Bridge-Based Active Domain Adaptation for Aspect Term Extraction

Zhuang Chen, Tieyun Qian*
School of Computer Science, Wuhan University, China
{zhchen18, qty}@whu.edu.cn

Abstract

As a fine-grained task, the annotation cost of aspect term extraction is extremely high. Recent attempts alleviate this issue using domain adaptation that transfers common knowledge across domains. Since most aspect terms are domain-specific, they cannot be transferred directly. Existing methods solve this problem by associating aspect terms with pivot words (we call this *passive domain adaptation* because the transfer of aspect terms relies on the links to pivots). However, all these methods need either manually labeled pivot words or expensive computing resources to build associations. In this paper, we propose a novel *active domain adaptation* method. Our goal is to transfer aspect terms by actively supplementing transferable knowledge. To this end, we construct syntactic bridges by *recognizing syntactic roles as pivots instead of as links to pivots*. We also build semantic bridges by *retrieving transferable semantic prototypes*. Extensive experiments show that our method significantly outperforms previous approaches.

1 Introduction

Aspect term extraction (ATE) is a fundamental task in aspect-based sentiment analysis. Given a review sentence "*The pizza here is also absolutely delicious.*", ATE aims to extract the term *pizza*. Recent studies define ATE as a sequence tagging task and propose supervised taggers (Wang et al., 2017; Xu et al., 2018). However, due to the high cost of token-level annotation, the lack of labeled data becomes the main obstacle (Chen and Qian, 2019).

To alleviate the data deficiency issue, unsupervised domain adaptation is proposed to transfer knowledge from the labeled *source* domain to the unlabeled *target* domain. Since ATE is a token-level task, it is natural to conduct token-level domain adaptation. Then a problem arises: many aspect terms are domain-specific and cannot be transferred directly. We present the proportion of source aspect terms that also appear in target test data in Figure 1. As can be seen, in distant transfer pairs like $\mathbb{R}\rightarrow\mathbb{L}$, only less than 10% of source aspect terms have appeared in target data. Even in a close pair $\mathbb{L}\rightarrow\mathbb{D}$, the proportion is no more than 40%. In other words, there is a wide discrepancy between the data from different domains, and many aspect terms have to be transferred under the guidance of proper references.

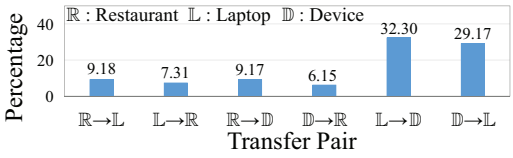


Figure 1: The proportion of source aspect terms that appear in target data. $\mathbb{R}$ (Restaurant), $\mathbb{L}$ (Laptop), and $\mathbb{D}$ (Device) are three datasets from different domains.

To solve this problem, previous studies try to associate aspect terms with specific pivot words[1]. We name these methods passive domain adaptation because the transfer of aspect terms is dependent on their links to the pivots. There are two types of methods along this line. (1) **Opinion terms as pivots**. Since aspect and opinion terms usually appear in pairs, it is straightforward to extract aspect terms with the indication from opinion terms. Early studies (Li et al., 2012; Ding et al., 2017) use common opinion seeds (e.g., *good*, *fancy*) and pre-defined rules (e.g., *good→amod→NN*) to extract aspect terms across domains. However, it is hard to collect a complete set of seeds or define high-quality rules, and thus these methods often produce inferior performance. Several studies (Wang and Pan, 2018, 2019b) manually annotate all opinion terms in reviews and design neural models to capture aspect-opinion relations via multi-task learning. While

*Corresponding author.

[1]Pivot words are words which behave in the same way for discriminative learning in both domains (Blitzer et al., 2006).

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 317–327
August 1–6, 2021.

getting improvements, these methods induce additional annotation costs. (2) **Context terms as pivots**. Since pre-trained language models (PLMs) like BERT represent words w.r.t their contexts, recent studies (Xu et al., 2019; Gong et al., 2020) leverage PLMs to transfer aspect terms with common context terms[2]. However, not all context terms qualify as pivots (e.g., *eat*). In addition, PLMs like BERT build word associations mainly based on semantic similarity in co-occurring contexts. For an aspect term like *pizza*, BERT tends to link it to *hamburger* via a flow like *pizza*→*eat*→*hamburger*. Consequently, it is hard for these methods to identify *keyboard* in the target domain based on the labeled term *pizza* in the source domain.

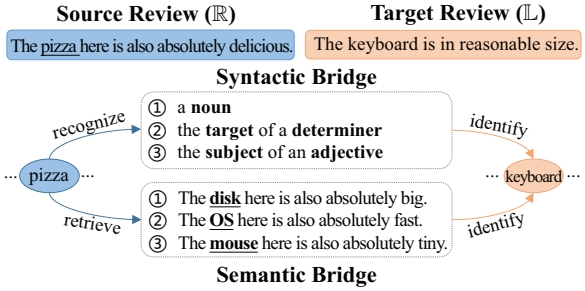


Figure 2: Illustration of syntactic and semantic bridges.

In this paper, we propose a novel active domain adaptation method. Concretely, we construct two types of bridges for all words, which can help transfer aspect terms across domains. An example in Figure 2 shows how to identify the unseen target term *keyboard* based on the source term *pizza*. (1) The **syntactic bridge** aims to recognize transferable syntactic roles for the words across domains. Though *pizza* and *keyboard* have almost no semantic relatedness, they often play a similar role in parse trees. In view of this, we treat the involved syntactic roles (including POS tag and dependency relations) of a certain word as its syntactic bridge. Previous studies also utilize dependency information. However, we differ our method from existing ones in that we do not use dependency relations to associate pivot words with aspect terms. Instead, we treat syntactic roles themselves as pivot features and do not need any manually annotated pivot words. (2) The **semantic bridge** moves one step further by retrieving transferable prototypes. Intuitively, if we correlate *pizza* with some prototype target terms like {*disk*, *OS*, *mouse*}, the domain discrepancy between the training and testing reviews can be largely reduced. Hence we regard the prototypes of a certain word as its semantic bridge and design a syntax-enhanced similarity metric to retrieve them. Compared with previous opinion and context term-based methods, building a semantic bridge directly links aspect terms across domains and only requires unlabeled source and target data.

Based on the syntactic/semantic bridges, we then develop an end-to-end tagger to fuse reviews with these transferable bridges. We conduct extensive experiments on three datasets. The results show that our method achieves a new state-of-the-art performance with a low computational cost.

