Cross-domain Sentiment Classification via Constructing Semantic Correlation

Xiaocong Wei, Hongfei Lin and Liang Yang

***Abstract*—Cross-domain sentiment classification trains robust classifiers across domains with the help of source domain labeled data. Sentiment is expressed differently in different domains. Sentiment terms that occur in a source domain may not appear in a different target domain. Such feature mismatches hinder cross-domain sentiment classification. Previous studies have addressed this problem by constructing a common feature representation or subspace; however, they have not considered semantic correlations between features. In this paper, we propose a cross-domain semantic correlation auto-correspondence method (CSCW) by capturing similar semantic features from different domains. First, our method uses sentiment-invariance words as features by considering their properties as strong sentiment indicators and their invariance across domains. Second, we extracted the top-*N* pivot features using a common frequency among source and target domains. These pivot features can then be employed to find semantically similar sentiment features from both domains. Third, for each pivot feature, by calculating the semantic similarity between non-pivot features and pivot features from either domain with the help of Word2Vec, we construct similar-pivot feature pairs that express similar sentiments but in different representations in either domain. Finally, we transform these pairs to align similar sentiment feature representations. This process avoids feature mismatches and reduces sentiment discrepancies between domains. The experimental results from testing on 12 source-target domain pairs of Amazon product reviews demonstrate that our method significantly outperforms previous approaches in sentiment classification.**

***Index Terms*—Cross-domain, Word2Vec, Sentiment classification, Transfer learning, Product review.**

I. INTRODUCTION

WITH the rapid development of the Internet, users increasingly choose to express their opinions about products or services they consume online. This has led to a vast amount of opinionated text that has attracted increasing attention from the information retrieval and natural language processing community. The ability to accurately estimate the sentiment expressed in a product review is important. When negative sentiment is abundantly associated with a certain product feature, the retailer can make plans to address the issue. Moreover, failing to detect such sentiments in reviews may result in decreasing sales. In online shopping, because consumers cannot see the products in a physical store, the opinions of other consumers are the only available subjective descriptions of the product. By automatically classifying product reviews based on the sentiment they express, we can help a potential consumer more easily understand the overall opinions of other consumers about the product. The performance of sentiment classification algorithms based on deep neural networks relies heavily on large-scale labeled training data; however, it is unrealistic to manually annotate such large amounts of data. Therefore, we urgently need a method that can automatically label sentiment from data in multiple domains.

In many situations, we may have plentiful labeled training data in a source domain but need to process text from a target domain that has a different distribution and no labeled data. It is costly to annotate data for each new domain. However, when we directly apply a classifier trained on a labeled source domain to predict an unlabeled target domain, it typically performs poorly. This poor performance is because many machine learning and data mining algorithms assume that the training and test data are from the same feature space and have the same distribution. Cross-domain sentiment classification focuses on the challenge of training a sentiment classifier from one or more labeled domains (source domains) and applying the trained classifier to a different, unlabeled domain (target domain).

A primary challenge facing cross-domain sentiment classification is feature mismatches. Sentiment classification is highly sensitive to the domain because different groups of consumers use different words to express their sentiments in different domains. Sentiment terms that appear in a source domain may not appear or appear only rarely in a target domain. As an example, suppose we wish to build a book review sentiment classifier. We possess few labeled book reviews, but labeled electronic reviews are abundant. In a book domain, a reviewer might use the words “exciting” or “graphic novel” to express positive sentiment and the words “boring” and “drowsy” to express negative sentiment. In contrast, in the electronics domain, reviewers might use words such as “durable” and “light” to express positive sentiment, while “expensive” and “short battery life” are examples of terms that often indicate negative sentiment. Considering both these domains, although “graphic novel” and “durable” both express positive sentiment, their representations are different.

Manuscript received July 17, 2016; revised March 18, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61572102, 61562080 and Dalian University of Foreign language Research Foundation under Grant 2014XJQN14.

X. C. Wei is with the Department of Computer Science and Technology, Dalian University of Technology, Dalian, 116024 China. She is also with the Department of Software Engineering, Dalian University of Foreign Language, Dalian, 116044 China. e-mail: weixiaocong@dlufl.edu.cn.

H. F. Lin is with the Department of Computer Science and Technology, Dalian University of Technology, Dalian, 116024 China. e-mail: hflin@dlut.edu.cn.

L. Yang is with the Department of Computer Science and Technology, Dalian University of Technology, Dalian, 116024 China. e-mail: yangliang@mail.dlut.edu.cn.

Moreover, the words "graphic novel" appear in the book domain but are unlikely to appear in the electronics domain.

