Applied Intelligence
https://doi.org/10.1007/s10489-022-03434-2

SentATN: learning sentence transferable embeddings for cross-domain sentiment classification

Kuai Dai[1] · **Xutao Li**[1] · **Xu Huang**[1] · **Yunming Ye**[1]

Accepted: 22 February 2022

Abstract
Cross-domain Sentiment Classification (CDSC) aims to exploit useful knowledge from the source domain to obtain a high-performance classifier on the target domain. Most of the existing methods for CDSC mainly concentrate on extracting domain-shared features, while ignoring the importance of domain-specific features. Besides, these approaches focus on reducing the discrepancy of the source domain and target domain on the word-level. As a result, they cannot fully capture the whole meaning of a sentence, which makes these methods unable to learn enough transferable features. To address these issues, we present a Sentence-level Attention Transfer Network (SentATN) for CDSC, with two distinctive characteristics. Firstly, we design an efficient encoder unit to extract domain-specific features of a sentence. Secondly, SentATN provides a sentence-level adversarial training method, which can better transfer sentiment across domains by capturing complete semantic information of a sentence. Comprehensive experiments have been conducted on extended Amazon review datasets, and the results show that the proposed SentATN performs significantly better than state-of-the-art methods.

Keywords Cross-domain sentiment classification · Sentence-level · Domain-shared features · Domain-specific features

1 Introduction

Sentiment classification is a common task in Natural Language Processing (NLP), which is used to recognize user emotional tendencies (positive or negative) [1, 2]. Emotional tendencies can be widely applied in public opinion analysis and recommendation systems, creating huge commercial value. However, the effectiveness of the existing methods for sentiment classification is extremely dependent on sufficient training samples. Moreover, these methods cannot adapt to the tasks from different domains. Hence, such methods cannot meet the needs of practical applications. Recently, Cross-Domain Sentiment Classification (CDSC) is paid much attention and becomes a promising research direction [3, 4]. Different from vanilla sentiment classification, CDSC aims at leveraging knowledge from a source domain (with sufficient labeled data) to train an effective classifier in a related target domain (with few or no labeled data). The task is quite challenging and a sentiment classifier trained in a source domain is unlikely to work well when directly applied to the target domain [5], because of the inherent discrepancy between the two domains.

In the literature, many approaches are proposed for CDSC. These methods can be roughly classified into two taxonomies, namely the traditional methods and deep learning methods. The Structural Correspondence Learning (SCL) [6] is firstly proposed to learn the correlation between pivots and non-pivots by designing multiple pivots prediction tasks. However, this method relies heavily on manually selected pivots. Hence, it is not suitable for large datasets. Recently, methods based on Deep Neural Networks (DNN) emerge as a quite promising direction for CDSC. For example, Stacked Denoising Autoencoders (SDA) [7] is proposed to capture domain-shared hidden features. However, SDA-based network structure is too simple and coarse to capture enough domain-shared features. Yu et al. [8] propose a new method by designing two auxiliary tasks to learn transferable representation with the Convolutional Neural Network (CNN). This method improves the performance for CDSC, but it also needs expensive manual identification. A series of models [9, 10] based on Memory Network are put forward to automatically identity the pivots by utilizing the attention mechanism

✉ Xutao Li
lixutao@hit.edu.cn

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen (HITSZ), Shenzhen, China

Published online: 09 April 2022

and adversarial training. Recently, with the great success of pre-trained models in various NLP tasks, a model based on BERT [11] with a new pre-trained task is proposed. It can capture domain-shared features and performs better. Later, a Wasserstein based Transfer Network (WTN) [4] is developed for CDSC, which can better capture domain-shared features by adversarial training with the Wasserstein distance loss.

Although some progress has been made with deep learning techniques in CDSC, the existing methods works have two important limitations. On the one hand, all of the previously mentioned methods merely focus on extracting domain-shared features. Intuitively, the more domain-shared features the model extracts, the better performance will be achieved in the target domain. However, existing methods cannot differentiate the importance of the domain-specific features, which limits their performance for CDSC. For instance, the same word may have opposite meanings in different domains during the process of transfer learning. The model trained from the source domain cannot accurately identify the semantic information of the same word in the target domain, which will degrade its performance. A concrete sample is shown in Fig. 1. The word 'fast' in the electronics domain reviews denote the positive emotion, while the word 'fast' in the kitchen domain reviews represent the negative emotion. Thus, it is necessary for the model to identify the domain-specific words first and then to learn the domain-shared features on this basis. On the other hand, the existing methods are mainly based on word-level transfer learning. Consequently, they cannot fully exploit the semantics of sentences in a document and fail to learn enough transferable features, especially for long documents.

To address the two key drawbacks, we propose a Sentence-level Attention Transfer Network (SentATN) for CDSC. By designing an effective semantic encoder unit and a sentence-level transfer mechanism, SentATN can better transfer sentiment across domains. Specifically, we present an efficient word encoder named Domain-Information Encoder (DI-Encoder) to obtain the sentence representation, which enables the model to learn sentence-level semantics and capture domain-specific information. Different from the conventional word encoder, DI-Encoder can associate each word in a sentence with domain-specific information, which helps to identify the domain-specific knowledge in the process of transfer training. Then, SentATN adopts sentence-attention mechanism and sentence positional encoding to form hierarchical representation of a document. Finally, SentATN can learn the domain-shared features by leveraging the Coral Loss [12] to reduce domain discrepancy. We summarize our main contributions as follows:

- We present the DI-Encoder, an efficient encoder unit that can extract domain-specific features of a sentence during the process of transfer learning.
- To the best of our knowledge, SentATN is the first model that reduces the domain discrepancy on the sentence-level for Cross-domain Sentiment Classification, which allows the model to extract more transferable features.
- Experimental results on the extended Amazon review dataset demonstrate our method is superior to state-of-the-art methods in terms of both effectiveness and efficiency for CDSC.

The remainder of this paper is organized as follows: some background knowledge and key techniques of NLP are described in Section 2. And Our approach SentATN is detailedly introduced in Section 3. Then experimental

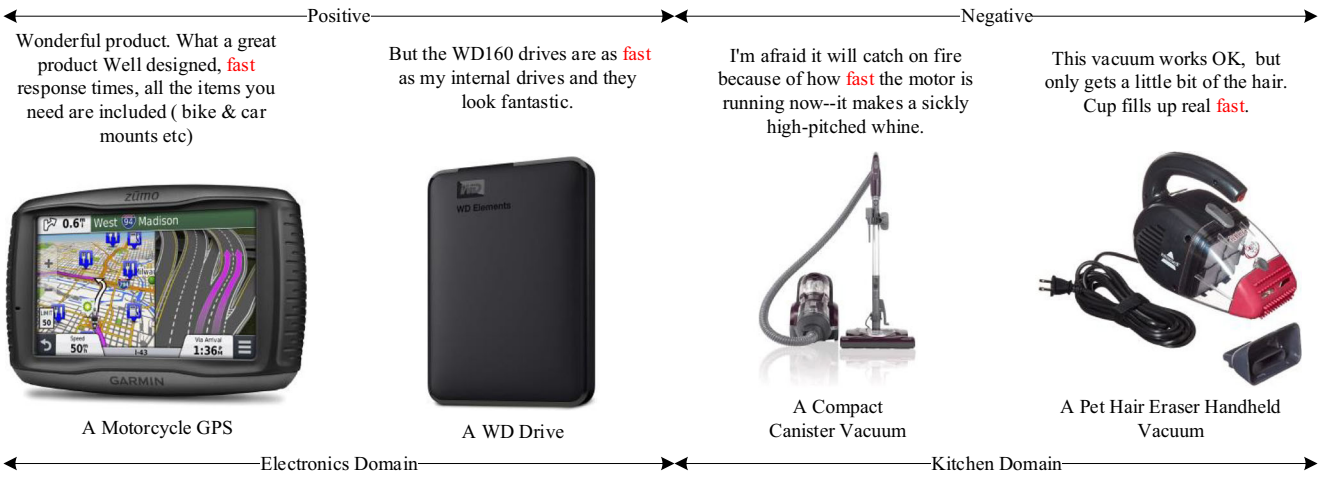

Fig. 1 This Figure shows four products with corresponding reviews on Amazon. The GPS and drive belong to the electronics domain, while the two different types of vacuums are in the kitchen domain. The word 'fast' in these reviews represents the opposite emotion across the two domains

