Bi-Transferring Deep Neural Networks for Domain Adaptation

Guangyou Zhou[1], **Zhiwen Xie**[1], **Jimmy Xiangji Huang**[2], **and Tingting He**[1]
[1] School of Computer, Central China Normal University, Wuhan 430079, China
[2] School of Information Technology, York University, Toronto, Canada
{gyzhou,xiezhiwen,tthe}@mail.ccnu.edu.cn jhuang@yorku.ca

Abstract

Sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of user generated sentiment data (e.g., reviews, blogs). Due to the mismatch among different domains, a sentiment classifier trained in one domain may not work well when directly applied to other domains. Thus, domain adaptation for sentiment classification algorithms are highly desirable to reduce the domain discrepancy and manual labeling costs. To address the above challenge, we propose a novel domain adaptation method, called Bi-Transferring Deep Neural Networks (BTDNNs). The proposed BTDNNs attempts to transfer the source domain examples to the target domain, and also transfer the target domain examples to the source domain. The linear transformation of BTDNNs ensures the feasibility of transferring between domains, and the distribution consistency between the transferred domain and the desirable domain is constrained with a linear data reconstruction manner. As a result, the transferred source domain is supervised and follows similar distribution as the target domain. Therefore, any supervised method can be used on the transferred source domain to train a classifier for sentiment classification in a target domain. We conduct experiments on a benchmark composed of reviews of 4 types of Amazon products. Experimental results show that our proposed approach significantly outperforms the several baseline methods, and achieves an accuracy which is competitive with the state-of-the-art method for domain adaptation.

1 Introduction

With the rise of social media (e.g., blogs and social networks etc.), more and more user generated sentiment data have been shared on the Web (Pang et al., 2002; Pang and Lee, 2008; Liu, 2012; Zhou et al., 2011). They exist in the form of user reviews on shopping or opinion sites, in posts of blogs/questions or customer feedbacks. This has created a surge of research in sentiment classification (or sentiment analysis), which aims to automatically determine the sentiment polarity (e.g., positive or negative) of user generated sentiment data (e.g., reviews, blogs, questions).

Machine learning algorithms have been proved promising and widely used for sentiment classification (Pang et al., 2002; Pang and Lee, 2008; Liu, 2012). However, the performance of these models relies on manually labeled training data. In many practical cases, we may have plentiful labeled data in the source domain, but very few or no labeled data in the target domain with a different data distribution. For example, we may have many labeled books reviews, but we are interested in detecting the polarity of electronics reviews. Reviews for different products might have different vocabularies, thus classifiers trained on one domain often fail to produce satisfactory results when transferring to another domain. This has motivated much research on cross-domain (domain adaptation) sentiment classification which transfers the knowledge from the source domain to the target domain (Thomas et al., 2006; Snyder and Barzilay, 2007; Blitzer et al., 2006; Blitzer et al., 2007; Daume III, 2007; Li and Zong, 2008; Li et al., 2009; Pan et al., 2010; Kumar et al., 2010; Glorot et al., 2011; Chen et al., 2011a; Chen et al., 2012; Li et al., 2012; Xia et al., 2013a; Li et al., 2013; Zhou et al., 2015a; Zhuang et al., 2015).

Depending on whether the labeled data are available for the target domain, cross-domain sen-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 322–332,
Berlin, Germany, August 7-12, 2016.

timent classification can be divided into two categories: supervised domain adaptation and unsupervised domain adaptation. In scenario of supervised domain adaptation, labeled data is available in the target domain but the number is usually too small to train a good sentiment classifier, while in unsupervised domain adaptation only unlabeled data is available in the target domain, which is more challenging. This work focuses on the unsupervised domain adaptation problem of which the essence is how to employ the unlabeled data of target domain to guide the model learning from the labeled source domain.

