Cross-Domain Sentiment Classification with Word Embeddings and Canonical Correlation Analysis

Ngo Xuan Bach
Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
bachnx@ptit.edu.vn

Vu Thanh Hai
FPT Software Research Lab, Vietnam
haivt2@fsoft.com.vn

Tu Minh Phuong
Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
phuongtm@ptit.edu.vn

ABSTRACT

A common approach for automatic sentiment classification is using classifiers trained on labeled text data (reviews, blog posts etc.) to predict the sentiment polarity of new data. Because people express sentiment differently in different domains, this approach requires annotated corpora for each domain. However, annotating data for every domain of interest is laborious and impractical. In this paper, we address the domain adaptation problem for sentiment classification. We explore the effect of generic methods for feature learning and feature subspace mapping, namely word embeddings and canonical correlation analysis (CCA), on cross-domain sentiment classifiers. We show that by using only such rather generic methods, it is possible to get results very competitive with those of sophisticated methods specially developed for the considered problem. An advantage of using word embeddings and CCA is their availability out-of-the-box, which is important for the applicability of the proposed method. Experiments on a widely used benchmark dataset shows that both word embeddings and CCA contribute to accuracy improvement and their combination provides the best results.

CCS Concepts

•**Computing methodologies** → **Lexical semantics;** Semi-supervised learning settings;

Keywords

Cross-domain sentiment classification; canonical correlation analysis; word embeddings

1. INTRODUCTION

Sentiment analysis is an attractive research topic with many applications in e-commerce, marketing, social analysis etc. [19]. The ability to detect and classify sentiment expressed in reviews, blog posts, and comments allows getting information about users' evaluation of products, brands, or events [8, 19]. Given a review, the task is to detect sentiment (if present) and classify it into several categories, for example "positive" or "negative". Sentiment classification is commonly cast as a supervised learning problem, where a collection of documents (reviews, comments) are annotated with sentiment labels and used for training a classifier [1, 2, 7, 19, 24].

A difficulty when applying sentiment classification is that people use different expressions and words to express sentiment for different domains. For example, we use words like "thrilling", "romantic", or "lengthy" to describe our opinion about movies but rarely use them for electronic devices. Instead, words like "reliable", "cheap", or "compacts" are often used in Electronics domain. Because of the differences in vocabularies and word meanings, a model trained for one domain may not make accurate predictions when directly applied to other domains. To create accurate classifiers, researchers and practitioners would need to annotate training data for every new domain of interest. Obviously, this is laborious and impractical, given the very large number of possible domains of interest. One way to alleviate this problem is domain adaptation, i.e. adapting a classifier trained on labeled data from one domain to classify sentiment for another domain. This setting is known as cross-domain sentiment classification and has received a considerable research interest [2, 3, 12, 23, 26, 30, 31, 32].

In this work, we address the problem of domain adaptation for sentiment classification with two techniques: 1) word embeddings and 2) feature space mapping via *Canonical Correlation Analysis* (CCA). With word embeddings, words or phrases are mapped to vectors of real numbers of low dimensions relative to the original vocabulary size. If the mapping is learned (in an unsupervised manner) from a large and generic enough corpus, words with similar semantic will tend to be close in the new vector space. In the context of sentiment classification, this means sentiment bearing words are close in the new vector space even if they are from different domains, provided they are learned from a large and diverged corpus. This should alleviate the vocabulary mismatch between domains.

In addition to word embeddings, we use CCA as a subspace mapping method to transform features from two domains to a common feature space before learning classification models. Specifically, given a source and a target domain, we identify domain-independent words and use CCA to find the mapping that makes the domain-independent words most correlated with domain-dependent words for

SoICT 2016, December 08-09, 2016, Ho Chi Minh, Viet Nam.

DOI: http://dx.doi.org/10.1145/3011077.3011104

each domain. By mapping the domain-independent and domain-dependent words to the new subspace recovered by CCA, we create new features to use with classifiers. The idea is that in the new feature subspaces, the domain-independent and domain-dependent features are correlated and thus the presence of the former is indicative of the later. In this way, the newly created domain-dependent features act as a bridge to reduce the gap between the two domains.

Previously, both word embeddings and feature space mapping have been used for sentiment classification [3, 20, 23, 28]. However, in previous work, special forms of embeddings were designed for sentiment classifiers, for example by training on corpora specially annotated for sentiment [28], or by specially designed embedding algorithms [3, 20]. The need for additional corpora or special algorithms requires additional effort to annotate and implement. In this work, we show that by combining only word vectors trained on generic corpora and a feature mapping method available in any statistical library it is possible to adapt sentiment classifiers and achieve improvements very competitive with specially designed algorithms. The use of generic corpora and generic algorithms is important for the applicability of domain adaptation in sentiment classification. To evaluate the proposed method, we performed experiments on a widely used cross-domain dataset provided by John Blitzer et al. [2]. Our method achieved results comparable with two influential methods, specially designed for cross-domain sentiment classification.