2 Related Work

Aspect Term Extraction Early researches for ATE mainly involve pre-defined rules (Hu and Liu, 2004; Popescu and Etzioni, 2005; Wu et al., 2009; Qiu et al., 2011) and hand-crafted features (Li et al., 2010; Liu et al., 2012, 2013; Chen et al., 2014). With the development of deep learning, supervised sequence taggers have become the mainstream due to their promising performance (Liu et al., 2015; Wang et al., 2016, 2017; Xu et al., 2018; Ma et al., 2019; Chen and Qian, 2020a). More recently, there emerge many studies that interact ATE with other tasks like aspect-level sentiment classification (Wang et al., 2018; He et al., 2019; Chen and Qian, 2020b). Since these methods highly depend on abundant domain-specific training data, they can hardly scale across the domains where labeled data is absent. Hence it would be more practical to develop unsupervised domain adaptation methods for ATE.

Domain Adaptation Many domain adaptation methods have been proposed to solve coarse-grained tasks like text classification (Blitzer et al., 2006; Ganin and Lempitsky, 2015; Guo et al., 2020). The basic idea in coarse-grained tasks is to transfer pivot words, which does not fit ATE well since most aspect terms are domain-specific non-pivot words. There have been a few attempts to this problem, which fall into two lines. (1) One is to model aspect-opinion relations. Early researches use common opinion seeds and pre-defined dependency link rules to build manual features (Jakob and Gurevych, 2010), conduct bootstrapping (Li et al., 2012), and create pseudo target labels (Ding et al., 2017). Due to the incompleteness of seeds and the inflexibility of rules, they often produce inferior performance. Subsequent studies (Wang and Pan, 2018, 2019a,b; Li et al., 2019) manually

[2]Context terms denote all words that are not aspect terms. Hence opinion terms form a subset of context terms.

annotate all opinion terms in reviews and design trainable neural models to capture the relations via multi-task learning. However, they induce extra annotation costs. (2) The other aims to find aspect-context relations. Xu et al. (2019) post-trains BERT on the cross-domain corpus to enhance its domain adaptation ability. Gong et al. (2020) and Pereg et al. (2020) further incorporate external syntactic information into BERT with auxiliary tasks or modified attention mechanisms, but they still rely on the prior knowledge in BERT. These methods often have more than 100M parameters and involve lots of computing power. Unlike all the aforementioned methods, we do not associate aspect terms with pivot words but actively transfer them via bridges.

3 Methodology

In this section, we first introduce the cross-domain ATE task. We then illustrate how to construct syntactic and semantic bridges. Lastly, we present the bridge-based sequence tagging.

3.1 Problem Statement

Given a review $x = \{x_1, ..., x_n\}$, we formulate ATE as a sequence tagging task that aims to predict a tag sequence $y = \{y_1, ..., y_n\}$, where each $y_i \in \{B, I, O\}$ denotes the *beginning of*, *inside of*, and *outside of* an aspect term. In this paper, we focus on the unsupervised domain adaptation for ATE, i.e., labeled training data is not available in the target domain. Specifically, given a set of labeled data $\mathcal{D}^S = \{(x_j^S, y_j^S)\}_{j=1}^{N_S}$ from the source domain and a set of unlabeled data $\mathcal{D}^U = \{(x_j^U)\}_{j=1}^{N_U}$ from the target domain, our goal is to predict labels y^T for the unseen target test data $\mathcal{D}^T = \{(x_j^T)\}_{j=1}^{N_T}$.

3.2 Bridge Construction

Given a review sentence x from either domain, we map it with a lookup table $\mathbb{E} \in \mathcal{R}^{d_e \times |V|}$, and generate word embeddings $\mathbf{E} = \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\} \in \mathcal{R}^{d_e \times n}$, where $|V|$ is the vocabulary size, and d_e is the embedding dimension. For cross-domain ATE, we construct bridges for reviews to help directly transfer aspect terms across two domains.

Syntactic Bridge In natural language, linguistic expressions are rich and flexible. In contrast, the syntactic structures are limited and are general across domains. Based on this observation, we propose to build connections between source and target words based on their syntactic roles (POS tags and dependency relations) rather than the lexical items. For example, from the parsing results in the upper part of Figure 3, the word *pizza* with a POS tag *NN* and dependency relations {*det*, *nsubj*} might be an aspect term, while those with the *RB* tag and *advmod* relation might not. Note the sentence "*The keyboard is in reasonable size.*" in the target domain has similar parsing results. Hence the syntactic roles can serve as supplementary evidence for recognizing aspect terms across domains.

Several prior studies (Wang and Pan, 2018, 2019b; Pereg et al., 2020) also make use of parsing results. However, they only use dependency relations to link words or to propagate word representations. For example, given a dependency $great \xrightarrow{nsubj} pizza$ in D^S, where *great* is a known pivot and *pizza* is an aspect term, the goal is to extract *keyboard* as an aspect from the target review *"The keyboard is great"* in D^T. The typical syntax based method Hier-Joint (Ding et al., 2017) first locates the pivot *great*, then utilizes the *nsubj* dependency to identify the term *keyboard*. Other methods like RNSCN (Wang and Pan, 2018) combine the embedding of the child node (*pizza*) with that of the parent node (*great*) according to the relation type, or reversely (depending on the specific design). It can be seen that the dependency relation *nsubj* here is only used as a link to the pivot.

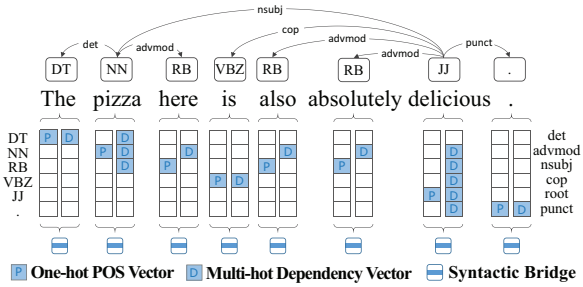


Figure 3: Construction of the syntactic bridge. If a POS tag or dependency relation is involved, its corresponding entry in the vector is set to 1, and otherwise 0.

We start in the opposite direction, i.e., we aim to fully exploit syntactic roles by recognizing themselves as pivots instead of treating them as links to pivots. To achieve this, we present a novel data structure to encode the POS and dependency information by grounding them into involved words. As shown in the lower part of Figure 3, for a word x_i, we use a **one-hot** vector $\boldsymbol{b}_{pos} \in \mathcal{R}^{N_{pos}}$ and a **multi-hot** vector $\boldsymbol{b}_{dep} \in \mathcal{R}^{N_{dep}}$ to represent its POS tag and dependency relation(s), where N_{pos} and N_{dep} are the number of tag/relation types. For

$\boldsymbol{b}_{dep}$, we merge all relations involved with x_i regardless of the direction (i.e., being the governor or dependent)[3].