results and analysis are shown in Section 4. Finally, the conclusion of our work is derived in Section 5.

2 Related work

In this section, we introduce the related work of Domain Adaption, Pre-trained Model, and Attention Mechanism, respectively.

2.1 Domain adaptation

Domain adaptation is proposed for transfer learning [13, 14], which aims at reducing the discrepancy between two domains [15–17]. The domain adaptation methods for CDSC can be roughly divided into two taxonomies, namely pivots-based methods [6, 8, 10, 18] and adversarial learning methods [11, 11, 19]. These pivots-based methods aim to learn domain-shared hidden states in the latent space with the aid of occurrence between pivots and non-pivots. They can be classified as traditional methods and deep learning-based methods. As a typical traditional method, Structural Correspondence Learning (SCL) [6] relies on manual selection for pivots and non-pivots. Differently, deep learning-based methods such as Stacked Denoising Auto-encoders (SDA) [7], and CNN-aux [8] capture pivots and non-pivots by automatically learning high-level feature representations of texts across domains. Although pivots recognized by deep learning methods are more accurate than traditional methods, their performance is still unreliable enough. As for the adversarial learning methods, they leverage the idea of Adversarial Generative Network (GAN) [20] to force neural networks to learn domain-shared knowledge by discriminating whether the given sample comes from the source domain or the target domain. For example, Ganin et al [18]. firstly apply the idea of GAN into CDSC and design a domain classifier, which can judge the current sample comes from which domain. The training procedure of this method is to fool the domain classifier to learn domain-shared features. Then, some adversarial learning-based methods like Adversarial Memory Network (AMN) [9], Hierarchical Attention Transfer Network (HATN) [10], and Interactive Attention Transfer Network (IATN) [19] are further proposed to strengthen ability for capturing domain-shared knowledge.

2.2 Pre-trained model

With the development of computing hardware, the pre-trained model with a very large amount of parameters becomes popular. Unlike the traditional model in NLP, the pre-trained model is trained on a very large-scale corpus and can more accurately capture semantic information. EMLo (Embedding from Language Model) [21], stacked by bidirectional LSTM units, is proposed to obtain high-level contextual representation. However, these LSTM units limit the feature extraction ability of EMLo due to the low training efficiency of LSTM. Then, GPT (Generative Pre-Training) [22] is proposed by replacing LSTM unit with Transformer [23] and obtains better performance. GPT is a one-directional language model and cannot capture the bi-directional semantic correlations. BERT (Bidirectional Encoder Representations from Transformers) [24], a Transformer-based bidirectional auto-encoder, fully takes advantage of Transformer's powerful feature extraction capability and high training efficiency to achieve state-of-the-art results in many NLP tasks. However, BERT is less effective in feature extraction of long text. To address the drawback, XLNet [25] is introduced by replacing Transformer with Transformer-XL [26] and shows significant improvement over long texts compared to BERT. In addition, there are many variants such as ALBERT [27], RoBERTa [28], which are all proposed to overcome drawbacks of the original BERT. The pre-trained models have become a hot topic in NLP.

2.3 Attention mechanism

Intuitively, different words in a sentence contribute different weights to semantic information. To consider this, the attention mechanism is proposed to assign weights to terms in the sentence and achieved good results in many NLP tasks such as machine translation [29], text classification [30–32]. Similarly, each sentence contributes differently to semantics of a whole document, while this cannot be effectively captured by a single attention mechanism. Thus, the hierarchical attention mechanism is proposed, divided into word attention mechanism and sentence attention mechanism. It extracts the interrelationships of words in sentences and sentences in documents respectively. A large number of experiments demonstrate that the hierarchical attention mechanism is superior to a single mechanism for long document tasks [30, 33, 34].

3 The proposed method

In this section, we introduce our method SentATN. Firstly, we introduce the problem and the formula symbols used in the paper in Section 3.1. Secondly, we introduce the overall architecture of the model in Section 3.2. Then, we introduce the training procedure of SentATN in Section 3.3. Finally, we provide an error boundary analysis of SentATN in Section 3.4 to show the superiority of our method.

3.1 Problem definition and notations

We assume that the notations D_s and D_t represent the source domain and target domain, respectively. D_s can be divided into two parts: labeled data D_s^l and unlabeled data D_s^u. For D_s^l, we have a set of samples $X_s^l = \{x_s^i\}_{i=1}^{N_s^l}$ and labels $\{y_s^i\}_{i=1}^{N_s^l}$. For D_s^u, we have $X_s^u = \{x_s^i\}_{i=N_s^l+1}^{N_s}$. Similarly, we have samples $X_t = \{x_t^j\}_{j=1}^{N_t}$ for unlabeled data D_t. The cross domain sentiment classification concentrates on training a classifier on D_s, which can obtain fairly good performance on D_t.

3.2 An overview of SentATN

The overall architecture of SentATN is shown in Fig. 2. Horizontally, SentATN can be divided into two modules: Encoding Module and Feature Extraction Module. Encoding Module and Feature Extraction Module are exploited to encode input documents and extract contextual information, respectively. Vertically, SentATN contains two subnetworks: S-NET and T-NET. S-NET and T-NET deal with the source domain data and target domain data, respectively, both of which share the parameter layers. Concretely, parameters of the DI-Encoder, the Sentence Positional Encoding, and the Sentence attention layer are shared by S-NET and T-NET. Sentences in the input documents will be processed into vectors by **DI-Encoder layer**, which leverages self-attention mechanism [23] to automatically identify the pivots and non-pivots in each sentence. Besides, a positional encoding is added to each corresponding sentence to learn the positional information in the document. Then, a GRU will encode these sentences to form contextual representation vectors. Furthermore, SentATN adopts **sentence attention mechanism** to compute the corresponding weight of each sentence in a document for document representation generation. Through the DI-Encoder layer and sentence attention mechanism, SentATN can produce the hierarchical structure document representation. Finally, SentATN employs the Coral loss [12] function to reduce discrepancy of the source domain and the target domain. Next, we will introduce these components of SentATN in detail.

3.2.1 DI-encoder layer

The overall architecture of DI-Encoder layer is shown in Fig. 3. We first propose to introduce the domain-specific information into the BERT unit to solve the problem that the domain-specific information in the sentence can be hardly captured.