The rest of the paper is organized as follows. In the next section we briefly review related work. In Section 3 we present the use of word vectors to extend the features used in cross-domain settings and describe how to further improve the results by using CCA. Section 4 presents experiments and results. The conclusion comes in Section 5.

2. RELATED WORK

Cross-domain sentiment classification is a hot research topic with a considerable number of papers published. In a pioneer work, Blitzer et al. [2] extend *structural correspondence learning* (SCL), a method for domain adaptation, to sentiment classification. SCL consists of several steps. First, it finds a set of pivots - words that occur in both domains. Second, linear predictors are trained to predict the presence/absence of pivots in a document. Finally, predicted pivots are added to original features and serve as input for a logistic regression model. The use of predicted pivots helps to reduce the mismatch between two domains.

Spectral feature alignment (SFA) [23] is another notable method. SFA first identify domain-independent and domain specific features. The method then creates a bipartite graph that connects the two types of features based on their co-occurrences. A spectral clustering algorithm is used to find cluster of connected features. Finally, cluster features are added to original features. In a recent work, Bollegala et al. [3] propose using word embeddings to resolve for domain mismatch. They design an algorithm able to learn embeddings that are sensitive to sentiment classification.

Deep learning has also been applied to cross-domain sentiment classification. Glorot et al. [11] use several denoising autoencoders stacked together to learn features from pooled reviews of different domains. They then use support vector machines (SVMs) trained on the source domain with the learned features to predict sentiment for the target domain.

Xia et al. [31] combine ensemble learning with subspace feature mapping for domain adaptation. The ensemble-learning component consist of several classifiers, each use only a subset of word features such as nouns, verbs, adjectives. Feature mapping is done via principal component analysis (PCA). They argue that different types of words suffer to domain mismatch to different extends and ensemble of classifiers using different word types can alleviate the domain mismatch problem, while PCA can help to find subspaces in which features from two domain have similar distributions.

In cross-domain sentiment classification, one usually has access to both labeled data in the source domain and unlabeled data in the target domain. This is the same setting as in semi-supervised learning and semi-supervised learning methods can be applied in a natural way. An example of method along this line is by Chen et al. [5], in which they use co-training for domain adaptation.

Beside studies that focus on domain adaptation for a single source domain, Wu and Huang [30] propose a method to transfer sentiment signals from multiple sources to a target domain.

3. PROPOSED METHOD

3.1 Method Overview

As shown in Figure 1, our method first employs several feature learning and feature subspace mapping techniques to achieve a rich feature set. Then a strong learning method is applied to the feature set to produce the classification model. There are three types of features as follows:

- **Raw features**: raw features are n-grams extracted from reviews. N-grams are fundamental yet effective features for most text classification problems.
- **Word embedding features**: word embedding features are low-dimensional features built from raw features. By learning from a large, generic collection of text, word embeddings are expected to produce similar representations for similar words in different domains.
- **CCA features**: CCA features are created by mapping domain-dependent words and domain-independent words to new subspaces, which act as a bridge between the domain-dependent words of the source and the target domains.

In the following, we describe how to produce word embedding features and CCA features as well as the learning method used in our framework.

3.2 Word Embeddings

Word representation is a basic step in most tasks in natural language processing. A traditional method for word representation is one-hot representation, which is based on the indices of the words (or n-grams) in a dictionary. In this method, each word (or n-gram) is represented as a vector with all zero elements except for an element with 1. Suppose we have a dictionary consisting of N words, the k^{th} word in the dictionary will be represented as an N-dimensional vector which consists of $(N-1)$ zero values and a 1 at the k^{th} position. Using this representation method, a phrase, a sentence, or a document can also be represented as an N-dimensional vector whose the value of the k^{th} element equals

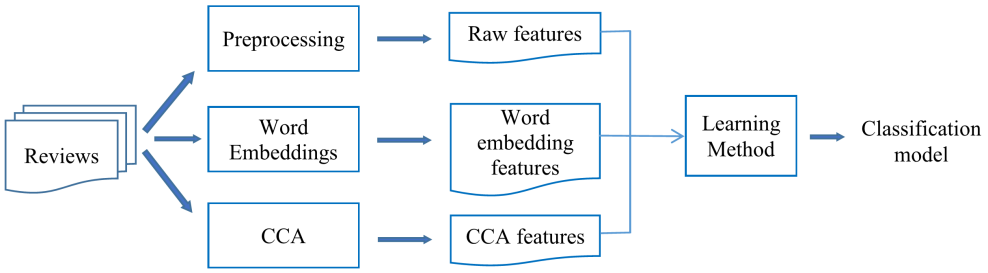

Figure 1: Overview of the proposed method.

