql, cheneh}@ustc.edu.cn*
- *Hao Qian, Biao Xiang, Qing Cui and Jun Zhou are with the Ant Financial Services Group, Hangzhou, Zhejiang 310000, China. E-mail: {qianhao.qh, xiangbiao.xb, cuiqing.cq, jun.zhoujun}@antgroup.com*

Corresponding author: Enhong Chen.

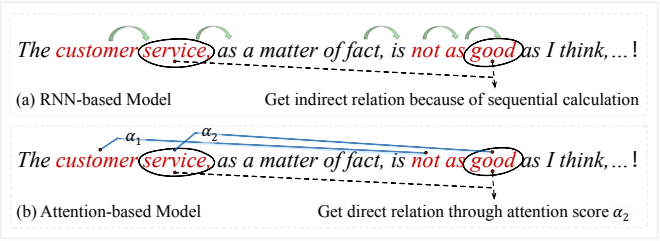


Fig. 2: **Comparison of the calculation process between RNN-based methods and Attention-based methods. Since (a) is calculated step by step, which may cause some information noise. In contrast, (b) directly calculates the relations between every two words through attention score α and can solve the problems above perfectly.**

Although significant improvement is brought by previous research, there still remain several challenges in most practical applications because of the following reasons. First, there is a huge amount of unlabelled data in some new domains (i.e., target domain), which are substantially different from the source domain. However, most existing methods focus on label-rich data and perform well on just one special domain. They usually lack universality and transferability when applying to a new domain. Thus, how to find an appropriate method to effectively aggregate both domains for comprehensive representation and make the model gain transferability is a nontrivial problem in the aspect-level sentiment analysis; Second, most deep neural networks learn features from domain-invariant to domain-specific, as the network becomes deeper, the feature transferability drops significantly. For example, DNN may learn the outline of an image in a shallow layer while learning more specific features in the deep layer. Hence, how to reduce feature bias in the deep layers is a significant problem for enhancing the transferability among domains. Third, different aspects of the sentence are associated with different contexts in the aspect-level sentiment analysis task. For instance, as shown in Fig. 2, the sentiment polarity of the aspect *"customer service"* should be determined by its corresponding aspect-aware sentiment words, i.e., *"not, good"*. However, sequential methods produce the representation of each word based on the previous one, which may introduce extra noise from unrelated words, e.g., *"the, matter, of, fact"*. Thus, it's a challenging problem to capture direct associations between aspects and aspect-aware sentiment words when understanding sentiment polarity of the aspects.

To cope with the aforementioned challenges, we propose a novel method named Efficient Adaptive Transfer Network (EATN), incorporating with multiple modules to solve the problems mentioned above in aspect-level sentiment analysis task. First, to integrate the two kinds of data sources and construct an adaptive method, we design a Domain Adaptation Module (DAM) to mine unlabelled data from the target domain. The key advantage of this module is that it contains two classification tasks, one of them referred as the domain task mainly for domain-invariant feature learning and the other one referred as sentiment task mainly for aspect-aware sentiment knowledge learning. Second, we design a novel Multiple-Kernel Maximum Mean Discrepancy (MK-MMD) based Multi-Layer Perceptron (MLP) for reducing feature discrepancy between different domains, which can simultaneously optimize the domain invariance and enhance the transferability of the features in the deep layer. Third, we design a novel Aspect-oriented Multi-head Attention mechanism to better extract the direct semantic relations among aspects and the aspect-aware sentiment words, i.e., as shown in Fig. 2(b). Finally, to evaluate our approach, we conduct extensive experiments on six real-world datasets with two granularities. The results demonstrate that our proposed approach significantly outperforms the state-of-the-art models in terms of classification performance and adaptation efficiency.

2 Related Work

In this section, we will introduce some research topics which are highly relevant to our work, i.e., aspect-level sentiment analysis, domain adaptation, and the multi-head attention.

Aspect-level Sentiment Analysis. Aspect-level sentiment analysis is a fine-grained sentiment classification task, which identifies the sentiment polarity of one specific aspect in the sentence. Some traditional approaches [6], [15] designed many rules-based models for aspect-level sentiment analysis. Nasukawa, et al. [5] first performed dependency parsing on the sentences, along with pre-defined rules to determine sentiment polarity about the aspects. Then, Jiang, et al. [7] improved the accuracy of sentiment classification by creating several target-dependent features based on the grammar structures of the sentences. In recent years, multiple Recurrent Neural Network based methods [10], [11], [12], [16] have been utilized for aspect-level sentiment classification problem and have shown its effectiveness in sequence modeling. Among them, Tang, et al. [17] approached this problem by developing a two-direction target-dependent LSTM (TD-LSTM) to model the left and right contexts of aspects. The last hidden states of these two LSTMs were concatenated to predict the sentiment label.

More recently, attention-based methods [18], [19] have been widely studied to enhance the influence of the aspects on the final representation for classification. Ma, et al. [12] developed an interactive attention network (IAN) to model the aspects and the sentence interactively. Fan, et al. [13] proposed a multi-grained attention network (MGAN), which is responsible for linking and fusing the words from the aspects and the contextual sentences. Li, et al. [14] designed a target-specific network (TNet) to better integrate the aspect information into the sentence representation. Xu, et al. [20] adopted bidirectional encoder representations from Transformers (BERT [21]) as a base model and proposed a joint post-training approach (BERT-PT) to enhancing both the domain and task knowledge. However, all of these supervised methods perform well in just one particular domain and lack generality ability, which makes it difficult to achieve the same performance in new unlabelled domains [22].

Domain Adaptation in NLP. In order to mitigate the applicability bottleneck from the domain shift, many domain adaptation methods have been proposed in the last decade. It can be simply treated as a standard semi-supervised problem in which the key idea is to transfer the common knowledge from the source domain to the target domain. In previous studies, Blitzer, et al. [23] designed a method,

which mainly utilized multiple shared features among domains to predict the final task. Pan, et al. [24] proposed a Spectral Feature Alignment (SFA) algorithm to solve the feature mismatch problem by aligning domain-specific words with the help of domain-independent words. Glorot, et al. [25] proposed a marginal stacking denoising autoencoder to improve the scalability and extract the domain-shared features from different domains. However, all the shallow methods mentioned above need to manually select some information such as shared or unshared features between the source domain and the target domain, which may bring huge labor expense.

Later, numerous researchers studied the neural network-based domain adaptation solutions [26], [27], [28]. For example, Yu, et al. [29] proposed two auxiliary tasks to learn sentence embedding based on convolutional neural network for cross-domain sentiment classification. Ganin, et al. [30] added adversarial mechanism into the training stage of deep neural networks, which is called domain-adversarial neural network (DANN). Along this line, Li, et al. [31] proposed an adversarial memory network that automatically identified common features. Li, et al. [32] also proposed a hierarchical attention transfer network (HATN) which paid different attention to the word-level and sentence-level sentiment to strengthen the final representation. Yang, et al. [33] devised an integrated approach which leverages the benefit of supervised deep neural networks as well as probabilistic generative models at the same time. Zhang, et al. [34] utilized the significant associations of the aspects in the sentence and proposed an interactive attention transfer network (IATN) for cross-domain sentiment classification. These researches realized the importance of domain adaptation, which has been extensively studied in other areas, e.g., image recognition and sentiment analysis. Unfortunately, most of the above domain adaptation researches focus on sentence-level instead of aspect-level sentiment analysis and they did not pay equal research attention to aspect-level sentiment analysis in domain adaptation scenario.

Multi-head Attention. Attention mechanism [10], [11] has become one of the most breakthrough technologies in recent years. However, the contextual vector obtained from traditional single attention mechanism usually focuses on one specific semantic subspace of the input sequence. Such a representation method can only reflect one semantic subspace of the input. However, most sentences, especially for long sentences, usually involve multiple semantic spaces. In order to solve this problem, the multi-head attention mechanism was designed and gained great success in many natural language processing (NLP) tasks, such as machine translation [35], [36], semantic role labelling [37], [38] and question answering [39], [40], [41]. The strength of multi-head attention lies in the rich expressiveness by using multiple attention functions in different representation subspaces. Besides, multi-head attention mechanism has been proved to be better than the traditional sequence model because of its effectiveness in extracting direct features between every two words [42]. Despite all the advantages, to the best of our knowledge, there is no prior work concentrate on mining the associations between the aspects and the aspect-aware sentiment words with a multi-head attention mechanism.