The fundamental challenge of cross-domain sentiment classification lies in that the source domain and the target domain have different data distribution. Recent work has investigated several techniques for alleviating the domain discrepancy: instance-weight adaptation (Huang et al., 2007; Jiang and Zhai, 2007; Li and Zong, 2008; Mansour et al., 2009; Dredze et al., 2010; Chen et al., 2011b; Chen et al., 2011a; Chen et al., 2012; Li et al., 2013; Xia et al., 2013a) and feature representation adaptation (Thomas et al., 2006; Snyder and Barzilay, 2007; Blitzer et al., 2006; Blitzer et al., 2007; Li et al., 2009; Pan et al., 2010; Zhou et al., 2015a; Zhuang et al., 2015). The first kind of methods assume that some training data in the source domain are very useful for the target domain and these data can be used to train models for the target domain after re-weighting. In contrast, feature representation approaches attempt to develop an adaptive feature representation that is effective in reducing the difference between domains.

Recently, some efforts have been initiated on learning robust feature representations with deep neural networks (DNNs) in the context of cross-domain sentiment classification (Glorot et al., 2011; Chen et al., 2012). Glorot et al. (2011) proposed to learn robust feature representations with stacked denoising auto-encoders (SDAs) (Vincent et al., 2008). Denoising auto-encoders are one-layer neural networks that are optimized to reconstruct input data from partial and random corruption. These denoisers can be stacked into deep learning architectures. The outputs of their intermediate layers are then used as input features for SVMs (Fan et al., 2008). Chen et al. (2012) proposed a marginalized SDA (mSDA) that addressed the two crucial limitations of SDAs: high

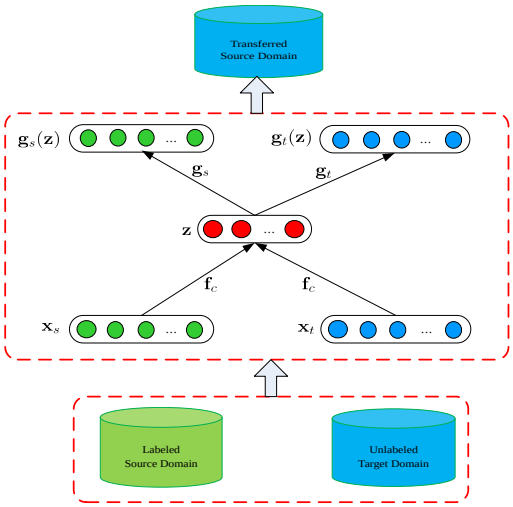

Figure 1: The framework of Bi-transferring Deep Neural Networks (BTDNNs). Through BTDNNs, a source domain example can be transferred to the target domain where it can be reconstructed by the target domain examples, and vice versa.

computational cost and lack of scalability to high-dimensional features. However, these methods learn the unified domain-invariable feature representations by combining the source domain data and that of the target domain data together, which cannot well characterize the domain-specific features as well as the commonality of domains.

To this end, we propose a Bi-Transferring Deep Neural Networks (BTDNNs) which can transfer the source domain examples to the target domain and also transfer the target domain examples to the source domain, as shown in Figure 1. In BTDNNs, the linear transformation makes the feasibility of transferring between domains, and the linear data reconstruction manner ensures the distribution consistency between the transferred domain and the desirable domain. Specifically, our BTDNNs has one common encoder $\mathbf{f}_c$, two decoders $\mathbf{g}_s$ and $\mathbf{g}_t$ which can map an example to the source domain and the target domain respectively. As a result, the source domain can be transferred to the target domain along with its sentiment label, and any supervised method can be used on the transferred source domain to train a classifier for sentiment classification in the target domain, as the transferred source domain data share the similar distribution as the target domain. Experimental results show that the proposed approach significantly outperforms several baselines, and achieves an accuracy which is competitive with the state-of-

the-art method for cross-domain sentiment classification.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes our proposed bi-transferring deep neural networks (BTDNNs). Section 4 presents the experimental results. In Section 5, we conclude with ideas for future research.

2 Related Work

Domain adaptation aims to generalize a classifier that is trained on a source domain, for which typically plenty of training data is available, to a target domain, for which data is scarce. Cross-domain generalization is important in many real applications, the key challenge is that data in the source and the target domain are often distributed differently.