Many existing approaches disentangle such domain discrepancies by constructing a common feature representation (subspace). For example, SCL [1] tries to map domain-dependent features to a common feature subspace using a map matrix. This method can automatically induce correspondences among features from different domains. SFA [2] uses "pivot features" as a bridge to align domain-specific features from both domains in a latent space using a learned projection matrix. Bollegala et al. [3] used a feature expansion method along with an automatically constructed sentiment-sensitive thesaurus to train and test a binary classifier. They also proposed a sentiment-sensitive word embedding learning method by constructing three objective functions: the distributional properties of pivots, label constraints in the source domain documents, and geometric properties in the unlabeled documents in both the source and target domains [4]. Zhou et al. [5] proposed a joint non-negative matrix factorization framework by linking heterogeneous input features with pivots for domain adaptation. It is crucial to be able to measure the semantic similarity between words precisely [6]; however, during the process of estimating feature-correspondence correlations, the methods mentioned above do not consider the semantic relationships between words. A feature that has been transformed to a subspace is not always accurate, which will affect the performance of the subspace. In contrast to the existing methods, we apply Word2Vec [7] to capture the semantic relationships between features from both the source and target domains and train a classifier using a common sentiment feature representation that can be shared across domains.

Most works focus on all the words appearing in the text. However, not all words contribute equally to sentiment classification; some are invariant across domains and some are domain-dependent. Xia et al. [8] first proposed a labeling adaptation method using a parts-of-speech (POS)-based feature ensemble (FE) that assigns different weights to different POS tags, giving higher weights to domain-independent parts such as adjectives and verbs and lower weights to domain-specific parts such as nouns. They also presented a PCA-based sample selection (PCA-SS) method for instance adaptation. Combining FE with PCA-SS for domain adaptation results in significant improvements compared to either FE or PCA-SS alone. In our work, verified by experiments, we found that adjectives are more invariant and are good indicators of sentiment classification. Based on these two characteristics, we adopted only adjectives as features.

The methods referred to above use only hand-crafted features extracted from text; thus, they rely on human insight into the problem and on linguists skilled in NLP. To address this cross-domain problem, several methods based on deep neural networks have been proposed in recent years [9]-[17]. It is assumed that the internal representation of a neural network contains no discriminative information about the raw input that is beneficial for cross-domain classification. However, these papers address only one specific task, such as mining consumption intention [9], learning good representation via deep architecture to reduce transfer loss (improving transfer ratio) [10]-[13], verifying the transferability of a deep neural network [14], improving representation learning [15], resolving online transfer learning [16], or proposing an architecture capable of multitask transfer learning [17]. The advantage of methods based on deep neural networks is that we can avoid hand-crafted features and automatically learn feature representations that can be shared across classes and tasks. Deep learning methods outperform conventional (non-deep) neural networks on large-scale corpuses; however, when a corpus is small, the accuracy of methods based on deep neural networks is lower than the accuracy of methods that do not use deep neural networks. Further studies of cross-domain product review sentiment classification on small datasets using deep neural networks are essential. Many existing classification methods based on conventional non-deep neural networks have been evaluated [18]-[19]. We address the problem using a support vector machine (SVM) classifier assisted by shallow neural networks (Word2Vec).

In this paper, similar to the literature above, we consider only a binary classification problem—i.e., whether a review is positive or negative. We use sentiment-invariant words as features and validate them experimentally. First, we split these features into pivot features and non-pivot features depending on whether the feature is a co-occurrence feature between the source domain and the target domain. Only the top-*N* frequent co-occurrence features are used as pivot features. Second, by calculating the semantic similarity of pivot features and non-pivot features from either domain with the help of Word2Vec, we also construct similar-pivot feature pairs. We transform the similar-pivot feature pairs from the source domain and target domain into a uniform representation to align different features that express similar sentiments. Finally, the SVM is trained to perform the classification task. We evaluate our method against existing state-of-the-art methods using data from Amazon product reviews. The experimental results demonstrate that our technique achieves the best results among the existing cross-domain methods.

The rest of this paper is organized as follows: Section 2 describes the problem and provides some definitions. The details of our solution are presented in Section 3. We describe a series of experiments conducted to evaluate the effectiveness of our proposed solution in Section 4. Section 5 concludes our work and outlines future work.

II. Problem Setting and Related Concepts

A. Problem Setting

Before giving a formal definition of the problem addressed in this paper, we first define some terms.

Domain: A domain D consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$. $\mathcal{X}$ is the space of all term vectors, and X is a specific learning sample. Generally, different domains have different feature spaces or different marginal probability distributions. In our paper, we consider only the case involving one source domain D_S and one target domain D_T .

Source domain: The source domain, $D_S = \{(X_{S_i}, Y_{S_i})\}_{i=1}^{n_S}$, refers to a set of labeled instances from a certain domain, where X_{S_i} is the i-th labeled instance. Here, X_{S_i} denotes one of the product reviews in the source domain, Y_{S_i} is the sentiment label for X_{S_i} , and $Y_{S_i} \in \{+1,-1\}$, where the sentiment labels +1 and -1 denote positive and negative sentiments, respectively. We use n_S to denote the number of labeled instances in the source domain. In our paper, n_S denotes the total number of product reviews in the source domain.

Target domain: The target domain, $D_T = \{(X_{T_i})\}_{i=1}^{n_T}$ refers to a set of unlabeled instances from a domain that is different from but related to the source domain. Here, X_{T_i} is the i-th unlabeled instance. In our paper, X_{T_i} denotes one of the product reviews in the target domain. The number of unlabeled instances in the target domain is n_T . In our paper, n_T denotes the total number of product reviews in the target domain.

