



Retrieve-and-Edit Domain Adaptation for End2End Aspect Based Sentiment Analysis

Zhuang Chen , Graduate Student Member, IEEE, and Tiejun Qian , Member, IEEE

Abstract—End-to-end aspect based sentiment analysis (E2E-ABSA) aims to jointly extract aspect terms and predict aspect-level sentiment for opinion reviews. Though supervised methods show effectiveness for E2E-ABSA tasks, the annotation cost is extremely high due to the necessity of fine-grained labels. Recent attempts alleviate this problem using the domain adaptation technique to transfer the word-level common knowledge across domains. However, the biggest issue in domain adaptation, i.e., how to transfer the domain-specific words like *pizza* and *delicious* in the source “Restaurant” to the target “Laptop” domain, has not been resolved. In this paper, we propose a novel domain adaptation method to address this issue by enhancing the transferability of domain-specific source words in a retrieve-and-edit way. Specifically, for all source words, we first retrieve the transferable prototypes from unlabeled target data via their syntactic and semantic roles. We then edit the source words to enhance their transferability by absorbing the knowledge carried in prototypes. Finally, we design an end-to-end framework to jointly accomplish cross-domain aspect term extraction and aspect-level sentiment classification. We conduct extensive experiments on four real-world datasets. The results prove that, by introducing transferable prototypes, our method significantly outperforms the state-of-the-art methods, achieving an absolute 3.95% F1 increase over the best baseline.

Index Terms—End-to-end aspect based sentiment analysis, domain adaptation, retrieve-and-edit, transferable prototypes.

I. INTRODUCTION

ASPECT based sentiment analysis (ABSA) is a fine-grained task that aims to summarize the opinions of users towards specific aspects in opinion reviews. With the rapid growth of the world wide web and social media, ABSA has been widely applied to various fields such as product review analysis, forum discussions, blog posts. ABSA mainly consists of two subtasks, i.e., aspect term extraction (ATE) and aspect-level sentiment classification (ASC). For example, given a sentence “*The pizza here is also absolutely delicious.*” ATE aims to extract the term *pizza* while ASC aims to classify its corresponding sentiment polarity *positive*. Considering that these two subtasks are highly correlated, recent studies propose to solve them in an end-to-end manner, i.e., E2E-ABSA.

Manuscript received July 6, 2021; revised January 4, 2022; accepted January 19, 2022. Date of publication January 25, 2022; date of current version February 7, 2022. This work was supported by NSFC under Projects 61572376, 62032016, and 61972291. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jing Huang. (Corresponding author: Tiejun Qian.)

The authors are with the School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China (e-mail: zhchen18@whu.edu.cn; qty@whu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2022.3146052

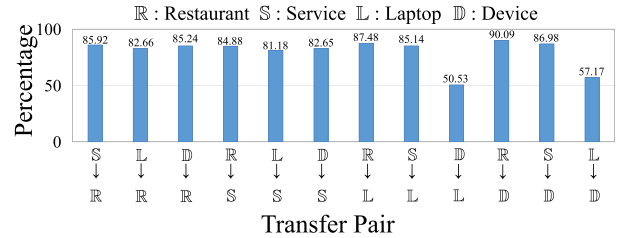


Fig. 1. The proportions of domain-specific aspect terms in \mathcal{D}^S . \mathbb{R} (Restaurant), \mathbb{L} (Laptop), \mathbb{D} (Device), and \mathbb{S} (Service) are four datasets from different domains.

Existing methods in E2E-ABSA fall into two types: collapsed tagging [1], [2] and joint training [3], [4]. The former constructs collapsed labels like $\{B\text{-}positive, I\text{-}negative, O\}$, where $\{B, I, O\}$ and $\{positive, negative\}$ are the labels for ATE and ASC, respectively. The latter constructs the multi-task learning framework that allows both privacy and interaction for the subtasks. All the above methods are supervised based and their performance highly relies on the abundant in-domain labeled data. However, as a fine-grained task, the annotation of labeled data in ABSA requires expert linguistic knowledge and is usually time- and resource- consuming [5], [6]. In addition, the amounts of annotated data are usually unbalanced across domains. For example, the hot research fields like restaurant reviews often contain more annotated data than the cold ones like online university comments.

To alleviate the data deficiency in E2E-ABSA, recent attempts are towards the unsupervised domain adaptation technique. The basic idea is to transfer the common knowledge from labeled source data (\mathcal{D}^S) to unseen target test data (\mathcal{D}^T) given some unlabeled target data (\mathcal{D}^U). Since ABSA is a fine-grained task with word-level annotations, it is necessary to conduct word-level domain adaptation. Then a problem naturally arises: many source words like *pizza* and *delicious* in \mathbb{R} (Restaurant) are domain-specific, and they are unable to be directly transferred to \mathbb{L} (Laptop). Fig. 1 presents the proportions of domain-specific aspect terms in \mathcal{D}^S under different transfer pairs. In distant transfer pairs like $\mathbb{R} \rightarrow \mathbb{L}$, more than 80% aspect terms in \mathcal{D}^S have not appeared in \mathcal{D}^T . Even in a close pair $\mathbb{L} \rightarrow \mathbb{D}$, the proportion of domain-specific aspect terms is more than 50%. In contrast, in normal in-domain ABSA settings, where train and test data are in the same domain, this proportion is often less than 20%. Due to the existence of domain-specific source words, for a model trained on \mathcal{D}^S , it is hard to adapt it to the target data \mathcal{D}^T .

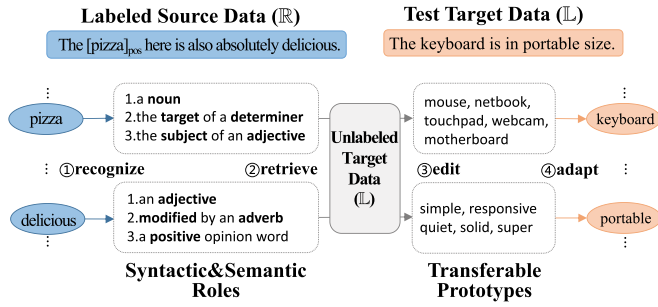


Fig. 2. Illustration of aligning domain-specific words across domains by retrieving and editing transferable prototypes.

Current domain adaptation studies try to address this problem in three ways. (1) Associating source words with specific pivot words₁. Early methods[8], [9] use common opinion seed (e.g., *good*) and pre-defined rules (e.g., *good*→*amod*→*NN*) as pivot to correlate domain-specific words in different domains. More recent studies [10], [11] manually annotate all opinion terms in reviews and design neural models to capture word-opinion relations. However, not all domain-specific words are accompanied by pivot words, and it is impossible to collect a complete pivot word set. (2) Setting adaptive weights for source words [12], [13]. By assigning higher weights to domain-invariant source words and lower weights to domain-specific ones, these methods can reduce the impact of untransferable source words. However, by this means, the domain-specific words in \mathcal{D}^S contribute little to the training process even if they are correctly labeled. (3) Generating pseudo labels for \mathcal{D}^U and then training the target classifier accordingly [14], [15]. Since domain-specific words in \mathcal{D}^S rarely appear in \mathcal{D}^U , their embedded knowledge cannot be captured by the generated pseudo labels. In summary, the untransferability of domain-specific words remains to be the biggest issue that prevents existing methods from achieving satisfactory performance.

In this paper, we propose a novel retrieve-and-edit domain adaptation method for the E2E-ABSA task. Our motivation is to retrieve appropriate target words that play similar syntactic and semantic roles with source words, and regard them as *transferable prototypes* to guide domain adaptation. By editing the source words with corresponding prototypes, we directly enhance their transferability and actively reduce the domain discrepancy. An example in Fig. 2 shows how to align domain-specific source words like *pizza* and *delicious* with unseen target words like *keyboard* and *portable*. We summarize our method into four steps.

- ① *Recognize*: For a given source word in \mathcal{D}^S , we first recognize its syntactic and semantic roles to represent it completely. Specifically, four different roles are used: part-of-speech tag, dependency relation, linguistic meaning, and sentiment polarity. For example, we can recognize *delicious* as an adjective modified by an adverb and a positive opinion word.

¹ Pivot words are words that behave in the same way for discriminative learning in both domains [7].

- ② *Retrieve*: Once obtaining the roles of a source word (e.g., *pizza*) in \mathcal{D}^S , we can retrieve several target words (e.g., *mouse* and *netbook*) from \mathcal{D}^U that play similar syntactic and semantic roles. Target words with high similarity scores will be considered as transferable prototypes to guide domain adaptation.
- ③ *Edit*: Given the source words from \mathcal{D}^S and the prototypes from \mathcal{D}^U , we then edit their representations accordingly. Concretely, we design a gating mechanism for source words to absorb the transferable knowledge carried by prototypes. As a result, the transferability of source words is directly enhanced and the domain discrepancy is actively narrowed. Moreover, we further make the editing process compatible with both pre-trained word embeddings like Word2Vec and pre-trained language encoders like BERT.
- ④ *Adapt*: Based on the edited words, we develop an end-to-end framework to jointly accomplish cross-domain aspect term extraction and aspect-level sentiment classification. Consequently, given training data from \mathcal{D}^S like *pizza* with *positive* polarity, our method can efficiently extract unseen target aspect terms in \mathcal{D}^T like *keyboard* and related *positive* sentiment polarity.

We conduct extensive experiments on four datasets with ten transfer pairs in total. The results show that, after introducing transferable prototypes, our method significantly outperforms the state-of-the-art baselines by a large margin.

II. RELATED WORK

In this section, we first review the literature for aspect based sentiment analysis, and then focus on the highly relevant work on domain adaptation and retrieve-and-edit methods.

A. Aspect-Based Sentiment Analysis

Most existing studies treat ABSA as a two-step task and develop separate methods for aspect term extraction (ATE) [8], [16]–[31], and aspect-level sentiment classification (ASC) [32]–[45], respectively. To obtain the complete ABSA performance, results from two steps must be merged together in a pipeline manner. In this way, the relation between ATE and ASC is totally neglected, and the errors from upstream ATE will be propagated to the downstream ASC. The overall performance of ABSA is not promising for pipeline methods.

