Cross-domain Sentiment Encoding through Stochastic Word Embedding

Yanbin Hao, Tingting Mu, *Member, IEEE,* Richang Hong, *Member, IEEE,*
Meng Wang, *Senior Member, IEEE,* Xueliang Liu, John Y. Goulermas, *Senior Member, IEEE*

Abstract—Sentiment analysis is an important topic concerning identification of feelings, attitudes, emotions and opinions from text. To automate such analysis, a large amount of example text needs to be manually annotated for model training. This is laborious and expensive, but the cross-domain technique is a key solution to reducing the cost by reusing annotated reviews across domains. However, its success largely relies on the learning of a robust common representation space across domains. In the recent years, significant effort has been invested to improve the cross-domain representation learning by designing increasingly more complex and elaborate model inputs and architectures. We support that it is not necessary to increase design complexity as this inevitably consumes more time in model training. Instead, we propose to explore the word polarity and occurrence information through a simple mapping and encode such information more accurately whilst managing lower computational costs. The proposed approach is unique and takes advantage of the stochastic embedding technique to tackle cross-domain sentiment alignment. Its effectiveness is benchmarked with over ten data tasks constructed from two review corpora and it is compared against ten classical and state-of-the-art methods.

Index Terms—Cross domain, sentiment classification, word/document embedding, similarity matrix, stochastic neighbor embedding.

1 Introduction

Sentiment classification plays a significant role in many applications related to opinion mining and sentiment analysis [1], such as opinion extraction and summarization [2], [3], review spam identification [4], user feeling analysis [5], contextual advertising [6], etc. The goal of sentiment classification is to automatically identify the sentiment polarity of a given text object, for instance, in terms of being positive, negative or neutral. Typical examples of such text objects include product reviews, which are generated by movie viewers, hotel customers, merchandise buyers, etc. The emotional tendency modeled through identifying sentiment polarity of the reviews can serve as a succinct yet informative indicator of the consumer attitude and opinion. This can potentially result in not only improved efficiency in the information sharing between the users, but also improved business solutions and services.

Focusing on reviews of a target product, a standard sentiment classifier can be built by training with a set of annotated example reviews of this product. Here, annotation refers to the process of assigning each review example a ground-truth sentiment polarity label. The sentiment polarities of new reviews for the same product can then be predicted by this trained classifier [7], [8]. Performance of such a system heavily relies on the availability and quality of the labeled example reviews. However, the process of manually annotating explosively growing online product reviews is very expensive and can be impractical. Therefore, there has been increasing interest on studying effective ways of reusing labeled reviews across different products. This is known as cross-domain sentiment classification, where a domain is referred to as a collection of reviews for a particular product.

A straightforward baseline approach for cross-domain sentiment classification is to directly apply a classifier trained using the labeled reviews of other products (source domain) to classify reviews of the target product (target domain), through comparing the words contained by the reviews (e.g., the bag-of-word features). Such an approach though, does not consider the fact that different sets of words can be used to express sentiment for different types of products. For example, people often use "excellent", "thrilling" and "boring" to express their opinions for books, while use "compact", "blurry" and "sharp" for electronics. Because of this, a sentiment classifier trained using book reviews performs poorly on classifying electronics reviews [9], as it does not consider the change of words for expressing sentiment across domains.

From the machine learning point of view, using different sets of words to express sentiments in different domains, is equivalent to training sentiment classifiers in different feature spaces. To enable a classifier trained in a source space to be usable in the target space, one effective solution is to create an alignment between the spaces by exploiting their shared and distinct characteristics. Previous research pursued space alignment and correspondence through analyzing the sentiment words that are commonly used across domains for expressing sentiments (known as pivots), and also domain- and topic-specific words that are uniquely associated to a particular domain [10]–[16]. Pivots behave

Y. Hao is with the Department of Computer Science, City University of Hong Kong, Hong Kong, 999077. Email: haoyanbin@hotmail.com.
T. Mu is with the School of Computer Science, The University of Manchester, Kilburn Building, Oxford Road, Manchester, UK, M13 9PL. Email: tingting.mu@manchester.ac.uk.
R. Hong, M. Wang and X. Liu are with the School of Computer and Information, Hefei University of Technology, Hefei, 230009, China. Email: {hongrc.hfut, eric.mengwang, liuxueliang1982}@gmail.com.
J.Y. Goulermas is with the Department of Computer Science, The University of Liverpool, Ashton Building, Liverpool, UK, L69 3BX. Email: j.y.goulermas@liverpool.ac.uk.

as universal sentiment indicators and always carry the same sentiment information in different domains while occurring frequently in all domains; typical examples include "excellent", "well" and "disappointing". Domain-specific words are more specialized at expressing sentiments in a particular domain. For instance, "sharp" is a domain-specific word mostly used in kitchenware reviews, and "realistic" in video game reviews. A typical strategy to establish space alignment is to map the original source and target spaces to a new common space by using the pivots as a bridge [10], [17]. The generated space aims at reduced mismatch in word usage and reduced gap between domain-specific words across domains. After embedding the reviews into the new space, a sentiment classifier trained using the labeled source reviews is expected to provide robust prediction performance for the target reviews.

Some existing cross-domain approaches cannot achieve effective domain transfer without supervision and require a small amount of labeled reviews in the target domain to boost the performance [10], [18], [19]. Some approaches [10], [17], although they compute new representation vectors for characterizing the reviews in the aligned space, are not adequately robust to provide stable performance on their own. These new representation vectors need to be combined with the original bag-of-word vectors for improvement. More recent works [11], [20] provide more effective ways of aligning word spaces of the source and target domains through spectral embedding and projection techniques. These lead to new review representations that can be independently used for cross-domain sentiment classification. Lately, there has also been rapid development of neural approaches for cross-domain sentiment classification, achieving high classification accuracy [21]–[24].

Comparing the domain adaption strategies used by various state-of-the-art techniques, we can see that, in addition to the main task of sentiment classification, they usually enhance their learning through preparing extra tasks like detecting whether a pivot co-occurs with a domain-specific word, whether the different versions of the same pivot word in different domains possess similar enough representation vectors, whether a review contains a pivot, or whether the reviews from the source and target domains can be distinguished in the representation space, and so on; we refer to these as the auxiliary learning tasks. The learning algorithms are mostly built on spectral approaches which explore and preserve inherent data structure through matrix decompositions [10], [11], [17], or neural networks which directly learn the review representations through structured processing of the content words based on different network architectures and exhaustive training [21], [23]–[25].

We argue that instead of creating many auxiliary learning tasks and constructing complex models with elaborate design and input configurations, satisfactory domain adaptation can be achieved by preserving simple polarity and occurrence information of words in reviews. These are actually parts of the classical information utilized in early cross-domain sentiment classification works, e.g., [10], [17], that however failed to achieve good performance. We support that the unsatisfactory performance of past endeavours were potentially caused by the employed spectral approaches that were incapable of preserving accurately the desired information in their embedding spaces. Similar observations on poor neighbor preservation ability of spectral embeddings are also reported in the data visualization field [26]. To tackle this issue, we propose a novel cross-domain sentiment representation learning model with its design inspired by the stochastic neighbor embedding method [27]. It is built upon a simple mapping architecture to ease the computational cost, but we propose a more sophisticated approach for optimizing the mapping variables to achieve more accurate similarity structure preservation. Specifically, it involves the following design elements:

- The mapping layer takes the standard word embeddings, which encode the word co-occurrence statistics collected from a large corpus of general English text, as the input, aiming at reducing the algorithm introduced bias by taking advantage of general language patterns.
- Various similarity structures between words and between reviews are explored, that result in multiple conditional probability matrices encoding polarity, co-occurrence and content information at both word and document levels. These matrices are designed to effectively capture the shared and distinct characteristics of the source and target domains through analyzing the special groups of pivots and domain-specific words.
- A composite Kullback-Leibler divergence score is designed to preserve these similarity structures in the mapped embedding space realizing a multi-objective optimization.

The experimental results show that our proposed approach significantly improves the quality of the cross-domain representations of both words and reviews, resulting in significantly improved sentiment classification performance. In many cases, our performance is close to that of the state-of-the-art deep learning techniques, but with much less demanding training procedures.

The remaining of this paper is organized as follows. Section 2 briefly reviews some representative works in the field. Section 3 explains the proposed algorithm, including pivot and domain-specific word selection, the construction of the loss functions at both word level and document level, as well as the multi-objective formulation and optimization of the aggregate training objective function. Finally, Section 4 compares and experimentally analyzes the proposed method, while Section 5 concludes the work.

2 Related Work

We briefly review the cross-domain sentiment classification techniques relevant to our work, and some recent deep learning approaches achieving state-of-the-art performance. A recent survey on this topic can be found in [28] and a task summary in [1]. One of the most classical works in cross-domain sentiment classification is structural correspondence learning (SCL) [10]. It first models the correlations between the pivots and other word features by building a set of pivot predictors, and then computes representation vectors for the reviews based on the singular value decomposition of the predictors' weights. Another representative work is spectral

feature alignment (SFA) [17], which maps the feature vector of a review into an aligned space computed by performing spectral embedding over the bipartite graph between the domain-specific and domain-independent words (e.g., pivots) built upon their co-occurrences.

Both SCL and SFA treat the computed representation vectors for the reviews as the additional features to complement their original bag-of-word vectors. This way of augmenting the original feature vector with additional features is referred to as feature expansion [14]. Another feature expansion method for cross-domain sentiment classification includes the topic and sentiment labels predicted for each word to a joint sentiment-topic model as the additional features [13]. Differently, the work in [14] includes a set of base words selected from a sentiment-sensitive thesaurus based on a ranking score as the additional features, where the scores can be used as the feature values. Instead of working with the direct co-occurrence counts, [29] obtains additional features by modeling a distribution-based association between a domain-independent word and a different word in each domain, where additional feature prediction is performed using a binary classifier.

In many cases, the representation vectors of the reviews or the predictions made over the reviews can be directly computed from the representation vectors of the words they contain, e.g., by adopting approaches as suggested in [11], [20], [30]. Therefore, development in cross-domain word representation learning greatly facilitates review sentiment analysis. There has been an increasing interest in this topic where researchers attempt to pursue more effective ways of reducing the mismatch between different domains while maintaining the distinct characteristics of each domain. For instance, the unsupervised cross-domain word representation learning [30] computes different versions of word representations for different domains, but attempts to bring the different versions close to each other for the pivots and meanwhile distinguishes between domain-specific words in terms of whether they appear in the local context of the pivots. In order to generate stronger representation vectors for the reviews, [11] improves their word representation learning technique by studying effective feature mapping rules by exploring not only information in the word space but also local geometry and closeness structure between friends, enemies and unlabeled reviews in the document space. Building upon the skip-gram model [31], [32] first learns word representations in the source domain using the standard skip-gram model. Then the word representation learning completes in the target domain using a modified skip-gram model with a regularization term added to the original loss. This work is similar to [11], [30] in the sense of bringing closer the source and target representation pivot vectors, but different in that such alignment is achieved in a controlled manner through designing a significance function. This function attempts to quantify the degree of knowledge transfer from the source domain to the target domain for each pivot. More recently, [20] proposes a projection method to modify the word representation vectors precomputed by a standard word embedding technique. Similar to other works, alignment of common words in different domains is taken into account, but additionally, a sentiment classification error is incorporated to the projection optimization objective function.