3 Efficient Adaptive Transfer Network

In this section, we first give the problem statement, followed by an overview of the framework. Then we explain three key components of EATN in detail, which are: 1) Embedding Module; 2) Aspect-oriented Multi-head Attention Module and 3) Domain Adaptation Module. Finally, we introduce the training strategy of the method.

3.1 Problem Statement

In this paper, we focus on the problem of unsupervised aspect-level sentiment analysis task in the domain adaptation scenario. Formally, we assume that there are two domains, one is source domain $\boldsymbol{\mathcal{D}}^s = \{x_i^s, a_i^s, y_i^s\}_{i=1}^{n_s}$ which has massive labelled data, where x_i^s is an item (e.g., review) and y_i^s is the associated sentiment label of its aspect a_i^s. Note that each aspect may contain several words. n_s represents the number of source domain data. The other is unlabelled target domain which is similarly defined as $\boldsymbol{\mathcal{D}}^t = \{x_j^t, a_j^t\}_{j=1}^{n_t}$, except the sentiment label is missing. We further assume that each item (i.e., x_i^s or x_j^t) at both domains consists of n context words denoted as $c = \{w_1^c, w_2^c, \ldots, w_n^c\}$ and the aspect contains m words denoted as $a = \{w_1^a, w_2^a, \ldots, w_m^a\}$. Note that one aspect may contain several words, such as "batter life" in the laptop domain. The goal is to train a robust model based on both labelled data in $\boldsymbol{\mathcal{D}}^s$ and unlabelled data in $\boldsymbol{\mathcal{D}}^t$ jointly, and adapt it to predict sentiment label of the aspects in the unlabelled target domain.

3.2 Overall Architecture of EATN

Since previous research either ignore the transferability between domains (e.g., aspect-level sentiment analysis methods) or ignore the impact of the aspect information (e.g., cross-domain sentiment classification methods). The goal of our model is to design a better cross-domain aspect-level sentiment classifier based on the domain adaptation technique. The overall model architecture is described in Fig. 3, EATN mainly contains three components: 1) *Embedding Module*: mapping each word into a low-dimensional real-value vector, and encoding sentence semantic in a more effective method, i.e., BERT; 2) *Aspect-oriented Multi-head Attention Module*: focusing on fully exploiting the information of the aspects as well as learning the deep semantic relationship between the aspects and contextual words effectively; 3) *Domain Adaptation Module (DAM)*: utilizing the data of both domains to train a model with two tasks and an auxiliary loss jointly, which makes the model transferable and sentiment aware at aspect-level. In what follows, we introduce how to achieve these components in detail.

3.3 Embedding Module

In this component, we introduce the details about the embedding process, which mainly consists of two parts: input preprocessing and word embedding.

1) Input Preprocessing: As we know, for each review, overall sentiment polarity is influenced by various aspects, which means that we have to face the situation where an item may include multiple aspects. However, our goal is to mine the specific aspect sentiment and all of the data needs to be processed in pairs (i.e., the aspect and the context

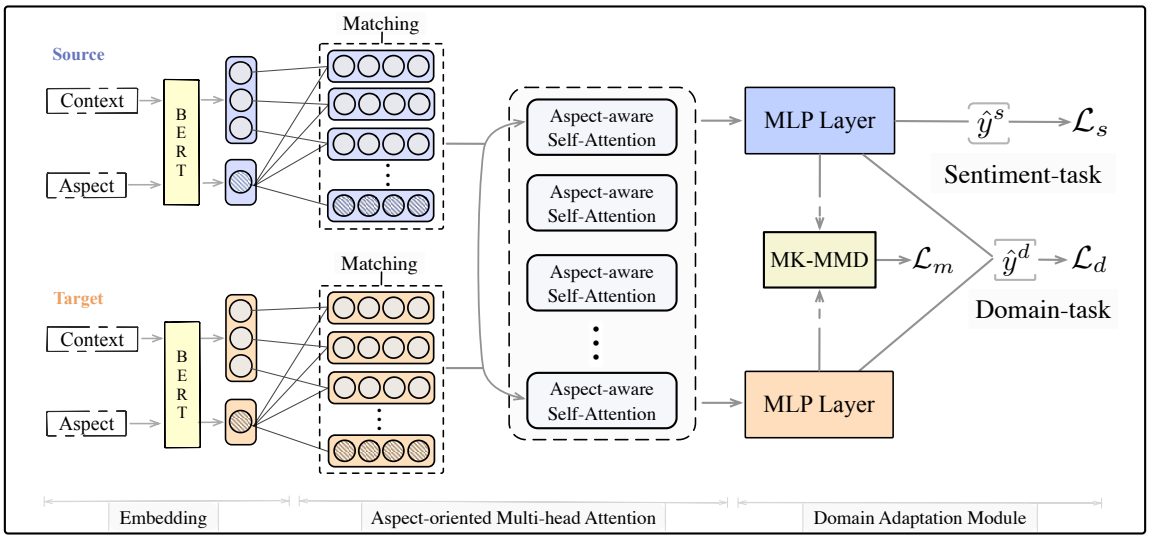


Fig. 3: **The overview of the proposed EATN model.**

correspond one-to-one, which are pre-given in the datasets). Thus, we split the sentence into multiple ones which each consist of just one aspect. For example, in the item *"The customer service, as a matter of fact, is not as good as I think, but I still like their food"*, we split it into two data items, which one is for aspect *"customer service"*–*"The $, as a matter of fact, is not as good as I think, but I still like their food"* and the other for aspect *"food"*–*"The customer service, as a matter of fact, is not as good as I think, but I still like their $"*. Note that the aspect words are replaced by the signal $ in the context part. Moreover, we also adopt "padding" operation to keep the input dimension consistent between domains.

2) Word Embedding: To represent semantic information of aspects and the context words better, we need to map each word into a low-dimensional real-value vector. Specifically, the inputs of our model are contextual word sequence $c = \{w_1^c, w_2^c, \ldots, w_n^c\}$ and corresponding aspect word sequence $a = \{w_1^a, w_2^a, \ldots, w_m^a\}$. There are several methods which can encode the original words into low-dimensional semantic embeddings. In this paper, we apply a popular pre-trained embedding method for word representation, i.e., Bidirectional Encoder Representations from Transformers (BERT[1]). BERT is a language model developed by Google for pre-training language representations, it contains two training methods: pre-training and fine-tuning. Pre-trained BERT models often show quite good performance on many tasks, e.g., sentiment analysis, machine translation. Here we adopt the large-scaled BERT model as the upstream feature extractor to learn the word's semantic embedding and froze most of its parameters to make the model more efficient.

To be specific, we take each word as input to get the contextual words embedding vectors $\boldsymbol{e}_c = \{e_1^c, e_2^c, \ldots, e_n^c\}$= BERT$(\{w_1^c, w_2^c, \ldots, w_n^c\})$ and the aspect word embedding vectors $\{e_1^a, e_2^a, \ldots, e_m^a\}$ = BERT$(\{w_1^a, w_2^a, \ldots, w_m^a\})$[2]. Note that, if the aspect is a single word like *"food"*, the aspect representation is the embedding of the word. While for the case where the aspect contains multiple words such as *"customer service"*, the aspect representation is the average of each word embedding [43]. We can denote the aspect embedding process as:

$$e_a = \begin{cases} e_1^a, \ \ if \ \ m = 1 \ , \\ (\sum_{\ell=1}^{m} e_\ell^a)/m \ , \ \ if \ \ m > 1 \ , \end{cases} \tag{1}$$

where m is the number of the aspect words, e_ℓ^a is the embedding of word ℓ in the aspect.

3.4 Aspect-oriented Multi-head Attention

Since it is beneficial to consider the impact between aspects and each contextual word, which can provide more information for aspect-level semantic understanding. In this subsection, we introduce how to model semantic relations between aspects and the contextual words more comprehensively. As Fig. 3 shows, this component consists of two parts, i.e., matching operation and aspect-aware self-attention.

1) Matching: To model the semantic relation between the aspects and it's context better, we leverage heuristic matching [44] between the aspect representations (i.e., e_a) and the contextual word representations (i.e., $e_1^c, e_2^c, \ldots, e_n^c$). In matching operation, we utilize three different calculations: concatenation, element-wise product and their difference. Concatenation is a simple and effective method to combine the two feature representations as well as retain all the information. The element-wise product is a certain measure of "similarity" of two words [45] and the difference can capture the degree of distributional inclusion in each dimension [46]. To be specific, we concatenate the aspects vector e_a, element-wise product vector and their difference vector with each contextual word vector e_i^c, $i \in$ (1, n) to retain the whole semantic information of the sentence. Also, to maintain the same latitude, we stack the aspect vector together as $\boldsymbol{h}_a$. Then, we denote the output result, i.e., H, as the input of

1. The detailed official introduction of BERT model can be found in following link: https://github.com/google-research/bert.