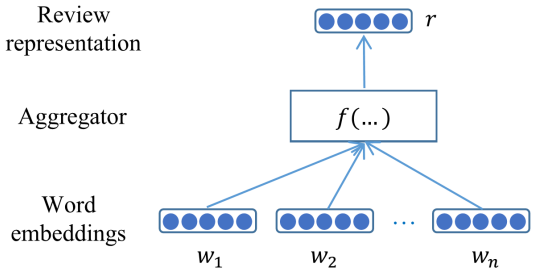

Figure 2: Review representation from word vectors.

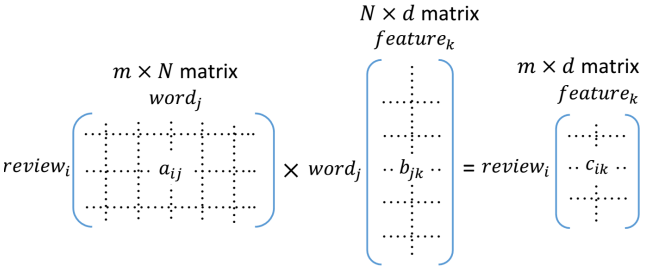

Figure 3: Aggregator function through matrix multiplication.

to the number of the corresponding word appearing in the phrase, the sentence, or the document. Such representation is usually referred as bag-of-word method.

In the cross-domain setting, classification models using one-hot representation may not perform well due to vocabulary mismatch. For example, "hilarious" is used in Movies domain but rarely appear in Electronics domain, while "reliable" is often used when reviewing electronic devices. In one-hot representations, the similarity between those two words is zero, although both are used in positive reviews.

Recently, several distributed representation methods, referred as word embeddings, have been developed to represent words [9, 10, 21, 25] from a large collection of raw text. The goal is to represent words by fix-size, dense, meaningful vectors. A word embedding method represents each word by an d-dimensional vector of real values. Usually, d is much smaller than N, the size of the dictionary. This representation is expected to capture the semantics of words and relations between related words [21, 25].

In this work, to capture the semantics of reviews we also exploit distributed representations of words. Figure 2 illustrates our method for review representation from word embeddings. First, each word w_i is represented by a d-dimensional vector $v(w_i)$. Then, an aggregator function is applied to represented word vectors to produce a vector representation for the review:

$$v(r) = f(v(w_1), v(w_2), \ldots, v(w_l)),$$

where l is the length in words of the review r.

Our aggregator function is accomplished through matrix multiplication of the review-word matrix, a $m \times N$ matrix whose element a_{ij} represents the frequency of the j^{th} word in the i^{th} review, and the word embedding matrix, a $N \times d$ matrix whose j^{th} row represents the word embedding of the j^{th} word, as shown in Figure 3. Here m, N, and d denote the number of reviews, the number of words or the size of the dictionary, and the number of dimensions in word embeddings, respectively. The output is a matrix of size $m \times d$, whose the i^{th} row represents the d-dimensional word embedding vector for the i^{th} review.

To represent words as vectors, i.e. word embeddings, we employ GloVe word representation method proposed by Pennington et al. [25]. The method combines the advantages of the two major model families in word representation, i.e., global matrix factorization and local context window methods, and efficiently leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. GloVe word vectors have been shown to be meaningful and efficient in a number of NLP tasks [25]. The authors of GloVe provide several sets of pre-trained word vectors, making it simple to implement our method.

3.3 CCA Features

As mentioned in the previous sections, the performance of classifiers often decreases in a new domain (called target domain) due to vocabulary mismatch between the original (called source domain) and the target domains. In this section, we describe how to use CCA to alleviate this problem.

Assume we are given reviews from a source domain and a target domain. Reviews from the source domain are labeled (e.g. as "positive", "negative", or "neutral") while reviews from the target domain are not. By $W^I = (w^I_1, w^I_2, \ldots, w^I_{|W^I|})$

we denote domain-independent words, i.e. the words that appear in both domain. By $W^s = (w_1^s, w_2^s, \ldots, w_{|W^s|}^s)$ (and $W^t = (w_1^t, w_2^t, \ldots, w_{|W^t|}^t)$) we denote source (and target) domain-specific or domain-dependent words, i.e. the words that appear in the source (and target) domain only or appear in one domain much more frequently than in the other one. Using these notations, a source domain review r^s is represented as