Deep learning models constitute another important group of techniques for cross-domain sentiment classification. Various neural network architectures are exploited to compute the representation vectors for reviews, such as, stacked auto-encoders (SAE) [33], [34], fully connected neural networks [21], convolutional neural networks (CNN) [25], and also recurrent neural networks (RNN) typically with long short-term memory (LSTM) units [23]. To enhance the expressive power of neural networks, there are various works investigating the use of recently developed attention and memory mechanisms [22], [24]. Additionally, a great deal of effort has been invested on the design of effective training methods for network optimization. Early works [33], [34] follow the de-noising auto-encoder (DAE) training [35] to obtain unified representation vectors for review in both source and target domains in an unsupervised manner. Subsequently, [21] proposes to replace the reconstruction error with the multi-kernel variant of maximum mean discrepancy to improve the DAE training, and further extends it to supervised training by adding a sentiment classification loss computed from the labeled reviews in the source domain. To achieve the goal of reducing the mismatch between the source and target domains, various auxiliary learning tasks are defined, in addition to classifying the labeled reviews in the source domain. For instance, [25] trains the network to predict whether a given sentence contains a pivot using the other words. Another major auxiliary task type is to distinguish the reviews in the source domain from those in the target domain based on the learned review representations. This task is usually taken into account by designing additional loss functions and (or) by conducting adversarial training [22]–[24], [36], [37].

3 Proposed Method

In general, the raw information we can directly collect for each review (referred to as a document) is a bag-of-words vector $\boldsymbol{x} = [x_1, \ldots, x_n]^T$, where n denotes the total number of words in the vocabulary, and x_t the number of occurrences that the tth word appears in this document[1]. Cross-domain sentiment classification operates by training a sentiment classifier using the labeled reviews from the source domain, and the classifier is expected to offer satisfactory classification performance for reviews in the target domain. The key to achieving this, is to learn robust representations (or called features) for reviews in both domains, so that the classifier can be transferred across domains without major performance loss.

3.1 Model Overview

Instead of learning representation vectors for both words and reviews, we follow the approach of computing representation vectors for reviews (referred to as document embedding vectors) directly from the representation vectors of words (referred to as word embedding vectors), with the

1. In this work, we simply use word frequencies to characterize each document. Alternative features, such as the term frequency-inverse document frequency (tf-idf) values can also be used.

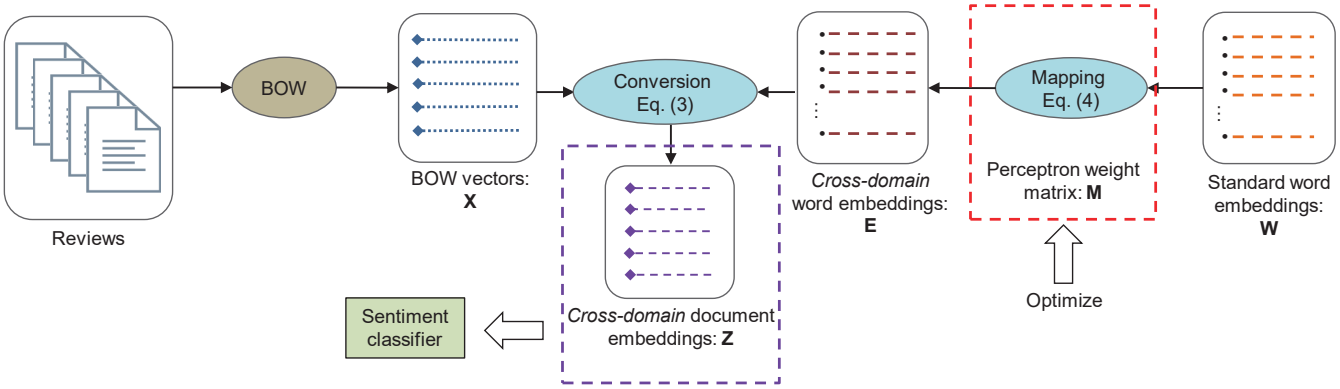


Fig. 1: Overall structure of the proposed cross-domain sentiment classification model, where BOW stands for "bag-of-words".

benefit of reduced number of model parameters to optimize. Specifically, letting $\boldsymbol{e}_t = [e_{t1}, \ldots, e_{td}]^T$ denote the d-dimensional embedding vector of the tth word in the vocabulary list, the document embedding vector $\boldsymbol{z} = [z_1, \ldots, z_d]^T$ is computed by

$$\boldsymbol{z} = \frac{\sum_{t=1}^{n} x_t \boldsymbol{e}_t}{\sum_{s=1}^{n} x_s}. \tag{1}$$

This is a weighted average of the word embedding vectors $\{\boldsymbol{e}_t\}_{t=1}^n$, where each element of $\boldsymbol{x}$ is used as the weight to favor more frequently appeared words.

Subsequently, the problem is reduced to the learning of $\{\boldsymbol{e}_t\}_{t=1}^n$ that can optimally support cross-domain sentiment classification; these vectors are referred to as the cross-domain word embedding vectors. Other than learning these vectors from scratch, a strategy similar to [20] is adopted. We compute $\{\boldsymbol{e}_t\}_{t=1}^n$ by modifying the standard word embeddings that are computed without considering domain adaptation. Let $\boldsymbol{w}_t = [w_{t1}, \ldots, w_{tk}]^T$ denote the standard embedding vector of the tth word in the vocabulary list. A single layer perceptron (SLP) is employed[2] to modify $\boldsymbol{w}_t$, resulting in the following cross-domain word embedding vector

$$\boldsymbol{e}_t = \text{sigmoid}\left(\mathbf{M}^T \boldsymbol{w}_t\right), \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{k \times d}$ denotes the perceptron weight matrix. As pointed out in [17], word co-occurrence statistics provide significant information for domain alignment. Therefore, we choose the GloVe embeddings [38] as our $\boldsymbol{w}_t$, which are computed from aggregated global word-to-word co-occurrence statistics captured in a large corpus of general English text.

The whole process can be expressed in matrix notation, by letting $\mathbf{E}$ denote the $n \times d$ cross-domain word embedding matrix with its rows storing $\{\boldsymbol{e}_t\}_{t=1}^n$, and $\mathbf{W}$ the $n \times k$ standard word embedding matrix with its rows storing $\{\boldsymbol{w}_t\}_{t=1}^n$. Given N denoting the total number of reviews in both domains, the $N \times n$ matrix $\mathbf{X}$ and $N \times d$ matrix $\mathbf{Z}$ store in their rows the bag-of-words vectors $\{\boldsymbol{x}_i\}_{i=1}^N$ and the document embedding vectors $\{\boldsymbol{z}_i\}_{i=1}^N$, respectively. Conversions between these matrices, equivalent to Eq. (1) and Eq. (2), are given as

$$\mathbf{Z} = \mathbf{\Lambda}(\mathbf{X})^{-1}\mathbf{X}\mathbf{E}, \tag{3}$$

$$\mathbf{E} = \text{sigmoid}(\mathbf{W}\mathbf{M}), \tag{4}$$

where $\mathbf{\Lambda}(\cdot)$ returns a diagonal matrix with diagonal elements being the row sums of the input matrix. In Figure 1 we present the main structure of our model, that demonstrates mappings between the document embeddings $\mathbf{Z}$, cross-domain word embeddings $\mathbf{E}$, and the standard word embeddings $\mathbf{W}$. The problem is finally reduced to the learning of the optimal weight matrix $\mathbf{M} \in \mathbb{R}^{k \times d}$ that should be tailored to cross-domain sentiment classification.

3.2 Alignment Loss in Word Space

A word selection process is first implemented to identify a set of g pivots and a set of f domain-specific words from all the words that appear in the corpus. A word alignment loss is designed to embed the selected pivot and domain-specific words in locations, such that distances between them reflect their sentiment, co-occurrence and semantic based similarities.

3.2.1 *Pivot and Domain-Specific Word Selection*

Built upon [39], a two-stage pivot selection strategy is applied, which also utilizes sentiment information offered by the labeled documents from the source domain. Firstly, the candidate pool of pivot words is pinned to a set of common words that appear in more than θ_c documents in each domain. Then, the pivot words are selected from the candidate pool based on an entropy measure conditioned on document sentiment polarity, as

$$H\left(\text{word}_i\right) = -\sum_{y \in Y} \log \frac{\text{count}\left(\text{word}_i, \mathfrak{D}_y\right)}{\text{count}(\text{word}_i, \mathfrak{D})}, \tag{5}$$

where Y denotes the total set of available sentiment labels, e.g., $Y = \{\text{positive}, \text{negative}\}$, $\mathfrak{D}$ denotes the total set of labeled documents in the source domain, and $\mathfrak{D}_y \subset \mathfrak{D}$ contains documents labeled as class $y \in Y$. The measure

2. The results have shown that the proposed method equipped with a simple but effective choice of SLP mapping is sufficient to obtain satisfactory performance improvement (see results reported in Tables 1 and 2). However, we would like to mention that the proposed method is general and can accommodate any type of continuous mapping of the form $\boldsymbol{e}_t = \boldsymbol{\phi}(\boldsymbol{w}_t)$, such as ones based on alternative neural network architectures and activation functions.

count$(\cdot,\cdot)$ denotes the number of occurrences of the left argument (word) appearing in the right argument (document set). According to the measure, the candidate words that appear more frequently in one sentiment class but not in the others, possess higher entropy values and are therefore selected as pivot words.

The remaining words, referred to as the non-pivot words, constitute the candidate pool of the domain-specific words. As argued in [17], among the non-pivot words, those that frequently co-occur with a pivot word in a domain, usually retain similarly rich sentiment information. Therefore, for each pivot word, we select the top θ_k non-pivot words that co-occur most frequently with it in each domain. By examining different pivot words under different domains, a total of f non-pivot words are selected and are treated as the domain-specific words.

3.2.2 Pivot Polarity Graph

We introduce the concept of pivot polarity for the universal sentiment indicators. It represents the role of a pivot in sentiment classification, and should remain unchanged across domains. It is defined according to whether the pivot is positively or negatively correlated with the review sentiment. We propose to collect the pivot polarities by a linear sentiment classifier[3] $y = \boldsymbol{a}^T\boldsymbol{x} + b$, where b is the bias parameter and $\boldsymbol{a}$ the weight vector, trained with the labeled reviews from the source domain each characterized by its bag-of-words vector. The weight vector $\boldsymbol{a}$ suggests the polarity information. Letting L_i denote the polarity of the ith pivot, we have

$$L_i = \begin{cases} +1, & \text{if } a_{I_i} > 0, \\ -1, & \text{if } a_{I_i} < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where I_i denotes the position of the ith pivot in the bag-of-words vector. The quantity $L_i = +1$ suggests this pivot is positively correlated with the review sentiment, and $L_i = -1$ denotes a negative correlation, while $L_i = 0$ suggests lack of contribution to sentiment discrimination. Although such information is learned in the source domain, it can be safely transferred to the target domain due to the universal role of pivots.

