mathematics

MDPI

Article

Dual-Word Embedding Model Considering Syntactic Information for Cross-Domain Sentiment Classification

Zihao Lu [1], Xiaohui Hu [2,*] and Yun Xue [2]

[1] School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006, China
[2] School of Electronics and Information Engineering, South China Normal University, Foshan 528225, China
* Correspondence: huxh@scnu.edu.cn

Abstract: The purpose of cross-domain sentiment classification (CDSC) is to fully utilize the rich labeled data in the source domain to help the target domain perform sentiment classification even when labeled data are insufficient. Most of the existing methods focus on obtaining domain transferable semantic information but ignore syntactic information. The performance of BERT may decrease because of domain transfer, and traditional word embeddings, such as word2vec, cannot obtain contextualized word vectors. Therefore, achieving the best results in CDSC is difficult when only BERT or word2vec is used. In this paper, we propose a Dual-word Embedding Model Considering Syntactic Information for Cross-domain Sentiment Classification. Specifically, we obtain dual-word embeddings using BERT and word2vec. After performing BERT embedding, we pay closer attention to semantic information, mainly using self-attention and TextCNN. After word2vec word embedding is obtained, the graph attention network is used to extract the syntactic information of the document, and the attention mechanism is used to focus on the important aspects. Experiments on two real-world datasets show that our model outperforms other strong baselines.

Keywords: cross-domain sentiment classification; word embedding; GAT

MSC: 68T50

Citation: Lu, Z.; Hu, X.; Xue, Y. Dual-Word Embedding Model Considering Syntactic Information for Cross-Domain Sentiment Classification. *Mathematics* **2022**, *10*, 4704. https://doi.org/10.3390/math10244704

Academic Editors: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 9 November 2022
Accepted: 9 December 2022
Published: 11 December 2022

1. Introduction

Sentiment classification is an important task in natural language processing, and it can help people make better decisions in daily life [1,2]. Over the past few decades, many machine learning methods have been introduced for classification tasks, such as logistic regression, collaborative representation, support vector machines, and neural networks [3–7]. With the development of the internet, a large number of user comments and other texts containing sentiment have been generated from different domains. However, the classical sentiment classification methods require that the training and testing data come from the same domain [8,9]. In addition, the training of deep networks relies on a large amount of labeled data, but texts in many domains lack sufficient labeled data. Cross-domain sentiment classification (CDSC) is a promising direction that can make full use of the rich labeled data in the source domain to assist the target domain with the lack of labeled data for sentiment classification.

Traditional word-level vector representations, such as word2vec [10], glove [11], and fastText [12], can use a single vector to represent all possible meanings of a word. This method results in providing the same representation for words that express different sentiment polarities in various domains. In recent years, pre-trained language models, such as ELMO [13] and BERT [14], have been widely used in natural language processing (NLP) tasks because they can obtain contextualized word embedding. Notably, BERT has achieved state-of-the-art results on many NLP tasks because of its strong language understanding capabilities. In cross-lingual tasks, multilingual BERT (mBERT) can share

part of its representation space between languages [15]. In addition, the mBERT language model has the ability to transfer syntactic knowledge cross-lingually, and can embed the dependency parse tree of sentences cross-lingually [16]. This shows that Bert parse trees have a strong ability to perform different tasks. However, some problems occur with directly fine-tuning BERT in CDSC tasks [17]. One of the pre-training tasks of BERT is to randomly MASK off 15% of the words, and when the words are filled back, various domains may fill back different words. In addition, because no labeled data exist in the target domain, fine-tuning only by the labeled data in the source domain reduces the performance because of different training and test distributions. Therefore, using BERT or word2vec only to obtain word vector embeddings in CDSC is insufficient. On the other hand, many current models aim to learn transferable semantic information in CDSC to predict the sentiment polarity of the target domain. However, in addition to semantic information, syntactic information is equally important. Therefore, extracting transferable syntactic information is important for CDSC tasks to better help target domain sentiment classification.

To solve the above problems, we propose a dual-word embedding model considering syntactic information for CDSC. The model performs dual-word embedding through BERT and word2vec to obtain rich word embedding information. Different from most previous models that only consider semantic information, we adopt dual-channel to obtain transferable semantic information and syntactic information. Semantic information is obtained by self-attention and TextCNN. Syntactic information is obtained through the graph attention network so that the aspects in the sentence can obtain syntactic information [18]. Then, the attention mechanism is used to pay attention to important aspects so that the syntactic information of aspects can play a role. Finally, domain-invariant features are obtained through adversarial training. The contributions of our study can be summarized as follows:

- A CDSC method is proposed using BERT and word2vec to obtain dual-word embeddings;
- Dual-channel feature extraction and adversarial training to obtain transferable semantic and syntactic information;
- Extensive experiments are conducted on two real-world datasets, and experimental results show that our model achieves better results compared to other strong baselines.