Cross-domain sentiment classification: We define cross-domain sentiment classification as the task of training a binary classifier on D_S to predict the sentiment label Y_{T_i} of a review X_{T_i} in the target domain D_T .

B. Related Concepts

We also present the following concepts:

Pivot feature (PF): Some features appear frequently in both source and target domains. For example, in sentiment classification, features such as “excellent” or “bad” express similar sentiments about a product regardless of domain. Such features represent common knowledge from both domains and are referred to as PFs.

Non-pivot feature (NPF): Any feature that is not a pivot feature is a non-pivot feature.

Similar-pivot feature (SPF): In one domain, when an NPF is similar to a PF, we call it a similar-pivot feature.

III. Methods

A. Feature Selection

POS tags such as verbs, nouns, adjectives and adverbs can be strong indicators of sentiment. A sentiment classifier may classify a product review as positive or negative depending on the sentiment expressed in the review. Previous work revealed a high correlation between the presence of adjectives and document sentiment [20]-[24]. For cross-domain sentiment classification, we assumed that adjectives are more domain invariant [8]. To verify this assumption, we selected adjectives, adverbs, nouns and verbs as features from the dataset depicted in Section IV.a. First, we performed an in-domain experiment using a training dataset and a test dataset from the same source domain and achieved an average accuracy of Ac_{in} . Then, we used a classifier trained on the source labeled data to classify target test data directly (without any domain adaptation) and achieved an average accuracy of Ac_{cross} . We used SVM as the classifier. Ac denotes the classification accuracy shift between them and can be computed as follows:

$$Ac = \left\| Ac_{in} - Ac_{cross} \right\| . \qquad (1)$$

TABLE I
The Accuracy Loss of Transfer Learning

POS	Ac
adjectives	0.0040
adverbs	0.0056
verbs	0.0119
nouns	0.0114

As shown in Table I, the cross-domain accuracy shifts the least for adjectives, demonstrating that adjectives vary only slightly across different domains—in other words, adjectives are sentiment-invariant words. Based on the above experiment, unlike prior methods, which used all words as features, we selected only adjectives as training features for classification to retain the strong sentiments in the reviews and to exclude features that amplify domain divergence.

B. Word2Vec

The source domain and the target domain typically include several frequently co-occurring sentiment words. For example, the word “awful”" appears frequently in both the book review domain and the DVD review domain. From observation, we can also find some correlations between sentiment features from different domains. For example, in the book domain, the NPF “horrible” is similar to the NPF “dreadful” in the DVD domain. In fact, “horrible” is similar to the co-occurrence feature “awful” and the term “dreadful” is also similar to the co-occurrence feature “awful.” Considering that “horrible” and “dreadful” are both similar to the same PF, they should express similar sentiment—but through different representations. We hypothesize that, with the help of the PF “awful,” we can extract “horrible” from the book domain and “dreadful” from the DVD domain using some method and then exploit their sentiment feature-correspondence correlations across both the book domain and the DVD domain. If words that express similar sentiments can be unified into a common representation, we can avoid the problem of feature mismatches and reduce domain divergence.

One of the most effective word representation methods, Word2Vec, was released by Google in 2013. Word2Vec can learn vector representations of words in a high-dimensional vector space based on deep neural networks. In its word embedding representations, semantically close words are likewise close in cosine distance in the lower dimensional vector space. To address the problem of feature mismatches,

TABLE II
THE ALGORITHM FOR WORD2VEC MODEL TRAINING

Input	D is a domain θ is the dimensionality of Word2Vec
Output	Word2Vec model Md 1.Remove punctuation using the Python NLTK toolkit. 2.Tokenize using the Python NLTK toolkit. 3.Train the Word2Vec model on the corpus above with θ. 4. Obtain Md.

we intend to apply Word2Vec to compute the similarity between features.

Our approach selects SPFs by calculating which NPF is most similar to a certain PF based on the Word2Vec model. First, we preprocess every domain corpus, remove punctuation and tokenize the data. Next, we train a model with Word2Vec on the corpus from each domain. The vector dimensionality is θ. The result is that, each domain has a Word2Vec model, denoted as $\{Md_l\}_{l=1}^{n}$, where n is the number of domains. The algorithm is displayed in Table II. The models are then prepared for feature similarity calculations.

C. Algorithm Overview

Cross-domain sentiment classification based on Word2Vec (CSCW) involves a source domain and a target domain in which both domains contain ample data, but only the source domain data are labeled. In this section, we describe our approach.