Recently, some supervised studies attempt to solve ABSA in an end-to-end manner where ATE and ASC are performed simultaneously, i.e., E2E-ABSA. Methods for E2E-ABSA fall into two types: collapsed tagging [1], [2], [46], [47] and joint training [3], [4], [48], [49]. The former combines the labels of ATE and ASC to construct collapsed labels. The subtasks need to share all trainable features without distinction, which is likely to confuse the learning process. Meanwhile, the latter constructs a multi-task learning framework where each subtask has independent labels and can have shared and private features. This allows the interactive relations among different subtasks to be modeled explicitly. Generally, joint training methods can

achieve better performance than both separate and collapsed tagging methods, and have become the new paradigm for ABSA.

Since the supervised methods highly depend on abundant domain-specific training data, they can hardly scale across the domains where labeled data is absent. Different from the above work, we focus on the unsupervised domain adaption for E2E-ABSA where in-domain labeled data is not available.

B. Domain Adaptation

Many domain adaptation methods have been proposed to solve coarse-grained tasks like text classification [7], [50], [51]. The basic idea in these tasks is to transfer only pivot words, which does not fit ABSA well since it requires fine-grained transfer for every labeled word. There have been a few attempts to cross-domain ABSA, which mainly fall into three types.

First, modeling the relation between source and pivot words. Early researches use common opinion seeds and pre-defined dependency link rules to build manual features [52] or conduct bootstrapping [9]. Due to the incompleteness of seeds and the inflexibility of rules, they often produce inferior performance. Subsequent studies [10]–[12], [53] manually annotate all opinion terms in reviews and design trainable neural models to capture the relations via multi-task learning. However, they still cannot correlate all domain-specific words with pivot words.

Second, adaptively reweighting source words [12], [13]. The basic idea here is to reduce the impact of domain-specific words in word-level domain adaptation. Therefore, they choose to assign higher weights to domain-invariant words and lower weights to domain-specific words. In this way, these methods could indirectly reduce the discrepancy across domains and hence improve the performance. However, they are equal to using a part of source training data only, and the expert annotations of domain-specific words in \mathcal{D}^S are underutilized.

Third, generating pseudo labels for unlabeled target data to train the task classifier [14], [15]. In this way, more target knowledge can be captured since these methods can be trained on in-domain data. Specifically, the pseudo labels can be derived either from pre-defined rules or from a model trained on source annotations using semi-supervised methods. However, in both ways, the pseudo labels only contain domain-invariant knowledge and have no benefit for transferring domain-specific words.

Unlike all the aforementioned methods, we propose a novel retrieve-and-edit domain adaptation method. By retrieving transferable prototypes and editing source words accordingly, we directly enhance the transferability of domain-specific words and narrow the domain discrepancy. In our previous study [54], we propose SynBridge and SemBridge which supplement syntactic and semantic information for domain adaptation, but they do not suit the E2E-ABSA task for several reasons. Firstly, they are designed for a different task, i.e., cross-domain aspect term extraction without polarity prediction. Secondly, their supplemental information ignores the polarity of words, which may be harmful to E2E-ABSA. Thirdly, they are only compatible with static Word2vec embeddings and unable to make use of the powerful pretrained language models like BERT. Hence their

performance for E2E-ABSA is limited as we will show in the experiments.

C. Retrieve-and-Edit Methods

Retrieve-and-edit methods focus on retrieving prototypes (or templates) according to certain metrics and then editing them to guide the learning process. The idea of prototypes originates from information retrieval (IR) approaches for sentence matching tasks like response generation [55], [56]. They aim to retrieve a related sample from the dataset as the counterpart of the input sample. More recently, several studies shed new light on this domain by deeply fusing prototypes with neural networks. Many of them use the task-dependent metrics [57], [58], common metrics such as Jaccard similarity [59]–[61], or existing tools like Lucene [60] to retrieve prototypes, and then input the prototypes into a neural model for generating outputs. [6], [62] follow another line, where the prototype is generated using a pre-trained translation or language model.

Different from the above prototypes, we retrieve prototypes via syntactic and semantic roles and adapt them for domain adaptation.

III. METHODOLOGY

In this section, we first introduce the domain adaptation problem for aspect based sentiment analysis. We then illustrate the proposed retrieve-and-edit method in detail.

A. Problem Statement and Model Overview

Given a review $x = \{x_1, \dots, x_n\}$, we formulate E2E-ABSA as a sequence tagging task that aims to jointly predict two tag sequences $ya = \{ya_1, \dots, ya_n\}$ and $ys = \{ys_1, \dots, ys_n\}$ for aspect term extraction and aspect-level sentiment classification, respectively. Each $ya_i \in \{B, I, O\}$ denotes the *beginning of*, *inside of*, and *outside of* an aspect term, while each $ys_i \in \{positive, neutral, negative\}$ denotes the sentiment polarity towards each word. Table I presents an example of annotations for E2E-ABSA. For convenience, we use y to denote $\{ya, ys\}$.

In this work, we focus on the unsupervised domain adaptation for E2E-ABSA, 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 $\mathcal{D}^T = \{(x_j^T)\}_{j=1}^{N_T}$.

We propose a novel model for E2E-ABSA which retrieves and edits prototypes to actively reduce the domain discrepancy. Specifically, our model consists of four steps: recognizing syntactic and semantic roles, retrieving prototypes via syntactic and semantic roles, editing words with prototypes, and jointly extracting aspect and classifying sentiment. The first three steps are designed for enhancing the transferability of source data, and the last step performs a joint training for two subtasks in E2E-ABSA based on the enhanced data.

TABLE I
AN EXAMPLE OF ANNOTATIONS FOR END-TO-END ASPECT BASED SENTIMENT ANALYSIS

Label y	Review x	The	pizza	here	is	also	absolutely	delicious
	Aspect Term Label $ya \in \{B, I, O\}$	O	B	O	O	O	O	O
	Aspect Sentiment Label $ys \in \{positive, neutral, negative\}$	-	positive	-	-	-	-	-

B. Recognize Syntactic and Semantic Roles

In natural language, linguistic expressions are rich and flexible yet they own some common properties. For example, *pizza* and *keyboard* both have the same part-of-speech (POS) tag *NN*, and *good* and *wonderful* convey similar semantics. Such commonalities can enhance the transferability across domains, which inspires our study. To characterize the commonalities of words from different domains, we present to recognize their syntactic and semantic roles in the sentence.

- *Part-of-speech tag (syntactic)*: Although domain-specific words in different domains express diverse semantic meanings, their POS tags are relatively fixed and enumerable. Therefore, for a source word x_i like *pizza*, we first recognize its POS tags (e.g., *NN*) according to the parsing results of the review sentence. Then we use a one-hot vector $\mathbf{v}_{pos}^S \in \mathcal{R}^{N_{pos}}$ to represent its POS tag, where N_{pos} is the number of tag types (we here temporarily omit the subscript i for convenience).
- *Dependency relation (syntactic)*: Similar to POS tags, dependency relations are also shared by words in different domains. However, it is more difficult to describe the involved relations completely since a word may correlate with several other words with different relation types and directions. Therefore, we present a novel data structure to encode dependency information by grounding them into involved words. For a source word x_i , we use a multi-hot vector $\mathbf{v}_{dep}^S \in \mathcal{R}^{N_{dep}}$ to represent its dependency relation(s), where N_{dep} are the number of relation types. Specifically, we merge all relations involved with x_i regardless of the direction (i.e., being the governor or dependent). This simplification almost has no side effects since if a word has a *NN* tag and *det* relation, it must be the governor.
- *Linguistic meaning (semantic)*: Although the distribution of words varies from one domain to another, many common words like determiners and prepositions are shared by almost all domains. For those common words, they can be retrieved for each other with similar linguistic meanings. Therefore, for a source word x_i , we lookup its word embeddings \mathbf{v}_{lin}^S with a pre-trained embedding matrix (e.g., Word2Vec).
- *Sentiment polarity (semantic)*: To classify the sentiment polarity towards a certain aspect term, it is necessary to distinguish the polarity of related opinion terms. However, positive and negative words (e.g., *good* and *bad*) are often close in the above three roles. Therefore, for a source word x_i , we further resort to external sentiment lexicons to assign one-hot polarity vectors \mathbf{v}_{plr}^S to represent its polarity (i.e., being *positive*, *negative*, or *neutral*).

After recognizing, we now have four representations for a word v in \mathcal{D}^S : $\{\mathbf{v}_{pos}^S, \mathbf{v}_{dep}^S, \mathbf{v}_{lin}^S, \mathbf{v}_{plr}^S\}$. Similarly, we can obtain

$\{\mathbf{v}_{pos}^U, \mathbf{v}_{dep}^U, \mathbf{v}_{lin}^U, \mathbf{v}_{plr}^U\}$ and $\{\mathbf{v}_{pos}^T, \mathbf{v}_{dep}^T, \mathbf{v}_{lin}^T, \mathbf{v}_{plr}^T\}$ for the word in \mathcal{D}^U and \mathcal{D}^T , respectively.

C. Retrieve Prototypes via Syntactic/Semantic Roles

To actively reduce the domain discrepancy, we propose to retrieve transferable prototypes to enhance the transferability of words in \mathcal{D}^S . Unlike previous methods that construct information flows like *pizza* \rightarrow *good* \rightarrow *keyboard* with the help of annotated pivot words, we aim to construct a direct flow like *pizza* \rightarrow *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 words like *{mouse, netbook, touchpad}* in \mathcal{D}^U for *pizza* and directly improve its relatedness with *keyboard*. We call these supplementary words transferable prototypes and will retrieve them for all words in \mathcal{D}^S to guide domain adaptation. Existing methods for retrieving prototypes (e.g., Jaccard similarity and search engine) calculate similarities based on semantic meanings. They are not suitable for domain adaptation because domain-specific words in different domains are often far away from each other in the semantic space. To address this problem, we propose to retrieve transferable prototypes via syntactic and semantic roles.

Before starting, we filter the words in \mathcal{D}^U by frequency and only preserve head words appearing more than τ times. We regard these words as candidate prototypes and build a prototype bank \tilde{V} from \mathcal{D}^U accordingly. Then for a query word $v \in V^S$ (vocabulary of \mathcal{D}^S), we want to find a prototype term $\tilde{v} \in \tilde{V}$ that plays similar syntactic and semantic roles in the target domain. Specifically, four similarities are calculated as follows.