Recent work has investigated several techniques for alleviating the difference in the context of cross-domain sentiment classification task. Blitzer et al. (2007) proposed a structural correspondence learning (SCL) algorithm to train a cross-domain sentiment classifier. SCL is motivated by a multi-task learning algorithm, alternating structural optimization (ASO), proposed by Ando and Zhang (2005). Given labeled data from a source domain and unlabeled data from both source and target domains, SCL attempts to model the relationship between "pivot features" and "non-pivot features". Pan et al. (2010) proposed a spectral feature alignment (SFA) algorithm to align the domain-specific words from the source and target domains into meaningful clusters, with the help of domain-independent words as a bridge. In the way, the cluster can be used to reduce the gap between domain-specific words of two domains. Dredze et al. (2010) combined classifier weights using confidence-weighted learning, which represented the covariance of the weight vectors. Xia et al. (2013a) proposed an instance selection and instance weighting method for cross-domain sentiment classification. After that, Xia et al. (2013b) proposed a feature ensemble plus sample selection method to further improve the sentiment classification adaptation. Zhou et al. (Zhou et al., 2015b) proposed to bridge the domain gap with the help of topical correspondence. Li et al. (2009) proposed to transfer common lexical knowledge across domains via matrix factorization techniques. Zhou et al. (2015a) further improved the matrix factorization techniques via a regularization term on the pivots and domain-specific words, ensuring that the pivots capture only correspondence aspects and the domain-specific words capture only individual aspects. Li and Zong (2008) proposed the multi-label consensus training approach which combined several base classifiers trained with SCL. Chen et al. (2012) proposed a domain adaptation algorithm based on sample and feature selection. Li et al. (2013) proposed an active learning algorithm for cross-domain sentiment classification. Xiao and Guo (2013) investigated the online active domain adaptation problem in a novel but practical setting where the labels can be acquired with a lower cost in the source domain than in the target domain.

There has also been research in exploring careful structuring of features or prior knowledge for domain adaptation. Daumé III (2007) proposed a kernel-mapping function which maps both source and target domains data to a high-dimensional feature space so that data points from the same domain are twice as similar as those form different domains. Dai et al. (2008) proposed translated learning which used a language model to link the class labels to the features in the source domain, which in turn is translated to the features in the target domain. Xia et al. (2010) proposed a POS-based ensemble model for cross-domain sentiment classification. Xiao et al. (2013) proposed a supervised representation learning method to tackle domain adaptation by inducing predictive latent features based on supervised word clustering. He et al. (2011) employed a joint sentiment-topic model for cross-domain sentiment classification; Bollegala et al. (2011) used a sentiment sensitive thesaurus to perform cross-domain sentiment classification. Xiao and Guo (2015) proposed to learn distributed state representations for cross-domain sequence predictions.

Recently, some efforts have been initiated on learning robust feature representations with deep neural networks (DNNs) for cross-domain natural language processing. Glorot et al. (2011) and Chen et al. (2012) proposed to use deep learning for cross-domain sentiment classification. Most recently, Yang and Eisenstein (2014) proposed an unsupervised domain adaptation method with marginalized structured dropout. Furthermore, Yang and Eisenstein (2015) proposed to use feature embeddings with metadata domain attributes for multi-domain adaptation. In this paper,

our proposed approach BTDNNs tackles the domain discrepancy with a linear data construction manner, which can effectively model the domain-specific features as well as the commonality of domains. Deep learning techniques have also been proposed to heterogeneous transfer learning (Socher et al., 2013; Zhou et al., 2014; Kan et al., 2015; Long et al., 2015), where knowledge is transferred from one modality to another based on the correspondences at hand. Our proposed framework can be considered as a more general case, where the bias of the correspondences between the source and target domains is constrained with a linear data reconstruction manner.

Besides, other researchers also explore the DNNs for sentiment analysis (Socher et al., 2011; Tang et al., 2014; Tang et al., 2015; Zhai and Zhang, 2016; Chandar et al., 2014). However, all these methods focus on the sentiment analysis without considering the domain discrepancy. In this paper, we focus on domain adaptation for sentiment classification with a different model formulation and task definition.