We encode the pivot polarity information using Euclidean distances between the cross-domain word representation vectors, by locating pivots with the same polarity in proximity to each other. For instance, the two pivots "excellent" and "well" with the same polarity will be mapped closeby in the learned representation space. Such closeness information can be stored in a pivot polarity graph. The g pivots are its nodes, and the adjacency matrix $\mathbf{R}_P = [r_{ij}^{(P)}]$ is defined as a binary one as

$$r_{ij}^{(P)} = \max(0, L_i L_j). \tag{7}$$

The truncation $\max(0,\cdot)$ makes sure that the negative relationship between pivots with opposite polarities is not encoded, because such information has been observed to be less reliable.

3. In this work, we just use an l_2-regularized logistic regression.

3.2.3 Co-occurrence Bipartite Graph

Additionally, we consider the linkage between pivots and domain-specific words. As explained in Section 3.2.1, each domain-specific word is selected according to a pivot by examining their overall co-occurrences in an entire domain. This necessitates a $g \times f$ binary link matrix $\mathbf{R}_D = [r_{ij}^{(D)}]$ between the pivots and domain-specific words, where

$$r_{ij}^{(D)} = \begin{cases} 1, & \text{if } j \overset{\theta_k}{\Rightarrow} i, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The notation $j \overset{\theta_k}{\Rightarrow} i$ indicates that the jth domain-specific word is among the top θ_k most co-occurring non-pivot words of the ith pivot in a domain. The link matrix results in a co-occurrence based bipartite graph between the pivots and domain-specific words. As before, we attempt to encode such link information through Euclidean distances. When two words are linked, they are to be mapped close to each other in the embedding space. For instance, the distance between the domain-specific word "sharp" and the pivot "excellent" will be small in that space, as they co-occur frequently in kitchenware reviews.

3.2.4 Stochastic Word Graph Preservation

In our model, we control the word locations in the embedding space using the pivot polarity graph $\mathbf{R}_P$ and the co-occurrence bipartite graph $\mathbf{R}_D$. Guided by $\mathbf{R}_P$, words mapped close to each other in the embedding space most likely share the same polarity. Further guided by $\mathbf{R}_D$, polarity information carried by the pivots is transferred to the domain-specific words. For instance, "excellent " and "sharp" are close, and therefore "sharp" can inherit the polarity label of "excellent". Below we explain how to achieve this based on the stochastic neighbor embedding technique.

To simplify the notations, $\mathbf{R}_P$ and $\mathbf{R}_D$ are combined into a single $(g+f)\times(g+f)$ binary matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_P & \mathbf{R}_D \\ \mathbf{R}_D^T & \mathbf{0}_{f\times f} \end{bmatrix}, \tag{9}$$

where $\mathbf{0}_{f\times f}$ denotes a zero matrix of size $f \times f$. To improve generalization, we further smoothen the binary matrix $\mathbf{R}$ with word semantic information gathered from general English text, by utilizing word similarities computed from the standard word embedding vectors $\{\boldsymbol{w}_i\}_{i=1}^{g+f}$. This results in a modified similarity matrix $\hat{\mathbf{R}} = [\hat{r}_{ij}]$ defined as

$$\hat{r}_{ij} = \frac{\alpha r_{ij}}{\sum_{t\neq i} r_{it}} + (1-\alpha)p_{ij}, \tag{10}$$

where

$$p_{ij} = \frac{\exp\left(-\frac{\|\boldsymbol{w}_i - \boldsymbol{w}_j\|_2^2}{2\sigma_i^2}\right)}{\sum\limits_{t\neq i}\exp\left(-\frac{\|\boldsymbol{w}_i - \boldsymbol{w}_t\|_2^2}{2\sigma_i^2}\right)}. \tag{11}$$

The parameter $0 \leq \alpha \leq 1$ controls the preference degree over the hard polarity and co-occurrence links and the soft semantic similarities, while σ_i controls how fast the similarity between two vectors vanishes as their Euclidean distance increases. We use the integer perplexity parameter K to set σ_i. The value of σ_i that offers a Shannon entropy $-\sum_{j\neq i} p_{ij}\log_2 p_{ij}$ closest to $\log_2 K$ is used, where K can

be interpreted as a smooth measure of the effective number of neighbors [27].

Preserving the similarity information contained by $\hat{\mathbf{R}}$ in the embedding space can be achieved by maximizing the matching degree between $\hat{\mathbf{R}}$ and an estimated similarity matrix $\mathbf{Q}_e = [q_{ij}^{(e)}]$ from the mapped embeddings given as

$$q_{ij}^{(e)} = \frac{\exp\left(-\|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2^2\right)}{\sum_{t\neq i}\exp\left(-\|\boldsymbol{e}_i - \boldsymbol{e}_t\|_2^2\right)}. \tag{12}$$

This is an effective way of formulating the estimation $\mathbf{Q}_e$, which computes the Euclidean distances between $\{\boldsymbol{e}_i\}_{i=1}^{g+f}$ and converts these to normalized similarities using a scaled Gaussian. The Kullback-Leibler (KL) divergence $\epsilon(\cdot,\cdot)$ can then be employed to measure the matching degree between the two matrices, resulting in our alignment loss function

$$L_A(\mathbf{M}) = \epsilon(\hat{\mathbf{R}}, \mathbf{Q}_e) = \tfrac{1}{2}\sum_{i\neq j}\frac{\hat{r}_{ij}}{\sum_{t\neq i}\hat{r}_{it}}\log\left(\frac{\frac{\hat{r}_{ij}}{\sum_{t\neq i}\hat{r}_{it}}}{q_{ij}^{(e)}}\right). \tag{13}$$

The operation $\frac{\hat{r}_{ij}}{\sum_{t\neq i}\hat{r}_{it}}$ normalizes $\hat{\mathbf{R}}$ to have unit row sum to match $\mathbf{Q}_e$ in that respect, so that the two matrices describe conditional probability distributions. The smaller the divergence is, the better the matching between $\hat{\mathbf{R}}$ and $\mathbf{Q}_e$ is.

3.3 Neighbor Loss in Document Space

In addition to the word-level modeling as in the above section, we further improve the learning by exploiting information at the document level. We study the labeled reviews from the source domain, unlabeled reviews from the source domain, and the ones from the target domain separately, with $I^{(\mathrm{S_l})}$, $I^{(\mathrm{S_u})}$ and $I^{(\mathrm{T})}$ correspondingly denoting their index sets. We also use $\mathrm{S_l}$, $\mathrm{S_u}$ and T as the superscript symbols in all relevant notations for clarity. Three individual neighboring graphs are constructed in Section 3.3.1 for reviews in $I^{(\mathrm{S_l})}$, $I^{(\mathrm{S_u})}$ and $I^{(\mathrm{T})}$. A stochastic embedding based loss function is developed in Section 3.3.2 to map reviews that possess the same sentiment polarity and (or) similar word content close to each other in the document embedding space.

3.3.1 Document Neighboring Graphs

Firstly, we construct the neighboring graph for the labeled reviews from the source domain, utilizing the friend and enemy concepts proposed in [40]. Its adjacency matrix is denoted by $\mathbf{S}^{(\mathrm{S_l})} = [s_{ij}^{(\mathrm{S_l})}]$. Among the objects that are within a local neighborhood of each other, friends are regarded those from the same class, whereas enemies the ones from different classes. We use the cosine coefficient to compute review similarity from their bag-of-words vectors, based on which a κ-nearest-neighbor (κ-NN) search is performed to identify friends and enemies. The friend reviews possess not only similar word content but also the same sentiment polarity, and are assigned the highest similarity value of 1. The enemy reviews possess different sentiment polarity but their word content is somehow similar, and are therefore most likely boundary cases with challenging classification. To enhance the class separability, we assign the lowest similarity value of 0 to these enemy reviews, so that they can be forcefully pulled away from each other in the document embedding space. Between the non-friend and non-enemy reviews, their original cosine similarities between 0 and 1 are assigned. This results in

$$s_{ij}^{(\mathrm{S_l})} = \begin{cases} 1, & \text{if } y_i = y_j, \text{ and reviews } i,j \\ & \text{are undirected } \kappa\text{-NNs}, \\ 0, & \text{if } y_i \neq y_j, \text{ and reviews } i,j \\ & \text{are undirected } \kappa\text{-NNs}, \\ \cos(\boldsymbol{x}_i^{(\mathrm{S_l})}, \boldsymbol{x}_j^{(\mathrm{S_l})}), & \text{otherwise}, \end{cases} \tag{14}$$

where $y_i \in Y$ denotes the sentiment class label of the corresponding review.

Then, we construct the neighboring graphs for the unlabeled reviews from the source and target domains, separately, of which their adjacency matrices are denoted as $\mathbf{S}^{(\mathrm{S_u})} = [s_{ij}^{(\mathrm{S_u})}]$ and $\mathbf{S}^{(\mathrm{T})} = [s_{ij}^{(\mathrm{T})}]$. These graphs are constructed solely based on their word content, by computing the cosine coefficient from their bag-of-words vectors, resulting in

$$s_{ij}^{(\mathrm{S_u})} = \cos(\boldsymbol{x}_i^{(\mathrm{S_u})}, \boldsymbol{x}_j^{(\mathrm{S_u})}), \tag{15}$$

$$s_{ij}^{(\mathrm{T})} = \cos(\boldsymbol{x}_i^{(\mathrm{T})}, \boldsymbol{x}_j^{(\mathrm{T})}). \tag{16}$$

3.3.2 Stochastic Document Graph Preservation

Similar to the word graph preservation in 3.2.4, we attempt to preserve the review neighboring structures encoded by $\mathbf{S}^{(\mathrm{S_u})}$, $\mathbf{S}^{(\mathrm{S_u})}$ and $\mathbf{S}^{(\mathrm{T})}$ through designing a document-level loss function. Because it is to be combined with the word-level alignment loss to form a multi-objective optimization problem, we need to modify $\mathbf{S}^{(\mathrm{S_u})}$, $\mathbf{S}^{(\mathrm{S_u})}$ and $\mathbf{S}^{(\mathrm{T})}$ so that they are in a scale comparable to the word similarity matrix $\hat{\mathbf{R}}$.