First, considering the two characteristics of domain invariance and strong sentiment indicators, we extract adjectives as sentiment-invariant features that can be used for classification as described in Section III.a. In our method, only adjectives are extracted as feature from both the source domain D_S and the target domain D_T. Following prior work on cross-domain sentiment classification, high-frequency features common to both domains are referred to as PFs. We rank these high-frequency features in descending order by their frequency and select the top-*N* features as the PFs. The main challenge of cross-domain sentiment classification is to avoid feature mismatches. Features that express similar sentiment can have different representations in different domains. The feature we use for training a classifier in the source domain may not occur or may occur only rarely in the target domain. Therefore, it is necessary to unify the representations. We intend to transform the features that have similar sentiment. With the help of the Word2Vec model trained on the source domain in Section III.b, we compute the most similar NPF for each PF as a domain SPF, and then compute the SPFs in the target domain in the same manner. Hence, each PF has an SPF pair. Because each feature of the pair is similar to the same PF, they are similar in sentiment. To reduce feature mismatches between the two domains, we transform the SPFs from the source domain and target domain into a unified form and then build a common subspace. For example, assume that X_S is a product review from the source domain that has adjectives as a feature, $X_S = \{x_{S_1}, x_{S_2}, \ldots, x_{s_i}, \ldots, x_{s_m}\}$, that X_T is one of the

TABLE III
THE ALGORITHM FOR CSCW

Input	$D_S = \{(X_{S_i}, Y_{S_i})\}_{i=1}^{n_S}$, $D_T = \{(X_{T_i})\}_{i=1}^{n_T}$, N Md_l^S, the Word2Vec model trained on the source domain Md_l^T, the Word2Vec model trained on the target domain
Output	$\{(Y_{T_i})\}_{i=1}^{n_T}$ 1. Extract adjectives as features from D_S and D_T 2. Select the top-*N* co-occurrence features as pivot features from D_S and D_T. $\{x_k\}_{k=1}^{N}$ 3. For *k* in 1......N, obtain the most similar non-pivot feature x_{S_i} for x_k from D_S based on Md_l^S, and obtain the most similar non-pivot feature x_{T_i} for x_k from D_T based on Md_l^T, transform every x_{S_i} appearing in D_S as $x_{S_i}\_x_{T_i}$. transform every x_{T_i} appearing in D_T as $x_{S_i}\_x_{T_i}$. 4. Obtain $D_S^{'} = \{(X_{S_i}^{'}, Y_{S_i})\}_{i=1}^{n_S}$, $D_T^{'} = \{(X_{T_i}^{'})\}_{i=1}^{n_T}$. 5. Train an SVM classifier M_{model} on $D_S^{'}$. 6. Predict $\{(Y_{T_i})\}_{i=1}^{n_T}$ for $D_T^{'} = \{(X_{T_i}^{'})\}_{i=1}^{n_T}$ with M_{model}.

product reviews from the target domain, which also has adjectives as a feature, and that $X_T = \{x_{T_1}, x_{T_2}, \ldots, x_{T_j}, \ldots, x_{T_n}\}$. Here, $\{x_k\}_{k=1}^{N}$ is a PF set. Calculating the similarity based on Word2Vec, x_{s_i} is an SPF that is most similar to x_k in the source domain and the SPF x_{T_j} is most similar to x_k in the target domain. Then, based on the common PF x_k, we can construct the similar-pivot pair (x_{s_i}, x_{T_j}). We can then transform every source domain instance where feature x_{s_i} appears as follows: $X_S^{'} = \{x_{S_1}, x_{S_2}, \ldots x_{S_i}\_x_{T_j}, \ldots, x_{S_m}\}$. In addition, in the target domain, we can transform every instance where feature x_{T_j} appears in the same manner: $X_T^{'} = \{x_{T_1}, x_{T_2}, \ldots x_{S_i}\_x_{T_j}, \ldots, x_{T_n}\}$. In this way, the features x_{s_i} and x_{T_j} that express similar sentiment are transformed to a unified form that can avoid feature mismatches between different domains. The new datasets are denoted as $D_S^{'} = \{(X_{S_i}^{'}, Y_{S_i})\}_{i=1}^{n_S}$ and $D_T^{'} = \{(X_{T_i}^{'})\}_{i=1}^{n_T}$. By training a classifier, M_{model}, on $D_S^{'}$. we can then use it to predict $\{(Y_{T_i})\}_{i=1}^{n_T}$. The algorithm is shown in Table III.

TABLE IV
NUMBER OF REVIEWS IN THE BENCHMARK DATASET

DOMAIN	REVIEWS	POSITIVE	NEGATIVE
books	2,000	1,000	1,000
DVDs	2,000	1,000	1,000
electronics	2,000	1,000	1,000
kitchen	2,000	1,000	1,000

IV. EXPERIMENTS

A. Dataset and Evaluation Metric

The benchmark dataset collected by Blitzer et al. [1] has been widely used in many cross-domain sentiment classification methods. It contains Amazon product reviews of four different product type domains: books (B), DVDs (D), electronics (E) and kitchen appliances (K). The dataset includes 1,000 positive reviews and 1,000 negative reviews in each domain, as listed in Table IV. Each review is assigned a -1 (negative) or a +1 (positive) label according to the rating score given by a product user. In this dataset, we can construct 12 cross-domain sentiment classification tasks: D→B, E→B, K→B, K→E, D→E, B→E, B→D, K→D, E→D, B→K, D→K and E→K in which the letter preceding the arrow corresponds to the source domain, and the letter after the arrow corresponds to the target domain.