- *Part-of-speech similarity*: In \mathcal{D}^S , v can appear with various contexts. Notice that many words (e.g., *like*) are polysemous and can exhibit different POS tags with different contexts. Therefore, we choose to summarize the global usages $\langle \mathbf{v}_{pos}^S \rangle$ of v by merging its POS embeddings in all reviews where v appear in \mathcal{D}^S :

$$\langle \mathbf{v}_{pos}^S \rangle = \{\mathbf{v}_{pos,j=1}^S | \mathbf{v}_{pos,j=2}^S | \dots | \mathbf{v}_{pos,j=N_S}^S\} \quad (1)$$

where $|$ is the dimension-wise OR operation and N_S is the number of reviews in \mathcal{D}^S . Similarly, we can obtain $\langle \mathbf{v}_{pos}^{\tilde{V}} \rangle$ for \tilde{v} :

$$\langle \mathbf{v}_{pos}^{\tilde{V}} \rangle = \{\mathbf{v}_{pos,j=1}^{\tilde{V}} | \mathbf{v}_{pos,j=2}^{\tilde{V}} | \dots | \mathbf{v}_{pos,j=N_U}^{\tilde{V}}\} \quad (2)$$

We then calculate the POS similarity between v and \tilde{v} :

$$pos.sim(v, \tilde{v}) = cosine(\langle \mathbf{v}_{pos}^S \rangle, \langle \mathbf{v}_{pos}^{\tilde{V}} \rangle) \quad (3)$$

where $c(\cdot, \cdot)$ is the cosine similarity.

- *Dependency similarity*: Following the steps in calculating the part-of-speech similarity, we can obtain the global usages $\langle \mathbf{v}_{dep}^S \rangle$ for v and $\langle \mathbf{v}_{dep}^{\tilde{V}} \rangle$ for \tilde{v} , then calculate

the dependency similarity between v and \tilde{v} :

$$dep.sim(v, \tilde{v}) = cosine(\langle \mathbf{v}_{dep}^S, \langle \mathbf{v}_{dep}^{\tilde{v}} \rangle \rangle) \quad (4)$$

- *Linguistic similarity*: For a static embedding look-up table, the word vectors \mathbf{v}_{lin}^S of v and $\mathbf{v}_{lin}^{\tilde{v}}$ of \tilde{v} is not varied with different contexts. Therefore, we can simply calculate the linguistic similarity between v and \tilde{v} :

$$lin.sim(v, \tilde{v}) = cosine(\mathbf{v}_{lin}^S, \mathbf{v}_{lin}^{\tilde{v}}) \quad (5)$$

Obviously, for a word that frequently appears in both source and target domains, its most probable prototype is itself since the linguistic similarity is 1.0. In this way, we can effectively suppress noisy prototypes for domain-invariant words.

- *Polarity similarity*: Just like the linguistic similarity, we can calculate the polarity similarity between v and \tilde{v} :

$$plr.sim(v, \tilde{v}) = cosine(\mathbf{v}_{plr}^S, \mathbf{v}_{plr}^{\tilde{v}}) \quad (6)$$

By adding this, we can further distinguish positive and negative opinion words that usually have high scores with each other in the above three similarities.

After calculating the four different similarities, we can obtain the overall role similarity score between v and \tilde{v} :

$$r.sim(v, \tilde{v}) = pos.sim \times dep.sim \times lin.sim \times plr.sim, \quad (7)$$

Consequently, we can obtain a $r.sim$ score matrix $\mathbf{M}^S \in \mathcal{R}^{|V^S| \times |\tilde{V}|}$. After ranking, for v , we select the top-K words $\{\tilde{v}_k\}_{k=1}^K$ along with their $r.sim$ scores $\{\tilde{s}_k\}_{k=1}^K$ from the prototype bank, and regard them as transferable prototypes.

Following the way for \mathcal{D}^S , we also retrieve prototypes for \mathcal{D}^U and \mathcal{D}^T using \tilde{V} . In this way, source and target words with the same prototypes can be directly correlated to each other. During testing, a minor challenge for \mathcal{D}^T is that we cannot obtain $\langle \mathbf{v}_{pos}^T \rangle$ and $\langle \mathbf{v}_{dep}^T \rangle$ since the whole test data is unseen. To address this issue, we reuse the score matrix $\mathbf{M}^U \in \mathcal{R}^{|V^U| \times |\tilde{V}|}$ of \mathcal{D}^U . For a word in \mathcal{D}^T , we select prototypes according to \mathbf{M}^U if it has appeared in \mathcal{D}^U . Otherwise, we temporarily use the local \mathbf{v}_{pos}^T and \mathbf{v}_{dep}^T of the current test sample to calculate the part-of-speech and dependency similarities. The retrieval process is a one-time job for each transfer pair and often finishes in ten seconds.

D. Edit Words With Prototypes

For words in a given review $x = \{x_1, \dots, x_n\}$, we aim to edit their representations with corresponding transferable prototypes $\{\{\tilde{v}_{1,k}\}_{k=1}^K, \dots, \{\tilde{v}_{n,k}\}_{k=1}^K\}$ to actively enhance the transferability. Considering that there are two types of text representations, i.e., pre-trained word embeddings like Word2Vec and pre-trained language models like BERT, we here show how to edit them with prototypes, respectively.

1) *Edit Pre-Trained Word Embeddings*: Given a pre-trained word embedding lookup table $\mathbb{E} \in \mathcal{R}^{d_e \times |V|}$, we map x to a set of word vectors $\{e_1, \dots, e_n\} \in \mathcal{R}^{d_e \times n}$, where $|V|$ is the dictionary size, and d_e is the embedding dimension. Similarly, for prototypes, we also get a set of word vectors

$\{\{\tilde{e}_{1,k}\}_{k=1}^K, \dots, \{\tilde{e}_{n,k}\}_{k=1}^K\}$. For each x_i , we first aggregate its prototypes to a single prototype vector \tilde{p}_i according to the similarity scores:

$$\tilde{p}_i = \sum_{k=1}^K \tilde{s}_{i,k} \cdot \tilde{e}_{i,k}. \quad (8)$$

Notice that prototypes have two properties. (1) They are domain-invariant and should be preserved. (2) They can help extract domain-invariant information from e_i . Therefore, we propose to enhance the embedding e_i of the word x_i with its prototype vector \tilde{p}_i . Specifically, we first calculate a dimension-wise gate g_i :

$$g_i = \sigma(\mathbf{W}_1(e_i \oplus \tilde{p}_i)), \quad (9)$$

where $\mathbf{W}_1 \in \mathcal{R}^{2d_e \times 2d_e}$, σ is the Sigmoid function, \oplus is concatenation. We then scale the concatenated vector $e_i \oplus \tilde{p}_i$ with g_i and obtain the prototype-enhanced word representation $r_i \in \mathcal{R}^{2d_e}$:

$$r_i = g_i \odot (e_i \oplus \tilde{p}_i), \quad (10)$$

where \odot is element-wise multiplication. Hereafter, we denote the method using pre-trained word embeddings as **TransProto^W**.

2) *Edit Pre-Trained Language Model*: When it comes to pre-trained language models (PLMs) like BERT, the word vectors are contextualized after interacting with other words in the review. Since prototypes $\{\tilde{v}_{i,k}\}_{k=1}^K$ for x_i are several individual words, how to make them compatible with the language encoder is a non-trivial problem. To begin, we map $\{x_1, \dots, x_n\}$ and $\{\{\tilde{v}_{1,k}\}_{k=1}^K, \dots, \{\tilde{v}_{n,k}\}_{k=1}^K\}$ with the embedding layer inside PLM, and obtain two sets of word vectors. A key problem here is that, due to the sub-word tokenizer inside PLM, each word/prototype may be tokenized into several sub-words (e.g., x_1 may be mapped to two sub-word vectors e_1 and e_2). For example, given a word *appetizer* and its prototype *netbook*, the corresponding sub-words in BERT are $\{app, \#\#eti, \#\#zer\}$ and $\{net, \#\#book\}$, respectively. The inconsistency of sub-word tokenization between the words and their prototypes induces further manipulation for editing.

To address it, we first average the sub-word vectors of each prototype and recover the complete vectors $\{\{\tilde{e}_{1,k}\}_{k=1}^K, \dots, \{\tilde{e}_{n,k}\}_{k=1}^K\}$. Then for each x_i , we can also aggregate its prototypes to a single prototype vector \tilde{p}_i following Eq.8. But as illustrated before, a word x_i may also be tokenized to several sub-words. In this case, we further duplicate \tilde{p}_i for certain times to match the number of sub-words for x_i . Consequently, we can obtain the tokenized vector list $\{e_1, \dots, e_{n'}\}$ for the review x and corresponding manipulated prototype vector list $\{p_1, \dots, p_{n'}\}$, where $n' \geq n$ is the total length of review x after sub-word tokenization.