Each cosine-based similarity is first converted to an angular distance, and then a similar trick to that used in Section 3.2.4 converts the distance value to a similarity value through a scaled Gaussian. Taking the labeled reviews in the source domain as an example, this gives the similarity

$$\hat{s}_{ij}^{(\mathrm{S_l})} = \frac{\exp\left(-\frac{\left(\frac{2}{\pi}\cos^{-1}\left(s_{ij}^{(\mathrm{S_l})}\right)\right)^2}{2\sigma_i^2}\right)}{\sum_{t\neq i}\exp\left(-\frac{\left(\frac{2}{\pi}\cos^{-1}\left(s_{it}^{(\mathrm{S_l})}\right)\right)^2}{2\sigma_i^2}\right)}. \tag{17}$$

The value of σ_i is controlled by the perplexity parameter K in a similar way as in Eq. (11), which is to find the closest σ_i offering $-\sum_{j\in I^{(\mathrm{S_l})}, j\neq i}\hat{s}_{ij}^{(\mathrm{S_l})}\log_2\hat{s}_{ij}^{(\mathrm{S_l})} = \log_2 K$. Similar conversions are followed for the other two document sets, and the similarities are denoted by $\hat{s}_{ij}^{(\mathrm{S_u})}$ and $\hat{s}_{ij}^{(\mathrm{T})}$. Finally, the three modified similarity matrices are denoted by $\hat{\mathbf{S}}^{(\mathrm{S_l})} = [\hat{s}_{ij}^{(\mathrm{S_l})}]$, $\hat{\mathbf{S}}^{(\mathrm{S_u})} = [\hat{s}_{ij}^{(\mathrm{S_u})}]$ and $\hat{\mathbf{S}}^{(\mathrm{T})} = [\hat{s}_{ij}^{(\mathrm{T})}]$.

The estimated review similarity in the document embedding space can be obtained in a similar way to Eq. (12), but using the document embedding vectors as the input. Let $\mathbf{Q}^{(\mathrm{S_l})} = [q_{ij}^{(\mathrm{S_l})}]$, $\mathbf{Q}^{(\mathrm{S_u})} = [q_{ij}^{(\mathrm{S_u})}]$ and $\mathbf{Q}^{(\mathrm{T})} = [q_{ij}^{(\mathrm{T})}]$ denote the three estimated similarity matrices for the three review sets.

Taking the labeled reviews from the source domain as an example, we have

$$q_{ij}^{(\mathrm{S_l})} = \frac{\exp\left(-\|\boldsymbol{z}_i^{(\mathrm{S_l})} - \boldsymbol{z}_j^{(\mathrm{S_l})}\|_2^2\right)}{\sum_{t\neq i}\exp\left(-\|\boldsymbol{z}_i^{(\mathrm{S_l})} - \boldsymbol{z}_t^{(\mathrm{S_l})}\|_2^2\right)}. \quad (18)$$

Identical approaches are followed for computing $q_{ij}^{(\mathrm{S_l})}$ and $q_{ij}^{(\mathrm{T})}$. The KL divergence scores are used to examine the matching degrees between the estimated similarity matrices $\mathbf{Q}^{(\mathrm{S_l})}$, $\mathbf{Q}^{(\mathrm{S_u})}$ and $\mathbf{Q}^{(\mathrm{T})}$ and the desired ones $\hat{\mathbf{S}}^{(\mathrm{S_l})}$, $\hat{\mathbf{S}}^{(\mathrm{S_u})}$ and $\hat{\mathbf{S}}^{(\mathrm{T})}$. This results in the following neighbor loss function

$$L_N(\mathbf{M}) = a_1\,\epsilon(\hat{\mathbf{S}}^{(\mathrm{S_l})}, \mathbf{Q}^{(\mathrm{S_l})}) + a_2\,\epsilon(\hat{\mathbf{S}}^{(\mathrm{S_u})}, \mathbf{Q}^{(\mathrm{S_u})}) + a_3\,\epsilon(\hat{\mathbf{S}}^{(\mathrm{T})}, \mathbf{Q}^{(\mathrm{T})}). \quad (19)$$

where $a_1, a_2, a_3 \geq 0$ are balancing parameters controlling the preference weights of the scores. Similar to Eq. (13), the smaller the loss is, the better matching is implied.

3.4 Model Optimization

The optimization of $\mathbf{M}$ is based on a multi-objective formulation combining the two proposed loss functions $L_A(\mathbf{M})$ and $L_N(\mathbf{M})$. By re-arranging the balancing parameters of different terms to have more convenient hyper-parameter control, we have the loss

$$L(\mathbf{M}) = \beta_1\,L_A(\mathbf{M}) + \beta_2\,\epsilon^{(\mathrm{S_l})}(\mathbf{M}) + \lambda(1-\beta_1-\beta_2)\,\epsilon^{(\mathrm{S_u})}(\mathbf{M}) + (1-\lambda)(1-\beta_1-\beta_2)\,\epsilon^{(\mathrm{T})}(\mathbf{M}) + \frac{\mu}{2}\|\mathbf{M}\|_F^2, \quad (20)$$

where $\epsilon^{(\mathrm{S_l})}$, $\epsilon^{(\mathrm{S_u})}$ and $\epsilon^{(\mathrm{T})}$ are shorthands for $\epsilon(\hat{\mathbf{S}}^{(\mathrm{S_l})}, \mathbf{Q}^{(\mathrm{S_l})})$, $\epsilon(\hat{\mathbf{S}}^{(\mathrm{S_u})}, \mathbf{Q}^{(\mathrm{S_u})})$ and $\epsilon(\hat{\mathbf{S}}^{(\mathrm{T})}, \mathbf{Q}^{(\mathrm{T})})$, $\beta_1, \beta_2, \lambda \in [0,1]$ are the balancing parameters, $\|\mathbf{M}\|_F^2$ is a Frobenius norm based regularization term, and $\mu \geq 0$ its control parameter. The matrices $\hat{\mathbf{R}}$, $\hat{\mathbf{S}}^{(\mathrm{S_l})}$, $\hat{\mathbf{S}}^{(\mathrm{S_u})}$ and $\hat{\mathbf{S}}^{(\mathrm{T})}$ are computed from the known information including the standard word embeddings $\mathbf{W}$, the bag-of-words review vectors $\mathbf{X}$ and the class information of the labeled reviews in the source domain. $\mathbf{Q}_e$, $\mathbf{Q}^{(\mathrm{S_l})}$, $\mathbf{Q}^{(\mathrm{S_u})}$ and $\mathbf{Q}^{(\mathrm{T})}$ are computed from the two cross-domain embedding matrices $\mathbf{E}$ and $\mathbf{Z}$, which are functions of the weight matrix $\mathbf{M}$. Figure 2 illustrates the information resources used to build the optimization architecture.

The differentiability of the objective $L(\mathbf{M})$ with respect to $\mathbf{M}$ allows a gradient descent algorithm to search for the optimal. For completeness, we list below the gradients in matrix form for the four losses L_A, $\epsilon^{(\mathrm{S_l})}$, $\epsilon^{(\mathrm{S_u})}$ and $\epsilon^{(\mathrm{T})}$ with respect to $\mathbf{M}$. Some basic operations are defined to simplify the gradient formulations

$$\mathbf{L}(\mathbf{A}) = \mathbf{\Lambda}(\mathbf{A}) - \mathbf{A}, \quad (21)$$
$$\mathbf{D}(\mathbf{A}) = \mathbf{\Lambda}(\mathbf{A})^{-1}\mathbf{A}, \quad (22)$$
$$\mathbf{\Gamma}(\mathbf{A}) = \mathbf{A} \circ (\mathbf{1}_{\mathbf{A}} - \mathbf{A}), \quad (23)$$
$$\mathbf{\Upsilon}(\mathbf{A}, \mathbf{B}) = \mathbf{A} - \mathbf{B} + \mathbf{A}^T - \mathbf{B}^T, \quad (24)$$

for any matrices $\mathbf{A}$ and $\mathbf{B}$ of the same size, $\mathbf{1}_{\mathbf{A}}$ a matrix of the same size as $\mathbf{A}$ with unity values, and $\circ$ being the Hadamard product. Following derivations similar to [27], we obtain

$$\frac{\partial L_A}{\partial \mathbf{M}} = \mathbf{W}_w^T\left((\mathbf{L}(\mathbf{\Upsilon}(\hat{\mathbf{R}}, \mathbf{Q}_e))\mathbf{E}_w) \circ \mathbf{\Gamma}(\mathbf{E}_w)\right), \quad (25)$$

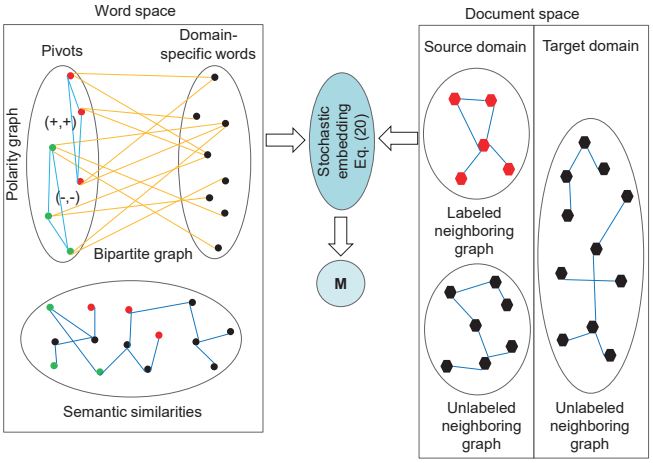


Fig. 2: Optimization framework of the proposed cross-domain sentiment classification model.

where $\mathbf{W}_w$ and $\mathbf{E}_w$ store the pre-trained and mapped word embedding vectors for the pivot and domain-specific words only. Similarly, the gradients of the other three scores are computed as

$$\frac{\partial \epsilon^{(*)}}{\partial \mathbf{M}} = \mathbf{W}^T\left((\mathbf{D}^T(\mathbf{X}^{(*)})\mathbf{L}(\mathbf{\Upsilon}(\hat{\mathbf{S}}^{(*)}, \mathbf{Q}^{(*)}))\mathbf{Z}^{(*)}) \circ \mathbf{\Gamma}(\mathbf{E})\right), \quad (26)$$

where the asterisk notationally represents the three cases for $\mathrm{S_l}$, $\mathrm{S_u}$ and T. The regularization term gradient is simply

$$\frac{\partial \|\mathbf{M}\|_F^2}{\partial \mathbf{M}} = 2\mathbf{M}. \quad (27)$$

To accelerate the training, the Delta-Bar-Delta algorithm [41] is employed to adaptively modify the learning rate during each iteration of the gradient descent update. At the $(t+1)$th iteration, the modified learning rate $\eta_{ij}^{(t+1)}$ for the ijth element of the matrix $\mathbf{M} = [m_{ij}]$ is given by

$$\eta_{ij}^{(t+1)} = \begin{cases} \eta_{ij}^{(\mathrm{t})} + \tau, & \text{if } \bar{\Delta}_{t-1}^{(ij)}\Delta_t^{(ij)} > 0, \\ (1-\xi)\,\eta_{ij}^{(\mathrm{t})}, & \text{if } \bar{\Delta}_{t-1}^{(ij)}\Delta_t^{(ij)} < 0, \\ \eta_{ij}^{(\mathrm{t})}, & \text{otherwise,} \end{cases} \quad (28)$$

where $\Delta_t^{(ij)}$ denotes the derivative $\frac{\partial \epsilon}{\partial m_{ij}}$ computed at the tth iteration. An averaged approximation is computed from the derivatives in the two previous iterations, according to

$$\bar{\Delta}_{t-1}^{(ij)} = (1-\delta)\,\Delta_{t-1}^{(ij)} + \delta\Delta_{t-2}^{(ij)}. \quad (29)$$

The learning parameters $\tau, \xi, \delta > 0$ are chosen by the user and follow the same setting as in [42]. The matrix $\boldsymbol{\eta}_t = [\eta_{ij}^{(\mathrm{t})}]$ stores all the learning rates for the weight parameters in the tth iteration. Implementation of the proposed algorithm (we refer to as CrossWord) is summarized in Algorithm 1.