To enlarge the learning capability, we project $\boldsymbol{b}_{pos}$ and $\boldsymbol{b}_{dep}$ to the same dimensionality with learnable weight matrices[4] and concatenate them to form the syntactic bridge $\boldsymbol{b}_{syn}$:

$$\boldsymbol{b}_{syn} = (\mathbf{W}_{pos} \times \boldsymbol{b}_{pos}) \oplus (\mathbf{W}_{dep} \times \boldsymbol{b}_{dep}), \quad (1)$$

where $\boldsymbol{b}_{syn} \in \mathcal{R}^{d_e}$ has the same dimensionality with the word embedding $\boldsymbol{e}$. In training, $\mathbf{W}_{pos}$ and $\mathbf{W}_{dep}$ get trained by labeled samples. In testing, we fix them and obtain $\boldsymbol{b}_{syn}$ for $\mathcal{D}^T$. By doing this, our proposed method well preserves two types of syntactic information throughout the entire learning process. As a result, we can take full advantage of their transferable information.

Semantic Bridge The semantic bridge takes the syntactic roles above as a basis but moves one step further to retrieve transferable prototypes. Unlike previous passive methods that construct information flows like *pizza→good→keyboard* via opinion terms or *pizza→offer→keyboard* via context terms, we aim to construct a direct flow like *pizza→keyboard*. For example, to transfer knowledge from *pizza* in $\mathcal{D}^S$ to *keyboard* in $\mathcal{D}^T$, we aim to introduce some supplementary target terms like {*disk*, *OS*, *mouse*} in $\mathcal{D}^U$ for *pizza* and directly improve its semantic relatedness with *keyboard*. We call these supplementary terms **prototypes** and will retrieve them to build the semantic bridges[5].

PLMs like BERT can find a set of semantically similar terms like {*hamburger*, *salad*} for *pizza*, which can also serve as prototypes. However, such prototypes are not suitable for the domain adaptation task, because aspect terms in one domain are often far away from those in another domain in the semantic space. To address this problem, we design a syntax-enhanced similarity metric to retrieve transferable semantic prototypes.

Before starting, we filter the words in $\mathcal{D}^U$ by frequency and only preserve those appearing more than τ times. We regard these words in unlabeled target data as candidate prototypes and build a prototype bank $\widetilde{V}$ from $\mathcal{D}^U$ accordingly. We then conduct retrieval following the procedure in Figure 4.

For a query word $v \in V^S$ (vocabulary of $\mathcal{D}^S$),

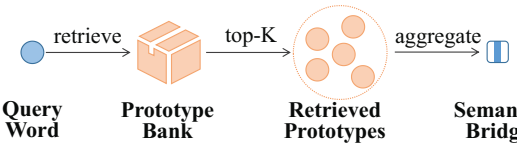


Figure 4: Construction of the semantic bridge. For a query word, the top-K prototypes are retrieved from the prototype bank and aggregated to its semantic bridge.

we want to find a prototype term $\widetilde{v} \in \widetilde{V}$ that play a similar syntactic role in the target domain. Specifically, we first summarize the global usages of v by merging its POS and dependency embeddings in all reviews where v appear in $\mathcal{D}^S$:

$$\begin{aligned} \boldsymbol{b}^g_{pos} &= \{\boldsymbol{b}_{pos,j=1} \mid \boldsymbol{b}_{pos,j=2} \mid ... \mid \boldsymbol{b}_{pos,j=N_S}\}, \\ \boldsymbol{b}^g_{dep} &= \{\boldsymbol{b}_{dep,j=1} \mid \boldsymbol{b}_{dep,j=2} \mid ... \mid \boldsymbol{b}_{dep,j=N_S}\}, \end{aligned} \quad (2)$$

where $\mid$ is the dimension-wise OR operation and N_S is the number of reviews in $\mathcal{D}^S$. Similarly, we can obtain $\widetilde{\boldsymbol{b}}^g_{pos}$ and $\widetilde{\boldsymbol{b}}^g_{dep}$ for $\widetilde{v}$. We then define the *syntax-enhanced similarity* between v and $\widetilde{v}$:

$$s.sim(v, \widetilde{v}) = c(\boldsymbol{b}^g_{pos}, \widetilde{\boldsymbol{b}}^g_{pos}) \times c(\boldsymbol{b}^g_{dep}, \widetilde{\boldsymbol{b}}^g_{dep}) \times c(\boldsymbol{e}, \widetilde{\boldsymbol{e}}), \quad (3)$$

where $\boldsymbol{e}$ and $\widetilde{\boldsymbol{e}}$ are word embeddings and $c(\cdot, \cdot)$ is the cosine similarity. Here the POS and dependency similarities are used to find similar syntactic roles, while the word similarity is used to reduce the noise of prototypes[6]. Consequently, we can obtain a $s.sim$ score matrix $\mathbf{M}^S \in \mathcal{R}^{|V^S| \times |\widetilde{V}|}$. After ranking, for v, we select the top-K words $\{\widetilde{v}_k\}_{k=1}^K$ with their $s.sim$ scores $\{\widetilde{s}_k\}_{k=1}^K$ from the prototype bank. Lastly, we aggregate these prototypes into the semantic bridge $\boldsymbol{b}_{sem}$ of v:

$$\boldsymbol{b}_{sem} = \sum_{k=1}^{K} \widetilde{s}_k \cdot \widetilde{\boldsymbol{e}}_k. \quad (4)$$

Following the way for $\mathcal{D}^S$, we also retrieve transferable prototypes for $\mathcal{D}^U$ and $\mathcal{D}^T$ using $\widetilde{V}$. In this way, source and target words with the same prototypes can be directly correlated to each other. For $\mathcal{D}^U$, we can generate a score matrix $\mathbf{M}^U \in \mathcal{R}^{|V^U| \times |\widetilde{V}|}$ by calculating the *s.sim* for all words in $\mathcal{D}^U$ and all candidate prototypes in $\widetilde{V}$. Then we can obtain the semantic bridge $\boldsymbol{b}_{sem}$ for each word in $\mathcal{D}^U$ in training. In testing, $\mathcal{D}^T$ is unseen and the global $\boldsymbol{b}^g_{pos}$/$\boldsymbol{b}^g_{dep}$ are not available. Therefore, for a word w in $\mathcal{D}^T$, we obtain $\boldsymbol{b}_{sem}$ using $\mathbf{M}^U$ if w has appeared in $\mathcal{D}^U$. Otherwise, we temporarily use the local $\boldsymbol{b}_{pos}$/$\boldsymbol{b}_{dep}$ of w in current tesing sample to replace the global $\boldsymbol{b}^g_{pos}$/$\boldsymbol{b}^g_{dep}$ and calculate the $s.sim$.

[3]This simplification almost has no side effects. If a word has a *NN* tag and *det* relation, it must be the governer.

[4]In all equations, $\mathbf{W}$ denotes a trainable weight matrix.

[5]We retrieve prototypes for all words in the review due to the existence of domain-specific context terms like *eat*.