3 Bi-Transferring Deep Neural Networks

3.1 Problem Definition

Given two domains X_s and X_t, where X_s and X_t are referred to a source domain and a target domain, respectively. Suppose we have a set of labeled sentiment examples as well as some unlabeled examples in the source domain X_s with size n_s, containing terms from a vocabulary $\mathcal{V}$ with size m. The examples in the source domain X_s can be represented as a term-document matrix $\mathbf{X}_s = \{\mathbf{x}_1^s, \cdots, \mathbf{x}_{n_s}^s\} \in \mathbb{R}^{m \times n_s}$, with their sentiment labels $\mathbf{y}_s = \{y_1^s, \cdots, \mathbf{y}_{n_s}^s\}$, where $\mathbf{x}_i^s \in \mathbb{R}^m$ is the feature representation of the i-th source domain example with a tf-idf weight of the corresponding term and $y_i^s \in \{+1, -1\}$ is its sentiment label.[1]

Similarly, suppose we have a set of unlabeled examples in the target domain X_t with size n_t, containing terms from a vocabulary $\mathcal{V}$ with size m. The examples in target domain X_t can also be represented as a term-document matrix $\mathbf{X}_t = \{\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_{n_t}^{(t)}\} \in \mathbb{R}^{m \times n_t}$, where each example denotes a tf-idf weight of the corresponding term. The task of cross-domain sentiment classification is to learn a robust classifier to predict the polarity of unseen examples from X_t. Note that we only consider one source domain and one target domain in this paper. However, our proposed algorithm is a general framework and can be easily adapted to multi-domain problems.

[1]We use upper case and lower case characters represent the matrices and vectors respectively throughout the paper.

3.2 Basic Auto-Encoder

An auto-encoder is an unsupervised neural network which is trained to reconstruct a given input vector from its latent representation (Bengio et al., 2007). It can be seen as a special neural network with three layers: the input layer, the latent layer, and the reconstruction layer. An auto-encoder contains two parts: encoder and decoder. The encoder, denoted as $\mathbf{f}$, attempts to map an input vector $\mathbf{x} \in \mathbb{R}^{m \times 1}$ to the latent representation $\mathbf{z} \in \mathbb{R}^{k \times 1}$, in which k is the number of neurons in the latent layer. Usually, $\mathbf{f}$ is a nonlinear function as follows:

$$\mathbf{z} = \mathbf{f}(\mathbf{x}) = s_e(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1}$$

where s_e is the activation function of the encoder, whose input is called the activation function, which is usually non-linear, such as sigmoid function or tanh function is a linear transform parameter, and $\mathbf{b} \in \mathbb{R}^{k \times 1}$ is the basis.

The decoder, denoted as $\mathbf{g}$, tries to map the latent representation $\mathbf{z}$ back to a reconstruction:

$$\mathbf{g}(z) = s_d(\mathbf{W}'\mathbf{z} + \mathbf{b}') \tag{2}$$

Similarly, s_d is the activation function of the decoder with parameters $\{\mathbf{W}', \mathbf{b}'\}$.

The training objective is the determination of parameters $\{\mathbf{W}, \mathbf{b}\}$ and $\{\mathbf{W}', \mathbf{b}'\}$ that minimize the average reconstruction errors:

$$\mathcal{L} = \min_{\mathbf{W},\mathbf{b},\mathbf{W}',\mathbf{b}'} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{g}(\mathbf{f}(\mathbf{x}_i)) \right\|_2^2 \tag{3}$$

where $\mathbf{x}_i$ represents the i-th one of N training examples. Parameters $\{\mathbf{W}, \mathbf{b}\}$ and $\{\mathbf{W}', \mathbf{b}'\}$ can be optimized by stochastic or mini-batch gradient descent. By minimizing the reconstruction error, we require the latent features should be able to reconstruct the original input as much as possible.

3.3 Bi-Transferring Deep Neural Networks

The traditional auto-encoder in subsection 3.2 attempts to reconstruct the input itself, which is usually used for feature representation learning. Nevertheless, our proposed bi-transferring deep neural networks (BTDNNs) attempts to transfer

examples between domains to deal with the domain discrepancy, with the inspiration of DNNs in computer vision (Kan et al., 2015). Motivated by the successful application in computer vision (Kan et al., 2015), we construct the architecture of BTDNNs with one encoder $\mathbf{f}_e$, and two decoders, $\mathbf{g}_s$ and $\mathbf{g}_t$ shown in Figure 1, which can transform an input example to the source domain and the target domain respectively.[2]

Specifically, the encoder $\mathbf{f}_c$ tries to map an input example $\mathbf{x}$ into the latent feature representation $\mathbf{z}$, which is common to both the source and target domains as follows:

$$\mathbf{z} = \mathbf{f}_c(\mathbf{x}) = s_e(\mathbf{W}_c\mathbf{x} + \mathbf{b}_c) \tag{4}$$