Specifically, we design a new mechanism to add a fixed pattern sentence containing domain-specific information to each sentence in the document. For instance, we preface 'This is the kitchen domain' at the beginning of the sentence that comes from the kitchen domain. Transformer [23] is the most important unit in BERT, which is proposed to improve the training speed for machine translations. Transformer introduces a new attention mechanism called self-attention mechanism, which will calculate the weight of any two tokens in a sentence. Based on these weights, we can obtain the encoding vector of each word in the input sentence.

Assume that document x has m sentences and x_i denotes embeddings for the ith sentence. For each sentence x_i, we can get three new embeddings X_Q, X_K and X_V by multiply three same-size matrices W_Q, W_K and W_V respectively.

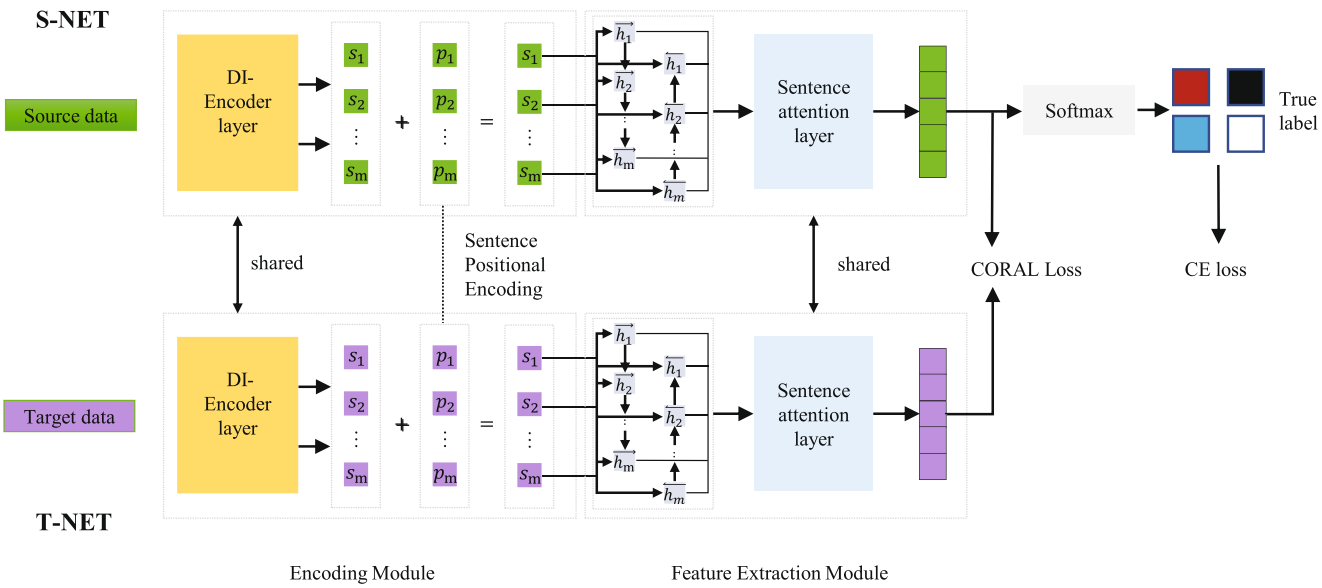

Fig. 2 The overall architecture of SentATN

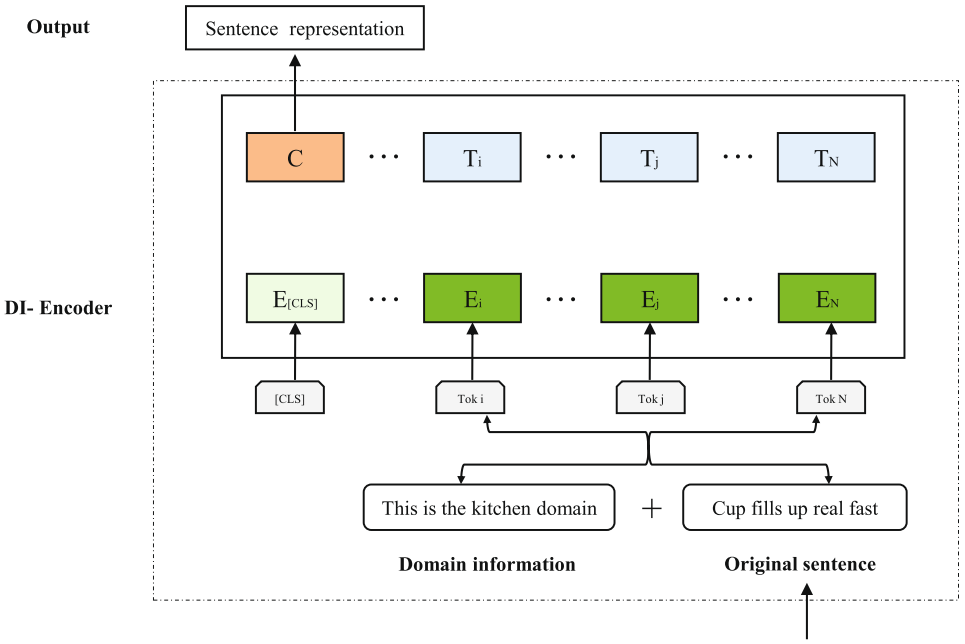

Fig. 3 The overall architecture of DI-Encoder Layer

The output of x_i after being processed by the self-attention mechanism is:

$$Att(x_i) = Att(X_Q, X_K, X_V) = softmax\left(\frac{X_Q X_K^{\mathrm{T}}}{\sqrt{d_k}}\right) X_V \tag{1}$$

where the d_k represents the dimension of X_Q.

We define an embedded vector x_D for the pattern sentence corresponding with x_i. For each input sentence, BERT adds [CLS] in front of the input sentence. We consider the first output vector of BERT as the representation of the input sentence. Then, we can define the process of the input sentence x_i by DI-Encoder layer:

$$s_i = f(Att(x_i \oplus x_D))[0], i \in [1, m] \tag{2}$$

where the f denotes the operation of the DI-Encoder and the [0] represents we select the first vector produced by the DI-Encoder.

3.2.2 Sentence positional encoding

Lots of previous studies have demonstrated that positional encodings are beneficial for text representation [23, 29, 35]. Positional encodings can be divided into fixed encodings and learned encodings depending on different tasks. To make better use of the location information of each sentence in the document, we designed a new positional encoding for each sentence. p^i represents the position vector of the ith sentence, and we add p^i to the embedded vector s^i for each sentence, i.e., $s^i = s^i + p^i, i \in [1, m]$, where p shared by the S-NET and T-NET is learned during the training process.