2. The detail of implementation can be found in following link: https://bert-as-service.readthedocs.io/en/latest/index.html

the aspect-aware self-attention. The process of the matching operation can be represented as follows:

$$\boldsymbol{h}_i = (e_i^c, e_a, e_i^c \odot e_a, e_i^c - e_a) \ , \tag{2}$$

$$\boldsymbol{h}_a = (e_a, e_a, e_a, e_a) \ , \tag{3}$$

$$H = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n, \boldsymbol{h}_a] \ . \tag{4}$$

2) Aspect-aware self-attention: After getting the combined representations, i.e., H, of contextual words and the corresponding aspects, we need to extract the deep semantic relationships between them. As we mentioned before, the traditional sequence model may cause additional noise and lead to information loss. In order to model the relationship between aspects and contextual words more accurately, we employ the attention mechanism, which allows the model to attend aspect-aware information. Besides that, as shown in Fig. 2(b), compared with traditional sequence models (e.g., LSTM, RNN and GRU), attention mechanism aims at modeling the relevance between each representation pairs, thus the aspects are allowed to build direct relationships with other aspect-aware sentiment words. Specially, the attention mechanism computes a query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$ of dimension d_q, d_q, d_v, which are linear projected by the input embedding representation (i.e., H):

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = H\boldsymbol{W}^Q, H\boldsymbol{W}^K, H\boldsymbol{W}^V, \tag{5}$$

where H is the output vector of the embedding module. $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, $\boldsymbol{W}^V$ represent parameter matrices. Therefore, we can get the attention output representation vector Z which is calculated as follows:

$$\begin{aligned} Z &= Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \\ &= softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_q}}\right)\boldsymbol{V} \ . \end{aligned} \tag{6}$$

Self-attention is an attention mechanism relating to different positions of the input sequence. Based on self-attention mechanism, multi-head attention can further jointly attend information from different representation subspaces to enhance the model's representation ability and improve the modeling performance. For example, in a sentence *"The customer service, as a matter of fact, is not as good as I think, but I still like their food"*, the user expresses negatively towards the aspect *"customer service"* at one subspace and positively towards the aspect *"food"* at another subspace. By applying multi-head attention in EATN, the network can model the different semantic dependencies between different aspects at different subspaces, thus selectively focusing on the aspect-oriented information in the feature learning process. Formally, multi-head attention mechanism consists of multiple heads that each head computes a unique scaled-dot product attention distribution. Furthermore, the multi-head attention mechanism has been proved with effectiveness in producing deep semantic representation in machine translation task [36]. To be specific, it performs multiple self-attention function t times to generate queries, keys, values matrices $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, $\boldsymbol{V}_i$ from $i = 1, 2, \ldots, t$. For each of the attention head, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are uniquely projected before the attention being computed by:

$$\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i = \boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V, \tag{7}$$

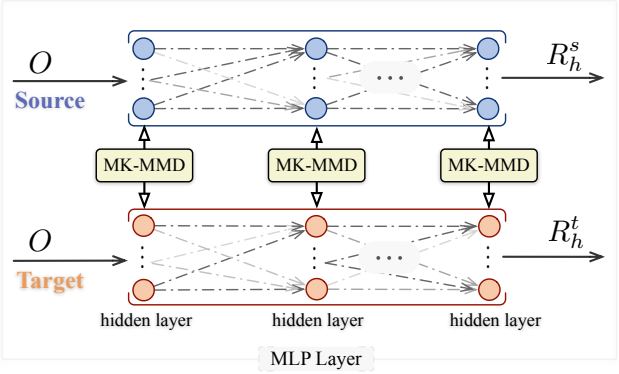


Fig. 4: **Inter-structure of MK-MMD based MLP for eliminating feature discrepancy and learning transferable features.**

$$Z_i = \ Attention(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) \ , \tag{8}$$

where $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, $\boldsymbol{V}_i$ are the query, key, and value representations of the *i-th* head, which are projected by matrices $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V$, respectively. Z_i is the output of the single attention head-i. Then, we concatenate the output states of attention head Z_i $(i = 1, 2, \ldots, t)$ to produce the final output state O:

$$O = [Z_1, Z_2 \ , \ldots, Z_t] \ . \tag{9}$$

where O represents the concatenation of vectors.

3.5 Domain Adaptation Module (DAM)

In this section, we introduce how EATN obtains transferability progressively through the DAM which mainly includes three components, i.e., the MLP layer with MK-MMD [47], sentiment classification, and domain classification.

1) MK-MMD based MLP Layer: As we demonstrated above, many DNN models [8], [9], [10] are proposed to automatically learn high-dimensional representations for aspects and their contexts [12]. Similarly, in order to build up deeper feature representations, the output state (i.e., O) of the previous layer is the input to MLP to extract the deep semantic relationships between the aspects and contextual words. There are h hidden layers in the MLP and each layer employ with a *Rectified Linear Unit* $(ReLu)$ activation function. The process is calculated by:

$$\begin{aligned} R_1 &= Relu(\boldsymbol{W}_1 O + \boldsymbol{b}_1) \ , \\ R_2 &= Relu(\boldsymbol{W}_2 R_1 + \boldsymbol{b}_2) \ , \\ &\ldots \qquad \ldots \qquad \ldots \\ R_h &= Relu(\boldsymbol{W}_h R_{h-1} + \boldsymbol{b}_h) \ , \end{aligned} \tag{10}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_h$ are trainable parameter matrices, $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_h$ are hidden unit biases and R_ℓ is the ℓ-th layer output representation of MLP, $\ell \in (1, h)$.

As mentioned above, our data are from two domains, in which one is labelled source domain and the other is the unlabelled target domain. Given a source domain item $(x^s, a^s) \in \boldsymbol{\mathcal{D}}^s$ and a target domain item $(x^t, a^t) \in \boldsymbol{\mathcal{D}}^t$, through the above operations, we get the source item representation R_ℓ^s and the target item representation R_ℓ^t. Until now, we get the representation of both domains by interacting the aspects with their contextual words in a more

efficient way. Then we take them into the following module to learn transferable knowledge and general features.

In standard neural networks, deep features eventually convert from the general to specific as the network becomes deeper. Thus, the transferability gap grows with the domain discrepancy and becomes particularly large [47], [48], [49]. In other words, the deep layers are tailored to their original domain at the expense of degraded performance on target domain, hence they cannot be directly transferred to the target domain with limited target supervision. In this paper, we are committed to making the feature distribution between the source domain and the target domain similarly under each hidden layer representations of the MLP, which can be learned by adding an MK-MMD adaptation regularizer across domains.

As shown in Fig. 4, for each hidden layer, we assume that the MK-MMD is defined as Reproducing Kernel Hilbert Space (RKHS) distance between the mean feature embeddings of p and q. Note that, the most important property is that $p = q$ if $d_k^2(p,q) = 0$ [50]. The distance between them which can be calculated by:

$$d_k^2(p,q) \triangleq \left\| \mathbf{E}_p\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2 , \tag{11}$$

where $\phi(\cdot)$ is the nonlinear feature mapping that induces $\mathcal{H}_k$. By using the kernel trick $k(\mathbf{x}^s, \mathbf{x}^t) = \langle \phi(\mathbf{x}^s), \phi(\mathbf{x}^t) \rangle$, we can compute $d_k^2(p,q)$ through the expectation of kernel functions instead of mapping function $\phi(\cdot)$. The equation can be formulated as:

$$\begin{aligned} d_k^2(p,q) = \mathbf{E}_{\mathbf{x}^s\mathbf{x}'^s} k\left(\mathbf{x}^s, \mathbf{x}'^s\right) + \mathbf{E}_{\mathbf{x}^t\mathbf{x}'^t} k\left(\mathbf{x}^t, \mathbf{x}'^t\right) \\ - 2\mathbf{E}_{\mathbf{x}^s\mathbf{x}^t} k\left(\mathbf{x}^s, \mathbf{x}^t\right) , \end{aligned} \tag{12}$$

where $\mathbf{x}^s, \mathbf{x}'^s \overset{iid}{\sim} p$; $\mathbf{x}^t, \mathbf{x}'^t \overset{iid}{\sim} q$. $\mathbf{x}^*$ represent samples from different domain distributions. However, this computation incurs a high complexity (i.e., $O(n^2)$). Therefore, we follow the unbiased estimate of MK-MMD which can be simply computed with linear complexity (i.e., $O(n)$) [47], [50]. The main calculate process is as follows:

$$d_k^2(p,q) = \frac{2}{n_s} \sum_{i=1}^{n_s/2} g_k\left(\mathbf{z}_i\right) . \tag{13}$$