[6]A domain-invariant word that appears frequently in both domains should preserve its own information. It will have a maximum similarity score with itself since $c(\boldsymbol{e}, \widetilde{\boldsymbol{e}}) = 1$.

3.3 Bridge-based Sequence Tagging

Based on the syntactic and semantic bridges, we now propose a lightweight end-to-end sequence tagger for aspect term extraction. As shown in Figure 5, the tagger receives a mixture of $\mathcal{D}^S$ and $\mathcal{D}^U$ for training and then makes predictions for $\mathcal{D}^T$ in testing. We then illustrate the details.

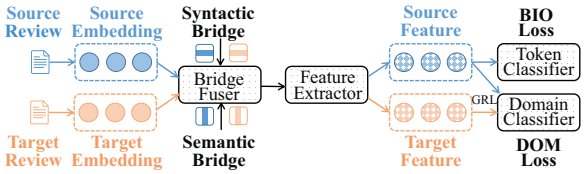


Figure 5: Training of bridge-based sequence tagging.

Bridge Fuser Our constructed bridges have two properties. (1) Bridges are domain-invariant and should be preserved. (2) Bridges can help extract domain-invariant information from $\boldsymbol{e}_i$. Therefore, we propose to enhance the embedding $\boldsymbol{e}_i$ of a word x_i with its transferable bridges $\boldsymbol{b}_{syn,i}$ and $\boldsymbol{b}_{sem,i}$. Specifically, we use a gating operation to fuse bridges. Take the syntactic bridge as an example, we first calculate a dimension-wise gate $\boldsymbol{g}_{syn,i}$:

$$\boldsymbol{g}_{syn,i} = \sigma\left(\mathbf{W}_{syn}(\boldsymbol{e_i} \oplus \boldsymbol{b}_{syn,i})\right), \tag{5}$$

where $\mathbf{W}_{syn} \in \mathcal{R}^{2d_e \times 2d_e}$, σ is the Sigmoid function, $\oplus$ is concatenation. We then scale the concatenated vector $\boldsymbol{e_i} \oplus \boldsymbol{b}_{syn,i}$ with $\boldsymbol{g}_{syn,i}$ and obtain the syntactic bridge enhanced embedding $\boldsymbol{e}_{syn,i}$:

$$\boldsymbol{e}_{syn,i} = \boldsymbol{g}_{syn,i} \odot (\boldsymbol{e_i} \oplus \boldsymbol{b}_{syn,i}), \tag{6}$$

where $\odot$ is an element-wise multiplication. The semantic bridge enhanced embedding $\boldsymbol{e}_{sem,i}$ can be calculated similarly. We term the model with $\boldsymbol{e}_i$, $\boldsymbol{e}_{syn,i}$, and $\boldsymbol{e}_{sem,i}$ input as **BaseTagger**, **SynBridge**, and **SemBridge**, respectively. Three types of embeddings are collectively called $\boldsymbol{e}_{input,i}$.

Feature Extractor Previous studies (Xu et al., 2018) show that low-level token features are insufficient for tagging terms. Therefore, we use a CNN encoder containing L stacked convolutional layers with ReLU activation to extract the high-level features $\boldsymbol{f_i} \in \mathcal{R}^{d_f}$:

$$\boldsymbol{f}_i^{l+1} = ReLU(\boldsymbol{f}_{i-c:i+c}^{l} * \mathbf{K}^l + b^l), \quad \boldsymbol{f}_i^0 = \boldsymbol{e}_{input,i}, \tag{7}$$

where $\mathbf{K} \in \mathcal{R}^{d_f \times (d_{input} \times ks)}$ is the kernel group, $ks = 2c + 1$ is the kernel size.

Token Classifier For recognizing aspect and opinion terms, we send $\boldsymbol{f}_i^L$ in the last layer to a token classifier:

$$\hat{y}_i = Softmax(\mathbf{W}_A \times \boldsymbol{f}_i^L), \tag{8}$$

where $\hat{y}_i$ is the prediction of the word x_i.

Domain Classifier Besides BIO tagging, we further enhance the domain-invariance of bridge-based features via domain adversarial training. Specifically, we first aggregate $\boldsymbol{f}_i^L$ to a global representation $\boldsymbol{f_g}$:

$$\boldsymbol{f}_g = MaxPool(\boldsymbol{f}_{1:n}^L). \tag{9}$$

Then we add a Gradient Reversal Layer (GRL) (Ganin and Lempitsky, 2015) to $\boldsymbol{f}_g$ with the scale coefficient λ and train a domain classifier to distinguish the domain that $\boldsymbol{f}_g$ belongs to:

$$\hat{y}_d = Softmax(\mathbf{W}_O \times MLP(GRL_\lambda(\boldsymbol{f}_g))), \tag{10}$$

where $\hat{y}_d$ is the domain prediction, and MLP contains L_D layers with ReLU activation.

Training Procedure In training, only samples from $\mathcal{D}^S$ have corresponding BIO labels y^S for token classification. The goal is to minimize the tagging loss for recognizing aspect terms:

$$\mathcal{L}_{BIO} = -\sum_{\mathcal{D}^S}\sum_{i=1}^{n} \ell(\hat{y}_i, y_i), \tag{11}$$

where ℓ is the cross-entropy loss function. On the other hand, the samples from $\mathcal{D}^S$ and $\mathcal{D}^U$ are used to train the domain classifier and minimize the following domain classification loss:

$$\mathcal{L}_{DOM} = -\sum_{\mathcal{D}^S \cup \mathcal{D}^U} \ell(\hat{y}_d, y_d), \tag{12}$$

where $y_d = 0$ for $\mathcal{D}^S$ and $y_d = 1$ for $\mathcal{D}^U$. The final loss for training the end-to-end tagger is defined as $\mathcal{L} = \mathcal{L}_{BIO} + \mathcal{L}_{DOM}$. Notice that $\mathcal{D}^T$ is only used in testing. There is no data leakage in training, and the task setting is strictly inductive.

4 Experiment

4.1 Experimental Setup

Datasets We use three conventional English datasets from different domains and construct six directed transfer pairs, where $\mathbb{R}$ and $\mathbb{L}$ are from SemEval 2014 and 2015 (Pontiki et al., 2014, 2015), and $\mathbb{D}$ is collected by Hu and Liu (2004). Following previous studies (Wang and Pan, 2018, 2019b; Pereg et al., 2020), we use three different splits and each split has a fixed train-test ratio 3:1. The detailed statistics of datasets are presented in Table 1[7].

Table 1: The statistics of datasets.

Dataset	Domain	Total	Train	Test
$\mathbb{R}$	Restaurant	5841	4381	1460
$\mathbb{L}$	Laptop	3845	2884	961
$\mathbb{D}$	Device	3836	2877	959

[7]Our code and data are available at https://github.com/NLPWM-WHU/BRIDGE.