3.2.3 Sentence attention layer

Next, we show how to generate document vector representation. Firstly, we map each sentence x_i into vector s_i by the previously introduced DI-Encoder layer. Then, we adopt a bidirectional GRU to encode s_i and obtain corresponding hidden states. The bidirectional GRU contains the forward $\overrightarrow{GRU}$, which deals with the input document x from x_1 to x_m and a backward $\overleftarrow{GRU}$, which deals with the input document x from x_m to x_1:

$$\overrightarrow{h_i} = \overrightarrow{GRU}(s_i), i \in [1, m] \tag{3}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i), i \in [1, m] \tag{4}$$

The contextual representation h_i of ith sentence can be obtained by concatenating the forward hidden state $\overrightarrow{h_i}$ and the backward hidden state $\overleftarrow{h_i}$:

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{5}$$

Furthermore, we need to assign a weight α_i to the ith sentence x_i in the document, because different sentences contribute unequally to the semantic information of the entire document. The weight α_i is calculated as follows:

$$\mu_i = tanh(W_d h_i + b_d) \tag{6}$$

$$\alpha_i = softmax(\mu_i^{\mathrm{T}} w_d) = \frac{exp(\mu_i^{\mathrm{T}} w_d)}{\sum_{i=1}^{m} exp(\mu_i^{\mathrm{T}} w_d)} \tag{7}$$

W_d and w_d are the weight matrices shared by each sentence in the document, and b_d is the bias. $tanh$ is a non-linear activation function. W_d and w_d have similar yet different functions. W_d is utilized to project h_i into μ_i, while w_d is employed to project μ_i into α_i. Notably, α_i denotes the importance of the corresponding sentence in the document and the final document representation d is computed as:

$$d = \sum_{i=1}^{m} \alpha_i h_i \tag{8}$$

3.3 Model training

The loss function for the training of SentATN can be divided into two parts. One is for sentiment classification, and the other is for minimizing the discrepancy between the source domain and target domain. We leverage cross-entropy loss function for sentiment classification and deep Coral loss to reduce differences between the source domain and target domain. Next, we introduce loss functions for sentiment classification and domain discrepancy in details, respectively.

3.3.1 Sentiment classification

The sentiment loss L_{sen} (a normalised negative log-likelihood loss) is to minimize the cross-entropy for the labeled data X_s^l in source domain:

$$L_{sen} = -\frac{1}{N_s^l} \sum_{i=1}^{N_s^l} y_i ln \hat{y_i} + (1 - y_i) ln(1 - \hat{y_i}) \tag{9}$$

where $y_i \in \{0, 1\}$ and $\hat{y_i}$ is the sentiment prediction for the ith source labeled sample, respectively.

3.3.2 Reduce Domain Discrepancy

During the training procedure, we select the same quantity samples from the source domain and target domain for each batch and compute the Coral loss to reduce the discrepancy between two domains.

Specifically, Coral loss function measures the discrepancy between the second-order statistics (covariances) of the source features and target features. $X_s^l = \{x_s^i\}_{i=1}^{N_s^l}$ and $X_t^u = \{x_t^i\}_{i=1}^{N_t^u}$ ($N_s^l = N_t^u$) denote the labeled documents in the source domain and the unlabeled documents in target domain, respectively. Suppose M_s and M_t represent the feature covariance matrices. The Coral Loss can be computed as:

$$L_{coral} = \frac{1}{4d^2} ||M_s - M_t||_F^2 \tag{10}$$

where $||.||_F^2$ denotes the squared matrix Frobenius norm, and d represents the dimensions of the input data.

The feature covariance matrices of the source domain and target domain are computed as:

$$M_s = \frac{(X_s^{l\mathrm{T}} X_s^l - (1/N_s^l)(I^{\mathrm{T}} X_s^l)^{\mathrm{T}} (I^{\mathrm{T}} X_s^l))}{N_s^l - 1} \tag{11}$$

$$M_t = \frac{(X_t^{u\mathrm{T}} X_t^u - (1/N_t^u)(I^{\mathrm{T}} X_t^u)^{\mathrm{T}} (I^{\mathrm{T}} X_t^u))}{N_t^u - 1} \tag{12}$$

where I represents the column vector whose all elements are equal to 1. Finally, the whole objective function for SentATN is:

$$L = L_{sen} + L_{coral} + \lambda L_2 = L_{sen} + L_{coral} + \lambda ||W||^2 \tag{13}$$

The regularization term L_2 is used to avoid the overfitting on the training data, and λ is the coefficient for L_2 regularization. The W represents parameters in SentATN.

The entire training process of SentATN is summarized as Algorithm 1.

Algorithm 1 Training the SentATN for cross domain sentiment classification.

Input and Output:
The source domain data X_s^l and target domain data X_t^u , and the target domain
sentiment label y_t are the input and the output of the SentATN, respectively.
Setting:
Setting the batch size t and the learning rates l for the SentATN.
Setting λ for the objective function L.
for number of training iterations **do**
for iter in batchs **do**
batch data $\{x_s^i, y_s^i\}_{i=1}^t$, $\{x_t^i\}_{i=1}^t$ sampled from X_s^l and X_t^u.
$X, Y = (x_s^i, x_t^i)_{i=1}^t, \{y_s^i\}_{i=1}^t$ Assume x demotes a document in X, $x = \{s_i\}_{i=1}^m$
for each sentence s_i in x
$s_i = DI - Encoder(x_i) + p^i$
the document representation
$d = SentenceAttentionLayer(\{s_i\}_{i=1}^{i=m})$
Then we have $D_s, D_t =$
$SentenceAttentionLayer(X)$ $X = (D_s, D_t)$
Update the SentATN's parameter W
$L(X, Y) = L_{sen}(D_s, \{y_s^i\}_{i=1}^t)$
$+ L_{Coral}(D_s, D_t) + \lambda ||W||^2$
$W = W - l * \frac{\partial L(X,Y)}{\partial W}$
if the SentATN no longer performs better in five future consecutive epochs.
break

3.4 Error boundary analysis

Moving forward, we provide an error boundary analysis of SentATN. Assume that H is the hypothesis class. Given source domain D_s and target domain D_t, we have:

$$\forall h \in H, \epsilon_t(h) \leq \epsilon_s(h) + \frac{1}{2} dis(D_s, D_t) + C \tag{14}$$

where $\epsilon_t(h)$ and $\epsilon_s(h)$ represent the error prediction of SentATN in the source domain and target domain respectively. $dis(D_s, D_t)$ represents the $\mathcal{H}\Delta\mathcal{H} - divergence$ [36] between D_s and D_t, and it can measure the discrepancy in distribution between two domains. C is a constant.

Obviously, it is easy to minimize the error rate of source domain $\epsilon_s(h)$ by sentiment loss L_{sen}. For $dis(D_s, D_t)$, SentATN utilizes Coral loss to reduce the discrepancy between the two domains during the model training process, which enables SentATN to learn the domain-shared knowledge. For the third term, C is a constant and can be disregarded. Therefore, SentATN can directly reduce $dis(D_s, D_t)$. Through this inequality, SentATN can indeed reduce the upper error limit of D_t, which further illustrates the effectiveness of SentATN.

4 Experiments

4.1 Dataset

All experiments in this study are conducted on the extended Amazon reviews dataset [6], which has been widely used in cross-domain sentiment classification. The original dataset contains four domains: Books (B), DVD (D), Electronics (E), Kitchen (K). In each domain, there are 2000 labeled reviews (1000 positives and 1000 negatives). The original dataset is too small to comprehensively evaluate the accurate performance of models for CDSC. Then, Li et al. [10] extended the dataset by adding a new domain Video (V) and increased the sample size of each domain from 2000 to 6000. Now there are 3000 positive and negative samples in each domain. In addition, unlabeled data is provided for each domain, 9750 unlabeled reviews for B, 11843 for D, 17009 for E, 13856 for K, 30180 for V. The detailed Statistics of the extended Amazon reviews dataset are shown in Table 1. Among these five domains, we construct 20 cross-domain sentiment classification tasks. For instance, $B \rightarrow D$, where the letter B denotes the source domain and D denotes the target domain D. For a fair comparison, we follow Li et al. [10] and randomly choose 2800 positive and 2800 negative reviews from the source domain B as the training data, the rest 400 reviews from the source domain B as the validation data, and the 6000 reviews from the target domain D as the test data. To more accurately reflect the performance of our model, we use 5-fold cross-validation on each transfer task for all the experiments.