Here, we denote $\mathbf{z}_i \triangleq \left(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t\right)$, and we evaluate kernel function k on each quad-tuple $\mathbf{z}_i$ through:

$$\begin{aligned} g_k\left(\mathbf{z}_i\right) \triangleq k\left(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s\right) + k\left(\mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t\right) \\ - k\left(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^t\right) - k\left(\mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t\right) , \end{aligned} \tag{14}$$

where k is the kernel function which denotes as Gaussian kernel in our paper. Finally, our purpose is to minimize the distance between the source and target domain, which can be represented as:

$$\mathcal{L}_m = \sum_{\ell=1}^{h} d_k^2\left(\boldsymbol{\mathcal{D}}_\ell^s, \boldsymbol{\mathcal{D}}_\ell^t\right), \tag{15}$$

where $\boldsymbol{\mathcal{D}}_\ell^* = \left\{R_{\ell_i}^*\right\}$ is the ℓ-th layer hidden representation for the source and target items, and $d_k^2\left(\boldsymbol{\mathcal{D}}_\ell^s, \boldsymbol{\mathcal{D}}_\ell^t\right)$ is the value of MK-MMD which is evaluated on the ℓ-th hidden layer representation of the MLP.

2) Sentiment Classification: The sentiment classifiers focus on mining aspect-aware semantic information in the contextual sentences. At the same time, it is also used to learn domain-shared features that contribute to aspect-level sentiment classification. As illustrated in the top part of prediction module, we treat the output (i.e., R_h^s) in the last layer of the MLP as the source data representation and feed it to the *softmax* layer for aspect-level sentiment classification. The probability of labelling aspect with sentiment polarity is computed by:

$$p_i^s = softmax(\boldsymbol{W}^s R_h^s + \boldsymbol{b}^s) . \tag{16}$$

where $i \in [1, C]$, C is the number of sentiment categories. p_i^s is the estimated probability for each class.

The goal of the sentiment classification is to minimize the cross-entropy loss for all the labelled data in the source domain. The loss function will train the network parameters to mine aspect-aware semantic features. The equation can be formulated as follows:

$$\mathcal{L}_s = -\frac{1}{n_s} \sum_{j=1}^{n_s} \sum_{i=1}^{C} \left(y_i^s ln p_i^s\right) , \tag{17}$$

where y_i^s denotes the groundtruth and n_s denotes the number of source domain data.

3) Domain Classification: This task simultaneously optimizes the domain invariance to learn domain-shared features and to facilitate knowledge transfer across domains, which makes our model become more universal and transferable. To be specific, we feed all feature representation R_h (i.e., R_h^s and R_h^t) into the $softmax$ layer for domain classification, which the goal is to identify whether a training example originates from the source or target domain. The formulation can be defined as follows:

$$\hat{y}^d = softmax(\boldsymbol{W}^d R_h + \boldsymbol{b}^d) . \tag{18}$$

The traditional training method is to minimize the classification error of the domain classifier so that the classifier can learn domain-specific features and better distinguish the difference between two domains. In this way, the classifier can learn domain-specific features. However, it is contrary to our purpose which the goal is to make the domain classifier learning domain-shared features and cannot discriminate between domains. Thus, we need to maximize the loss function of domain classifier, so that the discrepancy between the source domain and target domain can be minimized. However, this poses one problem to train the whole model: how to train two classifiers jointly when the training purpose contains both maximum (i.e., domain loss) and minimum (i.e., sentiment loss)? To eliminate this problem, we added a gradient reversal operation (i.e., GRL) [30], [32] to reverse the gradient direction in the back propagation of the domain classifier. Through the GRL operation, the EATN can learn domain-shared features by minimizing the loss function of the domain classifier instead of maximizing the loss function of it. The equation can be formulated as follows:

$$G(x) = x, \quad \frac{\partial G(x)}{\partial x} = -\lambda I , \tag{19}$$

where λ is a hyperparameter. Through the above operation, the domain classifier can be trained by minimizing the cross-

TABLE 1: **Statistics of SemEval.2014 and twitter datasets.**

Domains	Training Set Percentage (80%)			
	# Pos.	# Neg.	# Neu.	# Asp-T.
Restaurant	2,800	1,000	800	1,310
Laptop	1,300	970	600	920
Twitter	1,800	1,800	3,000	170

TABLE 2: **Statistics of YelpAspect datasets.**

Domains	Training Set Percentage (80%)			
	# Pos.	# Neg.	# Neu.	# Asp-C.
BeautySpa	30,000	15,000	30,000	38
Hotel	30,000	15,000	30,000	36
Restuarant-1	30,000	15,000	30,000	52

entropy loss for all data from the source domain and the target domain, the domain loss function is defined as:

$$\mathcal{L}_d = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i^d ln\hat{y}_i^d + (1-y_i^d)ln(1-\hat{y}_i^d)\right) , \quad (20)$$

where N is the sum of n_s and n_t. y_i^d, $\hat{y}_i^d$ denote the ground-truth and the prediction domain label for the i-th sample, respectively.

3.6 Training Strategy

Different from the traditional methods, our training process comprises three different parts. Based on the individual learning process defined above (i.e., MK-MMD based MLP, sentiment classification and domain classification), we conduct joint learning for them to optimize the parameters of both tasks and one auxiliary module. In order to avoid overfitting, we add a squared regularization and combine them into an entire objective function:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_d + \beta\mathcal{L}_m + \rho\mathcal{L}_{reg} , \quad (21)$$

where $\mathcal{L}_{reg}$ is l_2 regularization function which aims to add the sum of the squares of the parameters (i.e., parameters in sentiment classifier and parameters in domain classifier) to the loss function to make the model smoother and get better generalization ability. β and ρ are parameters to balance the loss terms. The regularization function is formulated as:

$$\mathcal{L}_{reg} = \|W^s\|_2^2 + \|b^s\|_2^2 + \|W^d\|_2^2 + \|b^d\|_2^2 . \quad (22)$$

The training goal of the joint learning is to minimize the loss function $\mathcal{L}$ with respect to the model parameters. Additionally, all the parameters are optimized by the standard back-propagation algorithm [51].

4 Experiments

4.1 Dataset Preparation

For reliability and authority of the experimental results, we conduct experiments on six real-world datasets with two granularities. The fine-grained datasets are for aspect-term sentiment polarity detection which gathers from SemEval 2014 Task 4 Subtask 2[3] and Twitter.com. And other coarse-grained datasets, i.e., the YelpAspect dataset from [52], are used for aspect-category sentiment polarity detection. The basic statistics are shown in Table 1 and Table 2.

4.1.1 Aspect-term

In this dataset, we aim to determine whether the polarity of aspect-term is positive, negative or neutral. It contain three subset, the first two are reviews from *Restaurants (R)* and *Laptop (L)* and the third one is *Twitter (T)* dataset, which is gathered by previous work [53]. Among them, each aspect with the contextual sentence is labelled by three sentiment polarities, namely positive, negative and neutral. Then, we conduct the cross-domain experiments between every two subsets in the aspect-term dataset, which means that we have six domain adaptation tasks in aspect-term dataset: ***R***→***L***, ***R***→***T***, ***L***→***R***, ***L***→***T***, ***T***→***R***, ***T***→***L***. For example, the notation *"**R**→**L**"* represents the task which transfers from the source domain *"Restaurant"* to the target domain *"Laptop"*.

4.1.2 Aspect-category

The dataset is gathered from [52] and provide a predefined set of aspect-categories (e.g., price, food). The goal is to determine the polarity (positive, negative or neutral) of each aspect category[4]. Note that, the aspect-categories are typically coarser than the aspect terms, and they do not necessarily occur as terms in the given sentence. Specifically, YelpAspect contains three domains: Restaurant (R1), Beautyspa (B), and Hotel (H). The statistics of the YelpAspect dataset are summarized in Table 2. With the same method, we can construct six domain adaptation task in aspect-category dataset: ***B***→***H***, ***B***→***R1***, ***H***→***B***, ***H***→***R1***, ***R1***→***B***, ***R1***→***H***. In the end, we adopt *Classification Accuracy* as the main metric to evaluate the performances of the classifiers which are widely used in previous works [10], [14].