$$r^s = (W^s, W^I).$$

Similarly, a target domain review r^t is represented as

$$r^t = (W^t, W^I).$$

Because W^t do not (or rarely) appear in the source domain, a classification model trained on source domain reviews will ignore words from W^t, thus loosing information from target-specific words. To reduce this negative effect, we will try to predict the presence of target specific words W^t based on the presence of domain independent words W^I. In this work, we achieve this by finding linear combinations of words in W^t, which are correlated with combinations of words in W^I. As an example, assume that when domain independent words w_1^I, w_2^I, w_3^I appear in a target review, target specific words w_2^t and w_4^t also appear with high probability. In other words, combination (w_1^I, w_2^I, w_3^I) is highly correlated with combination (w_2^t, w_4^t). In this case, the presence of the former is predictive of the latter and vice versa. Since the combination (w_1^I, w_2^I, w_3^I) is predictive of some target specific words, it is useful to guide the classification model to focus on them during training. We can do this by adding the combination of (w_1^I, w_2^I, w_3^I) to the feature set. Similarly, combinations of domain-independent words that are correlated with combinations of source dependent words are important and should be used as additional features because they allow to transfer information from source domain via independent words.

More generally, we seek to find all linear mappings of domain-independent and target-specific/source-specific features so that once mapped, the new, transformed features are correlated. Note that, for this step, we use unlabeled data from each domain to find mappings for this domain (for source domain data, we simply ignore labels). We use the transformed features to augment the original feature set by concatenating the two feature vectors. In the next sections, we describe how to use CCA to find such mappings, how to find domain-independent and domain-dependent features, and how to compute the new feature set.

3.3.1 Canonical Correlation Analysis

To save the space, in this section we only explain how to use CCA to find linear correlations between independent and target dependent features. The correlations between independent and source dependent features are recovered similarly.

CCA [13] is a multivariate technique that aims to describe the correlations between two groups of variables. In our case, the two groups of variables are domain-independent and target-specific features. CCA aims to find linear combinations of variables from each group so that the results are maximally correlated. In other words, it finds the two bases for each group in which the correlation matrix between them is diagonal and the correlations on the diagonal are maximized.

Given two sets of domain-independent and target-specific words, which are represented as two sets of columns $\mathbf{W}^{It} = [\mathbf{w}_1^{It}, \mathbf{w}_2^{It}, \ldots, \mathbf{w}_{|W^I|}^{It}]$ and $\mathbf{W}^t = [\mathbf{w}_1^t, \mathbf{w}_2^t, \ldots, \mathbf{w}_{|W^t|}^t]$ respectively. Each column $\mathbf{w}_j^{It}$ of size m contains the number of appearances of word w_j^I in m reviews, normalized to have zero mean. Similarly, each column $\mathbf{w}_j^t$ contains the number of appearances of word w_j^t in m reviews, normalized to have zero means. CCA learns the projection vectors $\mathbf{u}^{It} \in \mathbb{R}^{|W^I|}$ and $\mathbf{u}^t \in \mathbb{R}^{|W^t|}$ to maximize the correlation coefficient between $\mathbf{W}^{It}\mathbf{u}^{It}$ and $\mathbf{W}^t\mathbf{u}^t$, i.e.:

$$\max_{\mathbf{u}^{It}, \mathbf{u}^t} \rho = \frac{\mathbf{u}^{It\top}\mathbf{W}^{It\top}\mathbf{W}^t\mathbf{u}^t}{\sqrt{\mathbf{u}^{It\top}\mathbf{W}^{It\top}\mathbf{W}^{It}\mathbf{u}^{It}}\sqrt{\mathbf{u}^{t\top}\mathbf{W}^{t\top}\mathbf{W}^t\mathbf{u}^t}} \quad (1)$$

where $\top$ denotes matrix transpose.

This optimization problem can be solved as a generalized eigenvalue decomposition problem. Generally, up to $min(|W^I|, |W^t|)$ pairs of vectors $\mathbf{u}^{It} \in \mathbb{R}^{|W^I|}$ and $\mathbf{u}^t \in \mathbb{R}^{|W^t|}$ can be learned by maximizing the objective function in (1), so that each pair is orthogonal to previously found pairs. CCA is provided in many statistical packages or libraries like R or Matlab, which makes it simple to implement our method.

CCA and its variations have been exploited successfully in a number of tasks, including part-of-speech tagging [15], word representations [27], label representations [17], and lexical representations [16].

3.3.2 Dependent and Independent features

We now describe how to find domain-independent and domain-dependent features (or independent and dependent features for short). Recall that independent features are those that appear in both domain with similar frequencies while dependent features appear only in one domain or appear in one domain much more frequently than in the other one.