PLMs use transformers containing feed-forward networks and multi-head attentions to encode word vectors and generate contextualized representations. Therefore, we further present two schemas to edit word and prototype vectors, i.e., *edit-encode* or *encode-edit*. In the edit-encode schema, we first edit each e_i with p_i following Eq.9 and Eq.10, and generate an intermediate

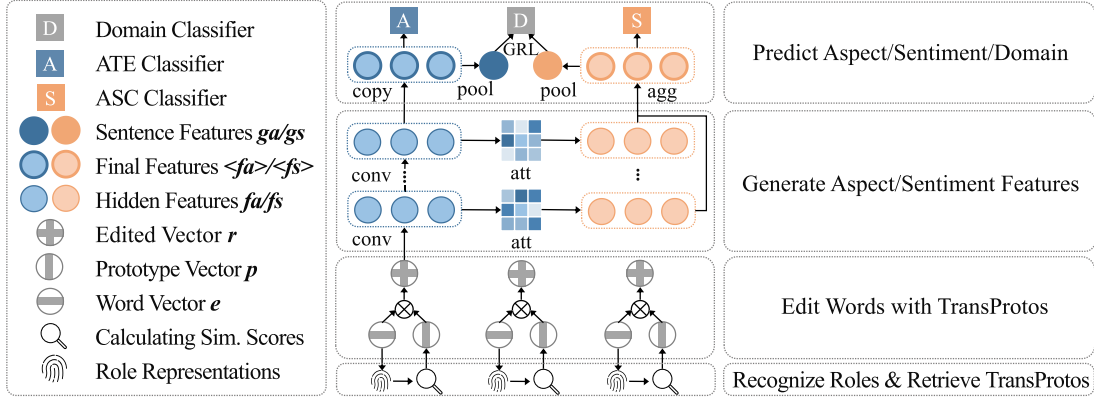


Fig. 3. End-to-end framework for cross-domain aspect based sentiment analysis.

prototype-enhanced representation \hat{r}_i . Then we send \hat{r}_i into the transformers and obtain the final word representations r_i :

$$\{r_1, \dots, r_n\} = PLM_Enc(\mathbf{W}_2 \times \{\hat{r}_1, \dots, \hat{r}_n\}), \quad (11)$$

where \mathbf{W}_2 here is used to reduce the dimensionality of r_i from $2d_e$ to d_e . In the encode-edit schema, we first treat e_i and p_i as two-way inputs towards transformers, and generate their representations separately:

$$\{\hat{e}_1, \dots, \hat{e}_{n'}\} = PLM_Enc(\{e_1, \dots, e_{n'}\}),$$

$$\{\hat{p}_1, \dots, \hat{p}_{n'}\} = PLM_Enc(\{p_1, \dots, p_{n'}\}). \quad (12)$$

After encoding, we edit \hat{e}_i and \hat{p}_i following Eq.9 and Eq.10 to generate the final word representations r_i . In practice, we use the encode-edit schema and will examine the difference between two schemas in experiments. Hereafter, we denote the method using pre-trained language models as **TransProto^B**.

E. Jointly Extract Aspect and Classify Sentiment

Based on the edited word representations, we propose an end-to-end framework for cross-domain aspect based sentiment analysis. The overall architecture is shown in Fig. 3.

1) *Generate Aspect/Sentiment Features*: Aspect based sentiment analysis consists of two subtasks, i.e., aspect term extraction (ATE) and aspect-level sentiment classification (ASC), and the key semantic information for two subtasks are intuitively different. For ATE, local N-gram features like determinizers (*the*) and conjunctions (*and*) are quite important for recognizing aspect terms. While for ASC, global dependency features from verbs (*enjoy*) and adjectives (*delicious*) to aspect terms are informative for classifying sentiment polarities. Therefore, we first extract local N-gram features $\mathbf{f}a_i \in \mathcal{R}^{d_f}$ for ATE using stacked convolutional layers:

$$\mathbf{f}a_i^l = ReLU(\mathbf{f}a_{i-c:i+c}^{l-1} * \mathbf{K}^{l-1} + b^{l-1}), \sim \mathbf{f}a_i^0 = r_i \quad (13)$$

where $\mathbf{K} \in \mathcal{R}^{d_f \times (2d_e \times ks)}$ is the kernel group, $ks = 2c + 1$ is the kernel size. We pad the left and right c positions with all zeros to guarantee that the output sequence has the equal length² n .

²Here we do not distinguish n and n' since they do not affect the learning process.

Based on $\mathbf{f}a_i$, we then calculate global sentiment features $\mathbf{f}s_i \in \mathcal{R}^{d_f}$ for ASC using the attention mechanism. Specifically, the attention score between the word x_i and every other word in each layer is calculated as follows:

$$\begin{aligned} word_score_{i,j}^l (i \neq j) &= (\mathbf{f}a_i^l)^T \times (\mathbf{f}a_j^l), \\ word_att_{i,j}^l &= \frac{\exp(word_score_{i,j}^l)}{\sum_{m=1}^n \exp(word_score_{i,m}^l)} \end{aligned} \quad (14)$$

To obtain $\mathbf{f}s_i$, we aggregate the features of other words w.r.t their attention scores:

$$\mathbf{f}s_i^l = \sum_{j=1 (j \neq i)}^n word_att_{i,j}^l \cdot \mathbf{f}a_j^l \quad (15)$$

2) *Predict Aspect/Sentiment/Domain*: After L layers of convolution and interaction, we can obtain the final word features for classification. For ATE, we set the final feature $\langle \mathbf{f}a_i \rangle = \mathbf{f}a_i^L$ since it has already included aspect information in different layers. For ASC, each $\mathbf{f}s_i^L$ only captures the interaction within each layer, thus we further aggregate them with an attention layer to generate the final feature $\langle \mathbf{f}s_i \rangle$:

$$\begin{aligned} layer_score_i^l &= \mathbf{W}_3 \times \mathbf{f}s_i^l, \\ layer_att_i^l &= \frac{\exp(layer_score_i^l)}{\sum_{m=1}^L \exp(layer_score_i^m)}, \\ \langle \mathbf{f}s_i \rangle &= \sum_{l=1}^n layer_att_i^l \cdot \mathbf{f}s_i^l. \end{aligned} \quad (16)$$

To obtain word-level aspect and sentiment prediction, we feed $\langle \mathbf{f}a_i \rangle$ and $\langle \mathbf{f}s_i \rangle$ to two classifiers, respectively:

$$\begin{aligned} \hat{y}a_i &= Softmax(\mathbf{W}_4 \times \langle \mathbf{f}a_i \rangle), \\ \hat{y}s_i &= Softmax(\mathbf{W}_5 \times \langle \mathbf{f}s_i \rangle). \end{aligned} \quad (17)$$

Besides word-level tagging, we further enhance the domain-invariance of word features via domain adversarial training (DAT). Specifically, we first aggregate $\langle \mathbf{f}a_i \rangle$ and $\langle \mathbf{f}s_i \rangle$ to global sentence representations $\mathbf{g}a$ and $\mathbf{g}s$, respectively:

$$\begin{aligned} \mathbf{g}a &= MaxPool(\langle \mathbf{f}a_{1:n}^L \rangle), \\ \mathbf{g}s &= MaxPool(\langle \mathbf{f}s_{1:n}^L \rangle). \end{aligned} \quad (18)$$

Then we add a Gradient Reversal Layer (GRL)[50] with the scale coefficient λ and train a domain classifier to distinguish the domain that ga and gs belong to:

$$\hat{y}_d = \text{Softmax}(\mathbf{W}_6 \times \text{MLP}(\text{GRL}_\lambda(\mathbf{ga} \oplus \mathbf{gs}))), \quad (19)$$

where \hat{y}_d is the domain prediction, \oplus is concatenation, and MLP contains L_D layers with ReLU activation. Optimizing the domain classifier will enhance its distinguishing ability and also encourage the aspect/sentiment features to be domain-invariant due to the reversed gradient.

3) *Training Procedure*: In training, only samples from \mathcal{D}^S have corresponding aspect and sentiment labels for word-level classification. The goal is to minimize the tagging loss for recognizing aspect terms and classifying sentiment polarities:

$$\mathcal{L}_{CLS} = - \sum_{\mathcal{D}^S} \sum_{i=1}^n \ell(\hat{y}_{a_i}, y_{a_i}) + \ell(\hat{y}_{s_i}, y_{s_i}), \quad (20)$$

where ℓ is the cross-entropy loss function.³ 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:

$$\mathcal{L}_{DOM} = - \sum_{\mathcal{D}^S \cup \mathcal{D}^U} \ell(\hat{y}_d, y_d), \quad (21)$$

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 framework is defined as $\mathcal{L} = \mathcal{L}_{CLS} + \mathcal{L}_{DOM}$. In testing, for each sample from \mathcal{D}^T , we first retrieve and edit its prototypes and then use the fixed framework to predict its aspect and sentiment. It is clear that there is no data leakage from \mathcal{D}^T in training, and the task setting is strictly inductive in this work.

IV. EXPERIMENT

In this section, we first present the experimental setup, then compare our proposed TransProto with the state-of-the-art baselines and investigate the impacts of components and hyperparameters.

A. Experimental Setup

1) *Datasets*: We use four English benchmark datasets from different domains: Restaurant (\mathbb{R}), Laptop (\mathbb{L}), Device (\mathbb{D}), and Service (\mathbb{S}). Specifically, \mathbb{R} contains reviews of the restaurant datasets from SemEval 2014, 2015, and 2016 ABSA tasks[63]–[65]. \mathbb{L} contains reviews of the laptop dataset from SemEval 2014 Task 4[63]. \mathbb{D} contains reviews for five different digital products including digital camera, cellular phone, MP3 player, and DVD player [66]. \mathbb{S} contains comments for web services like online universities [67]. The detailed statistics of datasets are presented in Table II.

Following previous studies[12], [13], [15], we construct 10 transfer pairs like $\mathbb{R} \rightarrow \mathbb{L}$ with the four datasets mentioned above but omit the pairs $\mathbb{D} \rightarrow \mathbb{L}$ and $\mathbb{L} \rightarrow \mathbb{D}$ since these two domains are very similar. For each transfer pair, we use the labeled training data from the source domain and unlabeled training data from

³When training ASC, only the predicted sentiment towards true aspect terms would be counted in loss.

TABLE II
THE STATISTICS OF DATASETS

Dataset	Domain	Sentences	Training	Testing
\mathbb{R}	Restaurant	3,900	2,481	1,419
\mathbb{L}	Laptop	1,869	1,458	411
\mathbb{D}	Device	1,437	954	483
\mathbb{S}	Service	2,153	1,433	720

the target domain to train the tagger. Then we use the labeled test data from the source domain as the development set to fine-tune hyperparameters and select best-performing checkpoints. Lastly, we evaluate the model on unseen test data from the target domain.

Settings: We pre-process each dataset by lowercasing all words. For recognizing syntactic roles, we use Stanford CoreNLP[68] for dependency parsing. There are $N_{pos}=45$ classes of POS tags and $N_{dep}=40$ classes of dependency relations in four datasets. For retrieving prototypes, we use the same *word2vec* embeddings as previous studies[10], [12], [15] to calculate the embedding similarity, and resort to two external sentiment lexicons[66], [69] to calculate the polarity similarity. The frequency threshold $\tau=5$, and the number of prototypes $K=10$. For editing words in TransProto^W, we use the *word2vec* embeddings mentioned above to generate static word vectors and set $d_e=100$. For editing words in TransProto^B, we use BERT-Cross[70] to generate contextual word vectors and set $d_e=768$.