The obtained solution for $\mathbf{M}$ results in a word embedding space tailored to the cross-domain sentiment classification task, where document embedding vectors of the reviews are computed and unify reviews in the source and target domains in one common feature space. Working in this feature space, a sentiment classifier trained in the source domain is expected to be robust for predictions in the target domain.

Algorithm 1 Pseudocode of CrossWord.

Input: Bag-of-words matrix $\mathbf{X}$ for documents in the source and target domains, sentiment class labels for the labeled documents in the source domain, standard word embedding matrix $\mathbf{W}$.

Output: Perceptron weight matrix $\mathbf{M}$.

Model parameters: Embedding dimension d, word selection parameters θ_c, θ_k, perplexity K, neighbor parameter κ, relevance weight α, balancing parameters λ, β_1, β_2 and regularization parameter μ.

Optimization parameters: Iteration number N_T, learning rate parameters τ, ξ, δ, and a momentum schedule $\zeta(t)$.

Word Selection: Select pivot and domain-specific words.

Initialization: Randomly initialize $\mathbf{M}$ and set $\mathbf{\Delta}_0^{(\mathbf{M})} = \mathbf{0}$.

for $t = 1$ to N_T **do**

Compute gradient $\frac{\partial L}{\partial \mathbf{M}}\big|_t$.

Update the mapping matrix by

$$\mathbf{\Delta}_t^{(\mathbf{M})} = \zeta(t)\, \mathbf{\Delta}_{t-1}^{(\mathbf{M})} - \boldsymbol{\eta}_t \circ \frac{\partial L}{\partial \mathbf{M}}\bigg|_t, \tag{30}$$

$$\mathbf{M}_{t+1} = \mathbf{M}_t + \mathbf{\Delta}_t^{(\mathbf{M})}. \tag{31}$$

end for

4 Experiments and Results

The experiments are performed under an unsupervised setting in the target domain, by training a sentiment classifier using only the labeled reviews from a source domain. Performance is evaluated using query reviews from a target domain. Ten existing methods are compared with the proposed one, each constructing representation vectors for characterizing the reviews in a different way. Among the competing methods, there are also three recently developed neural approaches, referred to as mSDA, DANN and HATN. A brief description of these competing methods is provided below:

• **Baseline1** uses a bag-of-word vector to characterize each review, where no domain adaptation is involved.

• **Baseline2** computes the document embedding vector for each review by Eq. (1), where the standard word embedding vectors trained by the GloVe model are used as $\boldsymbol{e}_t$, thus no domain adaptation is involved.

• **SCL** [10] constructs a concatenated representation vector for each review, includes the original bag-of-word vector and its projected vector learned to align the two domains.

• **SFA** [17] constructs a concatenated representation vector for each review, which includes the original bag-of-word vector and a mapped vector learned to align the domain-specific words.

• **SSE** [11] computes the document embedding vector for each review by Eq. (1), where its proposed sentiment sensitive word embedding vectors are used as $\boldsymbol{e}_t$ to achieve domain adaptation.

• **BLSE** [20] characterizes each review by averaging the mapped word embeddings through two cross-domain projection matrices.

• **MEDA** [43] computes the document embedding vector for each review by Eq. (1), where $\boldsymbol{e}_t$ is generated through an effective manifold feature learning method recently proposed by computer vision researchers.

• **mSDA** [34] constructs a concatenated representation vector for each review, which includes the original bag-of-word vector and hidden representation vectors computed by a marginalized staked de-noising auto-encoder.

• **DANN** [37] builds an end-to-end domain adversarial neural network, containing a sentiment classification layer and a domain classification layer optimized by adversarial training.

• **HATN** [24] builds an end-to-end hierarchical attention transfer network, which transfers word and sentence level emotion attentions across domains based on exploring characteristics of pivots and non-pivot words.

4.1 Datasets and Experimental Setup

Two benchmark review datasets are used to evaluate the cross-domain sentiment classification performance. The **DAT_A** dataset [44] contains product reviews collected from the Amazon website (domain A), movie reviews from IMDb (domain I) and restaurant reviews from Yelp (domain Y). Two sentiment polarities are studied: positive and negative. In each domain, there are 500 positive and 500 negative review samples available. From this dataset, we construct the six cross-domain sentiment classification tasks (expressed in the form of source domain → target domain): A→I, A→Y, I→A, I→Y, Y→A and Y→I.

The **DAT_B** dataset [10] contains product reviews on books (domain B), DVDs (domain D), electronics (domain E) and kitchenware (domain K), which are all collected from the Amazon website. Each review is awarded a rating score between 0 and 5. Reviews rated above 3 are considered positive, while below 3 negative. In each domain, there are 1,000 positive and 1,000 negative reviews, as well as thousands of unrated reviews without sentiment class labels. From this dataset, we construct the twelve cross-domain sentiment classification tasks: B→D, B→E, B→K, D→B, D→E, D→K, E→B, E→D, E→K, K→B, K→D and K→E. In each domain, in addition to the 2,000 labeled reviews, we randomly select another 2,000 unlabeled reviews to support the representation learning.

4.1.1 Experimental Setup

To assess the upper bound of the classification performance, a standard sentiment classifier is trained and tested using the labeled documents from the target domain. To enable a performance comparison between the cross-domain and standard sentiment classifiers, a hold-out strategy is used. Following a data split scheme similar to those used in previous works [11], [17], a randomly selected review subset (100 positive and 100 negative from DAT_A, and 200 positive and 200 negative from DAT_B) is used as the testing set in each target domain. The remaining reviews are used to train the standard sentiment classifier when needed. For the two neural approaches, mSDA and DANN, the top 5,000 uni-grams and bi-grams in the reviews are included to the vocabulary list. For mSDA, its corruption level is set to 0.5, and its resulting concatenated representation vector is of 30,000 dimensions, including 5,000 original features and 25,000 embedded features returned by its 5 hidden layers. To implement HATN, the GloVe embeddings are

TABLE 1: Comparison of sentiment classification accuracies (%) for different methods using the DAT_A dataset. The best performance is boldfaced and the second best is underlined.

Tasks	A→I (81.00)	A→Y (79.50)	I→A (84.50)	I→Y (79.50)	Y→A (84.50)	Y→I (81.00)
Baseline1	63.00	71.00	75.50	75.50	73.50	66.50
Baseline2	71.50	73.50	65.50	72.00	76.50	66.50
SCL0	55.50	62.50	61.50	54.50	62.00	57.50
SCL	66.00	73.00	76.00	75.00	74.00	66.50
SFA0	62.50	60.00	59.50	61.00	64.00	57.00
SFA1	66.00	68.50	77.00	75.00	73.00	67.00
SFA2	63.00	73.50	73.00	69.00	70.50	69.50
SSE	67.50	73.00	<u>77.50</u>	76.00	74.00	67.50
BLSE	77.00	<u>76.50</u>	75.50	72.50	77.00	75.50
MEDA0	<u>77.50</u>	72.50	<u>77.50</u>	<u>79.00</u>	79.50	73.50
MEDA1	75.00	74.50	77.00	<u>79.00</u>	<u>80.00</u>	<u>76.50</u>
Proposed	**80.50**	**79.00**	**80.00**	**79.50**	**83.50**	**77.00**

used to initialize the network. The dimensionality of the GloVe word embedding space is $k = 300$, trained on the general English corpus of Wikipedia 2014 and Gigaword 5. Following the same setting as in SFA [17] and SSE [11], the number of selected pivots is set to $g = 200$ for DAT_A and $g = 500$ for DAT_B. The dimensionality of the learned cross-domain word embedding space is set to $d = 100$, as it is observed to be the optimal setting for most methods. The sentiment classifier is trained using l_2-regularized logistic regression implemented using LIBLINEAR [45], with its penalty (or coefficient) parameter set to $c = 1$.

The optimization parameters for our proposed CrossWord model are set to $T = 500$, $\eta_{ij}^{(0)} = 0.1$, $\tau = 0.4$, $\xi = 0.4$, $\delta = 0.2$, and also

$$\zeta(t) = \begin{cases} 0.5, & \text{if } t < 150, \\ 0.8, & \text{otherwise,} \end{cases}$$

following the gradient descent update setting recommended in [42], [46]. The performance of CrossWord is not sensitive to the setting of the regularization parameter μ, as long as it is within a reasonable range, and therefore we fix it to $\mu = 0.01$, as suggested in [47]. We adopt the fixed setting of $\theta_c = 5$ (DAT_A) and $\theta_c = 10$ (DAT_B) for selecting the pivot word candidate pool, and $K = 15$ for the perplexity parameter. We choose $\theta_k = 5$, $\kappa = 5$, $\alpha = 0.3$, $\lambda = 0.8$ for the other algorithm parameters based on parameter tuning. The nearest neighbor number κ used by $\epsilon^{(\mathrm{S_l})}$ for identifying the friend and enemy documents is searched within $\{5, 10\}$ by setting $\beta_1 = 0$ and $\beta_2 = 1$. The two algorithm parameters used by L_A follow $\theta_k \in \{5, 10\}$ and $\alpha \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$ by setting $\beta_1 = 1$ and $\beta_2 = 0$. The balancing parameter λ for controlling the preference degree between the two unsupervised scores $\epsilon^{(\mathrm{S_u})}$ and $\epsilon^{(\mathrm{T})}$ is searched within $\{0, 0.2, \ldots, 0.8, 1\}$ under the setting of $\beta_1 = \beta_2 = 0$. After determining the setting of the other parameters, the two balancing parameters β_1 and β_2 are tuned first within $\{0, 0.2, \ldots, 0.8, 1\}$. Then, β_1 is further tuned within $\{0.05, 0.10, 0.15\}$ and β_2 within $\{0.2(1-\beta_1), 0.4(1-\beta_1), 0.6(1-\beta_1), 0.8(1-\beta_1)\}$, for which these fine-grained search ranges are set based on the previous rough tuning. During this process the other parameters remain fixed to their selected values.