Table 2: Comparison of different methods. Baselines with △ use annotated opinion terms. The best scores are in bold and the second best ones are underlined. Averaged results with † and ‡ are significantly better than BERT-Cross and BaseTagger ($p < 0.05$) based on one-tailed unpaired t-test, respectively. The upper bounds of three datasets (achieved by BaseTagger trained on in-domain labeled data) are 76.43 ($\mathbb{R}$), 75.60 ($\mathbb{L}$), and 57.10 ($\mathbb{D}$).

Type	Model	Embedding	$\mathbb{R}\rightarrow\mathbb{L}$	$\mathbb{L}\rightarrow\mathbb{R}$	$\mathbb{R}\rightarrow\mathbb{D}$	$\mathbb{D}\rightarrow\mathbb{R}$	$\mathbb{L}\rightarrow\mathbb{D}$	$\mathbb{D}\rightarrow\mathbb{L}$	AVG.
I	TCRF	Manual	19.72	28.19	21.07	6.59	29.96	24.22	21.63
	RAP	Manual	25.92	46.90	22.63	45.44	34.54	28.22	33.94
	SAL	Word2vec	29.03	44.57	22.82	38.89	38.82	47.25	36.90
	Hier-Joint	Word2vec	33.66	48.10	33.20	47.97	31.25	34.74	38.15
	RNSCN$^{\triangle}$	Word2vec	40.43	52.91	35.10	48.36	40.42	51.14	44.73
	TRNN$^{\triangle}$	Word2vec	40.15	53.78	37.33	51.17	41.19	51.66	45.88
	TIMN$^{\triangle}$	Word2vec	43.68	54.12	35.45	53.82	38.63	52.46	46.36
II	BERT-Base	BERT	33.89	42.74	35.30	36.86	43.54	46.06	39.73
	UDA	BERT	44.24	50.52	40.04	53.39	41.48	52.33	47.00
	SA-EXAL$^{\triangle}$	BERT	47.59	54.67	40.50	54.54	42.19	47.72	47.87
	BERT-Cross	BERT	46.30	51.60	**43.68**	53.15	44.22	50.04	48.17
III	BaseTagger	Word2vec	48.86	61.42	40.56	57.67	43.75	51.95	50.70†
	SynBridge	Word2vec	**51.53**	63.90	42.76	59.40	44.97	52.44	52.50†‡
	SemBridge	Word2vec	**51.53**	**65.96**	43.03	**60.61**	**45.37**	**53.77**	**53.38**†‡

Settings We pre-process each dataset by lowercasing all words. We use the same *word2vec* vectors as previous studies (Wang and Pan, 2018, 2019a,b) to generate word embeddings, and set the dimensionality d_e=100. In the syntactic bridge, we use Stanford CoreNLP (Manning et al., 2014) for dependency parsing. There are 45 classes of POS tags and 40 classes of dependency relations in three datasets. In the semantic bridge, we set the frequency threshold τ=5, the number of prototypes K=10. In the end-to-end tagger, we set the number of convolution layers L=4, and the kernel size ks of each layer is 3, 5, 5, 5, respectively, the number of MLP layers L_D=3, and dropout (Srivastava et al., 2014) is applied to layers' outputs with the probability 0.5. The dimensionality of features d_f=256, the scale coefficient of GRL λ=0.1. We train the tagger for 100 epochs using Adam optimizer (Kingma and Ba, 2015) with the learning rate 1e-4 and batch size 8 in a 1080Ti GPU.

Evaluation For each transfer pair, we use the labeled training data from the source domain and unlabeled training data from the target domain to train the tagger. Then we evaluate the tagger on unseen test data from the target domain. We use the mean F1-scores of aspect terms over three splits with three random seeds (i.e., nine runs for each transfer pair) for evaluation[8].

4.2 Compared Methods

We classify all models into three categories.

Type-I denotes the opinion term-based methods. TCRF (Jakob and Gurevych, 2010), RAP (Li et al., 2012), and Hier-Joint (Ding et al., 2017) use manually defined dependency rules. RNSCN and TRNN (Wang and Pan, 2018, 2019a) model dependency trees with trainable recursive networks. SAL (Li et al., 2019) and TIMN (Wang and Pan, 2019b) replace the dependency tree with trainable memory interaction.

Type-II denotes context term-based methods. BERT-Base uses vanilla base BERT (Devlin et al., 2019) for ATE. BERT-Cross (Xu et al., 2019) post-trains BERT on a combination of Yelp and Amazon corpus. UDA (Gong et al., 2020) and SA-EXAL (Pereg et al., 2020) incorporate syntactic information into BERT with auxiliary tasks and modified attention mechanisms[9].

Type-III denotes the proposed active domain adaptation strategy. BaseTagger is the tagger without bridges, while SynBridge and SemBridge use syntactic and semantic bridges, respectively.

4.3 Main Results

The comparison results for all methods are shown in Table 2. It is clear that our proposed model achieves a new state-of-the-art performance in terms of the average F1-scores. For example, SemBridge outperforms the best TIMN in Type-I by 7.02% and BERT-Cross in Type-II by 5.21%, respectively. We also notice that our BaseTagger already outperforms all baselines. We attribute this to the design of CNN feature extractor and domain adversarial training (DAT). CNN focuses on the N-gram feature rather than a single word and reduces the side effects of non-pivot aspect terms. DAT is applied to the sentence-level features, such that they are not misled by the common N-grams that are labeled both 0 and 1.

[8]The hyperparameter ranges are presented in Appendix A.

[9]Since SAL and UDA use extra aspect sentiment labels, we show how to make them fair competitors in Appendix B.

SynBridge and SemBridge further improve BaseTagger with a 1.80% and 2.68% absolute gain, respectively. This proves the effectiveness of our proposed active domain adaptation strategy. Meanwhile, SemBridge is a bit superior to SynBridge. The reasons are two-fold. (1) The semantic bridges come from prototype words that possess prior embedding knowledge and also contain syntactic information, while the syntactic bridges are merely trained from scratch. (2) The retrieved top-K terms make the supplementary information in SemBridge more diverse and abundant than that in SynBridge.

Among the baselines, early methods using common opinion seeds and pre-defined rules are inferior. Relying on annotated opinion terms, the methods like TIMN get some improvements but induce extra annotation costs. By incorporating pre-trained BERT with external dependency and cross-domain corpus, UDA, SA-EXAL, and BERT-Cross outperform previous methods, but they need high computational resources. In contrast, by using the static Word2vec embeddings, our model can outperform those with dynamic BERT representations. This is instructive for other researches in that there is still room for improvement by exploring the syntactic and semantic features beyond the popular BERT-based models[10].