The rest hyper-parameters are tuned on the development set. In joint training, the kernel size $ks=3$, the number of convolution layers $L=4$, the number of MLP layers in domain classifier $L_D=3$, the scale coefficient of GRL $\lambda=0.1$, and dropout [71] is applied to convolution layers' outputs with the probability 0.5. For TransProto^W/TransProto^B, we set $d_f=256/768$, and train the model for 50/15 epochs using Adam optimizer [72] with the learning rate $1e-4/3e-5$ and batch size 8 in a 1080Ti GPU, respectively.

Evaluation: We report F1-scores for both ATE and ABSA. To compute ATE-F1, the prediction would be considered correct if it exactly matches the label span of aspect terms. For ASBA-F1, the result for an aspect term would be considered correct only when both ATE and ASC results are correct. Following previous studies[3], [4], if an aspect term contains multiple words, we use the predicted sentiment of the first word as the ASC result. We run the experiments five times with random initialization and report the averaged results. The checkpoint achieving the maximum ABSA-F1 on the development set is used for evaluation on the test set.

B. Compared Methods

According to the input embeddings, we separate all baselines into Word2Vec- (Type-I) and BERT-based (Type-II) methods to conduct fair comparison for TransProto^W and TransProto^B. Below are baselines in Type-I.

- *Hier-Joint* [14]: first generates auxiliary labels for both source and target data according to manually designed

TABLE III

COMPARISON OF DIFFERENT METHODS FOR DIFFERENT TRANSFER PAIRS. THE BEST SCORES ARE IN **BLUE** AND THE SECOND BEST ONES ARE IN **ORANGE**. ALL RESULTS OF TRANSPROTO ARE AVERAGE SCORES OF 5 RUNS WITH RANDOM INITIALIZATION, AND THOSE WITH † AND ‡ ARE SIGNIFICANTLY BETTER THAN SEMBRIDGE AND UDA-CROSS ($p < 0.01$) BASED ON ONE-TAILED UNPAIRED T-TEST, RESPECTIVELY. FOR T-TEST, WE ALSO RERUN SOURCE CODES OF SEMBRIDGE AND UDA-CROSS FIVE TIMES

(a) Comparison results in terms of ABSA-F1.

Type	Model	Embedding	S→R	L→R	D→R	R→S	L→S	D→S	R→L	S→L	R→D	S→D	AVG.
I	Hier-Joint	Word2Vec	31.10	33.54	32.87	15.56	13.90	19.04	20.72	22.65	24.53	23.24	23.72
	RNSCN	Word2Vec	33.21	35.65	34.60	20.04	16.59	20.03	26.63	18.87	33.26	22.00	26.09
	AD-SAL	Word2Vec	41.03	43.04	41.01	28.01	27.20	26.62	34.13	27.04	35.44	33.56	33.71
	AHF	Word2Vec	46.00	44.49	44.58	35.62	33.79	35.95	34.03	32.98	38.77	39.34	38.55
	SynBridge	Word2Vec	46.75	49.25	49.50	25.48	24.70	29.39	38.78	31.93	41.77	44.18	38.17
	SemBridge	Word2Vec	47.98	51.21	50.54	27.24	25.60	28.08	39.29	32.23	42.62	44.71	38.95
II	BERT-Base	BERT	41.55	46.56	43.61	30.03	28.77	29.55	41.40	32.45	46.27	42.15	38.23
	BERT-Cross	BERT	57.74	54.72	28.70	35.43	36.18	14.43	51.68	40.63	56.20	53.63	42.93
	UDA-Base	BERT	56.20	53.17	55.46	32.89	40.83	44.62	43.27	37.78	47.90	53.22	46.53
	UDA-Cross	BERT	60.63	56.69	59.67	36.62	38.78	49.68	50.66	43.95	54.69	56.53	50.79
III	TransProto ^W	Word2Vec	50.82	53.04	52.94	32.12	31.58	30.28	42.05	32.71	43.37	40.98	40.99 [†]
	TransProto ^B	BERT	62.64	62.27	61.09	49.27	48.92	53.98	54.50	44.66	52.53	57.50	54.74 ^{†‡}

(b) Comparison results in terms of ATE-F1.

Type	Model	Embedding	S→R	L→R	D→R	R→S	L→S	D→S	R→L	S→L	R→D	S→D	AVG.
I	Hier-Joint	Word2Vec	46.39	48.61	42.96	27.18	25.22	29.28	34.11	33.02	34.81	35.00	35.66
	RNSCN	Word2Vec	48.89	52.19	50.39	30.41	31.21	35.50	47.23	34.03	46.16	32.41	40.84
	AD-SAL	Word2Vec	52.05	56.12	51.55	39.02	38.26	36.11	45.01	35.99	43.76	41.21	43.91
	AHF	Word2Vec	59.13	64.59	59.74	43.84	42.75	44.37	55.71	44.85	50.23	47.80	51.30
	SynBridge	Word2Vec	58.40	65.29	62.81	32.71	33.69	38.13	55.06	45.28	53.27	53.90	49.85
	SemBridge	Word2Vec	59.29	66.16	63.67	35.04	35.00	37.77	57.99	45.05	55.35	54.60	50.99
II	BERT-Base	BERT	60.95	60.60	61.73	34.65	37.93	45.69	54.97	49.07	53.71	54.54	51.38
	BERT-Cross	BERT	66.80	64.43	40.49	40.28	44.36	20.95	62.74	48.79	60.52	57.03	50.64
	UDA-Base	BERT	65.59	62.49	66.02	38.69	49.38	53.68	57.81	49.91	57.35	58.56	55.95
	UDA-Cross	BERT	68.40	66.10	67.49	41.17	46.04	55.88	64.96	53.27	60.39	59.72	58.34
III	TransProto ^W	Word2Vec	60.79	70.32	64.16	40.43	40.96	34.41	59.69	42.47	54.05	47.24	51.45
	TransProto ^B	BERT	71.66	75.33	69.38	57.52	60.32	61.32	73.21	56.64	57.85	61.52	64.47 ^{†‡}

rules, then trains an LSTM to predict these auxiliary labels along with gold-standard labels.

- *RNSCN* [10]: manually annotates all opinion terms and designs a trainable recursive network to model the dependency relations between aspect and opinion terms.
- *AD-SAL* [12] proposes to assign higher weights to domain-invariant words and lower weights to domain-specific words in the source data. It further designs a memory and LSTM based network to model the interaction between the aspect and opinion terms.
- *AHF* [15]: first generates pseudo target labels with an adaptive mean teacher network, then trains a student network with both accurate source labels and pseudo target labels.
- *SynBridge* [54]: supplements syntactic roles like POS tags and dependency relations to word representations for cross-domain aspect term extraction. We augment it with collapsed labels for E2E-ABSA without modifying its model.
- *SemBridge* [54]: uses syntactic roles to find transferable semantic knowledge for cross-domain aspect term extraction. We also augment it with collapsed labels.

Below are baselines in Type-II.

- *BERT-Base*: finetunes vanilla uncased base BERT[73] to predict collapsed labels for cross-domain E2E-ABSA.
- *BERT-Cross*: post-trains the vanilla BERT on the merged corpus from Yelp and Amazon reviews[70], then finetunes it to predict collapsed labels.

- *UDA-Base* [13]: first trains the vanilla BERT with POS tags and dependency relations via self-supervision, then reweights source samples according to their transferability. Lastly, it finetunes the trained BERT to make predictions.
- *UDA-Cross* [13]: replaces the vanilla BERT in UDA-Base with the post-trained BERT from BERT-Cross.

C. Main Results

The comparison results of ABSA-F1 are shown in Table III. It is clear that our proposed model achieves a new state-of-the-art performance, where TransProto^W outperforms SemBridge (best in Type-I) by 2.04% and TransProto^B outperforms UDA-Cross (best in Type-II) by 3.95%.

When inspecting the general performance of baselines in Type-I and Type-II, we can easily conclude that BERT-based methods are better than Word2Vec-based methods for cross-domain E2E-ABSA. Since BERT is pre-trained with large-scale external corpus across many domains, it can encode domain-invariant features effectively.

Among the baselines in Type-I, Hier-Joint and RNSCN also use parsing results to construct auxiliary tasks, but they stay at using syntactic roles to link aspect terms to pivot words. Obviously, training sets cannot include all pivot words and not all aspect terms are accompanied by pivot words. Therefore, they only achieve inferior performance. Different from them, TransProto^W moves a step further and utilizes syntactic roles for retrieving transferable prototypes. By supplementing target knowledge for

all source words, TransProto^W is able to extract more domain-invariant features from source data thus performs better than Hier-Joint and RNSCN. AD-SAL chooses to reweight source words to indirectly reduce the domain discrepancy, but the annotation information for those low-weight domain-specific words is underutilized. AHF generates pseudo labels for unlabeled target data and trains the task classifier accordingly, thus can acquire more target knowledge in training. However, the pseudo labels are inaccurate since they still derive from the model trained on source annotations, and they only contain information for domain-invariant words. SynBridge and SemBridge also supplement syntactic roles to enhance the transferability of source data, but they neglect the polarity similarity and the interaction between two subtasks. Conversely, by recognizing both syntactic and semantic roles, TransProto^W can generate more accurate prototypes and take domain-specific source words into account. Moreover, the joint training framework further improves the performance.

Among the baselines in Type-II, BERT-Cross performs better BERT-Base after being post-trained on the cross-domain corpus. UDA-Base and UDA-Cross further get improvements after reweighting source words and training BERT with syntactic role prediction tasks. But similar to Hier-Joint and AD-SAL, in UDA, the utilization of syntactic roles is shallow and the source annotations are underutilized. In contrast, by incorporating the BERT backbone, TransProto^B achieves 54.74% in averaged ABSA-F1 and makes cross-domain E2E-ABSA more practical than ever before. In Table III(b), we also present ATE-F1 results and the observation is consistent with ABSA-F1, where TransProto^B outperforms the best baseline UDA-Cross by 6.13%.