4.2 Comparison with Existing Approaches

We compare the cross-domain sentiment classification performance of different methods in Tables 1 and 2, evaluated using the DAT_A and DAT_B datasets. The neural network based methods are only experimented using the larger dataset DAT_B, because they usually offer superior performance when learning from a large amount of training samples. Each column of the table corresponds to a pair of source and target domains (source → target). In addition to the cross-domain performance, we also report the target domain performance obtained by a standard supervised learning within the target domain based on the bag-of-word vectors. The reported performance is shown parenthesized in the first row of each performance table. We report two SFA performances for different parameter settings of the logistic regression classifier: SFA1 has the same setting as the ones used by all the other methods ($c = 1$), and SFA2 follows a setting recommended in [17] ($c = 10,000$). We additionally report the performance of the document representation vectors learned by SCL and SFA on their own and without being combined with the bag-of-word vectors using the classifier parameter $c = 1$, which are referred to as SCL0 and SFA0, respectively. For MEDA, two embedding dimensions are experimented, including the recommended setting of 20 in [43] (referred to as MEDA0), and the same setting of 100 as used by the proposed method (referred to as MEDA1). In addition to experimenting with mSDA and DANN separately, we report the performance of a stacked network by connecting mSDA and DANN (referred to as mSDA-DANN), where the hidden representation vectors produced by mSDA is used as the input of DANN to boost the performance.

Overall, the proposed CrossWord offers the best performance amongst all the non-neural methods for almost all the evaluated cross-domain tasks, and similarly good performance to the top neural methods. In many tasks, CrossWord offers significant performance improvement over the second best method. Below, we summarize our observations related to the comparison between different methods:

• Review documents from different domains can use different sets of words to express the sentiment. Therefore, when using word occurrences as features to characterize the documents, the replacement of training samples collected in the target domain with those collected in a different domain can cause significant performance drop. This is evidenced by the difference between the first-row performance in parenthesis and the Baseline1 performance. For example, there is an 18% drop in B→E, as users use very different words to describe their opinions on books and on electronic products.

• The GloVe technique is based on word co-occurrence statistics. The resulting word vectors may be able to construct alignment between words such as "sharp" (sentiment-rich word in kitchenware reviews) and "thrill" (sentiment-rich word in book reviews) as they have quite high chances to co-occur with words such as "great", "satisfactory" in a very large corpus of general English text. Therefore, there is a chance to obtain a performance improvement by enhancing the bag-of-word document vectors with the GloVe embeddings. As expected, Baseline2 offers performance gains compared to Baseline1 for many assessed tasks (see tables).

TABLE 2: Comparison of sentiment classification accuracies (%) for different methods using the DAT_B dataset. The bottom section corresponds to neural network based approaches, while the top section includes the non-neural approaches. Within each section, the best performance is boldfaced and the second best is underlined.

Tasks	B→D (81.25)	B→E (83.25)	B→K (80.25)	D→B (80.75)	D→E (83.25)	D→K (80.25)	E→B (80.75)	E→D (81.25)	E→K (80.25)	K→B (80.75)	K→D (81.25)	K→E (83.25)
Baseline1	70.50	65.00	63.75	67.00	68.50	67.00	61.00	64.50	69.50	65.00	66.50	74.25
Baseline2	77.50	66.00	69.00	76.25	67.50	66.75	68.00	70.25	73.00	67.50	69.50	70.75
SCL0	71.00	64.75	57.50	74.25	65.25	65.75	64.25	64.50	72.00	61.50	58.00	72.50
SCL	76.50	66.75	65.50	74.75	68.50	66.75	60.75	64.50	72.50	64.75	66.75	74.00
SFA0	54.00	51.50	50.00	61.00	55.75	54.50	55.25	62.00	65.00	60.00	60.50	60.75
SFA1	70.25	63.50	65.00	68.25	68.75	66.50	61.50	65.00	70.50	65.00	67.00	74.25
SFA2	78.75	67.75	67.25	78.00	70.50	70.75	64.50	69.25	77.50	64.75	69.00	**79.50**
SSE	72.75	66.25	67.00	70.75	69.00	67.75	62.00	68.00	73.75	66.50	69.75	74.25
BLSE	75.25	70.00	62.50	78.00	69.75	67.25	69.50	**73.75**	74.00	74.25	73.25	74.75
MEDA0	76.50	62.75	66.75	73.75	71.50	67.75	64.25	70.50	72.75	53.25	64.25	71.25
MEDA1	75.25	53.25	57.25	74.75	67.00	69.00	62.00	70.00	74.50	53.50	61.25	70.00
Proposed	**81.75**	**71.25**	**71.25**	**80.25**	**73.75**	**75.00**	**71.00**	72.00	**77.75**	**74.50**	**73.75**	78.25
mSDA	78.50	**76.00**	**69.75**	77.75	**74.25**	72.25	67.75	72.75	**77.75**	70.50	73.00	**78.75**
DANN	73.00	68.00	66.75	**78.00**	70.75	70.50	66.00	70.25	73.75	65.50	71.00	76.75
mSDA-DANN	78.75	73.00	68.25	77.25	73.50	**72.50**	66.50	69.75	**77.75**	**74.00**	74.25	78.00
HATN	**82.50**	72.00	69.00	**78.00**	72.75	72.00	**71.50**	**77.75**	76.25	71.25	**78.50**	78.00

TABLE 3: Comparison of the averaged training time (10 iterations/epochs) for methods that adopt the gradient descent optimization using the DAT_B dataset. The reported results are recorded using Python running on a server with the use of GeForce-GTX-1080Ti-12GB GPU. The best performance is boldfaced and the second best is underlined.

Method	Time(s) for 10 iterations/epochs
BLSE	12.72s
DANN	6.04s
mSDA-DANN	17.08s
HATN	81.75s
Proposed	**1.67s**

However, the drawback of the GloVe embedding vectors is that they are not learned from specialized review text, and therefore are not sentiment sensitive in some domains. Nevertheless, these word vectors carry rich word co-occurrence information and provide a good starting point to work with. This is why they are used as the input for our model.

• The state-of-the-art techniques SCL, SFA, SSE and BLSE provide similarly good performance and there seems to be no consistent winner throughout the different tasks. SCL is based on singular value decomposition of a computed weight matrix, SFA and SSE are based on eigen-decomposition of a constructed similarity matrix, while BLSE jointly minimizes a mean squared error and a cross-entropy error. Both SCL and SFA compute new vectors for characterizing the documents. But these new vectors are not robust enough to be used on their own and have to be combined together with the original bag-of-word vectors to maintain a satisfactory performance. This is evidenced by the performance difference between their two versions (SCL0 vs. SCL, SFA0 vs. SFA). On the contrary, the new document vectors computed by SSE can be used on their own, offering similarly good performance to that of the combined vectors of SCL and SFA. This shows that SSE may be a more robust technique than SCL and SFA. Furthermore, the learned word embeddings by BLSE can also perform well when being used to generate document representations.

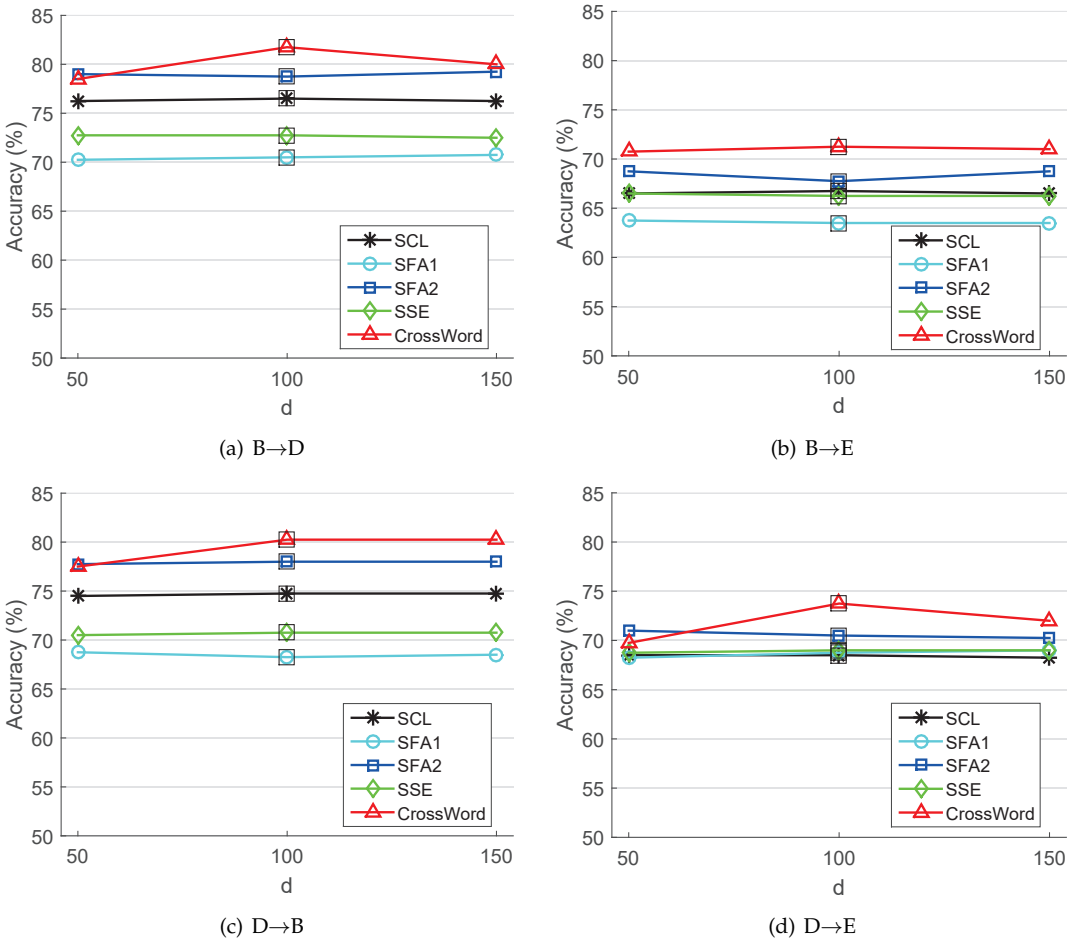
• Our proposed CrossWord model inherits the basic ideas from some modeling strategies of SSE, such as pivot word alignment, document separability enhancement and content-based neighbor preservation. Starting from these, it further improves the model design by proposing more sophisticated word alignment and document similarity matrix construction models, and also uses a more powerful similarity structure preservation technique based on stochastic neighbor embedding, which has been demonstrated to be more accurate in local neighbor preservation by various previous research [26], [46], [47]. Relying on this improved design, as shown in Tables 1 and 2, the proposed method offers significant performance improvement over SSE and the other state-of-the-art techniques.

• The proposed CrossWord provides similarly good performance to the best competing neural networks for most of the transfer tasks. It only fails to keep up with the best performing neural networks for 3 out of the 12 transfer cases, which are $B \rightarrow E$, $E \rightarrow D$ and $K \rightarrow D$. But it still manages to provide comparably good performance to the second best neural networks for these three cases. Moreover, we compare the training time of 10 iterations/epochs for the top-performing methods based on gradient descent optimization in Table 3. It shows that CrossWord possesses the highest training speed, particularly much higher than the top-performing HATN, under the same Python implementation and hardware environment.

4.3 Investigation of Model Behavior

To understand better the model behavior, we demonstrate the performance changes of the proposed method under different parameter settings, using the larger dataset DAT_B. Figure 3 compares the proposed and competing methods under different embedding dimension settings. It can be seen that CrossWord provides in general comparable or better performance than the competing methods for all the observed settings, and certainly the best performance when comparing under their own individual optimal settings.