2. Related Work

2.1. CDSC

CDSC aims to utilize the source domain with rich labeled data to help sentiment classification in the target domain without labeled data. The traditional CDSC method needs to manually select pivots. Blitzer et al. [19] proposed the structural correspondence learning (SCL) method. The most frequently used words in both domains are good predictors of source domain labels, so they select the set of pivot features that appear most frequently in both the source and target domains. Pan et al. [20] proposed spectral feature alignment (SFA) for CDSC. They want to associate the source domain with the target domain by aligning pivots with non-pivots. However, manually obtaining domain-invariant features through these traditional methods is a time-consuming and expensive process. With the rise of neural networks in recent years, many scholars have explored the application of deep learning in CDSC tasks. Among them, the domain adversarial neural network (DANN) [21] is explored to learn domain-invariant features in the min-max game between the domain classifier and the feature extractor through adversarial training. Li et al. [22] proposed a hierarchical attention transfer network (HATN) that can automatically capture pivots and non-pivots through hierarchical attention and auxiliary tasks. Zhang et al. [23] designed an interactive attention transfer network (IATN) that applies interactive attention to CDSC, considering the influence of aspects in sentences. Yang et al. [24] proposed a dual-channel mutual learning domain adaptive model. In recent years, BERT has been gradually applied to CDSC because of the advantages of the BERT pre-training model. Du et al. [17] designed a domain-aware BERT (BERT-DAAT) to apply BERT to unsupervised CDSC tasks. Du et al. [25] designed a Wasserstein-based transfer network (WTN) to obtain rich domain-invariant features. Fu et al. [26] paid closer attention to the intra-domain structure,

and they proposed domain adaptation with a contractible difference strategy. The successful application of the attention mechanism improves classification accuracy substantially. However, it is difficult to obtain syntactic information using attention. In this paper, we consider adding a graph attention network to obtain transferable syntactic information.

2.2. Graph Attention Work

Graph neural networks have received extensive attention from scholars in recent years because these networks allow the use of deep learning frameworks on graph structure data [27–30]. At present, many mature neural network models can work on regular network structures. Since the graph convolutional neural network (GCN) [31] was proposed as a deep convolutional learning paradigm for graph structure data, it has filled the gap in the development of deep learning for processing such data. To capture the dependencies between discontinuous and long-distance words in a document, Vashishth et al. [32] used GCN to characterize the dependency tree for each sentence in the document. However, the importance of each node in the graph should be different, and a graph convolutional neural network cannot deal with this situation. Therefore, some researchers have introduced the idea of attention mechanism into the graph convolutional neural network. Veličković proposed [33] graph attention network (GAT), which mainly improves GCN by using the attention mechanism to aggregate the characteristics of discriminated neighbor nodes. Therefore, compared with GCN, GAT can better handle dynamic graphs. Huang et al. [18] used GAT to establish dependencies between words. Although it is common to use GCN or GAT to obtain syntactic information in single-domain tasks, few people extract syntactic information in CDSC tasks.

2.3. Word Embedding

Word vector representations transform words in natural language into a form that the computer can recognize and understand [34]. We can obtain word vector representations by using word embedding methods, such as word2vec and glove. Nguyen et al. [35] applied a word2vec embedding model to construct a semantic vector for the plot content of each movie. Wang et al. [36] trained their personality classification model on a shared potential feature space by predictive text embedding. Naderalvojoud [37] et al. proposed two methods to create sentiment-aware word embeddings, improving on the pre-trained word embedding of the word2vec and gloVe models.

In recent years, BERT has received a lot of attention because it can learn contextualized word representations. BERT is a bidirectional variant of the multilayer transformer, which further integrates bidirectional representations. Jawahar et al. [38] revealed elements of the English language structure learned by BERT. They also demonstrated that BERT captures phrase-level information at the low layers, syntactic features at the intermediate layers, and semantic features at the high layers. In addition, the information at lower layers is diluted at higher layers. In this paper, we combine word2vec and BERT to obtain rich word vector information. In addition, in order to prevent the low-layer information from being diluted at the high-layer, we use the weighted sum of all layer information of BERT as the input vector.

3. Methodology

In this section, we introduce the framework of DWE in technical detail. First, we describe the problem and provide a model structure. Then, the training strategy is detailed.

3.1. Problem Definition

In the task of CDSC, we are given two domains, D_s and D_t, which denote a source domain and a target domain, respectively. A set of labeled data $\{X_s, Y_s\}$ is used in D_s, where $\{X_s, Y_s\} = \{x_i^s, y_i^s\}_{i=1}^{N_s}$ presents N_s labeled samples in D_s. We also have a set of unlabeled data $\{X_t\}$ in D_t, where $\{X_t\} = \{x_i^t\}_{i=1}^{N_t}$ presents N_t unlabeled samples in D_t.

The goal of the CDSC task is to utilize the source domain with rich labeled data to assist the target domain lacking labeled data for sentiment classification.

3.2. Model Structure

As shown in Figure 1, DWE mainly contains three parts: feature extraction module, domain discriminator, and sentiment classifier. The feature extraction module uses dual channels to obtain semantic information and syntactic information. The domain discriminator obtains domain-invariant features. The sentiment classifier uses the softmax activation function to obtain the probability of the sentiment label.