For each pair of cross-domain sentiment classification tasks, we evaluated the accuracy of our system's performance.

B. Baselines

To evaluate the effectiveness of CSCW, we compared our proposed method with several existing algorithms:

--SCL-MI is an improvement of SCL [1] that was proposed in [25]. SCL-MI exhibits beeter performance than SCL.

--SFA stands for spectral feature alignment and was proposed in [2]. It applies PFs as bridges to align domain-specific features from both domains in a latent space using a learned projection matrix.

--SS-FE was proposed in [8]. Similar to our work, it also adopts a POS-based method.

--CSCW is the method proposed in this paper.

C. Parameter Selection

We selected SVM as the base classification algorithm and used a linear kernel, conventional bag-of-words (BOW) representation and set the weight to tfidf. The number of PFs is N=800. Word2Vec adopts two main model architectures: a continuous bag-of-words (CBOW) model and a continuous skip-gram model. The CBOW model predicts the current word based on the context, and the skip-gram model predicts surrounding words given the current word. Our training algorithm used CBOW. Considering the size of the corpus, other parameters were set as follows: the dimensionality of the feature vector was θ =100; the maximum distance between the current and predicted word within a sentence was 5; the initial learning rate was 0.025, the random number generator (seed) was 1; no words were to be ignored; the threshold for configuring which higher-frequency words were randomly down-sampled is 0.001; we used negative sampling; the number of "noise words" that should be drawn was 5, using the mean of the context word vectors; weights were randomly initialized for increased training reproducibility; the number of iterations over the corpus was 5; the vocabulary was sorted by descending frequency before assigning word indexes; and the target size (in words) for the batches of examples passed to worker threads was 10,000.

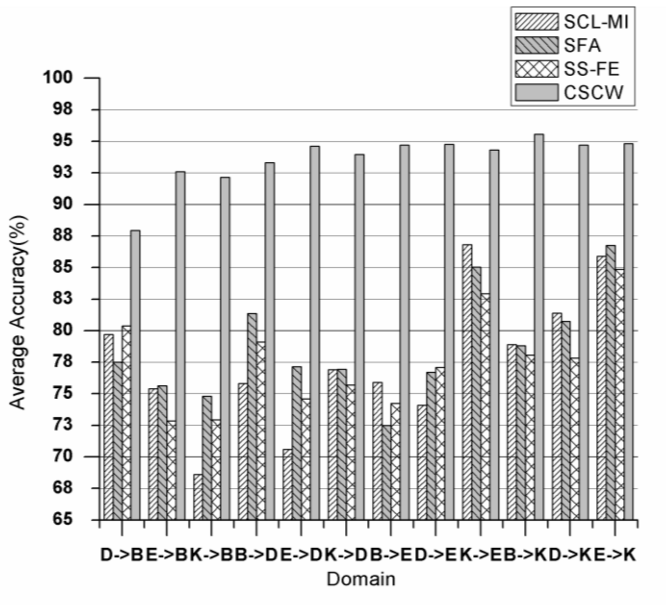

Fig. 1. Performance comparison of different methods.

D. Comparison Results

The performances of the different methods on each task are shown in Fig. 1, where each group of bars represents a cross-domain sentiment classification task. Each bar in a specific pattern represents a specific method. We compared the proposed method, CSCW, against three baselines: SCL-MI, SFA and SS-FE. The CSCW method outperforms all the baseline methods on all 12 domain pairs. There are two reasons for the dominance of our method. First, we select only adjectives as features because they are invariant across different domains and function as strong sentiment indicators. Second, we transform features from both domains to capture similar semantic relationships. This approach bridges the gap between domains by exploiting the sentiment semantic correlations between the domain PFs and the domain NPFs. In this way, we exploit the latent overlap in the level of sentiment semantics across domains and construct a shared latent-semantic subspace.

Performance comparisons with SFA and SCL-ML

SFA aligns domain-specific words from the source and target domains into clusters with the help of a PF to reduce the gap between the domain-specific words in the two domains. This method is based on a co-occurrence matrix. It cannot effectively capture similarities between words and neglects the semantic correlations between words. SCL-MI functions similarly. In the process of constructing correspondence correlation between different domains neither SCL-MI nor SFA consider semantic relationships; thus, they fail to find meaningful correspondence relationships between different domains. Although our work builds feature-correspondence correlations with the help of a PF in the same way, Word2Vec can learn the vector representations of words in the

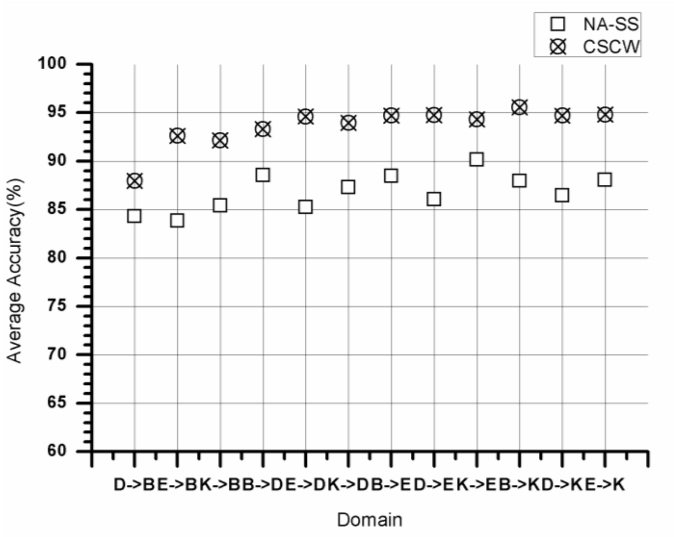

Fig. 2. Comparison diagram of NA-SS and CSCW.