4.2 Hyperparameters Setups

In the experiments, we randomly split each dataset into training (80%), validation (10%), and test (10%) sets. The parameters for all benchmark methods are initialized as in the corresponding papers, and are carefully tuned to achieve optimal performances. The learning rate for all models is tuned amongst [2e-5, 5e-5 and 1e-3] and the batch size is tested in [16, 32, 64]. All words of sentence and aspect are embedded in 300-dimension vectors. For our EATN model [5], the batch size, head number of multi-head attention, max-sequence length and the embedding size are set to 32, 12, 100 and 1024 respectively. The hidden unit sizes in 3-layer MLP are [128, 32, 8]. The parameter λ, β and ρ have been carefully adjusted, and final values are set to 1, 0.8 and 0.002 respectively. All weight matrices are randomly initialized by a uniform distribution $\mathcal{U}(-0.01, 0.01)$, and all bias matrices are initialized to zeros. Besides, we set the coefficient of l_2 normalization, the learning rate and the dropout rate as 10^{-4}, 10^{-4} and 0.1. We follow the standard procedures of

3. The detailed task and dataset introduction can be found in the following public link: http://alt.qcri.org/semeval2014/task4/.

4. More dataset description of the aspect-category can be find in the following link: https://alt.qcri.org/semeval2014/task4/index.php.

5. Our code is available via https://github.com/1146976048qq/EATN.

TABLE 3: **Experimental performance (accuracy) about Sem.2014 and Twitter datasets.**

Benchmarks	Res. →		Lap. →		Twitter. →	
	R→L	R→T	L→R	L→T	T→R	T→L
LSTM	0.6227*(-13.2%)*	0.5659*(-18.9%)*	0.5626*(-11.6%)*	0.5236*(-15.5%)*	0.5588*(-11.8%)*	0.5068*(-17.0%)*
TD-LSTM	0.6301*(-12.9%)*	0.5687*(-19.0%)*	0.5769*(-10.4%)*	0.5285*(-15.4%)*	0.5669*(-10.6%)*	0.5190*(-15.4%)*
ATAE	0.6023*(-15.3%)*	0.5839*(-17.1%)*	0.5652*(-12.2%)*	0.5326*(-16.1%)*	0.5727*(-9.8%)*	0.5243*(-14.8%)*
IAN	0.6280*(-14.1%)*	0.5814*(-18.7%)*	0.6371*(-8.0%)*	0.5664*(-15.0%)*	0.6031*(-8.1%)*	0.5444*(-15.9%)*
MemNet	0.6361*(-14.5%)*	0.5829*(-19.8%)*	0.6217*(-9.6%)*	0.6033*(-11.4%)*	0.6166*(-9.3%)*	0.5512*(-16.8%)*
AOA	0.6529*(-10.7%)*	0.5642*(-19.5%)*	0.6508*(-8.7%)*	0.6143*(-12.3%)*	0.6233*(-6.3%)*	0.5437*(-14.3%)*
MGNet	0.6541*(-11.1%)*	0.5920*(-17.3%)*	0.6631*(-8.3%)*	0.6210*(-12.5%)*	0.6267*(-7.1%)*	0.5489*(-15.0%)*
TNet	0.6642*(-12.0%)*	0.6099*(-17.4%)*	0.6711*(-8.3%)*	0.6364*(-11.7%)*	0.6360*(-7.5%)*	0.5880*(-13.8%)*
BERT-PT	0.6791*(-12.3%)*	0.6183*(-18.3%)*	0.6820*(-9.7%)*	0.6434*(-13.6%)*	0.6679*(-7.3%)*	0.6001*(-14.2%)*
SFA	0.6547	0.5924	0.6693	0.6101	0.6312	0.5832
HATN	0.6762	0.6121	0.6887	0.6383	0.6638	0.6044
IATN	0.6833	0.6169	0.6953	0.6490	0.6706	0.6101
EATN	**0.7087**	**0.6323**	**0.7162**	**0.6572**	**0.6822**	**0.6197**

MMD-based method [47], [50] and use Gaussian kernels as the kernel function. Finally, we optimize all the models with Adam optimizer [54].

4.3 Benchmark Methods

In order to comprehensively evaluate the performance of our model, we borrow several non-transfer approaches from aspect-level sentiment classification as well as some transfer methods from cross-domain sentiment classification for comparison. The methods are listed below.

- **LSTM** [16] utilizes neural network to learn the hidden states and obtain the averaged vector through mean pooling to predict the sentiment polarity.
- **TD-LSTM** [17] employs two LSTMs to model the left and right contexts of the target separately, then performs predictions based on concatenated context.
- **ATAE-LSTM** [10] is a simple LSTM model which learns attention embeddings and combines them with the hidden states to predict the polarity.
- **IAN** [12] interactively learns the coarse-grained attention between the aspects and sentences, then concatenate vectors for the final sentiment prediction.
- **MemNet** [9] adopts multi-hop attention on the word embeddings, learns the attention weights on context word vectors with respect to averaged query vector.
- **AOA** [19] models aspects and sentences in a joint way and explicitly captures the interaction between aspects and context sentences.
- **MGNet** [13] proposes a novel fine-grained attention method, which can capture the word-level interaction between the aspects and the contexts.
- **TNet** [14] employs a CNN layer to extract salient features and propose a component to generate specific representations of words in the sentence.
- **BERT-PT** [20] is a language model with a joint post-training approach which has been specially designed for aspect-level sentiment analysis task.

As mentioned in related work, there are some domain adaptation strategies in other relative tasks, such as cross-domain sentiment analysis. Thus, we choose some representative methods, which shows significant improvement in recent years, as benchmarks to further verify the ability of aspect-aware knowledge representation of EATN.

- **SFA** [24] interactively learns the coarse-grained attention between aspects and the contexts, then concatenate them to predict the final result.
- **HATN** [32] models the aspects and the sentences in a joint way and explicitly captures the interaction between them in both word and sentence level.
- **IATN** [34] proposes a novel fine-grained interactive attention method, which can capture the word-level feature interaction between the aspects and the contextual sentiment words.

Our benchmarks have a comprehensive coverage of the related models. In the experiments, all the benchmark methods are implemented by python (Pytorch) and are trained on a Linux server with two 2.20 GHz Intel Xeon E5-2650 CPUs and four Tesla K80 GPUs.

4.4 Results and Analysis

Considering the transferability and universality of aspect-level sentiment classifier is a relatively novel task. In order to better demonstrate the comprehensive performance of our proposed method, in this section, we cover multiple experimental results and the detailed description is as follows.

1) Overall performance. Table 3 and Table 4 show the main performance, i.e., classification accuracy, on each of the twelve tasks on different datasets. Table 5 shows some overall effects of other metrics of those methods.

For aspect-term performance shown in Table 3, BERT-PT has the best performance over all the non-transfer methods, i.e., aspect-level sentiment classifiers. This is consistent with previous work [20], [21], which indicates that the pre-trained model is more powerful in feature extraction and semantic representation. From the results, we observe that some specially designed attention-methods (e.g., TNet, MGNet, AOA) outperform memory-based methods (e.g., MemNet) and RNN-based methods (e.g., LSTM, ATAE). However, it's also impressive to observe that the performances of all