Figure 4 demonstrates the changes of CrossWord performance as observed during the parameter tuning process of α, employing the score function L_A only ($\beta_1 = 1$) and



(a) B→D (b) B→E

(c) D→B (d) D→E

Fig. 3: Performance changes for different methods and varying embedding length $d = 50, 100, 150$, assessed for four example tasks from DAT_B. The d setting used for reporting the performance in the tables is indicated by a square.

comparing two θ_k settings. The chosen setting of $\alpha = 0.3$ provides a generally good performance across different tasks.

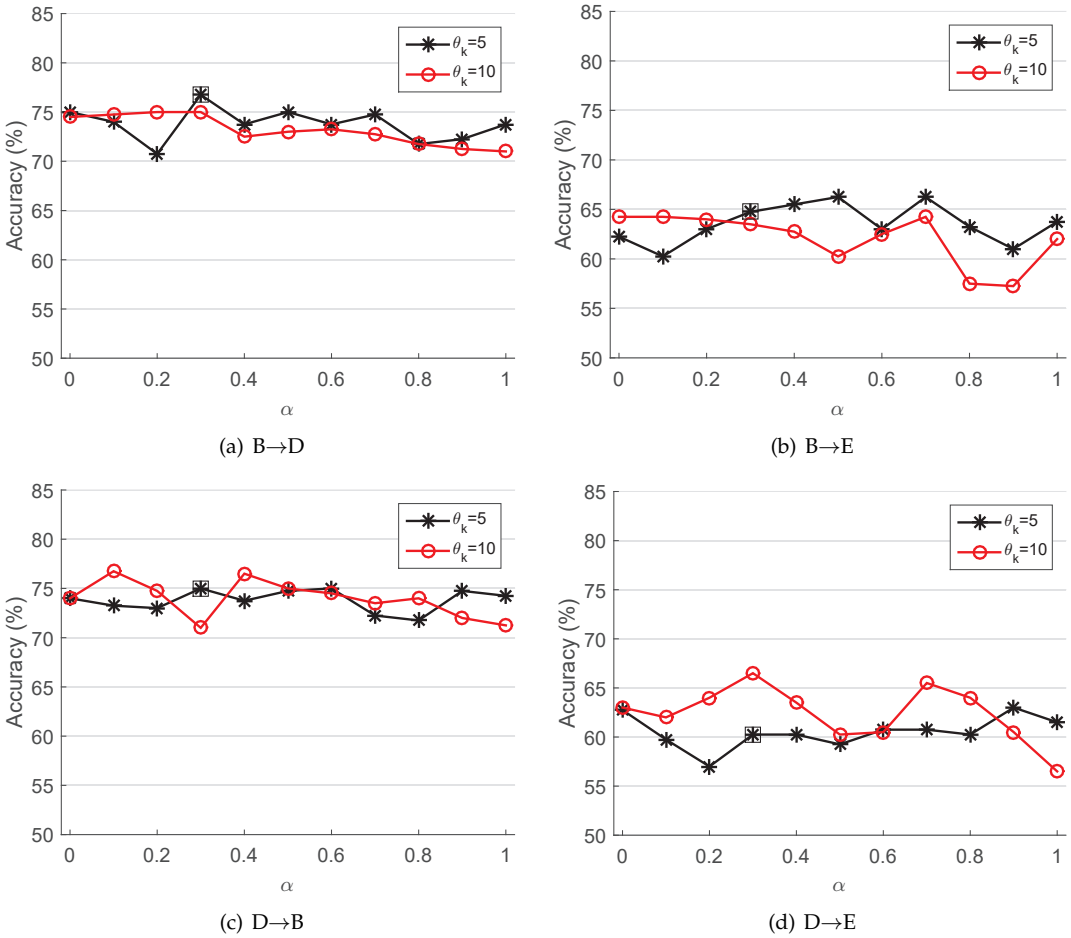
Figure 5 demonstrates the changes of CrossWord performance while varying κ, employing the score function $\epsilon^{(S_l)}$ only ($\beta_2 = 1$). It can be seen that, in general, a smaller size of local neighborhood is more reliable and provides better performance.

Figure 6 demonstrates the CrossWord performance changes with varying λ using the two score functions $\epsilon^{(S_u)}$ and $\epsilon^{(T)}$ ($\beta_1, \beta_2 = 0$). Increasing λ leads to an increased attention shift from the unlabeled documents in the source domain to the unlabeled ones in the target domain. The chosen setting $\lambda = 0.8$ results in higher attention degree over the domain-specific information. As explained in the previous section, after a rough tuning of β_1 and β_2 the setting of around 0.1 for β_1 provides a generally good performance. This means that around 10% attention is paid to the word-level score function L_A.

Figure 7 demonstrates the CrossWord performance while varying the attention degree to L_A from 5% to 15%, from 20% to 80% of the remaining attention to the score function $\epsilon^{(S_l)}$ and the rest for $\epsilon^{(S_u)}$ and $\epsilon^{(T)}$. The performance of the best competing method is displayed in Figure 7, where it can be seen that, for each example task there exist multiple setting combinations that offer satisfactory performance. The preference degree (controlled by β_2) between the source-domain separability and the unsupervised content information varies among tasks. Nevertheless, the performance change is not drastic, which shows that the proposed model is not demanding in tuning its hyper-parameters β_1 and β_2.

TABLE 4: Examples of selected pivot words and their corresponding domain-specific words, as well as their closeness ranking computed in the CrossWord embedding space for the task I→Y and DAT_A. T@n denotes that the given domain-specific word, ranks as the top nth word closest to the target pivot word based on examining the Euclidean distances between the computed word embedding vectors.

	Movie (Domain I)		Restaurant (Domain Y)	
Pivot	Domain-specific	Ranking	Domain-specific	Ranking
rather	superficial	T@4	letdown	T@2
almost	unrecognizable	T@5	empty	T@4
extremely	insincere	T@4	rude	T@17
recommended	fans	T@3	place	T@41
classic	cult	T@4	fantastic	T@29
imagination	ineptly	T@6	stretch	T@3
good	acting	T@5	service	T@6
glad	planned	T@3	unbelievable	T@33
attention	hold	T@3	waiter	T@16
special	effects	T@4	whatsoever	T@21



(a) B→D (b) B→E

(c) D→B (d) D→E

Fig. 4: Performance changes for CrossWord while varying $\alpha \in [0, 1]$, assessed under the settings $\theta_k = 5$ and $\theta_k = 10$, using four example tasks from DAT_B. The α setting used for reporting the performance in the tables is indicated by a square.

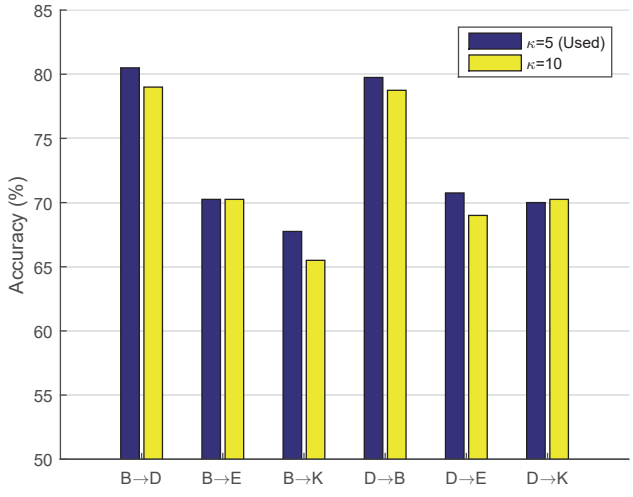


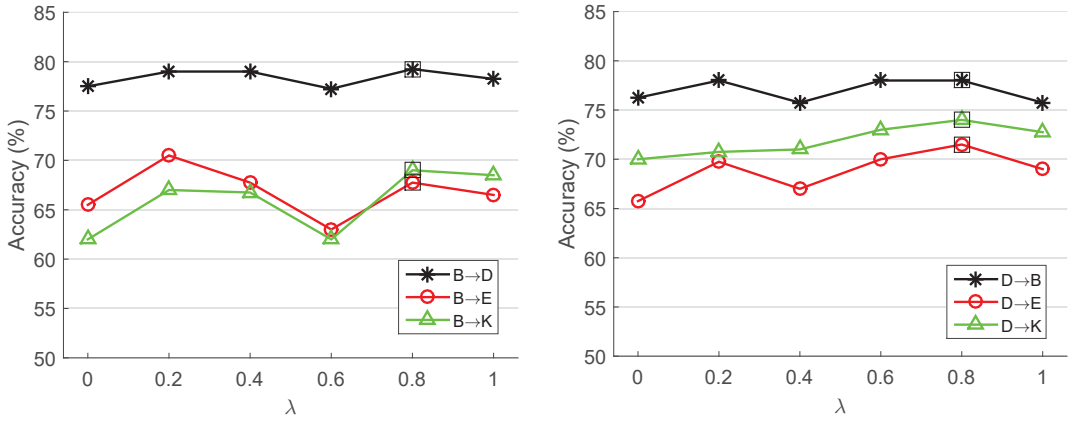
Fig. 5: Demonstration of the CrossWord performance under the two neighbor settings of $\kappa = 5$ and $\kappa = 10$, assessed for six example tasks from DAT_B.

4.4 Example Demonstration

Finally, in Table 4 we demonstrate examples of selected pivot words, and the top selected domain-specific word for each example pivot as well as its closeness ranking to its corresponding pivot word computed in the CrossWord embedding space. The examples are collected for the task of predicting sentiment polarity of restaurant reviews based on movie reviews. It can be seen from the table, that these selected pivot and domain-specific words carry sentiment information. Also, the domain-specific words that are connected to the same pivot word are indeed aligned in the embedding space, indicated by their top closeness rankings to that pivot word.