high-dimensional vector space and accurately capture the semantic relationships between features. Using Word2Vec, semantically close words are likewise close in cosine distance in the lower dimensional vector space. With the help of PFs and Word2Vec, we can accurately capture sentiment features that are semantically close to PFs from either domain. Hence, the features that express similar sentiments but that use different representations will be identified. Then, we transform these identified features into a uniform representation. Finally, the features that represent similar sentiments are aligned to their maximum extent. Through this process, additional similar sentiment knowledge is obtained to reduce feature mismatches and the gap between different domains, which is beneficial to cross-domain sentiment classification.

Unlike SCL-MI and SFA, which use all the words in a domain as features, we extract only adjectives as features. As verified by our experiments, adjectives are strong indicators of sentiment. By extracting only adjectives, we exclude other POS-tag features that can magnify domain divergence. As features, adjectives avoid amplifying domain divergence to the greatest extent, and they are invariant and generalize well across different domains.

Performance comparisons with SS-FE

SS-FE is a method based on POS-tag. First, it re-weights different POS-tag types and integrates them into a FE. Second, it integrates the FEs with the sample selection of PCA-SS. Unlike our work, it is a method that does not consider addressing feature mismatches from the aspect of instance adaptation. The results demonstrate that resolving feature mismatching is more effective than instance adaptation.

E. Effect of Feature Transformations Based on Word2Vec

To verify the ability to transform similar features based on Word2Vec, Fig. 2 shows a comparison between NA-SS and CSCW. The NA-SS method also uses adjectives as features and applies a classifier trained on labeled source data to classify the target test data directly without making any other domain adaptation. We can observe that CSCW performs significantly better on all 12 domains compared to NA-SS. This result demonstrates that when using adjectives as features, transformations based on Word2Vec makes improvements in advance. Thus, with the help of a PF, Word2Vec can accurately capture semantic relationships between NPFs and PFs from both domains and build meaningful correspondence relationships between them.

TABLE V
NUMBER OF PIVOT FEATURE AND ALL FEATURES BETWEEN DOMAINS

Task	D_S PF/ All features	D_T PF/ All features	D_S ratio (%)	D_T ratio (%)
B-D	28037/32986	27049/31630	85.00	85.52
B-E	22422/32986	15811/19055	67.97	82.98
B-K	21953/32986	14092/16452	66.55	85.66
D-E	22054/31630	15414/19055	69.72	80.90
D-K	21430/31630	13974/16452	67.75	84.94
K-E	13795/16452	15607/19055	83.85	81.91

TABLE VI
EXAMPLES OF PIVOT FEATURE AND SIMILAR-PIVOT FEATURES

Domain	PF	D_S SPF	D_T SPF
D→B	interesting	exciting	entertaining
	good	great	bad
	annoying	inferior	defective
	awful	sad	disappointing
E→K	simple	useful	useful
	flat	removable	unnecessary
	accurate	precise	identical
K→D	great	nice	wonderful
	perfect	ideal	fantastic
	appropriate	express	easy

We also observe that on the tasks D→B, B→D, E→K and K→E, CSCW improved less compared to NA-SS than on other tasks. To help explain these results, Table V lists the total number of PFs and the total number of all features that appear in each domain. The first column shows the task. The second and third columns display the total number of PFs and the total number of all features appearing in each domain. The fourth and fifth columns provide the ratios of the total number of PFs and all features between the source and target domain. We can see that the ratios for task B-D are D_s =85.00% and D_T =85.52%, while for task K-E the ratios are D_s =83.85% and D_T =81.91%. In these two task pairs, the quantity of the total PF is larger than in the other tasks. It demonstrates that having a greater number of total PFs is not necessarily better. Too many will excessively change the laws of raw text and lead to a decline in classification accuracy.

Table VI shows examples of PFs and SPFs extracted by our method. Our method captures similar sentiment features from either domain. It should be noted that words such as “great” and “bad” are similar in Word2Vec, even though they express opposite sentiments — probably because that they are syntactically equivalent. This similarity introduces negative effects in our work.