5 Analysis

5.1 What If There Is an OTE Task?

With the proposed active domain adaptation strategy, we do not need any manually labeled opinion terms for ATE. However, this does not mean that our method cannot handle opinion term extraction (i.e., OTE). In contrast, if the labeled opinion terms are provided in D^S, we can also conduct the OTE task for D^T by simply modifying the tagger. In specific, we add an opinion term prediction layer in Eq.8 and then extract aspect and opinion terms simultaneously. The results are shown in Table 3.

Obviously, our method again outperforms all baselines[11]. We find a small performance decrease in AVG-AS compared with that in Table 2. Similar results are also observed in BERT-Base. The reason is that the objective of ATE and OTE may interfere with each other without proper balancing and a sophisticated multi-task learning framework.

[10]We also make some explorations about combining SynBridge and SemBridge, please refer to Appendix C.

[11]Please refer to Appendix D for detailed results for all transfer pairs.

Table 3: Comparison of different methods. AVG-AS and AVG-OP are F1-scores for ATE and OTE averaged on all transfer pairs.

Model	AVG-AS	AVG-OP
RNSCN	44.73	67.44
TRNN-GRU	45.88	67.12
TIMN	46.36	68.21
BERT-Base	39.52	66.22
SA-EXAL	47.87	69.15
BERT-Cross	48.35	69.47
BaseTagger	50.12	71.73
SynBridge	51.86	71.73
SemBridge	**52.53**	**72.08**

5.2 Ablation Study

We conduct a series of ablation study to validate the effectiveness of our method. The results are shown in Table 4.

Table 4: Ablation study. The scores denote the decrease of performance after removing($-$) or replacing($\rightarrow$) a specific component.

Index	Model	Variant	AVG.
1	BaseTagger	$-\ \mathcal{L}_{DOM}$	1.94
2		CNN$\rightarrow$BiLSTM	8.47
3	SynBridge	$-\ \boldsymbol{b}_{pos}$	1.68
4		$-\ \boldsymbol{b}_{dep}$	1.49
5		$\boldsymbol{b}_{dep}\rightarrow$Tree-LSTM	3.97
6		$\boldsymbol{b}_{dep}\rightarrow$GCN	4.21
7	SemBridge	$-\ c(\boldsymbol{e}, \widetilde{\boldsymbol{e}})$	1.82
8		$-\ c(\boldsymbol{b}_{pos}, \widetilde{\boldsymbol{b}}_{pos})$	2.30
9		$-\ c(\boldsymbol{b}_{dep}, \widetilde{\boldsymbol{b}}_{dep})$	2.52

Results 1$\sim$2 conform to our previous discussion about BaseTagger that both CNN and domain adversarial training contribute to overall good performance. Results 3$\sim$6 show the effectiveness of POS and dependency embeddings in SynBridge. Specifically, in 5$\sim$6, we replace our proposed structure for dependency with frequently-used Tree-LSTM and GCN to model the dependency tree and find a significant drop in performance. Results 7$\sim$9 show the importance of all three types of similarity for retrieving prototypes in SemBridge.

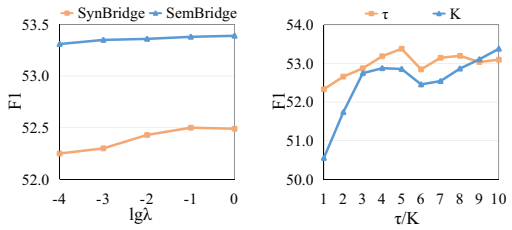
5.3 Parameter Study

There are three key hyperparameters in our method: the scale coefficient of GRL λ, the frequency threshold τ, and the number of prototypes K. We vary λ in the range $10^{-4} \sim 1.0$ and τ/K in $1 \sim 10$ to investigate their impacts and present the results in Figure 6.

In Figure 6(a), when increasing λ from 10^{-4} to 10^{-1}, we enlarge the scale of domain adversarial training in GRL and get small improvements. However, the performance does not keep rising when

Table 5: Case study. The left columns present the selected target testing examples, and the words in red are aspect terms. The right columns denote the extraction results of corresponding models.

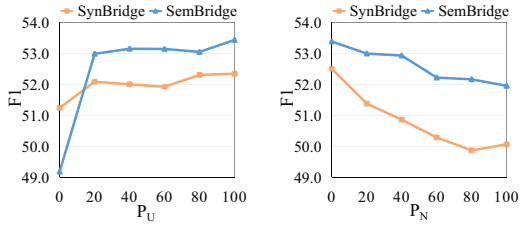
Pair	Example	RNSCN	BERT-Cross	SynBridge	SemBridge
$\mathbb{R}\rightarrow\mathbb{L}$	S1. it has usb ports, 1 sd memory card reader and an sd memory car expansion.	None ✗	card reader,✗ sd memory car expansion	usb ports, sd memory card reader, sd memory car expansion	usb ports, sd memory card reader, sd memory car expansion
$\mathbb{L}\rightarrow\mathbb{R}$	S2. The asparagus, truffle oil, parmesan bruschetta is a winner!	None ✗	asparagus, bruschetta ✗	asparagus,truffle oil parmesan bruschetta	asparagus,truffle oil parmesan bruschetta
$\mathbb{L}\rightarrow\mathbb{R}$	S3.They showed up 15 minutes after the tuna melt.	tuna melt ✗	None ✗	tuna melt ✗	tuna



(a) Impact of λ. (b) Impact of τ/K.

Figure 6: Impacts of hyperparameters λ, τ, and K.

$\lambda = 1.0$. This result shows that simply forcing non-pivots to transfer knowledge is not suitable for domain adaptation. In Figure 6(b), τ is used to balance diversity and accuracy. A low τ means that prototypes are diverse, but some of them are long-tail words and contribute little to the reduction of domain discrepancy. On the contrary, a high τ only preserves frequent prototypes, and some meaningful prototypes are filtered out. Therefore, a middle τ=5 is an appropriate choice. For K, the curve is generally upward when more prototypes are introduced. This trend is reasonable since more prototypes equal to more target information.



(a) Impact of P_U. (b) Impact of P_N.

Figure 7: Impacts of P_U and P_N.

In Figure 7, we further analyze the impacts of the percentage of unlabeled data P_U and the percentage of parsing noise P_N. For P_U, the performance is generally better when more unlabeled target data is introduced. Moreover, around 20%∼40% unlabeled data is enough to achieve satisfactory performance. Notice that SemBridge without unlabeled data will degenerate into BaseTagger since no prototypes can be retrieved. For P_N, we manually disturb the parsing results to observe the robustness of our method. Clearly, after introducing noises on parsing, the performance begins to degrade, but not by a large margin. Our method has the ability to resist parsing errors for two reasons. First, beyond syntactic roles, we also incorporate embedding similarity when retrieving prototypes (for SemBridge only). Second, the gating mechanism can further filter useless syntactic information and maintain the quality of word representations.