To distinguish between dependent and independent features in the source and target domains, we compute the mutual information between each feature and the domain label. Mutual information between two random variables is a measure of how much the presence of one variable tells about the presence of the other. The less the mutual information value between a feature and a domain is, the more likely the feature is independent to the domain. And vice versa, a large mutual information value means higher dependence. The mutual information MI between feature w_i and domain $D \in \{s, t\}$ is computed as:

$$MI(w_i, D) = \sum_{w_i} \sum_{D=s,t} p(w_i, D) \log\left(\frac{p(w_i, D)}{p(w_i)p(D)}\right)$$

where $p(w_i, D)$ is the probability of word w_i appearing in domain D; $p(w_i)$ and $p(D)$ are the probabilities of observing w_i and D, respectively.

We remove features that rarely appear in both domains. The remaining features are divided into three subsets: domain-independent W^I, source-dependet W^s, and target-dependent W^t as follows. First, features with the lowest MI values are selected to form W^I. The remaining features are further divided into W^s and W^t. To decide whether a feature belongs to W^s or W^t, we compute the numbers of times the feature appears in the source and the target domains. W^s consists of features with large values in the source domain

but small values in the target domain, while W^t contains features with small values in the source domain but large values in the target domain.

3.3.3 *Computing CCA features*

Having the dependent and independent features defined, the next step is to perform the CCA calculation on them. Recall that by $\mathbf{W}^{It}$ and $\mathbf{W}^t$ we denote the matrix representations of samples in the target domain using feature sets W^I and W^t, respectively. $\mathbf{W}^{It}$ is a matrix of size $m \times |W^I|$ and $\mathbf{W}^t$ is a matrix of size $m \times |W^t|$, where m is the number of samples in the target domain. Similarly, let $\mathbf{W}^{Is} \in \mathbb{R}^{n \times |W^I|}$ and $\mathbf{W}^s \in \mathbb{R}^{n \times |W^s|}$ be the matrix representations of samples in the source domain using feature sets W^I and W^s, respectively, where n is the number of samples in the source domain.

We apply CCA to learn correlations for each pair ($\mathbf{W}^{It}$, $\mathbf{W}^t$), and ($\mathbf{W}^{Is}$, $\mathbf{W}^s$) . Solving optimization problem in (1) for pair ($\mathbf{W}^{It}$, $\mathbf{W}^t$), we can find maximum $min(|W^I|, |W^t|)$ pairs of vector $\mathbf{u}^{It} \in \mathbb{R}^{|W^I|}$ and $\mathbf{u}^t \in \mathbb{R}^{|W^t|}$,from which we keep the first K_t vector pairs (which correspond to K_t largest eigenvectors), where K_t is a parameter. Let $\mathbf{U}^{It} = [\mathbf{u}_1^{It}, \mathbf{u}_2^{It}, \ldots, \mathbf{u}_{K_t}^{It},]$ be the matrix of size $|W^I| \times K_t$ formed by putting together the K_t vectors $\mathbf{u}^t$. We then form K_t transformed features by projecting the independent features to new bases represented by $\mathbf{U}^{It}$ as follows:

$$\mathbf{C}^t = \mathbf{W}^{It}\mathbf{U}^{It} \tag{2}$$

Similarly, performing CCA on ($\mathbf{W}^{Is}$, $\mathbf{W}^s$), we can find K_s projection vectors to form projection matrix$\mathbf{U}^{Is}$ and compute transformed features:

$$\mathbf{C}^s = \mathbf{W}^{Is}\mathbf{U}^{Is} \tag{3}$$

Note that, in general, K_t and K_s are different. However, for simplicity, in the experiments, we set $K_t = K_s = K$, where K is a parameter, whose value is chosen experimentally. Finally, we add all the new features computed by (2) and (3) to the original feature sets in both domains and use the new feature set for the classification models.

3.4 Learning Method

As the learning method, we choose Support Vector Machines (SVMs), a start-of-the-art statistical machine learning technique proposed by Vapnik et al. [6, 29]. SVMs have been demonstrated their performance on a number of problems in areas, including image processing, handwriting recognition, and statistical natural language processing. In the field of natural language processing, SVMs have been applied successfully to text categorization [14], word sense disambiguation [18], syntactic parsing [22], sentiment classification [1, 7, 24], among the others, and achieved very good results.

4. EXPERIMENTS

4.1 Data and Experimental Setup

We conducted experiments on the most popular dataset for cross-domain sentiment classification research introduced by Blitzer et al. [2]. The dataset contains product reviews collected from four different domains of Amazon.com, i.e., Books (B), DVD (D), Electronics (E), and Kitchen (K). Each domain consists of 1000 positive and 1000 negative labeled reviews[1].