We summarize the superiority of TransProto into three key points. (1) By retrieving and editing transferable prototypes, we actively reduce the domain discrepancy. Therefore, TransProto is able to capture more domain-invariant patterns in source training data and performs better in target test data than baselines. (2) In the joint training framework, we extract aspect and sentiment features separately according to the characteristics of corresponding tasks. Compared to baselines using collapsed labels, TransProto can explicitly model the interaction between ATE and ASC thus has a larger learning capability than baselines. (3) We introduce domain-adversarial training (DAT) to further narrow the domain gap. Different from baselines, we conduct DAT on global sentence features instead of local word features. Considering that many word segments appear in both source and target domains, they may puzzle the domain classifier since their domain labels are both 0 and 1. In contrast, the whole review sentence is definitely domain-specific and suitable for training the domain classifier.

D. Ablation Study

To validate the effectiveness of different components in TransProto, we conduct a series of ablation studies by removing or modifying each component and observe the performance variance. The results are shown in Table IV.

TABLE IV
ABLATION STUDY. THE SCORES DENOTE THE DECREASE OF AVERAGED ABSA-F1

Index	Variant	TransProto ^W	TransProto ^B
1	remove CNN	31.01	1.06
2	remove DAT	5.17	6.06
3	remove Prototype	1.92	1.99
4	only pos.sim	1.46	0.91
5	only dep.sim	1.55	0.19
6	only sem.sim	1.06	0.65
7	only plr.sim	3.86	0.29
8	use edit-encode	N/A	3.79

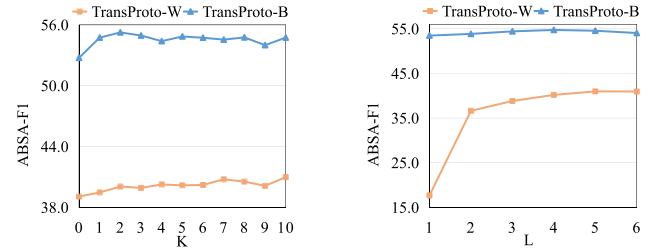


Fig. 4. Impacts of K and L .

In variant 1, we simplify the joint training framework by removing the convolutional layers in Section III-E1. The performance of TransProto^W drops dramatically since it almost loses all learning capability without convolutional layers. TransProto^B use BERT as the backbone, the decrease is relatively small but also observable. In variant 2, after removing domain-adversarial training (DAT) in Section III-E2, both models perform poorer than before due to the loss of knowledge in unlabeled target data. In variant 3, without prototypes, both models become weak in transferring domain-specific words and get performance decrease.

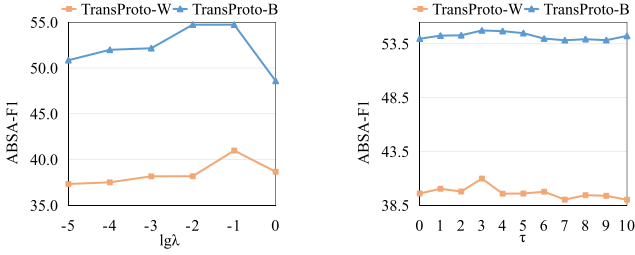
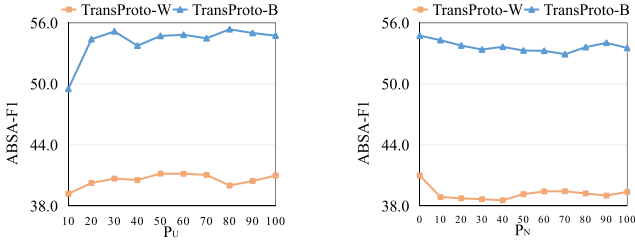
Variants 4~7 examine the importance of different syntactic and semantic roles by using each single similarity metric to retrieve prototypes. For TransProto^W, since static Word2Vec vectors are not informative enough, retrieving prototypes with a single metric only achieves inferior performance. For TransProto^B, the backbone BERT already captures most local POS information and contextual semantic meanings after being pre-trained, but it still lacks long-term dependency information and common sentiment knowledge. Therefore, supplementing dependency/polarity similarities can achieve comparable results.

In variant 8, we modify the editing schema in TransProto^B to the edit-encode schema, and get a large decrease in performance. The reason is that adding trainable modules in front of the transformers will disturb the pre-trained encoding process inside PLM.

E. Parameter Study

There are several key hyperparameters in TransProto, here we investigate their impacts by varying them in certain ranges and observing the performance trends.

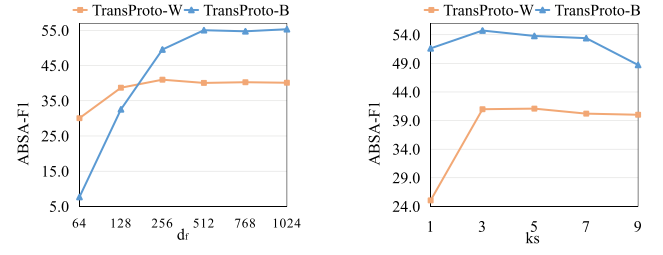
Fig. 4 shows the impacts of the number of transferable prototypes K (left) and the number of convolutional layers L (right). For K , when introducing more prototypes, the curve of

Fig. 5. Impacts of λ and τ .Fig. 6. Impacts of P_U and P_N .

TransProto^W is generally upward. This trend is reasonable since Word2Vec embeddings contain limited knowledge and more prototypes equal to more target information. Prototypes also bring improvements for TransProto^B, but only 1~3 prototypes are enough. The reasons are twofold. One is that PLMs already embed sufficient semantic knowledge and only need a few accurate prototypes to absorb target domain information. The other is that PLMs generate contextual representations via multi-head attention over words. In this case, the noise brought by latter prototypes with low similarity scores will be amplified and further deteriorates the generated representations. For L , stacking more convolutional layers means enlarging the learning capability, thus achieves better performance. Moreover, the curves begin to be flat (TransProto^W) or downward (TransProto^B) after 4 layers due to the redundancy of trainable parameters.

Fig. 5 shows the impacts of the coefficient of domain-adversarial training λ (left) and the frequency threshold in retrieval τ (right). For λ , too small values like $1e-5$ are not enough to align source and target domains effectively while too large values like 1 force the features to be meaningless. This result shows that simply forcing the model to transfer domain-specific words is not suitable for domain adaptation in E2E-ABSA. For τ , 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 $\tau=3$ is an appropriate choice.

Fig. 6 shows the impacts of the percentage of unlabeled data P_U (left) and the noise percentage of syntactic parsing P_N (right). For P_U , the performance is generally better when more unlabeled target data is used in prototype retrieval and domain-adversarial training. Moreover, around 70% (TransProto^W) or 80% (TransProto^B) unlabeled data is enough to achieve satisfactory performance. For P_N , we manually disturb the parsing results to observe the robustness of TransProto. Clearly, after introducing noises on parsing, the performance begins to degrade,

Fig. 7. Impacts of d_f and k_s .

but not by a large margin. TransProto has the ability to resist parsing errors for two reasons. First, beyond syntactic roles, we also incorporate linguistic and polarity similarities when retrieving prototypes. Second, the gating mechanism in editing can further filter useless syntactic information and maintain the quality of word representations.

Fig. 7 shows the impacts of the feature dimension of convolutional layers d_f (left) and the kernel size of convolutional layers k_s (right). For d_f , its value should match the dimension of input word representations (256 for TransProto^W and 768 for TransProto^B). A smaller value equals inadequate learning capability, while a larger value may cause over-fitting on source training data and deteriorate the cross-domain performance. For k_s , $k_s=1$ yields extremely poor performance for TransProto^W because the features are generated only by the current Word2Vec embeddings. The situation is slightly better for TransProto^B since the outputs of the BERT backbone are informative enough. Increasing k_s to 3 or 5 can widen the receptive field and remarkably boosts the performance. However, when further increasing k_s to 7 or 9, many irrelevant words are added as noises and thus deteriorate the performance.

V. DEEP ANALYSIS

In this section, we present an in-depth analysis for our TransProto including case studies and visualization.

A. Sample Prototype

In Table V, We first present several sample prototypes of TransProto from a loop of transfer pairs, i.e., $\mathbb{R} \rightarrow \mathbb{L}$, $\mathbb{L} \rightarrow \mathbb{S}$, $\mathbb{S} \rightarrow \mathbb{D}$, and $\mathbb{D} \rightarrow \mathbb{R}$. Here we first briefly review the content of data from different domains. \mathbb{R} mainly talks about foods and drinks in restaurants, \mathbb{L} is about the appearance and performance of laptops, \mathbb{S} contains reviews for online services like the financial consultants and university education, and \mathbb{D} is related to digital devices like MP3 and cameras.

Then we investigate the retrieved prototypes. For each transfer pair, we present three domain-specific words including an aspect term, an opinion term, and a context term along with their prototypes. In $\mathbb{R} \rightarrow \mathbb{L}$, TransProto correlates the domain-specific source aspect term *food* with typical target aspect terms like *machine* and *keyboard*. Besides, for the domain-specific opinion term *delicious*, its prototypes include both common positive opinion terms like *good* and typical target opinion terms like *clear*. Moreover, for the domain-specific context term *cook*, TransProto augments it with typical target context terms like

TABLE V
TOP-10 PROTOTYPES FROM DIFFERENT TRANSFER PAIRS. PROTOTYPES ARE
RANKED BY THEIR $r.sim$ SCORES

Pair	Term	Prototypes
\mathbb{R} ↓ \mathbb{L}	food	service,machine,product,keyboard,mouse,computer,netbook,touchpad,screen,performance
	delicious	good,amazing,wonderful,great,awesome,nice,perfect,beautiful,clear,excellent
	cook	run,use,play,load,connect,charge,plug,turn,go,look
\mathbb{L} ↓ \mathbb{S}	computer	system,server,university,program,school,phone,book,college,manager,room,office
	thin	small,high,short,quick,much,ok,full,average,plus,possible,other
	download	navigate,manage,teach,transfer,choose,execute,buy,digest,ask,write,create
\mathbb{S} ↓ \mathbb{D}	book	pocket,case,album,collection,device,artist,phone,cover,ipod,computer,explorer
	illiterate	useless,expensive,bad,poor,disappointed,loud,annoying,negative,difficult,sharp,low
	post	buy,say,get,purchase,look,go,take,see,return,try,know
\mathbb{D} ↓ \mathbb{R}	camera	item,restaurant,meal,plate,food,thing,glass,fish,scene,group,cuisine
	clear	beautiful,spectacular,pleasant,amazing,nice,impressive,good,easy,decent,helpful,attentive
	scan	open,check,choose,complete,write,visit,run,find,eat,clear,use

load and *connect*. Similar observations can be found in the rest transfer pairs, and the listed domain-specific source words are all enhanced by either typical target words or common domain-invariant words. Therefore, they can be easily transferred to the target domain under the guidance of transferable prototypes.