5.4 Case Study

To have a close look, we select a few samples from testing target data for a case study. S1 and S2 show the positive impacts of bridges. Due to the space limit, we illustrate S1 in detail. Since most words in S1 are domain-specific terms in $\mathbb{L}$, RNSCN fails to recognize any aspect terms by simply propagating word representations with dependency. BERT-Cross only extracts a part of aspect terms based on its prior knowledge. For our bridge-based method, SynBridge supplements syntactic roles {*nummod*, *compound*, *obj*, *conj*, *NNS*} for *port*. These syntactic roles also join the representation of *usb* and help to extract *usb ports* correctly. For SemBridge, the analysis is much straightforward. *usb* is the prototype of typical aspect terms in $\mathbb{R}$ like {*garlic*, *thai*, *banana*}, thus the tagger with semantic bridges can easily recognize *usb* as an aspect term.

S3 further illustrates how SemBridge helps recover from the wrong parsing results. Such results make two syntax based methods RNSCN and SynBridge stop working. In contrast, *tuna* is the prototype of noun words like {*nvidia*, *amd*, *blade*} in $\mathbb{L}$ and *melt* has the verb prototype like {*imagine*, *hang*, *relax*} in $\mathbb{R}$, thus SemBridge correctly extracts *tuna* and filters out *melt* in the same time.

In Table 6, We further present several sample prototypes of the training data from the transfer pairs $\mathbb{R}\rightarrow\mathbb{L}$ (upper three) and $\mathbb{L}\rightarrow\mathbb{R}$ (lower three) in SemBridge, where three terms on the left are aspect term, opinion term, and context term, respectively. For a source non-pivot term like *processor* in $\mathbb{L}$, SemBridge enhances it with typical target words like *soup* and *burger*. As a result, the domain discrepancy between the source and target data is largely reduced with the help of prototypes.

Table 6: Top-10 prototypes in SemBridge. Words are ranked by their $s.sim$ scores.

Term	Prototypes
food	machine,product,keyboard,netbook,service, computer,screen,value,touchpad,processor
delicious	amazing,wonderful,awesome,great,good,nice, fantastic,beautiful,perfect,lightweight
cook	use,load,plug,work,turn,break,charge,change, help,run
processor	soup,burger,meal,sauce,flavor,cheese,food, salad,seafood,fan
efficient	attentive,impressive,affordable,friendly,reasonable,pleasant,simple,courteous,helpful,hungry
freeze	eat,hang,stop,die,bring,stay,leave,start,give,keep

5.5 Analysis on Computational Cost

In practice, for any transfer pairs, the one-time construction of syntactic and semantic bridges can finish within 30 seconds. Therefore, we focus on the end-to-end training costs of SynBridge/SemBridge. We run five top-performing methods on the transfer pair $\mathbb{R}\rightarrow\mathbb{L}$ and present the trainable parameter number and running time per epoch of each method in Table 7. We can conclude that our proposed method maintains a quite low computational cost.

Table 7: Computational cost of each method.

	Parameter	Runtime
TIMN	0.8M	132s
BERT-Cross	109M	84s
BaseTagger	1.3M	11s
SynBridge/SemBridge	1.4M	12s

6 Conclusion

In this paper, we propose a novel active domain adaptation method for aspect term extraction. Unlike previous studies that conduct passive domain adaptation by associating aspect terms with pivots, we actively enhance the terms' transferability by constructing syntactic and semantic bridges for them. We then design a lightweight end-to-end tagger for bridge-based sequence tagging. Experiments on six transfer pairs demonstrate that our method achieves a new state-of-the-art performance with a quite low computational cost.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. The work described in this paper is supported by the NSFC projects (61572376, 91646206), and the 111 project (B07037).

References

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

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL*, pages 347–358.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *ACL*, pages 547–556.

Zhuang Chen and Tieyun Qian. 2020a. Enhancing aspect term extraction with soft prototypes. In *EMNLP*, pages 2107–2117. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2020b. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *ACL*, pages 3685–3694.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *AAAI*, pages 3436–3442.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189.

Chenggong Gong, Jianfei Yu, and Rui Xia. 2020. Unified feature and instance based domain adaptation for aspect-based sentiment analysis. In *EMNLP*, pages 7035–7045.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2020. Multi-source domain adaptation for text classification via distancenet-bandits. In *AAAI*, pages 7830–7838.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *ACL*, pages 504–515.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *EMNLP*, pages 1035–1045.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Yingju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *COLING*, pages 653–661.

Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain co-extraction of sentiment and topic lexicons. In *ACL*, pages 410–419.

Zheng Li, Xin Li, Ying Wei, Lidong Bing, Yu Zhang, and Qiang Yang. 2019. Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. In *EMNLP-IJCNLP*, pages 4589–4599.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *IJCAI*, pages 2134–2140.

Kang Liu, Liheng Xu, and Jun Zhao. 2012. Opinion target extraction using word-based translation model. In *EMNLP*, pages 1346–1356.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *ACL*, pages 3538–3547.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60.

Oren Pereg, Daniel Korat, and Moshe Wasserblat. 2020. Syntactically aware cross-domain aspect and opinion terms extraction. In *COLING*, pages 1772–1777.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval*, pages 486–495.

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

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *EMNLP*, pages 339–346.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Feixiang Wang, Man Lan, and Wenting Wang. 2018. Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning. In *IJCNN*, pages 1–8.

Wenya Wang and Sinno Jialin Pan. 2018. Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *ACL*, pages 2171–2181.

Wenya Wang and Sinno Jialin Pan. 2019a. Syntactically meaningful and transferable recursive neural networks for aspect and opinion extraction. *CL*, 45(4):705–736.

Wenya Wang and Sinno Jialin Pan. 2019b. Transferable interactive memory network for domain adaptation in fine-grained opinion extraction. In *AAAI*, pages 7192–7199.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*, pages 616–626.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*, pages 1533–1541.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*, pages 592–598.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *NAACL-HLT*, pages 2324–2335.

A Ranges of Hyperparameters

We present the hyperparameter ranges in Table 8. We select all hyperparameters via manual tuning.

Table 8: Ranges of Hyperparameters.

Hyperparameter	Range	Best
frequency threshold τ	1,2,3,4,5,6,7,8,9,10	5
number of prototypes K	1,2,3,4,5,6,7,8,9,10	10
number of CNN layers L	1,2,3,4,5	4
dimension of CNN features d_f	64, 128, 256	256
kernel size ks of CNN layer 1	3,5,7,9	3
kernel size ks of CNN layer 2	3,5,7,9	5
kernel size ks of CNN layer 3	3,5,7,9	5
kernel size ks of CNN layer 4	3,5,7,9	5
number of MLP layers L_D	1,2,3,4,5	3
the scale coefficient of GRL λ	$10^{[-4,-3,-2,-1,0]}$	10^{-1}

Table 9: Comparison of different methods when there is an OTE task. The best scores are in bold and the second best ones are underlined. AS and OP denote aspect and opinion F1-scores. Averaged results with * are significantly better than the best baseline BERT-Cross ($p < 0.01$) based on one-tailed unpaired t-test.