To conduct experiments, we randomly divided each domain into two sets: a training set of 1600 reviews and a test set of 400 reviews. Table 1 shows the dataset used in experiments in detail. We trained a classification model on the training set of a (source) domain and tested the model on the test set of another (target) domain. Totally, four domains produced 12 source-target scenarios. All classification models were trained using SVM[2]. The performance of classification systems was measured using accuracy:

$$accuracy = \frac{\#\text{of correctly classified reviews}}{\#\text{of reviews}}.$$

4.2 Models to Compare

We conducted experiments to compare the following models:

- **Baseline**: The model used only raw features, i.e., unigrams and bigrams.
- **WordVec**: The model used word embedding features combined with the raw features. The purpose of conducting experiments with this model is to evaluate the effectiveness of the word embedding features on cross-domain sentiment classification. In our experiments, we used 300-dimensional word vectors[3] trained from Wikipedia by Pennington et al. [25].
- **CCA**: The model used CCA features combined with the word embedding features and raw features. The purpose is to evaluate the effectiveness of CCA features on the task. When selecting dependent and independent features, we removed all features that appear less 3 times in both source and target domains. In our experiments, each domain has 2000 samples; the number of independent and dependent features were set to 1000 and 200, respectively; and K was set as the minimum rank of two input matrices of CCA computation.
- **In-Domain**: This model used the training set from the same domain with the test set. The model used only raw features. It provides experimental results of in-domain classification and compares to the results of cross-domain classification.

4.3 Results

Experimental results on four target domains, i.e. Books, DVD, Electronics, and Kitchen, are shown in Figure 4. In each chart, the red horizontal line demonstrates the result of the in-domain experiment. The letter on the left hand side of the arrow denotes the source domain, while the letter on the right hand side of the arrow denotes the target domain. Among 12 scenarios, the WordVec model achieved better results than the baseline in 9 cases and gave the same results in 2 other cases. The CCA model outperformed the baseline model in 10 cases. Comparing between WordVec and CCA

[1]Dataset available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[2]We used LIBSVM [4] with linear kernel. Software available at: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3]Available at: http://nlp.stanford.edu/projects/glove/

Table 1: Datasets used in experiments

Domain	Denoted by	#of training reviews	#of test reviews
Books	B	1600	400
DVD	D	1600	400
Electronics	E	1600	400
Kitchen	K	1600	400