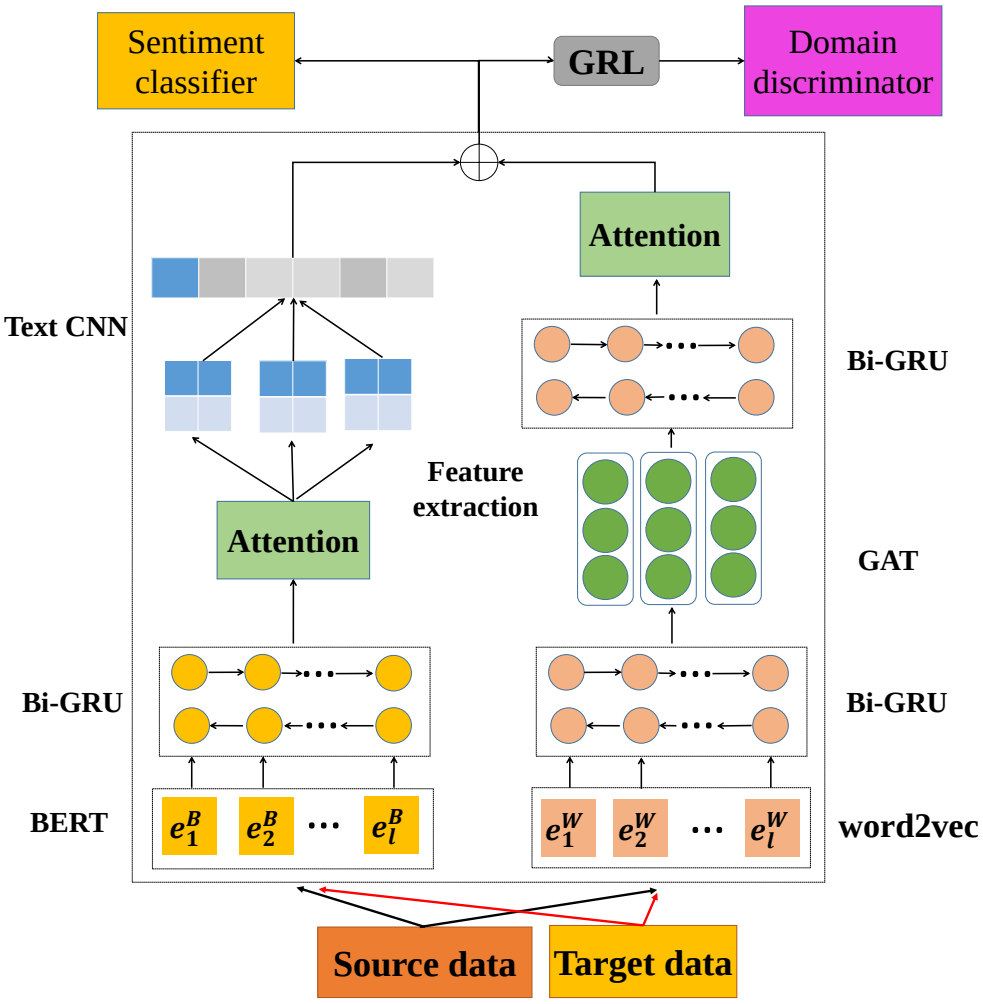

Figure 1. Model architecture.

3.3. Feature Extraction

To get rich word embedding information, we use BERT and word2vec to obtain dual word embedding. After obtaining different word embeddings, a dual channel is formed to extract transferable semantic information and syntactic information.

3.3.1. Bert Semantic Channel

In this channel, we mainly extract semantic information. We first use BERT to obtain word vectors. To prevent the loss of some information, unlike in the general final hidden state using the BERT structure, we apply an approach similar to that by Du et al. [25], using the weighted sum of all hidden states as the input vector. We define the nth hidden state of the mth layer as h_n^m. We suppose that a document contains S sentences with k words,

and w_i is the ith word of the input document. w_i is tokenized to q BPE (byte pair encoding) tokens $w_i = \{b_i^1, b_i^2, \ldots, \ldots, b_i^n\}$. The word vector obtained by BERT can be defined as

$$e_i^B = \sum_{m=1}^{\mathrm{L}} \alpha_m \cdot \frac{\sum_{n=1}^{q} h_n^m}{q} \tag{1}$$

where α_m and L are the weight coefficients of layer m and the number of hidden state layers of BERT, respectively. BiGRU is the variant of BiLSTM, which has the ability to learn long-term dependencies. We can use BiGRU to build sequential information about words or sentences. Thus, we then input the word vector into BiGRU to obtain the hidden states

$$h_i^B = BiGRU\left(e_i^B\right) \tag{2}$$

Different words in a sentence have different effects on sentence sentiment because these words express different semantic information. The attention mechanism can pay attention to the words that play an important role in sentence sentiment according to attention coefficient. In this paper, we use self-attention to calculate word-to-word associations in sentences, which can focus on words that have a stronger impact on sentence sentiment. Attention scores were calculated as follows:

$$g_i^B = tanh\left(W * h_i^B + b\right) \tag{3}$$

where W and b represent the learnable weight matrix and bias in the network, respectively.