TABLE 4: **Experimental performance (accuracy) about YelpAspect datasets.**

Benchmarks	**Bea. →**		**Hotel. →**		**Res1. →**	
	B→H	B→R1	H→B	H→R1	R1→B	R1→H
LSTM	0.6695*(-5.65%)*	0.6753*(-5.07%)*	0.6992*(-3.20%)*	0.7057*(-2.55%)*	0.6983*(-2.12%)*	0.6765*(-4.29%)*
TD-LSTM	0.6766*(-4.73%)*	0.6709*(-5.30%)*	0.6984*(-3.71%)*	0.7095*(-2.59%)*	0.7078*(-1.46%)*	0.6792*(-4.32%)*
ATAE	0.6795*(-5.37%)*	0.6776*(-5.56%)*	0.7039*(-4.19%)*	0.7105*(-3.52%)*	0.7083*(-2.26%)*	0.6870*(-4.38%)*
IAN	0.6805*(-2.07%)*	0.6626*(-3.84%)*	0.6963*(-4.06%)*	0.6933*(-4.36%)*	0.7112*(-2.14%)*	0.6828*(-4.96%)*
MemNet	0.6923*(-4.03%)*	0.6790*(-5.35%)*	0.6907*(-5.11%)*	0.7095*(-3.23%)*	0.7208*(-1.38%)*	0.6874*(-4.71%)*
AOA	0.6829*(-4.07%)*	0.6642*(-3.25%)*	0.6850*(-3.70%)*	0.7092*(-2.63%)*	0.7136*(-1.63%)*	0.6837*(-4.30%)*
MGNet	0.6907*(-3.59%)*	0.6903*(-3.62%)*	0.6986*(-4.63%)*	0.7192*(-2.58%)*	0.7216*(-0.41%)*	0.6916*(-3.39%)*
TNet	0.7007*(-4.06%)*	0.7110*(-3.02%)*	0.7211*(-3.66%)*	0.7251*(-3.27%)*	0.7291*(-1.73%)*	0.6985*(-4.78%)*
BERT-PT	0.7172*(-3.85%)*	0.7194*(-3.61%)*	0.7275*(-3.12%)*	0.7312*(-2.77%)*	0.7411*(-1.03%)*	0.7034*(-4.77%)*
SFA	0.6860	0.6722	0.6817	0.7015	0.7046	0.6824
HATN	0.6911	0.7138	0.7174	0.7208	0.7331	0.7088
IATN	0.7074	0.7209	0.7227	0.7264	0.7425	0.7101
EATN	**0.7218**	**0.7346**	**0.7410**	**0.7488**	**0.7526**	**0.7155**

TABLE 5: **Multiple results (i.e., average of all tasks) of different datasets.**

Evaluation	**Benchmark methods.**												
	LSTM	td-LSTM	ATAE	IAN	MNet	AOA	mgNet	TNet	BERT-PT	SFA	HATN	IATN	**EATN**
SemEval.2014 and Twitter.													
Accuracy	0.5567	0.5650	0.5635	0.5934	0.6019	0.6082	0.6176	0.6343	0.6484	0.6234	0.6472	0.6542	**0.6692**
F1-Value	0.4680	0.4778	0.4766	0.4913	0.4962	0.4974	0.5011	0.5079	0.5208	0.5004	0.5313	0.5379	**0.5416**
YelpAspect.													
Accuracy	0.6874	0.6904	0.6945	0.6879	0.6866	0.6897	0.7020	0.7142	0.7233	0.6881	0.7140	0.7217	**0.7356**
F1-Value	0.6076	0.6083	0.6112	0.5980	0.5922	0.6012	0.6167	0.6234	0.6424	0.6035	0.6259	0.6334	**0.6497**

nine non-transfer benchmark methods have a certain degree decline when performing it in new domains. For example, in Table 3 **R**→**L** task, LSTM method has achieved the performance of 75.57% in predicting the sentiment polarity in the restaurant domain (i.e., train and test in the same domain), while it reduces to 62.27% when adapting to the laptop domain (i.e., train in the restaurant domain and test in the laptop domain) for aspect-level sentiment classification. The content in parentheses (i.e., 13.2%) indicates the percentage of decline which caused by the domain discrepancy as assumed in Fig. 1. The phenomenons above can be explained as that these classifiers mainly concentrate on mining the aspect-level semantic information in a single domain instead of constituting more generic features that can be more easily adapted to new domains.

For the fairness of the experiments, we also adopt some domain adaption methods to verify the effectiveness of our proposed model. Among these benchmark methods, the rule-based method, i.e., SFA, performs worst since it adopts a simple distance function for fitting training samples. In contrast, HATN and IATN utilize the deep neural network for modeling the semantics of text, thus give better results than SFA. However, IATN performs worse than some aspect-level sentiment classifiers on several tasks such as **R**→**T**, **B**→**H** and **H**→**B**. A possible reason is that those domain adaptation methods usually focus on sentence-level sentiment analyzing, which is hard to extract aspect-aware semantics. Finally, we compare EATN with all benchmark methods. As shown in Table 3, it is clear to see that EATN is consistently better than all the benchmarks by a large margin at each task. Specifically, compared to the best aspect-level sentiment classifier (i.e., BERT-PT), the accuracy improvement of EATN on different tasks is between 1.5% and 3.5%. In addition, compared with the best adaptive model (IATN), EATN has also achieved a maximum improvement of 2.54% on **R**→**L** tasks. This suggests that EATN not only perform better in domain adaptation scenario but also gain powerful ability in aspect-aware semantic representation.

For the aspect-category performance shown in Table 4, we can observe the same trend of performance as the aspect-term tasks. To be specific, we can get several observations: 1) All the non-transfer methods from aspect-level sentiment classification have a certain decline, and the BERT-PT gains the best performance among them because of its powerful representation ability. 2) Most models with transfer capabilities are not as good as BERT-PT, the possible reason is that they cannot model aspect information. 3) The results also show that the proposed EATN model performs better than the benchmark methods in six subtasks when transferring from a source domain to a target domain. Note that, the aspect-category dataset is more generic than the aspect-term dataset because it is more coarse as we described in section 4.1.2. Thus, all the benchmark methods have a higher classification accuracy and the declined percent is a little lower than the performances in aspect-term tasks.

Moreover, from the average results in Table 5, we can get further observation that our proposed EATN performs better than other methods, i.e., aspect-level sentiment classifiers

TABLE 6: **Ablation performance (accuracy) of the EATN.**

Tasks / Methods	Restaurant.		Laptop.		Twitter.	
	R→L	R→T	L→R	L→T	T→R	T→L
(1) EATN	**0.7087**	**0.6323**	**0.7162**	**0.6572**	**0.6822**	**0.6197**
(2) w/ lstm	0.6960(-1.27%)	0.6210(-1.13%)	0.7024(-1.38%)	0.6514(-0.58%)	0.6799(-0.23%)	0.6162(-0.35%)
(3) w/o domain	0.6817(-2.70%)	0.6194(-1.29%)	0.6866(-2.96%)	0.6481(-0.91%)	0.6717(-1.05%)	0.6113(-0.85%)
(4) w/o mk-mmd	0.6875(-2.12%)	0.6223(-1.01%)	0.6915(-2.47%)	0.6497(-0.75%)	0.6710(-1.12%)	0.6092(-1.04%)

and domain adaptation methods, on average classification accuracy, reaching to 66.92% and 73.56% in SemEval.2014 dataset and YelpAspect dataset respectively. To be specific, the average accuracy of EATN is 1.50% higher than the best benchmark 65.42% (i.e., IATN) in aspect-term dataset and 1.33% higher than the best benchmark (i.e., BERT-PT) in aspect-category dataset. For other metrics, i.e., F1-value, the observations are similar. EATN consistently outperforms the state-of-the-art benchmark method by 2.22% in the aspect-term dataset and 3.33% in the aspect-category dataset. In conclusion, the results in Table 3, Table 4 and Table 5 indicate that our proposed method outperforms benchmark methods on diverse transfer tasks. And EATN is more effective and accurate for aspect-level sentiment analysis, more generic and transferable in domain adaptation scenario.

2) Ablation Study. In order to investigate the relevance of each component, we present multiple ablation studies of our EATN model on the more popular benchmark datasets, i.e., SemEval.2014 and Twitter. In what follows, we describe the variants of EATN approach and examine how each of them affects the final prediction performance:

- ***EATN*** $_{w/\ lstm}$: the variant utilizes LSTM instead of the aspect-oriented multi-head attention mechanism to learn the hidden representation.
- ***EATN*** $_{w/o\ domain}$: is a variant method of the EATN which removes the domain classifier (i.e., without using the domain classification task).
- ***EATN*** $_{w/o\ mk-mmd}$: the variant removes the MK-MMD and directly binds the output of MLP with the prediction tasks.

The results are shown in Table 6. Generally, all three factors contribute a certain degree of the improvement to EATN. Specifically, the performance of EATN significantly decreases when replacing the aspect-oriented multi-head attention with LSTM as shown in Table 6 (2). It verifies that the aspect-oriented multi-head attention has a stronger ability to learn the relationships between the aspects and its contextual sentiment words than RNN-based methods (e.g., LSTM), which is consistent with our previous assumption in Fig. 2. Besides, we wonder whether the domain classification task or the MK-MMD module is enough for improving the final performance. Thus, we remove them separately to verify it. The results in Table 6 (3)-(4) show that there is a significant drop in the performance of ***EATN*** $_{w/o\ domain}$ and ***EATN*** $_{w/o\ mk-mmd}$. Further, the results also suggest that both parts are extremely significant for the final classification, which means that domain classifier can learn domain-invariant features[6] between domains comprehensively and MK-MMD can align features across domains. Overall, both of them are indispensable for EATN to achieve excellent performance in cross-domain aspect-level sentiment analysis.