The decoder $\mathbf{g}_s$ attempts to map the latent representation to the source domain, and the decoder $\mathbf{g}_t$ attempts to map the latent representation to the target domain as follows:

$$\mathbf{g}_s(\mathbf{x}) = s_d(\mathbf{W}_s\mathbf{z} + \mathbf{b}_s) \tag{5}$$

$$\mathbf{g}_t(\mathbf{x}) = s_d(\mathbf{W}_t\mathbf{z} + \mathbf{b}_t) \tag{6}$$

where $s_e(\cdot)$ and $s_d(\cdot)$ are the element-wise nonlinear activation function, e.g., sigmoid or tanh function, $\mathbf{W}_c$ and $\mathbf{b}_c$ are the parameters for encoder $\mathbf{f}_c$, $\mathbf{W}_s$ and $\mathbf{b}_s$ are the parameters for decoder $\mathbf{g}_s$, $\mathbf{W}_t$ and $\mathbf{b}_t$ are the parameters for decoder $\mathbf{g}_t$.

Following the literature (Kan et al., 2015), we attempt to map the source domain examples $\mathbf{X}_s$ to the source domain (e.g., $\mathbf{X}_s$ itself) with an encoder $\mathbf{f}_c$ and a decoder $\mathbf{g}_s$. Similarly, given an encoder $\mathbf{f}_c$ and a decoder $\mathbf{g}_t$, we aim to map the source domain examples $\mathbf{X}_s$ to the target domain. Although it is unknown what the mapped examples look like, they are expected to follow the similar distribution as the target domain. This kind of distribution consistency between two domains can be characterized from the perspective of a linear data reconstruction manner.

The two domains $\mathbf{X}_s$ and $\mathbf{X}_t$ can be generally reconstructed from each other, and their distances can be used to measure the domain discrepancy. Following the literature (He et al., 2012), BTDNNs attempt to represent a transferred source domain $\mathbf{g}_t(\mathbf{f}_c(\mathbf{x}_i^s))$ with a linear reconstruction function from the target domain:

$$\|\mathbf{g}_t(\mathbf{f}_c(\mathbf{x}_i^s)) - \mathbf{X}_t\beta_i^t)\|_2^2 \tag{7}$$

[2]In the implementation, we use the stacked denoising auto-encoders (SDA) (Vincent et al., 2008) to model the source and the target domain data.

where β_i^t is the coefficients for the reconstruction of transferred source domain examples. Equation (7) enforces that each example of transferred domain is consistent with that of target domain, which ensures that the transferred source domain follows the similar distribution as the target domain. The overall objective for the examples of source domain $\mathbf{X}_s$ can be formulated as below:

$$\min_{\mathbf{f}_c,\mathbf{g}_s,\mathbf{g}_t,\beta_i^s} \|\mathbf{X}_s - \mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_s))\|_2^2 + \|\mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_s)) - \mathbf{X}_t\mathbf{B}_t)\|_2^2$$
$$s.t.\ \|\beta_i^t\|_2^2 < \tau,\ \ \mathbf{B}_t = [\beta_1^t, \beta_2^t, \cdots, \beta_{n_s}^t]^T \in \mathbb{R}^{n_s \times n_t}$$

where $\mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_s) = [\mathbf{g}_s(\mathbf{f}_c(\mathbf{x}_1^s)), \cdots, \mathbf{g}_s(\mathbf{f}_c(\mathbf{x}_{n_t}^s))]$ and $\mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_s) = [\mathbf{g}_t(\mathbf{f}_c(\mathbf{x}_1^t)), \mathbf{g}_t(\mathbf{f}_c(\mathbf{x}_{n_s}^t))]$. The same simplifications are used hereinafter if without misunderstanding.