4.2 Hyperparameters setting

For the data preprocessing, the documents are split into sentences and each sentence is tokenized into words each by NLTK [37]. The number of sentences in each document and the length of each sentence are set to 20 and 25, respectively. The weights in networks are randomly initialized from a uniform distribution $U(-0.01, 0.01)$. We adopt ADAM [38] optimizer for training. The learning rate is set to 5×10^{-4} and the batch size is set to 40. We conduct early stopping based on the validation set during the training process. The training process of SentATN will be terminated if the model no longer performs better in the future consecutive five epochs. Specifically, the output vector dimension of DI-Encoder is 768, and the hidden size of GRU is 300, which is the same as the dimension of Google word2vec [39]. These experiments are conducted on a GPU server with a CPU i7-7700 at 3.60GHz, a 32G memory, an NVIDIA RTX1080Ti. Our model SentATN is implemented with the Pytorch architecture. The hyperparameters setting is summarize in Table 2.

4.3 Model comparisons

In this subsection, we compare SentATN with state-of-the-art models to validate the effectiveness of our model.

Table 1 Statistics of the extended Amazon reviews dataset including the number of training, testing, and unlabeled reviews for each domain

Domain	Train	Validation	Unlab	Avgslen	Avgsnum
Books	5600	400	9750	19.16	9.69
DVD	5600	400	11843	18.24	10.76
Electronics	5600	400	17009	15.63	7.76
Kitchen	5600	400	13856	14.61	7.21
Video	5600	400	30180	17.59	9.77

Avgslen and Avgsnum denote the average length of each sentence and the average number of sentences in a document, respectively

Table 2 Hyperparameters setting for SentATN

Optimizer	Documents splitting	Hyperparameters
Adam		lr = 5*1e-4
	sentence num=20	batch size = 40
	sentebce length=25	output size of DI-Encoder = 768
		hidden size of GRU = 300

Additionally, some variants of our model are also compared for analyzing the impacts of individual components.

4.3.1 State-of-the-art models

As for a comparison, we utilize the following state-of-the-art methods as baseline methods.

- **CNN-aux** [8]: Two auxiliary prediction tasks are designed to help the CNN encoder in the model generate a domain-share representation.
- **BiLSTMAtt** [40]: This method leverage Bi-direction LSTM and attention mechanism to learn domain-shared knowledge.
- **DAmSDA** [18]: This method utilizes Neural Network to learn the domain-shared features with the gradient reversal layer and a domain classifier.
- **Capsule** [41]: The Capsule Network is a variant of CNN, which has a natural advantage in capturing domain-shared knowledge with capsules.
- **AMN** [9]: Two memory networks in the model are proposed to help the model automatically recognize pivots and non-pivots.
- **HAN** [30]: HAN can capture more features through Hierarchical Attention for long documents.
- **HATN** [10]: HATN contains two subnetworks: ”P-net” and ”NP-net”. It can simultaneously capture pivots and non-pivots with hierarchical attention and better capture domain-shared features.
- **CCHAN** [42]: The model adds a 3-layer convolution structure into HAN, which allows this model to learn high-level features across the source domain and target domain.
- **WTN** [4]: WTN exploits wasserstein distance to compute the discrepancy between the source domain and target domain, which can better capture domain-shared features by adversarial training.

4.3.2 Variants of Our method

To analyze the influence of key components in SentATN, some variants of SenTATN are obtained by ablating them.

- **SentATN-DI**: The variant is obtained by removing Domain-Information encoding for each sentence based on the SentATN.
- **SentAN**: This variant can be acquired by removing the adversarial training.
- **SentAN-**: The variant is obtained by simultaneously removing the sentence positional encoding and adversarial training.
- **WordATN**: We use original BERT to generate each word embedding in each sentence. Then, we apply the word attention mechanism, which is the same as previous sentence attention mechanism to obtain sentence representation. The other part of the model is the same as SentATN.
- **WordATN***: The only difference from WordATN is that the 300-dimensional word2vec vectors are employed to initialize word embedding.

4.4 Experimental analysis

Table 3 reports the classification accuracies of all state-of-the-art methods and SentATN. From Table 3, we observe that the proposed model SentATN consistently achieves the best performance on most transfer tasks. First of all, the CNN-aux performs worst in these approaches due to the poor ability of CNN to capture long-range dependency. Besides, the CNN-aux relies on manual operations to identify pivots, which further degrade the training efficiency of this method. BiLSTMAtt performs slightly better than CNN-aux by 0.3% on average. However, BiLSTMAtt performs significantly better than CNN-aux on some transfer tasks such as $D \rightarrow K$, $D \rightarrow V$. DAmSDA focuses on capturing domain-shared representation by a gradient reversal layer and performs worse than SentATN by 5.39% on average. Capsule outperforms CNN-aux by 0.72% and performs worse than SentATN by 5.05%, which suggests that Capsule indeed has the advantage in learning domain-shared knowledge over CNN. SentATN outperforms AMN by 4.96% on average, in which two memory networks are proposed to automatically recognize pivots and non-pivots. HAN outperforms these four methods especially BiLSTMAtt, which fully indicates the significance of hierarchical feature representation for long documents. HATN and CCHAN are both hierarchical attention network-based models and achieve significant improvements compared to previous methods. The reason is that HATN can learn the relationship between pivots and non-pivots and CCHAN can learn more general domain-shared representation by additional convolutional layer. SentATN outperforms HATN by 1.14%, CCHAN by 0.83% on average, respectively. WTN utilizes wasserstein distance to compute the discrepancy between the source domain and target domain, learning domain-shared representation. SentATN outperforms WTN by 0.118% on average. However, all these approaches focus on capturing domain-shared knowledge while neglecting the domain-