TABLE 7: **Runtime of BERT-PT, IATN and EATN. Specifically, "S" represents the training time (seconds) for a single epoch, "E" denotes the number of epochs to converge, and "T" is the total training time (minutes).**

Methods	**R→L task.**			**B→H task.**		
	S	**E**	**T**	**S**	**E**	**T**
(1) BERT-PT	86s	28	2428s	21m	23	568m
(3) IATN	15s	30	464s	6m	28	176m
(4) EATN	14s	24	359s	5m	25	147m

4.5 Efficient Analysis

To show the EATN is efficient not only in learning transferable knowledge but also in operating efficiency, we further investigated the training process of best benchmark models BERT-PT, IATN as well as our EATN model.

1) Loss Convergence. We investigated the training process of the neural models BERT-PT, IATN and EATN model. The losses of best benchmarks on the training set throughout the training process on two tasks are shown in Fig. 5. To be specific, the left part shows the training loss of the "**R→L**" task and the right part shows the loss of "**B→H**" task. The result in Fig. 5 shows similar observations as before, which demonstrates the effectiveness of the proposed EATN framework again. Clearly, from the results, the EATN model converges faster than the other two models, and also achieve a lower loss on both tasks. Thus, we can get the conclusion that the EATN model has superior ability and efficiency to extract domain-shared features across domains.

2) Runtime Comparison. We also compared overall runtime of three methods. From results that shown in Table 7, we can first observe that the training time of a single epoch in the EATN model perform better than the others. Second, compare with other models, the total training time of the EATN model is less. In particular, in aspect-category dataset, the EATN only needs 147 minutes to achieve the optimal performance, while IATN and BERT-PT need about 176 and

6. The domain adversarial loss $\mathcal{L}_d$ (i.e., equation 20) can enforce EATN to learn domain-shared features which the domain classifier cannot discriminate across domains.

TABLE 8: **Example study of three items from laptop and restaurant.**

	Cases are shown as below	Models and Prediction		
Example One.		BERT-PT	IATN	EATN
Aspects:	"the cord" (Ground-truth: **Neutral**)	Pos.	Pos.	Neu.
Sentence:	I charge it at night and skip taking "$" with me because of the good battery life.	(×)	(×)	(✔)
Example Two.				
Aspects:	"patches" (Ground-truth: **Negtive**)	Neg.	Pos.	Neg.
Sentence:	2nd Best computer in the world only one way this computer might become the best is that it needs to upgrade "$" to make less easier for people to hack into.	(✔)	(×)	(✔)
Example Three.				
Aspects:	"meal" (Ground-truth: **Postive**)	Neg.	Pos.	Pos.
Sentence:	I just wonder how you can have such a delicious "$" for such little money.	(×)	(✔)	(✔)

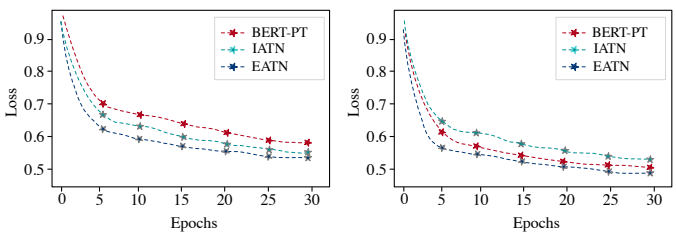


Fig. 5: **The loss curves of BERT-PT, IATN best benchmarks and our EATN on the asepct-term dataset (i.e., R→L task shown in the left) and the aspect-category dataset (i.e., B→H task shown in the right).**

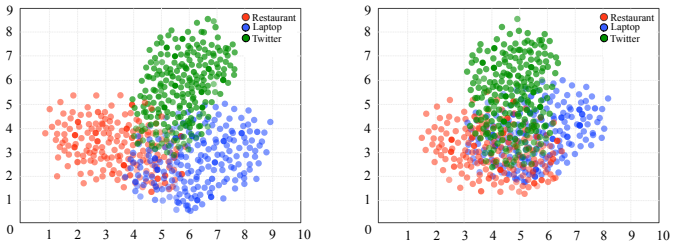


Fig. 6: **Feature distribution visualization of three domains, i.e., *Laptop*, *Restaurant*, and *Twitter*. The images on the left is data distribution randomly sampled after the original data embedding, while the right image is feature distribution (i.e., output of MLP) sampled after domain adaptation network. In each group (e.g., every two colors), the original and domain-adapted distribution images are shown from left to right, and their feature distributions become more similar.**

568 minutes, respectively. The possible reason is that the IATN (LSTM-based) can not be paralleled during training process, and the BERT-PT is a little difficult to fine-turn, relatively. In summary, the observations above show significant advantages of our EATN model in training efficiency.

4.6 Case Study

Table 8 displays three examples and the auxiliary information provided by BERT-PT, IATN and EATN model. Recall that the EATN model can encode the aspect-aware semantic information to gain a more efficient aspect-level sentiment classifier as well as gain remarkable transferability among domains. To better demonstrate this viewpoint, we randomly sample three aspect-sentence pairs (examples) from two datasets (i.e., Laptop and Restaurant). Among them, the first example is a review from the laptop domain. We can observe that the aspect (i.e., the cord) has no associated sentiment word and the sentiment polarity should be neutral. However, in models BERT-PT and IATN, it was misclassified as positive because of the unrelated sentiment word "good", which indicates that IATN concentrates more on the whole sentences' sentiment polarity instead of aspects' sentiment polarity. Example two which the whole sentiment polarity is positive can also demonstrate this viewpoint. Moreover, through Example One and Example Three, we find that the performance of BERT-PT is not very well when there are multiple sentiment words in one sentence, which indicate that BERT-PT has insufficient ability to extract aspects-aware information. Overall, the results demonstrate that the EATN facilitates the performance through multiple modules and is superior in aspect-level sentiment prediction.

4.7 Visualization of Distribution

From the above experiments, it is apparent that the EATN model can effectively gain outstanding performance than the compared methods. To intuitively show the transferability of the proposed method, in this subsection, we visualize the feature distributions of the bottleneck layer from three datasets (i.e., *Laptop*, *Restaurant* and *Twitter*) learned by the original embedding and our transfer network respectively in Fig. 6. From the original word representation (i.e., the left subfigure) to final feature representation (i.e., the right subfigure), the feature distributions between the source domain and target domains become more indistinguishable through Domain Adaptation Module. Moreover, the feature distributions between Laptop domain (i.e., blue color) and Restaurant domain (i.e., red color) is more similar than Laptop domain and Twitter domain (i.e., green color) since they come from the same general domain, i.e., Amazon.com, and focus on the reviews toward products. We also believe this is the reason why the transfer task $\boldsymbol{L}\rightarrow\boldsymbol{R}$ performs better than the transfer task $\boldsymbol{L}\rightarrow\boldsymbol{T}$. In summary, the visualization result indicates that EATN is able to match the complex structures of the source and the target data distributions, thus learning more transferable features for domain adaptation.

5 Conclusions

In this paper, we presented an Efficient Adaptive Transfer Network (EATN), a novel domain adaptation approach for aspect-level sentiment analysis. Unlike previous methods that only match the feature extracted from the source labeled domain, the proposed approach further exploits the inherent semantic relationship across domains by considering the transferable knowledge. Specifically, we designed a Domain Adaptation Module to ensure that EATN can learn domain-invariant and semantic features. Then, we devised a novel MLP module to further enhance the transferability of features in the deep neural networks. Finally, we designed an aspect-oriented multi-head attention to extracting the direct semantic relationships between the aspects and the contextual words. Extensive experiments on six real-world datasets demonstrated the effectiveness of our model. We hope this work can help boost more researches for aspect-level sentiment analysis in the domain adaptation scenario. In the future, we will try to integrate into more domains for semantic domain adaptation.

6 Acknowledgments

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101), and the National Natural Science Foundation of China (Grants No. 61922073, 61727809 and U20A20229). Qi Liu acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299).

References

[1] D. Bertero, F. B. Siddique, C.-S. Wu, Y. Wan, R. H. Y. Chan, and P. Fung, "Real-time speech emotion and sentiment recognition for interactive dialogue systems," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1042–1047.