Furthermore, we normalized the attention scores by using the softmax activation function to generate the attention coefficients α_i^B for each word

$$\alpha_i^B = \frac{exp\left(g_i^B\right)}{\sum_{i=1}^{n} exp\left(g_i^B\right)} \tag{4}$$

The attention coefficient is combined with the hidden state obtained by BiGRU to obtain the sentence vector s^B

$$s^B = \sum_{j=1}^{k} \alpha_j^B \cdot h_j^B \tag{5}$$

where $\cdot$ indicates the element-wise product. After obtaining sentence vectors, TextCNN [39] is used to further extract important semantic information that mainly includes convolution layer and pooling layer. First, we input the sentence vector to the convolution layer and the convolution operation involves the filter w_{cnn}

$$c^B = F\left(w_{cnn} \circ s^B + b_{cnn}\right) \tag{6}$$

where $\circ$ represents the convolution operation, b_{cnn} is the bias term, and F is a nonlinear function such as Relu. Then, max pooling is performed to retain important features. Finally, dropout prevents overfitting to obtain the sentence representation of the semantic channel. The relevant formulas are the following:

$$c_p^B = Maxpooling\left(c^B\right) \tag{7}$$

$$d^B = dropout\left(c_p^B\right) \tag{8}$$

3.3.2. Word2vec Syntax Channel

In this channel, we first use word2vec to obtain the word vector representation

$$e_i^w = word2vec(w_i) \tag{9}$$

Then, input the word vector into BiGRU to extract the sentence representation. The hidden output of BiGRU can be expressed as follows:

$$h_i^w = BiGRU(e_i^w) \tag{10}$$

To obtain syntactic information, the syntax dependency tree of the given sentence is built in advance, and then the tree structure is converted into a graph structure in which each node represents a word. Given a dependency graph with N nodes, the node representation is computed by aggregating the hidden states of the neighborhood. After l layers of GAT, the last layer outputs the syntactic representation. The output of the ith node at layer l is defined as g_i^l, and g_i^0 indicates the initial node status, $g_i^0 = h_i^w$. The node update process is as follows:

$$e_{ij}^l = leakyRelu\left(\alpha^{l^T}\left(W_g^l g_i^l \| W_g^l g_j^l\right)\right) \tag{11}$$

$$\alpha_{ij}^l = \frac{exp\left(e_{ij}^l\right)}{\sum_{k\in N(k)} e_{ik}^l} \tag{12}$$

$$g_i^{l+1} = \sigma\left(\sum_{j\in N(i)} \alpha_{ij}^l W_g^l g_i^l\right) \tag{13}$$

where W_g^l and α^{l^T} are trainable weight matrices and weight vectors, respectively. $\|$ represents vector concatenation. e_{ij}^l is the raw attention score between the ith and jth nodes. $N(i)$ is the set of all adjacent nodes. α_{ij}^l is the normalized attention weight. σ denotes a Relu activation function. For simplicity, we can write such feature propagation process as

$$g_i^{l+1} = GAT\left(g_i^l, A, \theta_l\right) \tag{14}$$

where A is the graph adjacent matrix and θ_l is the set of parameters at layer l. Finally, we input the syntactic representation into BiGRU and Attention. BiGRU can build the long-term dependencies of sentences in a document. Attention mechanism can make the syntactic information of important aspects in syntactic representation play a more critical role. Thus, we obtain the final representation of the syntactic channel:

$$H_i^w = BiGRU(g_i^w) \tag{15}$$

$$\alpha_i^w = \frac{exp\left(tanh\left(W_w H_i^w + b_w\right)\right)}{\sum_{i=1}^n exp\left(tanh\left(W_w H_i^w + b_w\right)\right)} \tag{16}$$

$$d^w = \sum_{j=1}^k \alpha_j^w \cdot H_j^w \tag{17}$$

where α_i^w, $\cdot$, W_w and b_w represent the attention weight, the element-wise product, the learnable weight matrix, and bias in the network, respectively.

3.3.3. Final Document Representation

The final document representation is obtained by concatenating the document representation of the two channels as follows:

$$d = \left[d^B, d^w\right] \tag{18}$$

3.4. Sentiment Classifier

The ultimate goal of our task is to predict sentiment labels. In this module, we use the softmax activation function to obtain the sentiment prediction label for the document

$$y = softmax\left(W_y d + b_y\right) \tag{19}$$

where W_y and b_y represent the learnable weight matrix and bias, respectively.

3.5. Domain Discriminator

The purpose of the domain discriminator (D) is to enable the feature extractor (FE) to learn domain-invariant representations. We consider using adversarial training. The domain discriminator tries to find out which domain the document vector comes from, while the feature extractor aims to deceive the domain discriminator so that it cannot distinguish which domain the document comes from and achieve the purpose of domain information transfer. The domain discriminator regards the document representation obtained by the feature extractor as input and outputs the probability that the document comes from the source domain. If a document belongs to the source domain, we set $r_i = 1$. For the target domain, we set $r_i = 0$. To better solve this problem, we introduce a gradient reversal layer (GRL) that can reverse the gradient direction during training. We can treat the gradient reversal layer as a pseudo function $G(x)$. Through the domain discriminator, we can obtain domain-invariant features. Formally, the domain discriminator performs a min-max game to optimize the parameters Θ_{FE} and Θ_D as follows:

$$\tilde{d} = G(d) \tag{20}$$

$$y'_d = softmax\left(W_d \tilde{d} + b_d\right) \tag{21}$$

$$\Theta_{FE}, \Theta_D = \underset{\Theta_{FE}}{argmax} \underset{\Theta_D}{min} L_{dom} \tag{22}$$

$$L_{dom} = -\left(r_i In y'_d + (1 - r_i) In(1 - y'_d)\right) \tag{23}$$

where L_{dom}, Θ_{FE}, and Θ_D represent the domain loss, parameters of the feature extractor, and parameters of the domain discriminator, respectively.