Table 3 The classification accuracy for CDSC on the Amazon reviews dataset

S - T	CNN-aux	BiLSTMAtt	DAmSDA	Capsule	AMN	HAN	HATN	CCHAN	WTN	SentATN
B-D	0.8442	0.8562	0.8612	0.8513	0.8562	0.8462	0.8707	0.8721	0.8800	**0.8815**
B-E	0.8063	0.7968	0.7902	0.8278	0.8055	0.8212	0.8575	0.8649	0.8552	**0.8773**
B-K	0.8338	0.8157	0.8105	0.8298	0.8188	0.8053	0.8703	0.8740	0.8738	**0.8845**
B-V	0.8443	0.8472	0.8498	0.8607	0.8725	0.8487	0.8780	0.8827	0.8847	**0.8852**
D-B	0.8307	0.8525	0.8517	0.8407	0.8453	0.8717	0.8778	0.8783	0.8855	**0.8987**
D-E	0.8035	0.8003	0.7617	0.7923	0.8042	0.8260	0.8632	**0.8717**	0.8568	0.8635
D-K	0.8168	0.8390	0.8260	0.8023	0.8167	0.8573	0.8747	0.8782	0.8747	**0.8838**
D-V	0.8587	0.8883	0.8380	0.8902	0.8740	0.8867	0.8912	0.9012	**0.9032**	0.8945
E-B	0.7738	0.7842	0.7992	0.7733	0.7752	0.8067	0.8403	0.8436	0.8345	**0.8583**
E-D	0.7907	0.7932	0.8263	0.8148	0.8053	0.7915	0.8432	0.8439	0.8300	**0.8505**
E-K	0.8715	0.8855	0.8580	0.8872	0.8783	0.8977	0.9008	0.9056	0.9093	**0.9240**
E-V	0.7878	0.7617	0.8170	0.7803	0.8212	0.8195	0.8418	**0.8476**	0.8303	0.8432
K-B	0.7847	0.8108	0.8055	0.7983	0.7905	0.8435	0.8488	0.8498	0.8447	**0.8678**
K-D	0.7907	0.7818	0.8218	0.7970	0.7950	0.8155	0.8472	0.8487	0.8373	**0.8502**
K-E	0.8673	0.8737	0.8800	0.8832	0.8668	0.8693	0.8933	0.8973	**0.9102**	0.9068
K-V	0.7882	0.8058	0.8147	0.8257	0.8215	0.8020	0.8485	0.8551	0.8482	**0.8637**
V-B	0.8148	0.8517	0.8300	0.8202	0.8350	0.8385	0.8710	0.8712	0.8700	**0.8813**
V-D	0.8525	0.8345	0.8590	0.8642	0.8688	0.8707	0.8790	0.8826	**0.8883**	0.8847
V-E	0.8232	0.7953	0.7767	0.7785	0.7968	0.7733	0.8598	0.8604	0.8420	**0.8678**
V-K	0.8128	0.7820	0.7952	0.8220	0.8098	0.8225	0.8645	0.8542	0.8545	**0.8827**
AVE	0.8198	0.8228	0.8236	0.8270	0.8279	0.8357	0.8661	0.8692	0.8657	**0.8775**

Bold entries signify the highest result for each transfer task

specific information. The domain-specific information in a sentence establishes significant obstacles in the transfer training process of HATN and CCHAN. However, the efficient DI-Encoder in SentATN can overcome the issue. In addition to the efficiency advantage, the proposed SentATN reduces the discrepancy in the distribution of source and target domains in the sentence-level, which helps it yield better performance.

To validate the effectiveness of the key building blocks in SentATN, we compare our method with its variants SentATN-DI, SentAN, and SentAN-. The result is shown in Fig. 4. We can see that the overall performance of SentATN is significantly better than its three variants and the SentAN- variant is the worst. Besides, we observe that the transfer performance of SentATN is better than SentATN-DI in 20 transfer tasks. This means

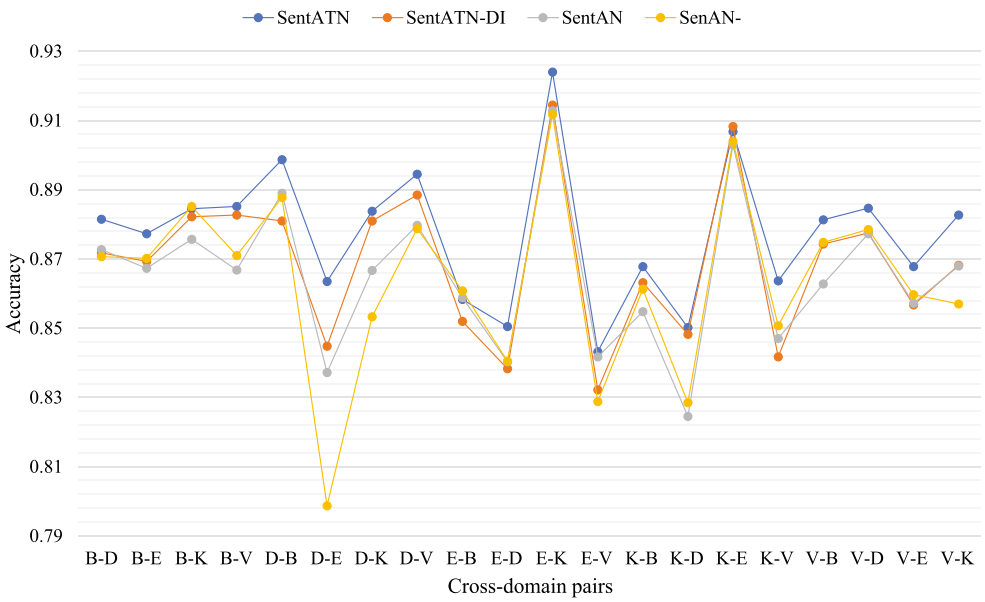

Fig. 4 The classification accuracy of variants from SentATN for CDSC on the Amazon reviews dataset

that DI-Encoder can effectively identify domain-specific information, which implies the effectiveness of introducing external domain-specific information in CDSC. SentAN's average performance is worse than SentATN and the observation suggests that leveraging target domain data to narrow the discrepancy in two domains is beneficial for CDSC. Furthermore, SentAN's average performance outperforms SentAN-, which validates the effectiveness of the positional encoding. SentAN is significantly better in some transfer tasks than SentAN- such as $D \rightarrow E$. This also shows that sentence-level positional features can also remarkably improve transfer capability.

Next, we compare SentATN with WordATN and WordATN*, respectively. The two models are very similar. WordATN* employs static word2vec vectors, while WordATN adopts original BERT to learn the embedding vectors of words. Figure 5 shows the results on 20 transfer tasks. We can see that SentATN significantly outperforms WordATN and WordATN* on all transfer tasks, which validates the effectiveness of the sentence-level transfer for CDSC. Besides, WordATN performs much better than WordATN* on most transfer tasks, which implies that BERT vectors contain more accurate semantic information than the static word vectors.

4.5 Training efficiency

Our proposed model SentATN is a sentence-level training method, while the other state-of-the-art methods such as WTN, CCHAN, HATN are word-level training methods. To compare the efficiency of the sentence-level training and word-level training, we compare the training curves of SentATN and WordATN on the $B \rightarrow D$ task. The model training will be terminated if the accuracy of the model in the validation set no longer improves in five consecutive training epoch. Figure 6 shows the results. From this figure, it can be obviously seen that the convergence speed of the SentATN is much faster than WordATN. Besides, we know SentATN performs much better than WordATN in previous experiments. Hence, we can conclude that SentATN is not only better than other word-level methods in performance but also much better in training efficiency.