[2] X. Kong, B. Li, G. Neubig, E. Hovy, and Y. Yang, "An adversarial approach to high-quality, sentiment-controlled neural dialogue generation," *arXiv preprint arXiv:1901.07129*, 2019.

[3] J. Wang, C. Sun, S. Li, X. Liu, L. Si, M. Zhang, and G. Zhou, "Aspect sentiment classification towards question-answering with reinforced bidirectional attention network," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3548–3557.

[4] L. Huang, Z. Dou, Y. Hu, and R. Huang, "Textual analysis for online reviews: A polymerization topic sentiment model," *IEEE Access*, 2019.

[5] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd international conference on Knowledge capture*. ACM, 2003, pp. 70–77.

[6] X. Ding and B. Liu, "The utility of linguistic rules in opinion mining," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 811–812.

[7] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 151–160.

[8] T. H. Nguyen and K. Shirai, "Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2509–2514.

[9] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," *arXiv preprint arXiv:1605.08900*, 2016.

[10] Y. Wang, M. Huang, L. Zhao *et al.*, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[11] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang, "Aspect-level sentiment classification with heat (hierarchical attention) network," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 97–106.

[12] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," *arXiv preprint arXiv:1709.00893*, 2017.

[13] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3433–3442.

[14] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," *arXiv preprint arXiv:1805.01086*, 2018.

[15] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "Nrc-canada-2014: Detecting aspects and sentiment in customer reviews," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 437–442.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," *arXiv preprint arXiv:1512.01100*, 2015.

[18] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 452–461.

[19] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 2018, pp. 197–206.

[20] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2015.

[23] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2006, pp. 120–128.

[24] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 751–760.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 513–520.

[26] M. Hu, Y. Wu, S. Zhao, H. Guo, R. Cheng, and Z. Su, "Domain-invariant feature distillation for cross-domain sentiment classification," *arXiv preprint arXiv:1908.09122*, 2019.

[27] Y. Wu and Y. Guo, "Dual adversarial co-learning for multi-domain text classification," *arXiv preprint arXiv:1909.08203*, 2019.

[28] M. Long, J. Wang, G. Ding, S. J. Pan, and S. Y. Philip, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076–1089, 2013.

[29] J. Yu and J. Jiang, "Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 236–246.

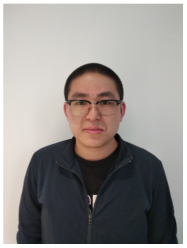
[30] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[31] Z. Li, Y. Zhang, Y. Wei, Y. Wu, and Q. Yang, "End-to-end adversarial memory network for cross-domain sentiment classification." in *IJCAI*, 2017, pp. 2237–2243.

[32] Z. Li, Y. Wei, Y. Zhang, and Q. Yang, "Hierarchical attention transfer network for cross-domain sentiment classification," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[33] M. Yang, W. Yin, Q. Qu, W. Tu, Y. Shen, and X. Chen, "Neural attentive network for cross-domain aspect-level sentiment classification," *IEEE Transactions on Affective Computing*, 2019.
[34] K. Zhang, H. Zhang, Q. Liu, H. Zhao, H. Zhu, and E. Chen, "Interactive attention transfer network for cross-domain sentiment classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5773–5780.
[35] T. Domhan, "How much attention do you need? a granular analysis of neural machine translation architectures," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1799–1808.
[36] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 287–297.
[37] E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum, "Linguistically-informed self-attention for semantic role labeling," *arXiv preprint arXiv:1804.08199*, 2018.
[38] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, "Deep semantic role labeling with self-attention," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[39] C. Tao, S. Gao, M. Shang, W. Wu, D. Zhao, and R. Yan, "Get the point of my utterance! learning towards effective responses with multi-head attention mechanism." in *IJCAI*, 2018, pp. 4418–4424.
[40] K.-M. Kim, S.-H. Choi, J.-H. Kim, and B.-T. Zhang, "Multimodal dual attention memory for video story question answering," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 673–688.
[41] W. Du, B. Li, M. Yang, Q. Qu, and Y. Shen, "A multi-task learning approach for answer selection: A study and a chinese law dataset," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9935–9936.
[42] X. Zhang, S. Li, L. Sha, and H. Wang, "Attentive interactive neural networks for answer selection in community question answering." in *AAAI*, 2017, pp. 3525–3531.
[43] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
[44] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," *arXiv preprint arXiv:1609.06038*, 2016.
[45] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," *arXiv preprint arXiv:1512.08422*, 2015.
[46] J. Weeds, D. Clarke, J. Reffin, D. Weir, and B. Keller, "Learning to distinguish hypernyms and co-hyponyms," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2249–2259. [Online]. Available: https://www.aclweb.org/anthology/C14-1212
[47] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.
[48] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
[49] M. Long, J. Wang, Y. Cao, J. Sun, and S. Y. Philip, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2027–2040, 2016.
[50] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Advances in neural information processing systems*, 2012, pp. 1205–1213.
[51] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
[52] Z. Li, Y. Wei, Y. Zhang, X. Zhang, X. Li, and Q. Yang, "Exploiting coarse-to-fine task transfer for aspect-level sentiment classification," *arXiv preprint arXiv:1811.10999*, 2018.
[53] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, vol. 2, 2014, pp. 49–54.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Kai Zhang is a Ph.D. candidate at School of Data Science, University of Science and Technology of China (USTC) His general area of research is data mining and natural language processing focusing on sentiment classification, aspect-level sentiment classification and transfer learning. He has won the CCML 2019 Best Student Paper Award. He has published papers in referred conference proceedings, such as AAAI, WSDM and CCML.

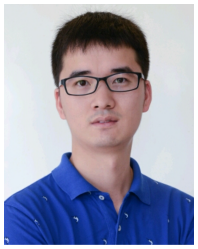
Qi Liu is a professor at University of Science and Technology of China (USTC). He received the Ph.D. degree in the School of Computer Science and Technology of USTC. His general area of research is data mining and knowledge discovery. He has published prolifically in referred journals and conference proceedings, e.g., TKDE, TOIS, TKDD, TIST, KDD, IJCAI, AAAI, ICDM, SDM and CIKM. He has served regularly in the program committees of a number of conferences, and is a reviewer for the leading academic journals in his fields. He is a member of ACM and IEEE. Dr. Liu is the recipient of the KDD 2018 Best Student Paper Award (Research) and the ICDM 2011 Best Research Paper Award. He is supported by the Young Elite Scientist Sponsorship Program of CAST and the Youth Innovation Promotion Association of CAS.



Hao Qian received the Ph.D. degree from Princeton University. His work focuses on the natural language processing and recommender system, in which he is interested in extracting information from the vast unsupervised text data and optimizing the personalized recommender system. He joined Ant Financial Services Group to work on the online learning personalized recommender system.

Biao Xiang received the Ph.D. degree in Computer Science from University of Science and Technology of China (USTC). He is an algorithm expert in Alibaba Group. His research interest include social network, web search and recommendation system. He has published prolifically in referred journals and conference proceedings, such as TKDD, SIGIR, AAAI, CIKM, Recsys.

Qing Cui received the Ph.D. degree in Computational Mathematics from Tsinghua University in 2015. He is an algorithm expert in Ant Financial Services Group, Hangzhou, China. His research interests include machine learning, recommender system, computational advertising and interpretable machine learning. He has published prolifically in referred conference proceedings, such as TKDD, SIGIR, AAAI, CIKM, WSDM.



Jun Zhou is currently a Senior Staff Engineer at Ant Financial. His research mainly focuses on machine learning and data mining. He has participated in the development of several distributed systems and machine learning platforms in Alibaba and Ant Financial, such as Apsaras (Distributed Operating System), MaxCompute (Big Data Platform), and KunPeng (Parameter Server). He is a member of IEEE. He has published more than 40 papers in top-tier machine learning and data mining conferences, including VLDB, WWW, SIGIR, NeurIPS, AAAI, IJCAI, and KDD.

Enhong Chen is a professor and vice dean of the School of Computer Science at University of Science and Technology of China (USTC). He received the Ph.D. degree from USTC. His general area of research includes data mining and machine learning, social network analysis and recommender systems. He has published more than 100 papers in refereed conferences and journals, including IEEE Trans. KDE, IEEE Trans. MC, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, SDM. He received the Best Application Paper Award on KDD-2008, the Best Student Paper Award on KDD-2018 (Research), the Best Research Paper Award on ICDM2011 and Best of SDM-2015. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. He is a senior member of the IEEE.