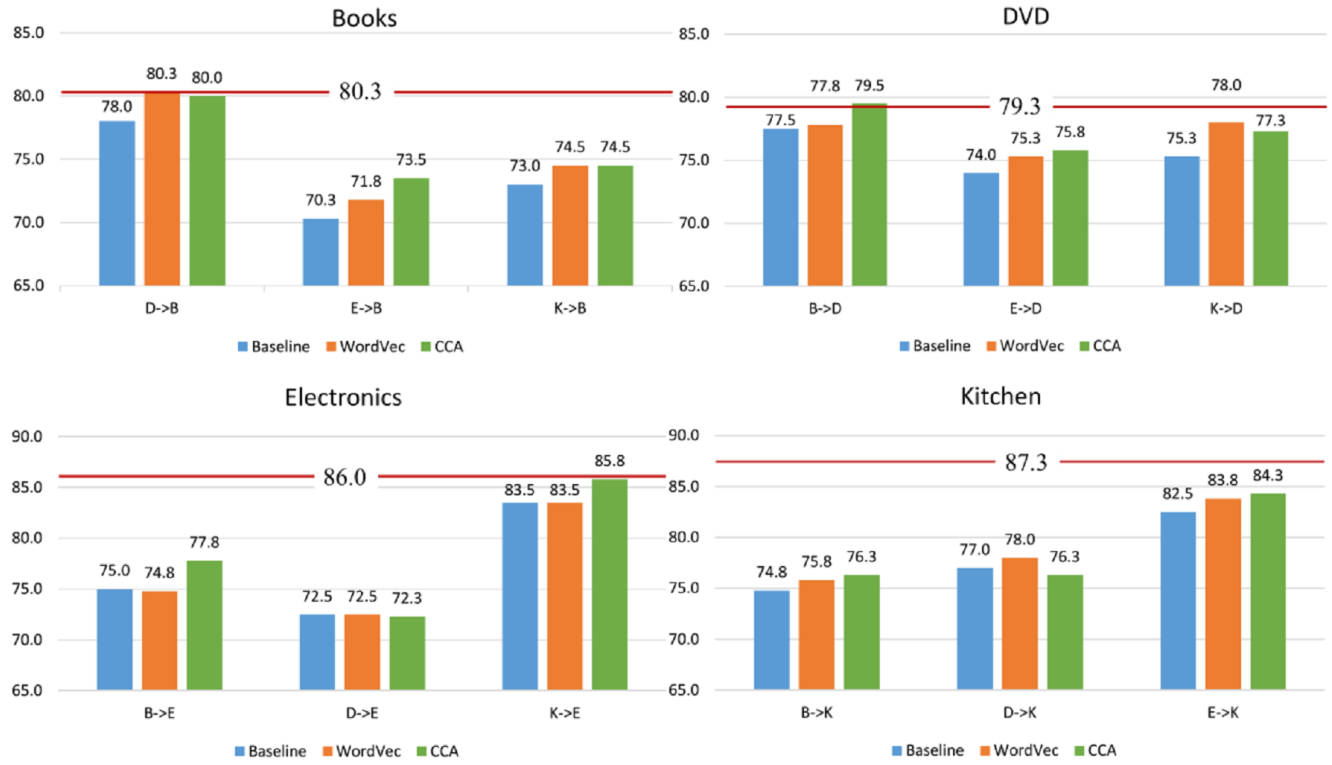

Figure 4: Experimental results of the proposed method in 12 source-target domain pairs.

models, the CCA model got better results in 7 cases and the same results in another case.

On average, the CCA model achieved 77.8% accuracy, while the baseline model and the WordVec model got 76.1% and 77.2% accuracy, respectively. Among four target domains, our method achieved the comparative results with the In-Domain model in 3 cases, i.e, Books, DVD, and Electronics.

4.4 Comparison with Previous Work

We compare experimental results of our model with the work of Blitzer et al. [2] and Xia et al. [31], which use the same dataset with similar settings to our work[4]. Experimental results are shown in Figure 5. Once again, in each chart, the letter on the left hand side of the arrow denotes the source domain, while the letter on the right hand side of the arrow denotes the target domain. Although our method uses only techniques available out-of-the-box, the results are comparable with methods specially designed and implemented for domain adaptation. Our model got better results than the work of Blitzer et al. in 5 scenarios and better results than the work of Xia et al. in 7 scenarios among 12 source-target pairs. On average, our model achieved 77.8% accuracy in comparison with 78.0% and 77.6% accuracy of Blizer et al. and Xia et al., respectively.

[4]Both Blitzer et al. [2] and Xia et al. [31] use 1600 reviews for training and 400 reviews for testing for each source-target domain pair like our work.

5. CONCLUSION

We have presented a simple yet generic and effective method for cross-domain sentiment classification. Our method employs feature learning and feature subspace mapping, i.e. word embeddings and canonical correlation analysis (CCA), to deal with the vocabulary mismatch problem between the source and target domains. Experiments on 12 source-target domain pairs showed that our method achieved competitive results with sophisticated methods specially developed for cross-domain sentiment classification. Due to the availability out-of-the-box of feature transfer learning methods, our method is easy to adapt to other natural language processing tasks.