Models	$\mathbb{R}\rightarrow\mathbb{L}$		$\mathbb{L}\rightarrow\mathbb{R}$		$\mathbb{R}\rightarrow\mathbb{D}$		$\mathbb{D}\rightarrow\mathbb{R}$		$\mathbb{L}\rightarrow\mathbb{D}$		$\mathbb{D}\rightarrow\mathbb{L}$		AVG.	
	AS	OP	AS	OP	AS	OP	AS	OP	AS	OP	AS	OP	AS	OP
RNSCN	40.43	65.85	52.91	72.51	35.10	60.17	48.36	73.75	40.42	61.15	51.14	71.18	44.73	67.44
TRNN	40.15	65.63	53.78	73.40	37.33	60.32	51.17	74.37	41.19	60.20	51.66	68.79	45.88	67.12
TIMN	43.68	68.44	54.12	73.69	35.45	59.05	53.82	76.52	38.63	62.22	52.46	69.32	46.36	68.21
BERT-Base	34.70	73.84	37.07	80.12	37.17	64.52	40.54	60.45	43.45	59.59	44.19	58.77	39.52	66.22
SA-EXAL	47.59	**75.79**	54.67	80.05	40.50	63.33	54.54	71.57	42.19	60.19	47.72	63.98	47.87	69.15
BERT-Cross	44.00	75.38	54.31	**81.97**	43.12	**66.57**	51.97	70.58	44.35	58.49	50.01	63.81	48.35	69.47
BaseTagger	47.78	70.61	58.39	79.53	39.71	63.63	57.56	80.18	44.49	64.14	52.77	72.30	50.12	71.73*
SynBridge	50.59	70.74	60.94	79.86	42.42	63.37	59.92	79.88	**45.30**	**64.22**	51.97	72.33	51.86*	71.73*
SemBridge	**50.67**	71.51	**63.04**	80.48	**43.34**	63.46	**60.19**	**80.21**	44.91	64.15	**53.02**	**72.63**	**52.53***	**72.08***

B Modification of SAL and UDA

Since SAL and UDA are designed for end-to-end cross-domain aspect-based sentiment analysis, they have access to the aspect sentiment labels in training. As previous studies show, aspect term extraction and aspect-level sentiment classification can benefit each other. Therefore, it is unfair to directly compare our method with SAL and UDA.

We choose to modify SAL and UDA and make them fair competitors. We degrade the collapsed tags {B-POS, I-POS, B-NEG, I-NEG, B-NEU, I-NEU, O} to {B, I, O} thus remove the aspect-level sentiment classification task. Following other BERT-based methods, we use BERT-Base as the backbone of UDA.

C Can We Combine SynBridge and SemBridge?

Since SynBridge and SemBridge contain transferable syntactic and semantic information, it is intuitive to combine them for a better performance than either individual model. Here we apply a very simple operation for combination.

For a word x_i with embedding $\boldsymbol{e}_i$, we first obtain its syntactic and semantic bridges $\boldsymbol{b}_{syn,i}$ and $\boldsymbol{b}_{sem,i}$, and merge them into a combined bridge:

$$\boldsymbol{b}_{com,i} = (\mathbf{W}_{syn} \times \boldsymbol{b}_{syn,i}) + (\mathbf{W}_{sem} \times \boldsymbol{b}_{sem,i}), \quad (13)$$

Then we conduct a similar gating operation and get the combined bridge enhanced embedding $\boldsymbol{e}_{com,i}$:

$$\begin{aligned} \boldsymbol{g}_{com,i} &= \sigma\left(\mathbf{W}_{com}(\boldsymbol{e_i} \oplus \boldsymbol{b}_{com,i})\right) \\ \boldsymbol{e}_{com,i} &= \boldsymbol{g}_{com,i} \odot (\boldsymbol{e_i} \oplus \boldsymbol{b}_{com,i}), \end{aligned} \quad (14)$$

Lastly, we regard $\boldsymbol{e}_{com,i}$ as the input of tagger and make predictions for aspect terms. We term this model **ComBridge** and present the results in Table 10.

Table 10: Comparison of different bridge-based methods. The best scores are in bold and the second best ones are underlined.

Model	$\mathbb{R}\rightarrow\mathbb{L}$	$\mathbb{L}\rightarrow\mathbb{R}$	$\mathbb{R}\rightarrow\mathbb{D}$	$\mathbb{D}\rightarrow\mathbb{R}$	$\mathbb{L}\rightarrow\mathbb{D}$	$\mathbb{D}\rightarrow\mathbb{L}$	AVG.
BaseTagger	48.86	61.42	40.56	57.67	43.75	51.95	50.70
SynBridge	51.53	63.90	42.76	59.40	44.97	52.44	52.50
SemBridge	51.53	65.96	**43.03**	60.61	**45.39**	**53.77**	53.38
ComBridge	**53.32**	**66.20**	42.56	**60.99**	44.74	53.32	**53.52**

ComBridge slightly outperforms SemBridge and achieves the optimal results in all bridge-based methods. The small improvement is explicable since SemBridge already contains most of the syntactic information in SynBridge and we do not use any sophisticated methods in combination.

D Detailed Results for an Additional OTE Task

When opinion terms are labeled, our method can also conduct aspect term extraction and opinion term extraction simultaneously. For recognizing aspect and opinion terms, we only need to add an opinion term prediction layer:

$$\begin{aligned} \hat{y}_{a,i} &= Softmax(\mathbf{W}_A \times \boldsymbol{f}_i^L), \\ \hat{y}_{o,i} &= Softmax(\mathbf{W}_O \times \boldsymbol{f}_i^L), \end{aligned} \quad (15)$$

where $\hat{y}_{a,i}$ / $\hat{y}_{o,i}$ are the predictions of $\{B, I, O\}$ for the aspect / opinion terms. And the resulted BIO loss is calculated as follow:

$$\mathcal{L}_{BIO} = -\sum_{\mathcal{D}^S}\sum_{i=1}^{n} \ell(\hat{y}_{a,i}, y_{a,i}) + \ell(\hat{y}_{o,i}, y_{o,i}) \quad (16)$$

where ℓ is the cross-entropy loss function.

We present the detailed results in Table 9. Obviously, our proposed SynBridge and SemBridge outperform other baselines in both aspect and opinion F1-scores.