3.6. Training Strategy

We apply the cross-entropy loss function to the sentiment classifier to obtain the sentiment classification loss

$$L_{sen} = -\left(y' In y + (1 - y') In(1 - y)\right) \tag{24}$$

where y' represents the ground truth of the sentiment label. Furthermore, we obtain our total loss function

$$L_{total} = L_{sen} + L_{dom} + \rho L_{reg} \tag{25}$$

$$L_{reg} = \lambda \|\theta\|^2 \tag{26}$$

where L_{reg}, ρ, λ, θ represents the L_2 regularization term which can avoid overfitting, regularization parameter, hyperparameters, and all parameters in the network, respectively. The regularization term can automatically weaken unimportant feature variables, automatically extract important feature variables from many feature variables, and reduce the magnitude of feature variables.

4. Experiment

4.1. Datasets

To verify the effectiveness of the proposed model, we used two datasets which are obtained from Amazon product reviews. Dataset 1 has been widely used in CDSC tasks. It contains reviews from four different domains: Books (B), DVDs (D), Electronics (E),

and Kitchen (K). A total of 2000 labeled data are in each domain, consisting of 1000 positive reviews and 1000 negative reviews. We selected 800 positive and 800 negative reviews in the source domain as the training data; 1600 in the target domain for domain classification; and the remaining 200 positive reviews and 200 negative reviews in the target domain as the test data. Table 1 records the details of Dataset 1.

Dataset 2, constructed by He et al. [40], contains data for three sentiment labels, namely, positive, neutral, and negative, so this dataset is more convincing. Dataset 2 also contains data from four domains: Book (BK), Beauty (BT), Music (M), and Electronics (E). Each domain has two types of data: Set 1 and Set 2. Set 1 is balanced, with 2000 data for each sentiment label, while Set 2 is unbalanced. For Dataset 2, we choose processing similar to that used by Du et al. [25], using balanced Set 1 as the training data of the source domain, and using unbalanced data Set 2 as the training data of the target domain. We selected 1200 reviews from the training set of the source domain as the development set. The balanced data Set 1 from the target domain is used as the test set. Table 2 presents an overview of the datasets.

Table 1. Statistics of Dataset 1.

Domain	Positive	Negative	Vocabulary
Books	1000	1000	26,278
DVD	1000	1000	26,940
Electronics	1000	1000	13,256
Kitchen	1000	1000	11,187

Table 2. Statistics of Dataset 2.

Domain		Positive	Negative	Neutral
Book	Set1	2000	2000	2000
	Set2	4824	513	663
Beauty	Set1	2000	2000	2000
	Set2	4709	616	675
Music	Set1	2000	2000	2000
	Set2	4441	785	774
Electronics	Set1	2000	2000	2000
	Set2	4817	694	489

4.2. Experiment Setup

In the experiment, we use the common word2vec and BERT to obtain dual-word embedding. First, we use 300-dimensional word2vec vectors as one of the word embeddings, which are trained on 100 billion words from Google News. Then, we fine-tune it during the training. We use uniform distribution U (−0.25,0.25) to randomly initialize words outside the vocabulary. In addition, we use BERT with 12 layers, 768 hidden units, 12 self-attention heads, and 110 million parameters as another word embedding. The dimension of the attention vector is set to 200. The dimension of the feature representation in each field and the maximum word number of every review are set to 200. The weight matrix in the network is randomly initialized from the uniform distribution U (−0.01,0.01). The dropout rate is 0.5 to prevent overfitting. The number of GAT layers is set to 3, and Adam algorithm is used as the optimizer.

4.3. Experimental Results

Following previous studies, we apply the accuracy rate as the evaluation standard. The accuracy rate is the percentage of correctly classified data in the total data. The best

results are highlighted in bold. We compare the proposed model DWE with some classic baselines as follows:

- DANN [21]: The model is trained using the domain adversarial network approach, including GRL for domain obfuscation;
- AuxNN [41]: The model uses auxiliary tasks for CDSC;
- AMN [42]: The model is based on memory network and the adversarial training method to obtain domain-invariant features;
- DAS [40]: It uses feature adaptation and semi-supervised learning to improve classifiers while minimizing domain divergence;
- HATN [22]: The hierarchical attention network is used for CDSC, and pivots and non-pivots features are extracted to assist classification tasks;
- IATN [23]: Interactive attention mechanism is used to connect sentences with important aspects;
- WTN [25]: A Wasserstein-based transfer network is used to obtain domain-invariant features;
- PTASM [43]: The attention-sharing mechanism and parameter transferring method are used for CDSC;
- DWE w/o BERT: The BERT word embedding is removed from our proposed model;
- DWE w/o word2vec: The word2vec word embedding is removed from our proposed model.