B. Case Study

To have a close look, we select ten samples from $\mathbb{S} \rightarrow \mathbb{R}$ for a case study and present the results of different methods in Table VI. Here we aim to investigate the impacts of transferable prototypes without the interference of powerful PLMs like BERT. Hence we choose AHF, TransProto^W without prototypes (denoted as NoProto), and the vanilla TransProto^W as the competitors.

Both baselines (AHF and NoProto) make use of the unlabeled target data and are equipped with domain-adversarial training. However, due to the large discrepancy across domains, they still fail to recognize correct target aspect terms in some cases.

1) The baselines cannot extract domain-specific aspect terms that have not appeared in the unlabeled target data. For example, *sweets* in S1, *curried casseroles* in S2, *mole sauce* in S3, and *meatball parm* in S4 only appear in the target test data, hence the baselines fail to extract them correctly. However, through prototypes, TransProto correlates *sweets* with *stocks*, *curried casseroles* with *e-pinions*, *mole sauce* with *materials*, and *meatball parm* with *equities*. As a result, TransProto^W can easily identify those aspect terms.

2) The baselines may encounter problems for extracting multi-word aspect terms completely. In the examples S5~S8, baselines only extract parts of aspect terms, e.g., *Chicken* in S5, *specials* in S6, *soup* in S7, and *cakes* in S8. In contrast, TransProto^W

Golden Standard : [teas] _{POS} [sweets] _{POS}
AHF The teas are great and all the sweets are homemade .
NoProto The [teas] _{POS} are great and all the sweets are homemade .
TransProto^W The [teas] _{POS} are great and all the [sweets] _{POS} are homemade .

Fig. 8. Detailed analysis of the example S1. Markers under words denote high attention scores w.r.t corresponding aspect terms, i.e., the blue triangle for *tea* and the orange square for *sweet*.

augments *teriyaki* with *e-trades*, *pasta* with *stocks*, *barley* with *cash*, and *crab* with *education*. Hence the entire aspect phrases are successfully extracted.

3) The baselines may wrongly extract non-aspect terms. Since AHF generates inaccurate pseudo labels, the task classifier may get misled in training and extracts non-aspect terms like *midtown area* in S9 and *location* in S10. In TransProto^W, only accurate source labels are used and the source data have been augmented with transferable knowledge, thus its ability in distinguishing aspect terms is stronger than the baselines.

In Fig. 8, we further present the detailed analysis of S1. Since the target aspect terms *tea* and *sweets* have never appeared in \mathbb{S} , AHF fails to recognize them correctly. For NoProto, it only extracts *teas* but ignores *sweets*. The reason maybe that *great* is a domain-invariant opinion word and it guides the extraction of *teas* via the attention mechanism. While for *sweets*, its corresponding opinion term *homemade* is a domain-specific word in \mathbb{R} thus is not captured by NoProto trained on \mathbb{S} . Moreover, NoProto assigns a high attention score to *sweets* because its embedding acts like a positive adjective. After introducing prototypes, TransProto^W successfully extracts both aspect terms and corresponding sentiment polarity. Specifically, *teas* is the prototype for many training words in \mathbb{S} like *discount* and *e-mails*, thus TransProto^W already captures its aspect characteristics after being trained on \mathbb{S} . While for *sweets*, prototypes like *pastas* and *desserts* correlate it with training words like *stocks* and *websites*. Similarly, *homemade* is linked to training words like *quick* and *instant*. Therefore, it is easy for TransProto^W to make correct predictions after absorbing knowledge from both labeled source data and unlabeled target data.

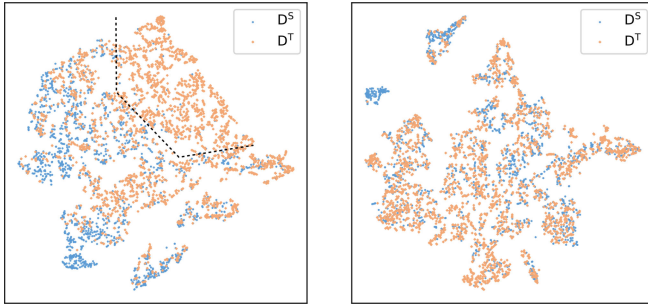
C. Visualization of Domain Discrepancy

To illustrate the effectiveness of TransProto, we visualize the final features for domain-specific aspect terms in \mathcal{D}^S and \mathcal{D}^T (i.e., not shared across domains) in $\mathbb{S} \rightarrow \mathbb{R}$. In Fig. 9(a), we remove transferable prototypes and domain-adversarial training from TransProto^W. Obviously, we can draw a boundary line for target features that are not covered by source training data, thus the corresponding target aspect terms are hard to be recognized. In contrast, almost all target points are accompanied by source points and there is no clear boundary in Fig. 9(b). This comparison clearly demonstrates that TransProto can actively reduce the domain discrepancy.

TABLE VI

CASE STUDY. THE LEFT COLUMN PRESENTS THE SELECTED EXAMPLES, AND THE THREE COLUMNS ON THE RIGHT DENOTE THE EXTRACTION RESULTS OF CORRESPONDING MODELS. WORDS IN RED ARE ASPECT TERMS WITH THE SUBSCRIPTS DENOTING THEIR SENTIMENT POLARITIES, AND “NONE” DENOTES THAT NO ASPECT TERMS ARE EXTRACTED

Examples	AHF	NoProto	TransProto ^W
S1.The [teas] _{POS} are great and all the [sweets] _{POS} are homemade.	NONE ✗	[teas] _{POS} ✗	[teas] _{POS} [sweets] _{POS}
S2.One caveat, some of the [curried casseroles] _{NEG} can be a trifle harsh.	NONE ✗	NONE ✗	[curried casseroles] _{NEG}
S3.The [fajita] _{NEG} we tried was tasteless and burned and the [mole sauce] _{NEG} was way too sweet.	NONE ✗	[fajita] _{NEG} ✗	[fajita] _{NEG} [mole sauce] _{NEG}
S4.I think the [meatball parm] _{POS} is good.	[parm] _{POS} ✗	NONE ✗	[meatball parm] _{POS}
S5.[Chicken teriyaki] _{NEG} had tomato or pimentos on top??	[Chicken] _{NEG} ✗	NONE ✗	[Chicken teriyaki] _{NEG}
S6.Both a number of the [appetizer] _{POS} and [pasta specials] _{POS} were amazing.	[specials] _{POS} ✗	[specials] _{POS} ✗	[appetizer] _{POS} [pasta specials] _{POS}
S7.The [mushroom barley soup] _{POS} is amazing.	[soup] _{POS} ✗	[mushroom] _{POS} ✗ [soup] _{POS}	[mushroom barley soup] _{POS}
S8.The [crab cakes] _{POS} are delicious and the [bbq rib] _{POS} was perfect.	[crab] _{POS} , [cakes] _{POS} , [bbq rib] _{POS} ✗	[cakes] _{POS} ✗ [bbq rib] _{POS} ✗	[crab cakes] _{POS} [bbq rib] _{POS}
S9.Really lovely [dining experience] _{POS} in the midst of buzzing midtown area .	[dining experience] _{POS} [midtown area] _{POS} ✗	[dining experience] _{POS} ✗	[dining experience] _{POS}
S10.This particular location certainly uses substandard [meats] _{NEG} .	[location] _{POS} [meats] _{NEG} ✗	NONE ✗	[meats] _{NEG}



(a) -Prototype, -DAT

(b) TransProto^W

Fig. 9. Visualization of domain-specific aspect terms.

TABLE VII
RETRIEVAL TIME FOR TRANSFER PAIRS

Transfer Pair	Retrieval Time (s)		
	Train	Dev	Test
S→R	4.05	5.06	7.95
L→R	3.98	3.25	7.95
D→R	3.70	3.62	7.95
R→S	3.42	6.81	4.26
L→S	2.99	2.76	4.26
D→S	2.75	3.00	4.26
R→L	3.20	6.39	2.63
S→L	2.85	4.21	2.63
R→D	2.48	5.75	2.44
S→D	2.18	3.53	2.44

D. Analysis on Computational Cost

The computational cost of TransProto consists of two parts. The first is the one-time retrieval process before training. In Table VII, we present the retrieval time for all transfer pairs. Obviously, the retrieval processes are very efficient and all finish within ten seconds.

The second is the training cost of the joint training framework. Here we run four 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 VIII. We can conclude that TransProto does not introduce much more computational cost than the top-performing baselines.

TABLE VIII
COMPUTATIONAL COST OF EACH METHOD

	Parameter Number	Runtime per Epoch
AHF	2.7M	11s
TransProto ^W	4M	12s
UDA-Cross	110M	150s
TransProto ^B	156M	162s

VI. CONCLUSION

In this paper, we propose a retrieve-and-edit domain adaptation method TransProto for aspect based sentiment analysis. By enhancing the transferability of source words with prototypes, TransProto can effectively transfer domain-specific words. Specifically, we first retrieve transferable prototypes from unlabeled target data via syntactic and semantic roles. Then we edit source words to absorb the transferable knowledge carried by prototypes and enhance their transferability. Lastly, we design a joint training framework to accomplish cross-domain aspect term extraction and aspect-level sentiment classification. Experimental results on four real-world datasets prove the effectiveness of our TransProto by comparison with the state-of-the-art baselines.

In the future, we plan to investigate other methods for retrieving and editing prototypes, and generalize the prototypes to more NLP tasks like named entity recognition and relation extraction.