4.6 Case study

Here we present a case study to show that how our model SentATN can capture domain-specific information during the transfer learning process. For instance, the sentence *"I'm afraid it will catch on fire because of how fast the motor is running now–it makes a sickly high-pitched whine."* is from the kitchen domain, and it is encoded by the DI-Encoder to obtain the sentence representation. DI-Encoder is stacked by many self-attention and feed-forward layers, and we visualize parameters in a single self-attention layer to analyze dependency between words in this sentence learned by this DI-Encoder. First, we analyze contributions of each token in this sentence for forming the whole sentence representation. The corresponding weight scores of different tokens are shown in Fig. 7. The token *DI* represents the fixed pattern sentence "This is the Kitchen Domain". Intuitively, we can see the domain information plays an important role in forming the whole sentence representation, and thus the representation for this sentence contains domain-specific semantic information in the latent

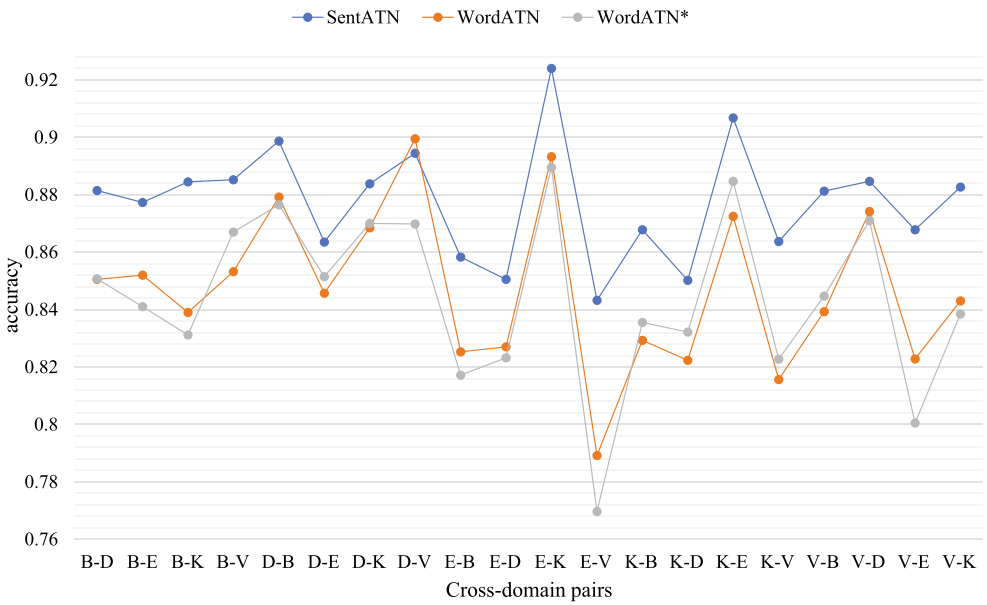

Fig. 5 The classification accuracy of SentATN and its variants (based on the word-level) for CDSC on the Amazon reviews dataset

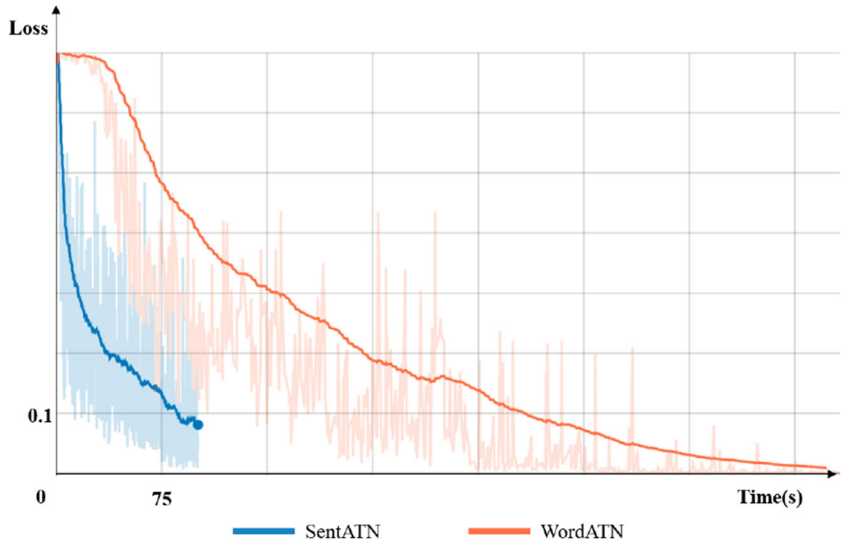

Fig. 6 Model training curve of WordATN and SentATN on the $B \rightarrow D$ task

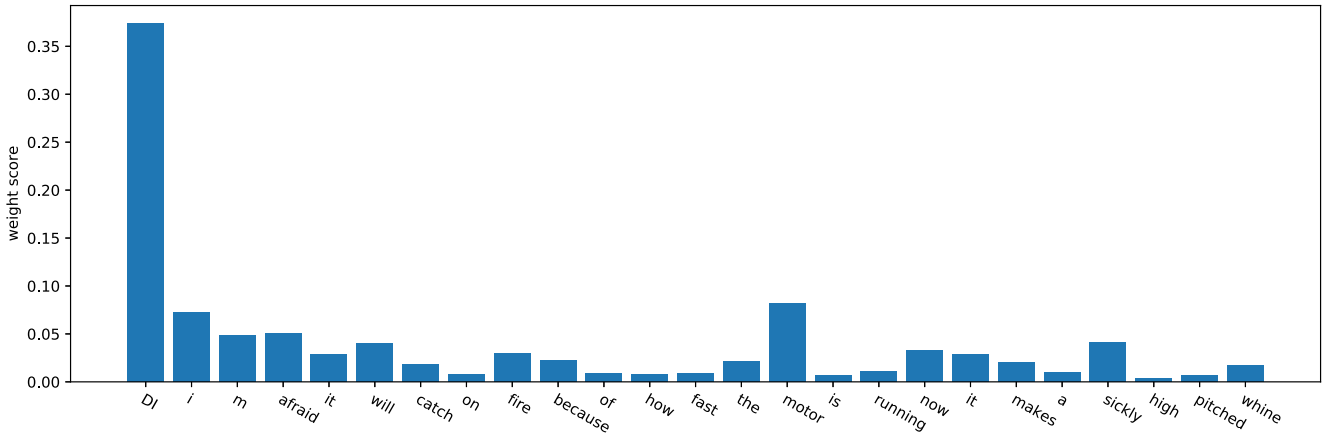

Fig. 7 The contributions of each token (including the domain information) for forming the whole sentence representation

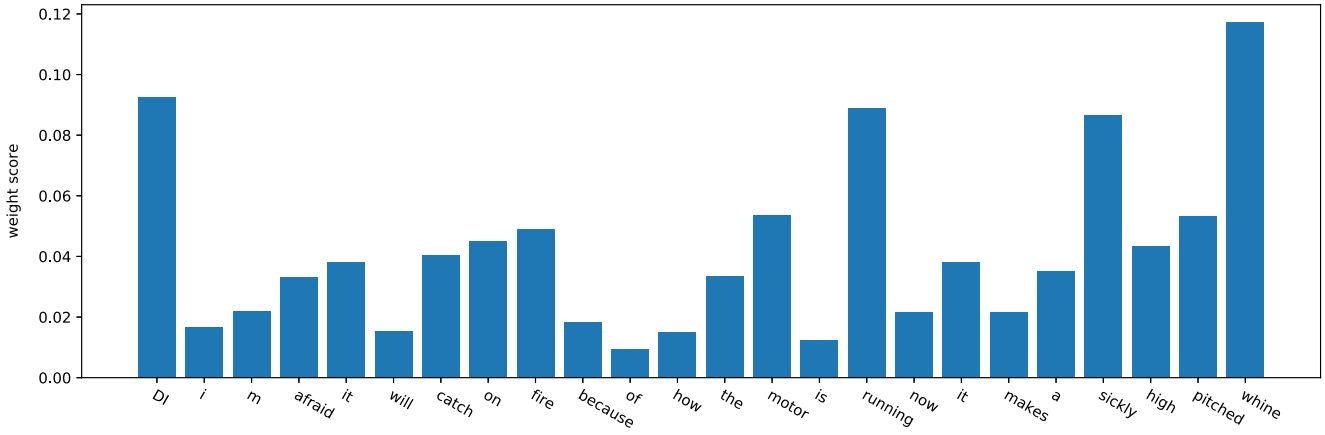

Fig. 8 The contributions of each token (not including the word *fast*) for forming the representation of the word *fast*

space. Second, we further show the impact of the domain information on each word such as *fast*. The weight scores of tokens in forming the representation of the word *fast* are shown in Fig. 8. With the same method, the domain information also makes corresponding contributions in obtaining the representation. In this DI-Encoder, each token obtains the semantic representation under the guidance of the domain information. Finally, the high-level domain-specific representation of this sentence can be obtained by the stacked self-attention and feed-forward layers in DI-Encoder. This analysis fully demonstrates that SentATN can obtain the semantic representation of each word under the guidance of domain information and further capture domain-specific features of a sentence.