Similarly, for the examples of target domain $\mathbf{X}_t$, with encoder $\mathbf{f}_c$ and decoder $\mathbf{g}_t$ they should be mapped on the target domain. Also, with encoder $\mathbf{f}_c$ and decoder $\mathbf{g}_s$ they should be mapped to the source domain, where they can be reconstructed by the source domain examples from the point of view of a linear data reconstruction manner (He et al., 2012), so as to ensure a similar distribution between the source domain and the transferred target domain. The overall objective for the examples of target domain $\mathbf{X}_t$ can be written as:

$$\min_{\mathbf{f}_c,\mathbf{g}_s,\mathbf{g}_t,\beta_i^t} \|\mathbf{X}_t - \mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_t))\|_2^2 + \|\mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_t)) - \mathbf{X}_s\mathbf{B}_s)\|_2^2$$
$$s.t.\ \|\beta_j^s\|_2^2 < \tau,\ \ \mathbf{B}_s = [\beta_1^s, \beta_2^s, \cdots, \beta_{n_t}^s]^T \in \mathbb{R}^{n_t \times n_s}$$

Combining the above equations, the overall objective of BTDNNs can be formulated as follows:

$$\begin{aligned}\min_{\mathbf{f}_c,\mathbf{g}_s,\mathbf{g}_t,\mathbf{B}_s,\mathbf{B}_t} & \|\mathbf{X}_s - \mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_s))\|_2^2 + \|\mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_s)) - \mathbf{X}_t\mathbf{B}_t)\|_2^2 \\ & + \|\mathbf{X}_t - \mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_t))\|_2^2 + \|\mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_t)) - \mathbf{X}_s\mathbf{B}_s)\|_2^2 \\ & + \gamma\Big(\sum_{i=1}^{n_s}\|\beta_i^t\|_2^2 + \sum_{j=1}^{n_t}\|\beta_j^s\|_2^2\Big)\end{aligned} \tag{8}$$

where γ is a regularization parameter controlling the amount of shrinkage. With the optimization of equation (8), our proposed approach BTDNNs can map any input examples to the source and target domains respectively. Especially, the source domain examples $\mathbf{X}_s$ can transferred to the target domain along with their sentiment labels. The transferred source domain data $\mathbf{g}_t(\mathbf{f}_s(\mathbf{X}_s))$ share the similar distribution as the target domain, so any supervised method can be used to learn a classifier for sentiment classification in the target domain. In this paper, a linear support vector machine (SVM) (Fan et al., 2008) is employed for building sentiment classification models.

3.4 Learning Algorithm

Note that the optimization problem in equation (8) is not convex in variables $\{\mathbf{f}_c, \mathbf{g}_s, \mathbf{g}_t, \mathbf{B}_s, \mathbf{B}_t\}$ together. However, when considering one variable at a time, the cost function turns out to be convex. For example, given $\{\mathbf{g}_s, \mathbf{g}_t, \mathbf{B}_s, \mathbf{B}_t\}$, the cost function is a convex function w.r.t. $\mathbf{f}_c$. Therefore, although we cannot expect to get a global minimum of the above problem, we shall develop a simple and efficient optimization algorithm via alternative iterations.

3.4.1 Optimize $\{\mathbf{f}_c, \mathbf{g}_s, \mathbf{g}_t\}$ given $\{\mathbf{B}_s, \mathbf{B}_t\}$

When $\mathbf{B}_s$ and $\mathbf{B}_t$ are fixed, the objective function in equation (8) can be formulated as:

$$\min_{\mathbf{f}_c,\mathbf{g}_s,\mathbf{g}_t} \|\mathbf{X}_s - \mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_s))\|_2^2 + \|\mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_s)) - \bar{\mathbf{X}}_t)\|_2^2 + \|\mathbf{X}_t - \mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_t))\|_2^2 + \|\mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_t)) - \bar{\mathbf{X}}_s)\|_2^2 \quad (9)$$

where $\bar{\mathbf{X}}_s = \mathbf{X}_s\mathbf{B}_s$ and $\bar{\mathbf{X}}_t = \mathbf{X}_t\mathbf{B}_t$. Equation (9) can easily optimized by gradient descent as the basic auto-encoder (Bengio et al., 2007).