ACKNOWLEDGMENT

The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

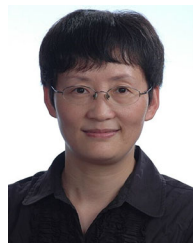
- [1] X. Li, L. Bing, P. Li, and W. Lam, “A unified model for opinion target extraction and target sentiment prediction,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6714–6721.
- [2] F. Wang, M. Lan, and W. Wang, “Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning,” in *Proc. Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.

- [3] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An interactive multi-task learning network for end-to-end aspect-based sentiment analysis," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 504–515.
- [4] Z. Chen and T. Qian, "Relation-aware collaborative learning for unified aspect-based sentiment analysis," in *Proc. Assoc. Comput. Linguistics*, 2020, pp. 3685–3694.
- [5] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Exploiting document knowledge for aspect-level sentiment classification," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 579–585.
- [6] Z. Chen and T. Qian, "Enhancing aspect term extraction with soft prototypes," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2107–2117.
- [7] J. Blitzer, R. T. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 120–128.
- [8] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.
- [9] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proc. Assoc. Comput. Linguistics*, 2012, pp. 410–419.
- [10] W. Wang and S. J. Pan, "Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 2171–2181.
- [11] W. Wang and S. J. Pan, "Transferable interactive memory network for domain adaptation in fine-grained opinion extraction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7192–7199.
- [12] Z. Li, X. Li, Y. Wei, L. Bing, Y. Zhang, and Q. Yang, "Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning," in *EMNLP-IJCNLP*, 2019, pp. 4589–4599.
- [13] C. Gong, J. Yu, and R. Xia, "Unified feature and instance based domain adaptation for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 7035–7045.
- [14] Y. Ding, J. Yu, and J. Jiang, "Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 3436–3442.
- [15] Y. Zhou, F. Zhu, P. Song, J. Han, T. Guo, and S. Hu, "An adaptive hybrid framework for cross-domain aspect-based sentiment analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 120–128.
- [16] A. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2005, pp. 339–346.
- [17] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2009, pp. 1533–1541.
- [18] F. Li *et al.*, "Structure-aware review mining and summarization," in *Proc. 23rd Int. Conf. Computat. Linguistics*, 2010, pp. 653–661.
- [19] K. Liu, L. Xu, and J. Zhao, "Opinion target extraction using word-based translation model," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2012, pp. 1346–1356.
- [20] Z. Chen, A. Mukherjee, and B. Liu, "Aspect extraction with automated prior knowledge learning," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 347–358.
- [21] M. Chernyshevich, "IHS R&D belarus: Cross-domain extraction of product features using CRF," in *SemEval Proc. 23rd Int. Conf. Computat. Linguistics*, 2014, pp. 309–313.
- [22] Z. Toh and W. Wang, "DLIREC: Aspect term extraction and term polarity classification system," in *SemEval Proc. 23rd Int. Conf. Comput. Linguistics*, 2014, pp. 235–240.
- [23] I. S. Vicente, X. Saralegi, and R. Agerri, "ELIXA: A modular and flexible ABSA platform," in *SemEval North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 748–752.
- [24] P. Liu, S. R. Joty, and H. M. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1433–1443.
- [25] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, "Improving opinion aspect extraction using semantic similarity and aspect associations," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2986–2992.
- [26] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, and M. Zhou, "Unsupervised word and dependency path embeddings for aspect term extraction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2979–2985.
- [27] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 616–626.
- [28] O. D. Clercq, E. Lefever, G. Jacobs, T. Carpels, and V. Hoste, "Towards an integrated pipeline for aspect-based sentiment analysis in various domains," in *WASSA, Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 136–142.
- [29] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Double embeddings and CNN-based sequence labeling for aspect extraction," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 592–598.
- [30] J. Yu, J. Jiang, and R. Xia, "Global inference for aspect and opinion terms co-extraction based on multi-task neural networks," *Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 168–177, 2019.
- [31] H. Luo, T. Li, B. Liu, B. Wang, and H. Unger, "Improving aspect term extraction with bidirectional dependency tree representation," *Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 7, pp. 1201–1212, 2019.
- [32] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proc. Assoc. Comput. Linguistics*, 2011, pp. 151–160.
- [33] S. Mohammad, S. Kiritchenko, and X. Zhu, "NRC- Canada: Building the state-of-the-art in sentiment analysis of tweets," in *North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 321–327.
- [34] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC- Canada-2014: Detecting aspects and sentiment in customer reviews," in *SemEval, Coling*, 2014, pp. 437–442.
- [35] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 49–54.
- [36] D. Vo and Y. Zhang, "Target-dependent twitter sentiment classification with rich automatic features," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1347–1353.
- [37] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence compression for aspect-based sentiment analysis," *Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2111–2124, 2015.
- [38] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 4068–4074.
- [39] P. Zhu and T. Qian, "Enhanced aspect level sentiment classification with auxiliary memory," in *Proc. 27th Int. Conf. Computat. Linguistics*, 2018, pp. 1077–1087.
- [40] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 547–556.
- [41] P. Zhu, Z. Chen, H. Zheng, and T. Qian, "Aspect aware learning for aspect category sentiment analysis," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 6, pp. 55:1–55:21, 2019.
- [42] B. Zhang, X. Li, X. Xu, K. Leung, Z. Chen, and Y. Ye, "Knowledge guided capsule attention network for aspect-based sentiment analysis," *Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2538–2551, 2020.
- [43] P. Lin, M. Yang, and J. Lai, "Deep selective memory network with selective attention and inter-aspect modeling for aspect level sentiment classification," *Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1093–1106, 2021.
- [44] X. Hou *et al.*, "Graph ensemble learning over multiple dependency trees for aspect-level sentiment classification," in *North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 2884–2894.
- [45] J. Dai, H. Yan, T. Sun, P. Liu, and X. Qiu, "Does syntax matter? A strong baseline for aspect-based sentiment analysis with roberta," in *North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 1816–1829.
- [46] M. Mitchell, J. Aguilar, T. Wilson, and B. V. Durme, "Open domain targeted sentiment," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1643–1654.
- [47] M. Zhang, Y. Zhang, and D. Vo, "Neural networks for open domain targeted sentiment," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 612–621.
- [48] H. Luo, T. Li, B. Liu, and J. Zhang, "DOER: Dual cross-shared RNN for aspect term-polarity co-extraction," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 591–601.
- [49] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," in *Proc. Assoc. Comput. Linguistics/IJCNLP*, 2021, pp. 2416–2429.
- [50] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, F. R. Bach and D. M. Blei, Eds., 2015, pp. 1180–1189.
- [51] H. Guo, R. Pasunuru, and M. Bansal, "Multi-source domain adaptation for text classification via distancenet-bandits," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7830–7838.

- [52] N. Jakob and I. Gurevych, "Extracting opinion targets in a single and cross-domain setting with conditional random fields," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 1035–1045.
- [53] W. Wang and S. J. Pan, "Syntactically meaningful and transferable recursive neural networks for aspect and opinion extraction," *Computat. Linguistics*, vol. 45, no. 4, pp. 705–736, 2019.
- [54] Z. Chen and T. Qian, "Bridge-based active domain adaptation for aspect term extraction," in *Proc. Assoc. Comput. Linguistics*, 2021, pp. 317–327.
- [55] Z. Ji, Z. Lu, and H. Li, "An information retrieval approach to short text conversation," 2014, *arXiv: 1408.6988*.
- [56] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.
- [57] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang, "Generating sentences by editing prototypes," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 437–450, 2018.
- [58] T. B. Hashimoto, K. Guu, Y. Oren, and P. Liang, "A retrieve-and-edit framework for predicting structured outputs," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 10073–10083.
- [59] J. Gu, Y. Wang, K. Cho, and V. O. K. Li, "Search engine guided neural machine translation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5133–5140.
- [60] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *Proc. Assoc. Comput. Linguistics*, 2018, pp. 152–161.
- [61] Y. Wu, F. Wei, S. Huang, Y. Wang, Z. Li, and M. Zhou, "Response generation by context-aware prototype editing," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7281–7288.
- [62] Y. Wang *et al.*, "Neural machine translation with soft prototype," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 6313–6322.
- [63] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval.*, 2014, pp. 27–35.
- [64] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval.*, 2015, pp. 486–495.
- [65] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 19–30.
- [66] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, W. Kim, R. Kohavi, J. Gehrke, and W. Du Mouchel, Eds., 2004, pp. 168–177.
- [67] C. Toprak, N. Jakob, and I. Gurevych, "Sentence and expression level annotation of opinions in user-generated discourse," in *Proc. Assoc. for Computational Linguistics*, J. Hajic, S. Carberry, and S. Clark, Eds., 2010, pp. 575–584.
- [68] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. Assoc. Comput. Linguistics*, 2014, pp. 55–60.
- [69] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2005, pp. 347–354.
- [70] H. Xu, B. Liu, L. Shu, and P. S. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 2324–2335.
- [71] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014, *arXiv:1412.6980*.
- [73] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.



Zhuang Chen (Graduate Student Member, IEEE) received the B.S. and M.Eng. degrees in electronic science and technology from the Huazhong University of Science and Technology, Wuhan, China. He is currently working toward the Ph.D. degree with the School of Computer Science, Wuhan University, Wuhan, China. He has authored or coauthored several papers in leading conferences and journals, such as ACL, EMNLP, and TKDD. His current research interests include information extraction, sentiment analysis, and domain adaptation. He was a Reviewer of ACL, EMNLP, and NAACL.



Tiejun Qian (Member, IEEE) received the B.S. degree in computer science from the Wuhan University of Technology, Wuhan, China, in 1991, and the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2006. She is currently a Professor with the School of Computer Science, Wuhan University, Wuhan, China. She has authored or coauthored more than 80 papers in leading conferences and top journals, including ACL, AAAI, SIGIR, CIKM, TKDE, TOIS, and TKDD. Her current research interests include text mining, web mining, and natural language processing. She is a Member of ACM and CCF. She was a Program Committee Member of many premium conferences, such as WWW, AAAI, IJCAI, ACL, EMNLP, ICDM, and CIKM.