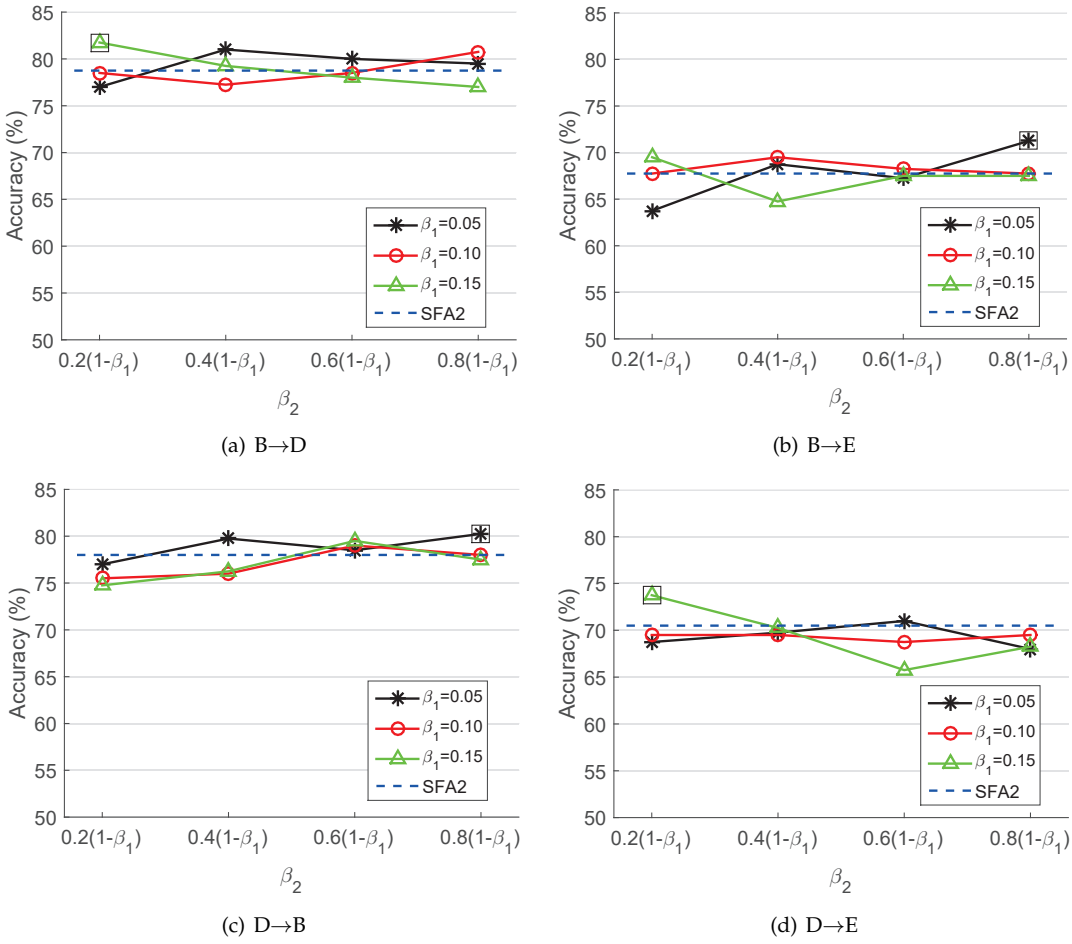
5 Conclusion

We have proposed CrossWord, a novel cross-domain embedding technique. It effectively creates an alignment between the source and target feature spaces. Low-dimensional document embedding vectors computed from its resulting embedding vectors are sufficient for constructing a robust sentiment classifier that can be shared across domains. CrossWord offers a more accurate modeling of probabilistic similarity relationships between the pivot and domain-specific words, and also between the labeled reviews in the source domain and the unlabeled reviews in both domains. It also provides a more accurate preservation of the desired similarity structures in the embedding



(a) (b)

Fig. 6: Performance changes for CrossWord while varying $\lambda \in [0, 1]$, assessed under different example tasks from DAT_B. The λ setting used for reporting the performance in the tables is indicated by a square.



(a) B→D (b) B→E

(c) D→B (d) D→E

Fig. 7: Performance changes for CrossWord while varying β_1 and β_2 using different example tasks from DAT_B. Best performing competing method is also reported as a dashed line. The β_1, β_2 setting used for reporting the performance in the tables is indicated by a square.

space, achieved through the use of the stochastic neighbor embedding technique. Furthermore, CrossWord attempts to reduce algorithm bias by learning a mapping function from a standard word embedding space, learned from a general English corpus, to the desired cross-domain embedding space. Extensive experimental results have demonstrated the superior performance of the proposed method over various classical and state-of-the-art algorithms.

REFERENCES

[1] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[2] L.-W. Ku, Y.-T. Liang, H.-H. Chen *et al.*, "Opinion extraction, summarization and tracking in news and blog corpora." in *AAAI spring symposium: Computational approaches to analyzing weblogs*, vol. 100107, 2006.

[3] S. Liu, X. Cheng, F. Li, and F. Li, "Tasc: topic-adaptive sentiment classification on dynamic tweets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1696–1709, 2015.

[4] F. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, 2011, p. 2488.

[5] P. H. Calais Guerra, A. Veloso, W. Meira Jr, and V. Almeida, "From bias to opinion: a transfer-learning approach to real-time sentiment analysis," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 150–158.

[6] T.-k. Fan and C.-h. Chang, "Sentiment-oriented contextual advertising," *Knowledge and Information Systems*, vol. 23, no. 3, p. 321, 2010.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[8] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 760–10 773, 2009.

[9] X. Huang, Y. Rao, H. Xie, T.-L. Wong, and F. L. Wang, "Cross-domain sentiment classification via topic-related tradaboost." in *AAAI*, 2017, pp. 4939–4940.

[10] J. Blitzer, M. Dredze, F. Pereira *et al.*, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL*, vol. 7, 2007, pp. 440–447.

[11] D. Bollegala, T. Mu, and J. Y. Goulermas, "Cross-domain sentiment classification using sentiment sensitive embeddings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 398–410, 2016.

[12] S. Gao and H. Li, "A cross-domain adaptation method for sentiment classification using probabilistic latent analysis," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 1047–1052.

[13] Y. He, C. Lin, and H. Alani, "Automatically extracting polarity-bearing topics for cross-domain sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 2011, pp. 123–131.

[14] D. Bollegala, D. Weir, and J. Carroll, "Cross-domain sentiment classification using a sentiment sensitive thesaurus," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 8, pp. 1719–1731, 2013.

[15] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 2012, pp. 410–419.

[16] X. Cui, N. Al-Bazzaz, D. Bollegala, and F. Coenen, "A comparative study of pivot selection strategies for unsupervised cross-domain sentiment classification," *The Knowledge Engineering Review*, vol. 33, p. e5, 2018.

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

[18] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 12, pp. 2240–2249, 2014.

[19] F. Li, S. Wang, S. Liu, and M. Zhang, "Suit: A supervised user-item based topic model for sentiment analysis." in *AAAI*, vol. 14, 2014, pp. 1636–1642.

[20] J. Barnes, R. Klinger, and S. Schulte im Walde, "Projecting embeddings for domain adaptation: Joint modeling of sentiment in diverse domains," in *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics*, 2018.

[21] M. Long, J. Wang, Y. Cao, J. Sun, and S. Y. Philip, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2027–2040, 2016.

[22] Z. Li, Y. Zhang, Y. Wei, Y. Wu, and Q. Yang, "End-to-end adversarial memory network for cross-domain sentiment classification," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017.

[23] J. Ji, C. Luo, X. Chen, L. Yu, and P. Li, "Cross-domain sentiment classification via a bifurcated-lstm," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018, pp. 681–693.

[24] Z. Li, Y. Wei, Y. Zhang, and Q. Yang, "Hierarchical attention transfer network for cross-domain sentiment classification," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, Lousiana, USA, February 2–7, 2018*, 2018.

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

[26] T. Mu, Y. J. Goulermas, and S. Ananiadou, "Data visualization with structural control of global cohort and local data neighborhoods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[27] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *NIPS*, vol. 15, 2002, pp. 833–840.

[28] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, 2017.

[29] L. Wang, J. Niu, H. Song, and M. Atiquzzaman, "Sentirelated: A cross-domain sentiment classification algorithm for short texts through sentiment related index," *Journal of Network and Computer Applications*, vol. 101, pp. 111–119, 2018.

[30] D. Bollegala, T. Maehara, and K. ichi Kawarabayashi, "Unsupervised cross-domain word representation learning," in *Proc. of the Annual Conference of the Association for Computational Linguistics (ACL) and the 7th International Joint Conferences on Natural Language Processing (IJCNLP)*, 2015.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[32] W. Yang, W. Lu, and V. Zheng, "A simple regularization-based algorithm for learning cross-domain word embeddings," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2898–2904.

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

[34] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 1627–1634.

[35] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[36] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, 2015, pp. 1180–1189.

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

[38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural*

Language Processing (EMNLP), 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[39] J. Blitzer, "Domain adaptation of natural language processing systems," Ph.D. dissertation, University of Pennsylvania, 2008.

[40] T. Mu, J. Jiang, Y. Wang, and J. Y. Goulermas, "Adaptive data embedding framework for multiclass classification," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1291–1303, 2012.

[41] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural networks*, vol. 1, no. 4, pp. 295–307, 1988.

[42] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[43] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18, 2018, pp. 402–410.

[44] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 597–606.

[45] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/liblinear

[46] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 1–14, 2017.

[47] Y. Hao, T. Mu, J. Y. Goulermas, J. Jiang, R. Hong, and M. Wang, "Unsupervised t-distributed video hashing and its deep hashing extension," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5531–5544, 2017.

Y. Hao received the B.E. and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 2012 and 2017, respectively. He is currently a Senior Research Associate in the Department of Computer Science at City University of Hong Kong. His research interests mainly include machine learning and multimedia data analysis, such as large-scale multimedia indexing and retrieval, multimedia data embedding, and video hyperlinking.

T. Mu (M'05) received the B.Eng. degree in electronic engineering and information science from the Special Class for the Gifted Young, University of Science and Technology, Hefei, China, in 2004, and the Ph.D. degree in electrical engineering and electronics from the University of Liverpool in 2008. She is currently a Lecturer in the School of Computer Science, the University of Manchester. Her research interests include machine learning, mathematical modeling and optimization, with applications to language and vision understanding.

R. Hong received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008. He was a Research Fellow with the School of Computing, National University of Singapore, from 2008 to 2010. He is currently a Professor with the Hefei University of Technology, Hefei, China. He has co-authored over 70 publications in his research interests, which include multimedia content analysis and social media. He is a member of the ACM and the Executive Committee Member of the ACM SIGMM China Chapter. He was a recipient of the Best Paper Award in the ACM Multimedia 2010, the Best Paper Award in the ACM ICMR 2015 and the Honorable Mention of the IEEE Trans. on Multimedia Best Paper Award. He served as an Associate Editor of the Information Sciences and Signal Processing Elsevier, and the Technical Program Chair of the MMM 2016.

M. Wang received the B.E. and Ph.D. degrees in the special class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively. He is currently a Professor with the Hefei University of Technology, China. He has authored over 200 book chapters, journal, and conference papers in these areas. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He was a recipient of the ACM SIGMM Rising Star Award 2014. He is an Associate Editor of the IEEE Trans. on Knowledge Discovery and Engineering, the IEEE Trans. on Circuits and Systems for Video Technology, and the IEEE Trans. on Neural Networks and Learning Systems.

X. Liu received his Ph.D. degree (March 2013) from EURECOM France. He is currently working as a faculty of School of Computer and Information in Hefei University of Technology. His current research interests include multimedia processing, and social event analysis.

J. Y. Goulermas (M'98, S'10) obtained the B.Sc.(1st class) degree in computation from the University of Manchester (UMIST), in 1994, and the M.Sc. and Ph.D. degrees from the Control Systems Center, UMIST, in 1996 and 2000, respectively. He is currently a Professor in the Department of Computer Science at the University of Liverpool. His current research interests include machine learning, combinatorial data analysis, data visualization as well as mathematical modeling. He has worked with various application areas including image/video analysis, biomedical engineering and biomechanics, industrial monitoring and control, and security. He is a senior member of the IEEE and an Associate Editor of the IEEE Trans. on Neural Networks and Learning Systems.