Table 3 records the classification accuracy of different models on Dataset 1. The results show that our proposed model DWE achieves the best performance on 11 cross-domain pairs. Our model outperforms DANN by 12.24%, AMN by 9.64%, DAS by 9.44%, HATN by 6.74%, IATN by 5.64%, WTN by 1.14%, and PTASM by 0.44% on average. DAS uses entropy minimization and self-integration methods to refine its classifier, which improves the experimental results compared with DANN and AMN. The addition of attention has greatly improved HATN and IATN compared with DAS, reflecting the effectiveness of the attention mechanism. Both WTN and PTASM have applied BERT to CDSC, which has been greatly improved compared with previous methods. WTN is based on Wasserstein distance as a domain discrepancy learning module, while PTASM uses an attention transfer mechanism and hierarchical attention to improve target domain classification. Different from previous methods, our proposed model uses dual-word embedding to make up for the deficiency of single word embedding. Our model also considers both transferable semantic information and syntactic information, which may be the reason for the improvement of our model.

Table 3. Classification accuracy of various models on Dataset 1.

S → T	DANN	AMN	DAS	HATN	IATN	WTN	PTASM	DWE
B → D	0.8330	0.8450	0.8390	0.8590	0.8680	0.9090	0.9012	**0.9150**
B → K	0.7920	0.8090	0.8220	0.8470	0.8590	0.8840	0.9060	**0.9100**
B → E	0.7730	0.8030	0.8120	0.8490	0.8650	0.8960	0.9010	**0.9075**
D → B	0.8050	0.8360	0.8190	0.8600	0.8700	0.9080	0.8990	**0.9125**
D → E	0.7980	0.8050	0.8160	0.8510	0.8690	**0.9150**	0.9110	**0.9150**
D → K	0.8080	0.8160	0.8140	0.8580	0.8580	0.8910	0.9080	**0.9100**
K → B	0.7490	0.8010	0.8020	0.8260	0.8470	0.9160	0.9210	**0.9250**
K → E	0.8320	0.8540	0.8590	0.8640	0.8760	0.9190	0.9190	**0.9200**
K → D	0.7680	0.8120	0.8150	0.8400	0.8440	0.8890	0.9140	**0.9150**
E → K	0.8380	0.8580	0.8490	0.8760	0.8870	**0.9320**	0.9170	0.9300
E → B	0.7350	0.7740	0.7970	0.8060	0.8180	0.9010	0.9140	**0.9175**
E → D	0.7790	0.8170	0.8020	0.8380	0.8410	0.8920	0.9070	**0.9075**
Average	0.7930	0.8190	0.8210	0.8480	0.8590	0.9040	0.9110	**0.9154**

Furthermore, we also compare our proposed model DWE with other baseline models on Dataset 2 and conduct ablation experiments simultaneously.

Table 4 records the classification accuracy on Dataset 2. We can see that our model DWE has the best performance among all cross-domain pairs. Our model outperforms AuxNN by 9.5%, DAS by 6.5%, and WTN by 3.8% on average, which demonstrates the effectiveness of our proposed model. On the other hand, after removing BERT word embedding and word2vec word embedding, the average performance decreases by 8.9% and 1.9%, respectively, which demonstrates the validation of the proposed dual-word embedding. The possible reason is that the single word embedding causes the model to lose part of the information, especially after removing the BERT word embedding, where a large amount of context-related information is lost.

Table 4. Classification accuracy of various models on Dataset 2.

S → T	AuxNN	DAS	WTN	DWE w/o BERT	DWE w/o word2vec	DWE
BK → BT	0.478	0.547	0.576	0.5160	0.558	**0.588**
BK → E	0.482	0.539	0.579	0.504	0.559	**0.587**
BK → M	0.488	0.535	0.582	0.551	0.587	**0.603**
BT → BK	0.585	0.633	0.640	0.550	0.643	**0.655**
BT → E	0.591	0.598	0.631	0.571	0.650	**0.654**
BT → M	0.536	0.560	0.576	0.534	0.600	**0.615**
M → BK	0.582	0.608	0.623	0.591	0.686	**0.692**
M → BT	0.469	0.497	0.545	0.499	0.588	**0.595**
M → E	0.494	0.529	0.545	0.485	0.583	**0.603**
E → BK	0.577	0.552	0.588	0.570	0.579	**0.646**
E → BT	0.544	0.560	0.590	0.544	0.644	**0.654**
E → M	0.523	0.554	0.561	0.505	0.577	**0.592**
Average	0.529	0.559	0.586	0.535	0.605	**0.624**

4.4. Case Study

To demonstrate the role of the proposed DWE model, we selected a piece of data from BK as our case analysis and compared it with WTN when BT was the source domain and BK was the target domain. Figure 2 shows the attention weights of the DWE and WTN for the sample. The darker the color, the higher the attention weight.

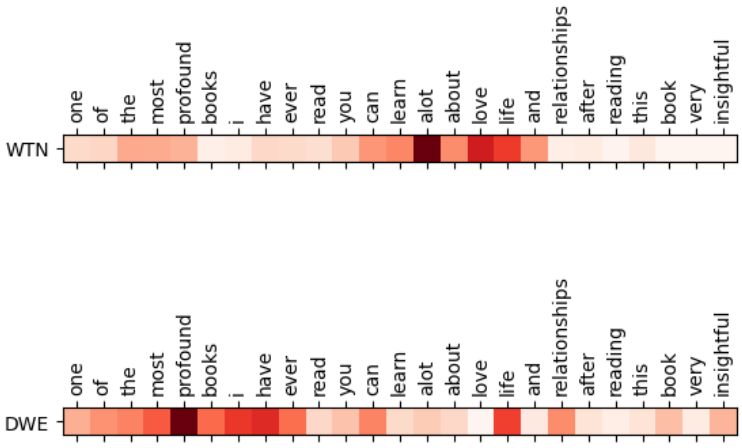

Figure 2. Case Study of Book Domain.