3.4.2 Optimize $\{\mathbf{B}_s, \mathbf{B}_t\}$ given $\{\mathbf{f}_c, \mathbf{g}_s, \mathbf{g}_t\}$

When $\{\mathbf{f}_c, \mathbf{g}_s, \mathbf{g}_t\}$ are fixed, the objective function in equation (8) can be written as:

$$\min_{\mathbf{B}_s,\mathbf{B}_t} \|\mathbf{G}_t - \mathbf{X}_t\mathbf{B}_t)\|_2^2 + \|\mathbf{G}_s - \mathbf{X}_s\mathbf{B}_s)\|_2^2 + \gamma\Big(\sum_{i=1}^{n_s}\|\beta_i^t\|_2^2 + \sum_{j=1}^{n_t}\|\beta_j^s\|_2^2\Big)$$

where $\mathbf{g}_s(\mathbf{f}_c(\mathbf{X}_t)) = \mathbf{G}_s = [\mathbf{g}_1^s, \cdots, \mathbf{g}_{n_t}^s]$ and $\mathbf{g}_t(\mathbf{f}_c(\mathbf{X}_s)) = \mathbf{G}_t = [\mathbf{g}_1^t, \cdots, \mathbf{g}_{n_s}^t]$. Since $\mathbf{G}_s$ and $\mathbf{G}_t$ are independent with each other, so they can be optimized independently. The optimization of $\mathbf{G}_s$ with other variables fixed is a least squares problem with ℓ_2-regularization. It can also be decomposed into n_t optimization problems, with each corresponding to one β_j^s and can be solved in parallel:

$$\min_{\beta_j^s} \|\mathbf{g}_j^s - \mathbf{X}_s\beta_j^s\|_2^2 + \gamma\|\beta_j^s\|_2^2 \quad (10)$$

for $j = 1, 2, \cdots, n_t$. It is a standard ℓ_2-regularized least squares problem and the solution is:

$$\beta_j^s = \left(\mathbf{X}_s^T\mathbf{X}_s + \gamma\mathbf{I}\right)^{-1}\mathbf{X}_s^T\mathbf{g}_j^s \quad (11)$$

where $\mathbf{I}$ is an identity matrix with all entries equal to 1.

Similarly, The optimization of $\mathbf{G}_t$ can also be decomposed into n_s ℓ_2-regularized least squares problems and the solution of each one is:

$$\beta_i^t = \left(\mathbf{X}_t^T\mathbf{X}_t + \gamma\mathbf{I}\right)^{-1}\mathbf{X}_t^T\mathbf{g}_i^t \quad (12)$$

for $i = 1, 2, \cdots, n_s$. We repeat the above equations until $\mathbf{f}_c$, $\mathbf{g}_s$, $\mathbf{g}_t$, $\mathbf{B}_s$ and $\mathbf{B}_t$ converge or a maximum number of iterations is exceeded.

3.5 Algorithm Complexity

In this section, we analyze the computational complexity of the learning algorithm described in equations (9), (11) and (12). Besides expressing the complexity of the algorithm using big O notation, we also count the number of arithmetic operations to provide more details about the run time. Computational complexity of learning matrix $\mathbf{G}_s$ is $O(m \times n_s \times k)$ per iteration. Similarly, for each iteration, learning matrices $\mathbf{G}_t$ takes $O(m \times n_t \times k)$. Learning matrices $\mathbf{B}_s$ and $\mathbf{B}_t$ takes $O(m^2 \times n_s)$ and $O(m^2 \times n_t)$ operations per iteration. In real applications, we have $k \ll m$. Therefore, the overall complexity of the algorithm, dominated by computation of matrices $\mathbf{B}_s$ and $\mathbf{B}_t$, is $O(m^2 \times n)$ where $n = \max(n_s, n_t)$.

4 Experiments

4.1 Data Set

Domain adaptation for sentiment classification has been widely studied in the NLP community. A large majority experiments are performed on the benchmark made of reviews of Amazon products gathered by Blitzer et al. (2006). This data set contains 4 different domains: Book (B), DVDs (D), Electronics (E) and Kitchen (K). For simplicity and comparability, we follow the convention of (Blitzer et al., 2006; Pan et al., 2010; Glorot et al., 2011; Xiao and Guo, 2013) and only consider the binary classification problem whether a review is positive (higher than 3 stars) or negative (3 stars or lower). There are 1000 positive and 1000 negative reviews for each domain, as well as approximately 4,000 unlabeled reviews (varying slightly between domains). The positive and negative reviews are also exactly balanced.