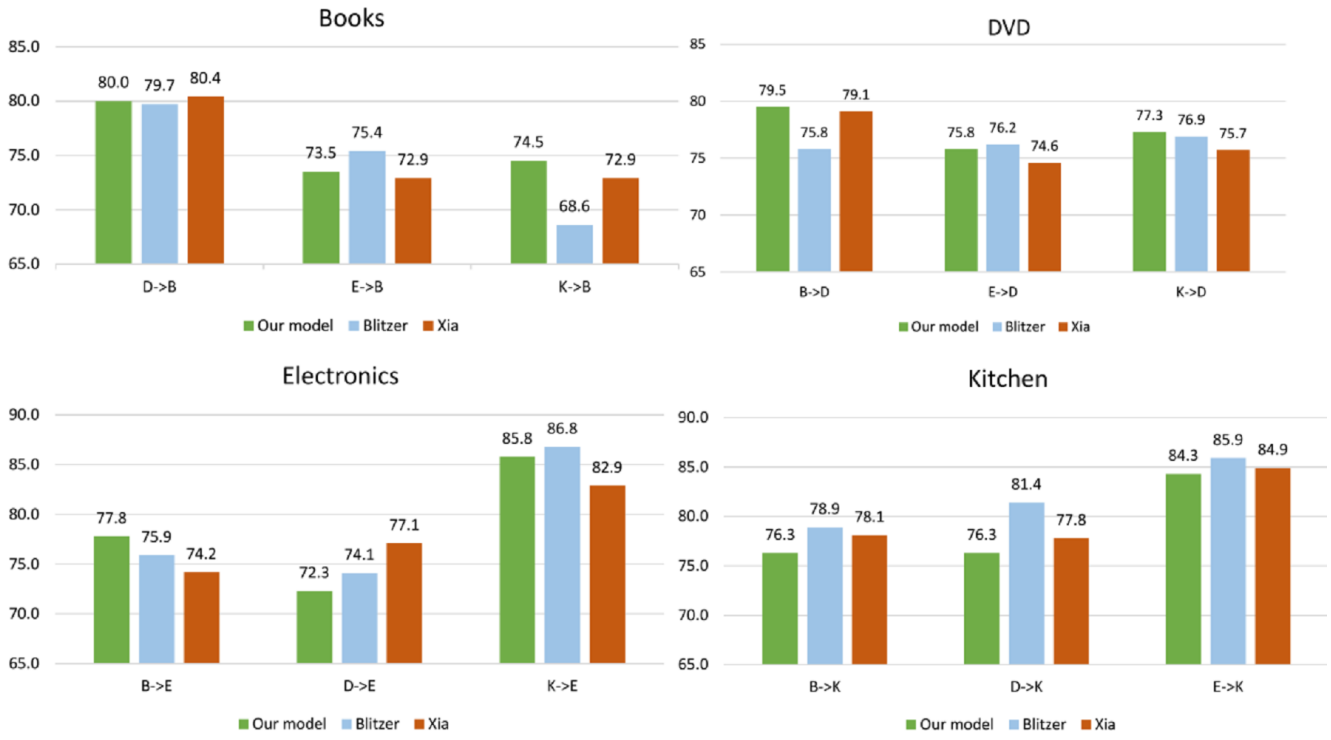

Figure 5: Experimental results in comparison with previous work.

6. ACKNOWLEDGMENTS

This work was supported in part by FPT Software. We are grateful to Nguyen Thai Thuy Chung for his support in data preparation

7. REFERENCES

[1] N. X. Bach and T. M. Phuong. Leveraging user ratings for resource-poor sentiment classification. In *Proceedings of the 19th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES)*, pages 322–331, 2015.

[2] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447, 2007.

[3] D. Bollegala, T. Mu, and J. Y. Goulermas. Cross-domain sentiment classification using sentiment sensitive embeddings. *IEEE Transaction on Knowledge and Data Engineering*, 28(2), 2016.

[4] C. C. Chang and C. J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[5] M. Chen, K. Weinberger, and J. Blitzer. Co-training for domain adaptation. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, 2011.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] N. T. Duyen, N. X. Bach, and T. M. Phuong. An empirical study on sentiment analysis for vietnamese. In *Proceedings of the International Conference on Advanced Technologies for Communications (ATC), Special session on Computational Science and Computational Intelligence (CSCI)*, pages 309–314, 2014.

[8] X. Fang and J. Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(5), 2015.

[9] M. Faruqui and C. Dyer. Non-distributional word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 464–469, 2015.

[10] M. Faruqui, Y. Tsvetkov, D. Yogatama, C. Dyer, and N. Smith. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1491–1500, 2015.

[11] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classi cation: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 513–520, 2011.

[12] Y. He, C. Lin, and H. Alani. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 123–131, 2011.

[13] H. Hotelling. Relations between two sets of variates.

Biometrika, 28(3):321–377, 1936.

[14] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, 1998.

[15] Y. B. Kim, B. Snyder, and R. Sarikaya. Part-of-speech taggers for low-resource languages using cca features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1292–1302, 2015.

[16] Y. B. Kim, K. Stratos, X. Liu, and R. Sarikaya. Compact lexicon selection with spectral methods. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 806–811, 2015.

[17] Y. B. Kim, K. Stratos, R. Sarikaya, and M. Jeong. New transfer learning techniques for disparate label sets. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 473–482, 2015.

[18] Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for wordsense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural LanguageProcessing (EMNLP)*, pages 41–48, 2002.

[19] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[20] A. Maas, R. Daly, P. Pham, D. Huang, A. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150, 2011.

[21] T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[22] J. Nivre, J. Hall, J. Nilsson, G. Eryigit, and S. Marinov. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Conference on Computational Natural LanguageLearning (CoNLL)*, pages 221–225, 2006.

[23] S. J. Pan, X. Ni, J. 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 (WWW)*, pages 751–760, 2010.

[24] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[25] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[26] N. Ponomareva and M. Thelwall. Do neighbours help?: An exploration of graph-based algorithms for cross-domain sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 655–665, 2012.

[27] P. Rastogi, B. V. Durme, and R. Arora. Multiview lsa: Representation learning via generalized cca. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 556–566, 2015.

[28] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 1631–1642, 2013.

[29] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[30] F. Wu and Y. Huang. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 301–310, 2016.

[31] R. Xia, C. Zong, X. Hu, and E. Cambria. Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18, 2013.

[32] G. Zhou, Y. Zhou, X. Guo, X. Tu, and T. He. Cross-domain sentiment classification via topical correspondence transfer. *Neurocomputing*, 159:298–305, 2015.