F. Effect of POS-tag

To verify that adjectives are more domain invariant than other POS tags, we conducted an additional experiment called multi-POS. It should be noted that multi-POS uses the same

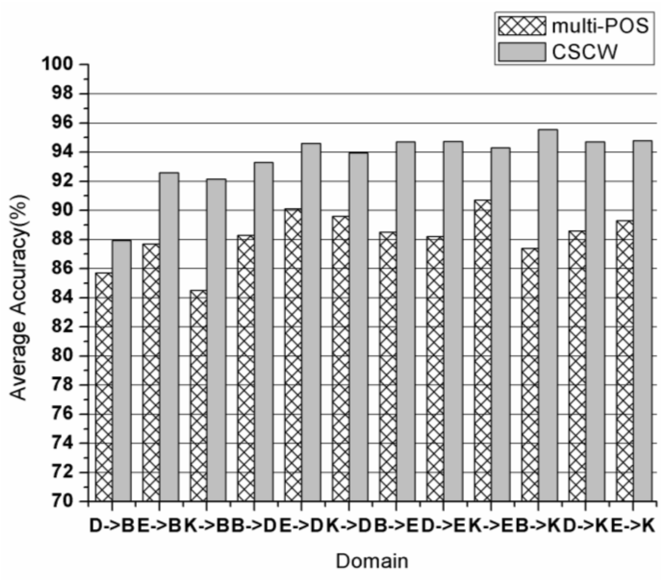

Fig. 3. Comparison diagram of multi-POS and CSCW.

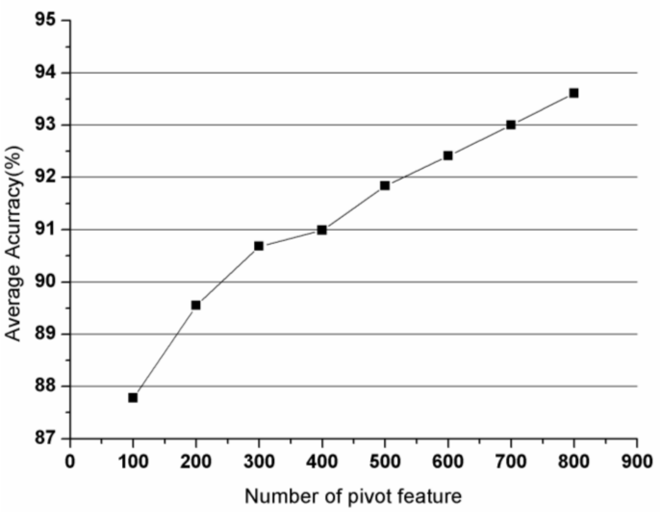

Fig. 4. Effect of number of pivot features.

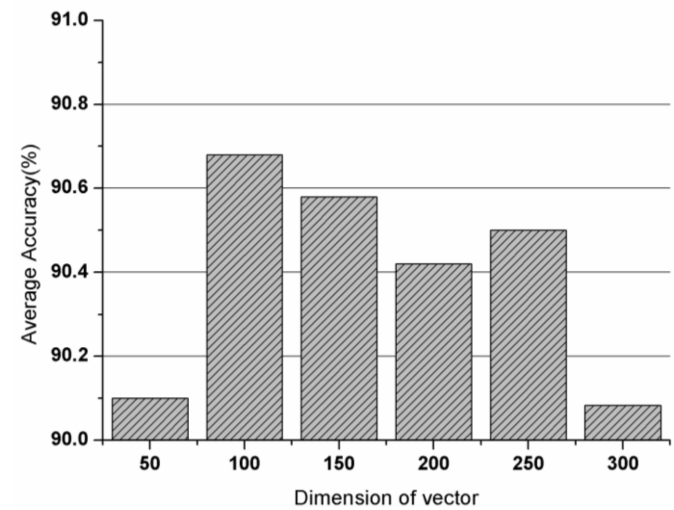

Fig. 5. Effect of word vector dimensionality.

method as CSCW but with a different selection of features; multi-POS includes verbs, nouns and adverbs as well as adjectives. Fig. 3 shows how CSCW compares with multi-POS. This experiment demonstrates that including verbs, nouns and adverbs as feature leads to worse predictive performance for sentiment classification because these word types magnify domain divergence. As features, adjectives avoid amplifying domain divergence, are invariant and generalize well across different domains.

G. Parameter Sensitivity Analysis

This section discusses the effect of varying the number of PFs and the dimensionality of the word vector.

Fig. 4 shows the effect of the number of PFs on CSCW's performance. We fixed the Word2Vec vector dimensionality to θ =100 and changed the number of PFs from 100 to 800 with a step size of 100. The average accuracy rises as we increase the number of PFs, demonstrating that with more PFs, we can construct more feature-correspondence correlations and, therefore, reduce sentiment feature mismatches across domains. Thus, we can better bridge the gap among different domains. Generally, with more PFs, the contribution to CSCW is greater.

Fig. 5 shows the effect of varying the word vector dimensionality on CSCW's performance. We fixed the number of PFs to N=300 and changed the Word2Vec vector dimensionality, θ , from 50 to 300 with a step size of 50. Fig. 5 shows that as the dimensionality of the word vector increases, the average accuracy also increases. In general, θ is between 100 and 300. For our corpus, the best performance is attained when the dimensionality θ =100. Increasing the dimensionality further results in instability. We can conclude that the selection of the Word2Vec dimensionality parameter should be based on the size of corpus.