Figure 2 shows that the WTN model focuses more on "alot" and "love," while our proposed DWE model focuses more on the most important sentiment word "profound".

The main reason may be that the syntactic module we added allows "profound" and "book" to establish a syntactic connection, thereby focusing on the more important sentiment word.

4.5. Visualization of Feature Representation

In this section, we visualize the data in two cross-domain pairs, namely, M → BK and BT → E in Dataset 2. Figure 3 shows the feature representation of M as the source domain and BK as the target domain. Figure 4 shows the feature representation of BT as the source domain and E as the target domain.

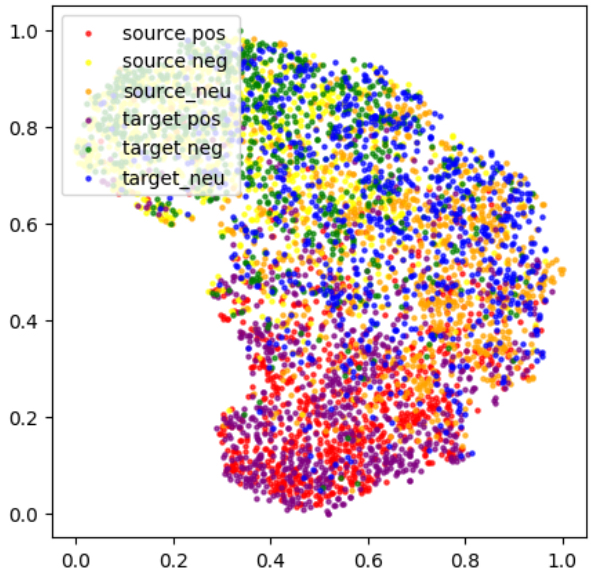

Figure 3. Visualization of feature representation on M → BK.

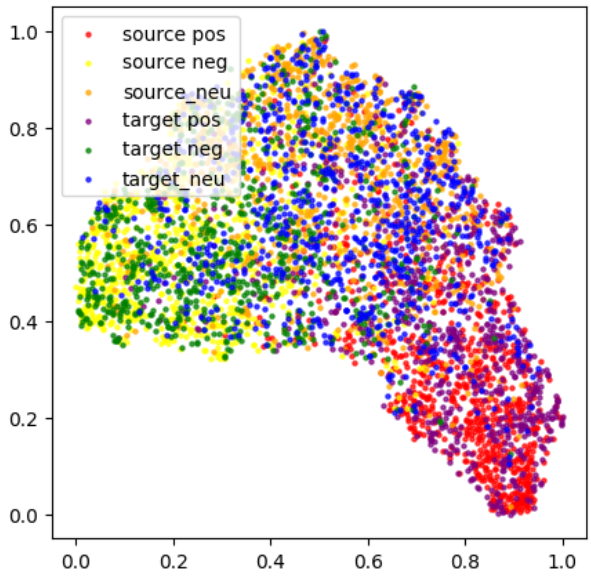

Figure 4. Visualization of feature representation on BT → E.

Figures 3 and 4 show that the sample features of two different domains are aligned. No obvious boundary exists between the two domains, and distinguishing between them is difficult. This condition shows that the two domains can share the learned feature representation, and the information from the source domain can be transferred to the target domain.

5. Conclusions

In this paper, we proposed a dual-word embedding model considering syntactic information for CDSC. The dual-word embedding is obtained through BERT and word2vec; then, the transferable syntactic information and semantic information are obtained by combining dual channel and adversarial training. Experiments showed that our model achieved better results on two real-world datasets. In future work, we will apply the model to cross-domain aspect-based sentiment analysis.