5 Conclusion

In this paper, we propose a new model SentATN by designing DI-Encoder and conducting transfer training on the sentence-level for CDSC. And a series of experiments are conducted on extended Amazon review datasets, which thoroughly validates the effectiveness and superiority of the proposed model. To the best of our knowledge, this study is the first to focus on extracting domain-specific features and investigate the transfer capability of the model in the sentence level for CDSC. In this work, we design the DI-Encoder to model the domain-specific information. Due to the complicated dependency relations between the pivots and non-pivots, we think it is interesting to integrate the graph neural network into our method to better exploit the domain-specific information in the future.

Acknowledgments This research was supported in part by the National Key R&D Program of China, 2018YFB2101100, 2018YFB2101101, and NSFC under Grant No. 61972111.

References

1. Deborah SA, Mirnalinee TT, Rajendram SM (2021) Emotion analysis on text using multiple kernel gaussian... Neural Process Lett 53(2):1187–1203
2. Parcheta Z, Sanchis-Trilles G, Casacuberta F, Rendahl R (2021) Combining embeddings of input data for text classification. Neural Process Lett 53:3123–3153
3. Zhang B, Xu X, Yang M, Chen X, Ye Y (2018) Cross-domain sentiment classification by capsule network with semantic rules. IEEE Access 6:58284–58294
4. Du Y, He M, Wang L, Zhang H (2020) Wasserstein based transfer network for cross-domain sentiment classification. Knowl-Based Syst 204:106162
5. Sharma R, Bhattacharyya P, Dandapat S, Bhatt HS (2018) Identifying transferable information across domains for cross-domain sentiment classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 968–978
6. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. The Association for Computational Linguistics
7. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning, pp 513–520
8. Yu J, Jiang J (2016) Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing. The Association for Computational Linguistics, pp 236–246
9. Li Z, Zhang Y, Wei Y, Wu Y, Yang Q (2017) End-to-end adversarial memory network for cross-domain sentiment classification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp 2237–2243
10. Li Z, Wei Y, Zhang Y, Yang Q (2018) Hierarchical attention transfer network for cross-domain sentiment classification. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp 5852–5859
11. Du C, Sun H, Wang J, Qi Q, Liao J (2020) Adversarial and domain-aware bert for cross-domain sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp 4019–4028
12. Sun B, Saenko K (2016) Deep coral: Correlation alignment for deep domain adaptation. In: Computer Vision - ECCV 2016 Workshops. Springer, pp 443–450
13. Fu C, Huang H, Chen X, Tian Y, Zhao J (2021) Learn-to-share: A hardware-friendly transfer learning framework exploiting computation and parameter sharing. In: International Conference on Machine Learning. PMLR, pp 3469–3479
14. Lashkaripour A, Rodriguez C, Mehdipour N, Mardian R, McIntyre D, Ortiz L, Campbell J, Densmore D (2021) Machine learning enables design automation of microfluidic flow-focusing droplet generation. Nat Commun 12(1):1–14
15. Huang X, Paul M (2019) Neural temporality adaptation for document classification: Diachronic word embeddings and domain adaptation models. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 4113–4123
16. Li Z, Peng X, Zhang M, Wang R, Si L (2019) Semi-supervised domain adaptation for dependency parsing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 2386–2395
17. Shu R, Bui HH, Narui H, Ermon S (2018) A dirt-t approach to unsupervised domain adaptation. In: Proceedings of the 6th International Conference on Learning Representations
18. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096–2030
19. Zhang K, Zhang H, Liu Q, Zhao H, Zhu H, Chen E (2019) Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 5773–5780
20. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems

21. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp 2227–2237
22. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8)
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
24. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp 4171–4186
25. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: Generalized autoregressive pretraining for language understanding. Adv Neural Inf Process Syst 32:5753–5763
26. Dai Z, Yang Z, Yang Y, Carbonell JG, Le Q, Salakhutdinov R (2019) Transformer-xl: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 2978–2988
27. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2020) ALBERT: A lite BERT for self-supervised learning of language representations. In: 8th International Conference on Learning Representations. OpenReview.net
28. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692
29. Sukhbaatar S, Weston J, Fergus R et al (2015) End-to-end memory networks. In: Advances in neural information processing systems, pp 2440–2448
30. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. The Association for Computational Linguistics, pp 1480–1489
31. Yue C, Cao H, Xu G, Dong Y (2021) Collaborative attention neural network for multi-domain sentiment classification. Appl Intell 51(6):3174–3188
32. Liao W, Zeng B, Yin X, Wei P (2021) An improved aspect-category sentiment analysis model for text sentiment analysis based on roberta. Appl Intell 51(6):3522–3533
33. Hovy D, Yang D (2021) The importance of modeling social factors of language: Theory and practice. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 588–602
34. Tan Z, Chen J, Kang Q, Zhou M, Abusorrah A, Sedraoui K (2021) Dynamic embedding projection-gated convolutional neural networks for text classification. IEEE Transactions on Neural Networks and Learning Systems
35. Zhou J, Huang JX, Hu QV, He L (2020) Is position important? deep multi-task learning for aspect-based sentiment analysis. Appl Intell 50(10):3367–3378
36. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. Mach Learn 79(1):151–175
37. Bird S, Klein E, Loper E (2009) Natural language processing with python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc."
38. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations
39. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of the 1st International Conference on Learning Representations
40. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, pp 3104–3112
41. Yin H, Liu P, Zhu Z, Li W, Wang Q (2019) Capsule network with identifying transferable knowledge for cross-domain sentiment classification. IEEE Access 7:153171–153182
42. Manshu T, Xuemin Z (2019) Cchan: An end to end model for cross domain sentiment classification. IEEE Access 7:50232–50239

Kuai Dai is working toward the Ph.D. degree with the Harbin Institute of Technology, Shenzhen, China. His research interests include natural language processing and data mining.

Xutao Li is currently an Associate Professor with the Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. degree in computer science from the Harbin Institute of Technology, Shenzhen, China. His research interests include data mining, machine learning, social network analysis, and remote sensing data analysis. He is the corresponding author of this article.

Xu Huang is working toward the Ph.D. degree with the Harbin Institute of Technology, Shenzhen, China. His research interests include data mining.

Yunming Ye is currently a Professor with the Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China. His research interests include data mining, text mining, and ensemble learning algorithms.