H. Error Analysis

To provide a road map for future work in cross-domain text sentiment classification, we analyzed the errors produced by our method. For example, the following is the raw text from a misclassified positive review of a DVD product:

"*R.J. The Raccoon (Bruce Willis) was just looking for... Extremely entertaining and fantastic CGI animated comedy from... provides great realistic CGI and laughs for everybody. There's also heart in this movie and shows the meaning of what it is like to have friends and be one's true self instead of being selfish! ... it's a wonderful and hilarious movie that can be enjoyed by both kids and adults...Highly recommended! The best DreamWorks animated movie...*".

For the B→D task, our method transforms the raw text into the following pattern:

"*bear crimson_entired large constant_later professional carb_new fantastic fastpaced_great realistic worthwhile_true selfish hammy latter_best scary_wonderful hilarious dvd fastpaced_good audio commercial animated*".

We can see that some of the useful adjectives such as "entertaining" and "best" are not identified via the NLTK tools. This result demonstrates that a more accurate part-of-speech tagger could improve the results of our method. Because our method relies solely on extracted adjectives as features, these omissions mean that our method ignores meaningful features that are not adjectives, such as "laugh", "enjoy", and "highly recommended". These features are verbs that are also indicators of sentiment polarity and that would be meaningful in cross-domain sentiment classification. However, not all verbs are useful for sentiment polarity—for example "study", "drink" and "exploit." If sentiment features can be identified correctly regardless of their POS, we could enlarge the common shared sentiment subspace across domains. These two factors affect classification performance and result in misclassifications.

V. Conclusion

In this paper, we propose a cross-domain semantic correlation feature-auto-correspondence method for cross-domain sentiment classification called CSCW. Because adjectives are strong indicators of sentiment and vary only slightly across different domains, we selected adjectives as sentiment-invariant features. First, we chose PFs by exploiting their common frequency between a source domain and a target domain. Second, with the help of Word2Vec and PFs, we selected SPFs and transformed them in each domain to unify the features that have similar sentiments in both domains. In this way, we were able to reduce feature mismatches between different domains, which improves the accuracy of a classifier trained on the source domain and tested on the target domain. The results of experiments showed that the proposed method can outperform competitive baseline approaches; CSCW achieved the best sentiment classification accuracies for all the tested cross-domain pairs.

In the future, we plan to study other classification tasks in addition to sentiment classification. We also plan to test our technique on a larger and more varied set of domains with the help of a Word2Vec model trained on Wikipedia or Google News.

References

[1] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006, pp. 120–128.

[2] 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.

[3] 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.

[4] D. Bollegala, T. Mu, 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.

[5] G. Zhou, T. He, W. Wu, and X. T. Hu, "Linking heterogeneous input features with pivots for domain adaptation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 1419–1425.

[6] D. Huynh, D. Tran, and W. Ma, "Contextual analysis for the representation of words," *IAENG International Journal of Computer Science*, vol. 41, no. 2, pp. 148-152, 2014.

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

[8] R. Xia, C. Zong, X. Hu, and E. Cambria, "Feature ensemble plus sample selection: domain adaptation for sentiment classification," *Intelligent Systems, IEEE*, vol. 28, no. 3, pp. 10–18, 2013.

[9] X. Ding, T. Liu, J. Duan, and J. Y. Nie, "Mining user consumption intention from social media using domain adaptive convolutional neural network." in *AAAI*, 2015, pp. 2389–2395.

[10] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for largescale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.

[11] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *Unsupervised and Transfer Learning Challenges in Machine Learning*, vol. 7, pp. 19-39, 2012.

[12] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. J. Goodfellow, et al., "Unsupervised and transfer learning challenge: a deep learning approach." *ICML Unsupervised and Transfer Learning*, vol. 27, pp. 97–110, 2012.

[13] B. Liu, M. Huang, J. Sun, and X. Zhu, "Incorporating domain and sentiment supervision in representation learning for domain adaptation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 1277–1283.

[14] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, et al., "How transferable are neural networks in nlp applications?" *arXiv preprint arXiv:1603.06111*, 2016.

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, et al., "Domain-adversarial training of neural networks," *arXiv preprint arXiv:1505.07818*, 2015.

[16] M. Seera and C. P. Lim, "Transfer learning using the online fuzzy min–max neural network," *Neural Computing and Applications*, vol. 25, no. 2, pp. 469–480, 2014.

[17] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[18] J. T. Lalis, "A New Multiclass Classification Method for Objects with Geometric Attributes Using Simple Linear Regression," *IAENG International Journal of Computer Science*, vol. 43, no. 2, pp. 198–203, 2016.

[19] J. M. Yang, Z. Y. Liu, and Z. Y. Qu, "Clustering of Words Based on Relative Contribution for Text Categorization," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 207-219, 2013.

[20] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, 1997, pp. 174–181.

[21] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 417–424.

[22] 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*, 2002, pp. 79–86.

[23] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[24] J. Wiebe, "Learning subjective adjectives from corpora," in *AAAI/IAAI*, 2000, pp. 735–740.

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