Author Contributions: Conceptualization, Z.L. and Y.X.; methodology, Z.L.; formal analysis, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, X.H.; supervision, Y.X. and X.H.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Characteristic Innovation Projects of Guangdong Colleges and Universities (Nos. 2018KTSCX049) and the Science and Technology Plan Project of Guangzhou under Grant Nos. 202102080258 and 201903010013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167.
2. Tang, D.; Qin, B.; Liu, T. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
3. Gou, J.; He, X.; Lu, J.; Ma, H.; Ou, W.; Yuan, Y. A class-specific mean vector-based weighted competitive and collaborative representation method for classification. *Neural Netw.* **2022**, *150*, 12–27. [CrossRef] [PubMed]
4. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004.
5. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. *arXiv* **2002**, arXiv:cs/0205070.
6. Gou, J.; Yuan, X.; Du, L.; Xia, S.; Yi, Z. Hierarchical Graph Augmented Deep Collaborative Dictionary Learning for Classification. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25308–25322. [CrossRef]
7. Zhang, K.; Zhang, K.; Zhang, M.; Zhao, H.; Liu, Q.; Wu, W.; Chen, E. Incorporating Dynamic Semantics into Pre-Trained Language Model for Aspect-based Sentiment Analysis. *arXiv* **2022**, arXiv:2203.16369.
8. Cambria, E.; Das, D.B.; Yopadhyay, S.; Feraco, A. Affective computing and sentiment analysis. In *A Practical Guide to Sentiment Analysis*; Springer: Cham, Switzerland, 2017; pp. 1–10.
9. Wang, D.; Jing, B.; Lu, C.; Wu, J.; Liu, G.; Du, C.; Zhuang, F. Coarse alignment of topic and sentiment: A unified model for cross-lingual sentiment classification. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *32*, 736–747. [CrossRef]
10. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, . [CrossRef]
11. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
12. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
13. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
15. Chi, E.A.; Hewitt, J.; Manning, C.D. Finding universal grammatical relations in multilingual BERT. *arXiv* **2020**, arXiv:2005.04511.
16. Guarasci, R.; Silvestri, S.; De Pietro, G.; Fujita, H.; Esposito, M. BERT syntactic transfer: A computational experiment on Italian, French and English languages. *Comput. Speech Lang.* **2022**, *71*, 101261. [CrossRef]
17. Du, C.; Sun, H.; Wang, J.; Qi, Q.; Liao, J. Adversarial and domain-aware bert for cross-domain sentiment analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
18. Huang, B.; Carley, K.M. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv* **2019**, arXiv:1909.02606.

19. Blitzer, J.; Dredze, M.; Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 23–30 June 2007; pp. 440–447.
20. Pan, S.J.; Ni, X.; Sun, J.-T.; Yang, Q.; Chen, Z. Cross-domain sentiment classification via spectral feature alignment. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010.
21. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 2096–2030.
22. Zhang, K.; Zhang, H.; Liu, Q.; Zhao, H.; Zhu, H.; Chen, E. Hierarchical Attention Transfer Network for Cross-domain Sentiment Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
23. Zhang, K.; Zhang, H.; Liu, Q.; Zhao, H.; Zhu, H.; Chen, E. Interactive attention transfer network for cross-domain sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33.
24. Yang, C.; Zhou, B.; Hu, X.; Chen, J.; Cai, Q.; Xue, Y. Dual-Channel Domain Adaptation Model. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Melbourne, VIC, Australia, 14–17 December 2021.
25. Du, Y.; He, M.; Wang, L.; Zhang, H. Wasserstein based transfer network for cross-domain sentiment classification. *Knowl.-Based Syst.* **2020**, *204*, 106162. [CrossRef]
26. Fu, Y.; Liu, Y. Domain adaptation with a shrinkable discrepancy strategy for cross-domain sentiment classification. *Neurocomputing* **2022**, *494*, 56–66. [CrossRef]
27. Wu, M.; Pan, S.; Zhu, X.; Zhou, C.; Pan, L. Domain-adversarial graph neural networks for text classification. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019.
28. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
29. Zhu, S.; Zhou, C.; Pan, S.; Zhu, X.; Wang, B. Relation structure-aware heterogeneous graph neural network. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019.
30. Zhu, S.; Zhou, L.; Pan, S.; Zhou, C.; Yan, G.; Wang, B. GSSNN: Graph smoothing splines neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34.
31. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
32. Vashishth, S.; Dasgupta, S.S.; Ray, S.N.; Talukdar, P. Dating documents using graph convolution networks. *arXiv* **2019**, arXiv:1902.00175.
33. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
34. Zhou, M.; Liu, D.; Zheng, Y.; Zhu, Q.; Guo, P. A text sentiment classification model using double word embedding methods. *Multimed. Tools Appl.* **2020**, *81*, 18993–19012. [CrossRef]
35. Vuong Nguyen, L.; Nguyen, T.H.; Jung, J.J.; Camacho, D. Extending collaborative filtering recommendation using word embedding: A hybrid approach. *Concurr. Comput. Pract. Exp.* **2021**, e6232. [CrossRef]
36. Wang, H.; Zuo, Y.; Li, H.; Wu, J. Cross-domain recommendation with user personality. *Knowl.-Based Syst.* **2021**, *213*, 106664. [CrossRef]
37. Naderalvojoud, B.; Sezer, E.A. Sentiment aware word embeddings using refinement and senti-contextualized learning approach. *Neurocomputing* **2020**, *405*, 149–160. [CrossRef]
38. Jawahar, G.; Sagot, B.; Seddah, D. What does BERT learn about the structure of language? In Proceedings of the ACL 2019-57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
39. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
40. He, R.; Lee, W.S.; Ng, H.T.; Dahlmeier, D. Adaptive semi-supervised learning for cross-domain sentiment classification. *arXiv* **2018**, arXiv:1809.00530.
41. Yu, J.; Jiang, J. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016.
42. Li, Z.; Zhang, Y.; Wei, Y.; Wu, Y.; Yang, Q. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, Australia, 19–25 August 2017.
43. Zhao, C.; Wang, S.; Li, D.; Liu, X.; Yang, X.; Liu, J. Cross-domain sentiment classification via parameter transferring and attention sharing mechanism. *Inf. Sci.* **2021**, *578*, 281–296. [CrossRef]