Following the literature (Pan et al., 2010), 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, E → K, where the word before an arrow corresponds with the source domain and the word after an arrow corresponds with the target domain. To be fair to other algorithms that we compare to, we use the raw bag-of-words unigram/bigram features as their input and pre-process with tf-idf (Blitzer et al., 2006). Table 1 presents the statistics of the data set.

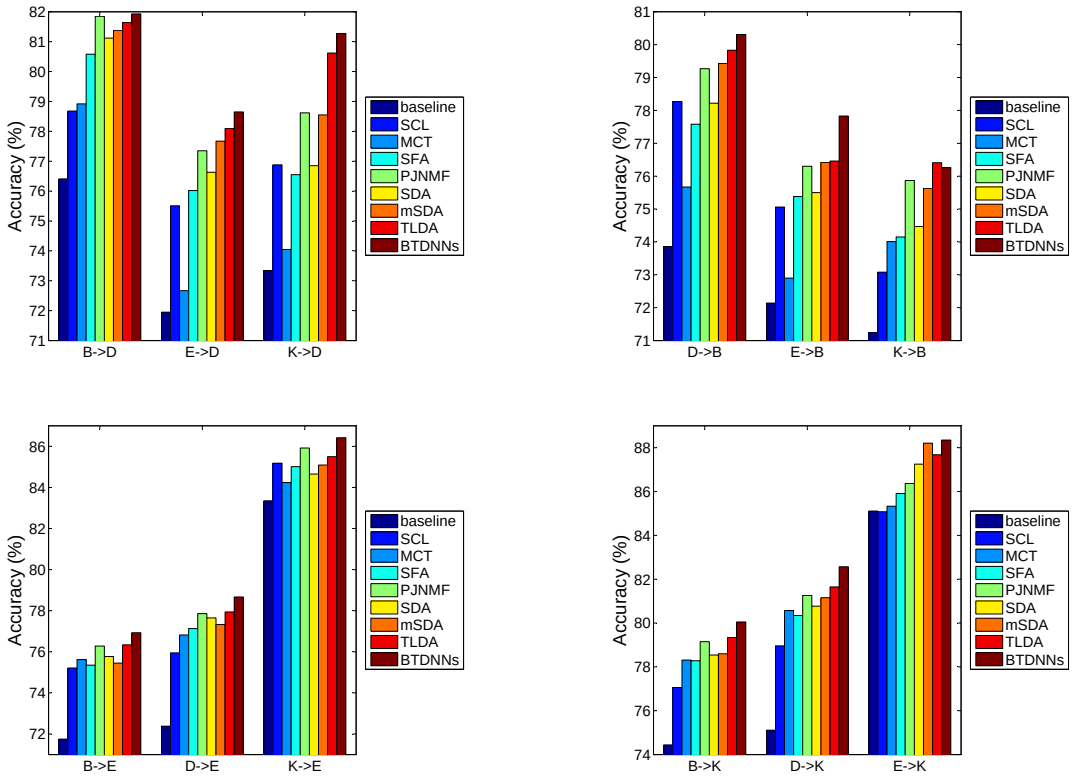

Figure 2: Average results for cross-domain sentiment classification on the Amazon product benchmark of 4 domains.

Domain	#Train	#Test	#Unlab.	% Neg.
Books	1600	400	4465	50%
DVDs	1600	400	5945	50%
Electronics	1600	400	5681	50%
Kitchen	1600	400	3586	50%

Table 1: Amazon review statistics. This table depicts the number of training, testing and unlabeled reviews for each domain, as well as the portion of negative training reviews of the data set.

4.2 Compared Methods

As a *baseline* method, we train a linear SVM (Fan et al., 2008) on the raw bag-of-words representation of the labeled *source* domain and test it on the *target* domain. In the original paper regarding the benchmark data set, Blitzer et al. (2006) adapted Structural Correspondence Learning (SCL) for sentiment analysis. Li and Zong (2008) proposed the Multi-label Consensus Training (MCT) approach which combined several base classifiers trained with SCL. Pan et al. (2010) first used a Spectral Feature Alignment (SFA) algorithm to align words from the source and target domains to help bridge the gap between them. Zhou et al. (2015a) proposed a method called PJNMF, which linked heterogeneous input features with pivots via joint non-negative matrix factorization.

Recently, some efforts have been initiated on learning robust feature representations with DNNs for cross-domain sentiment classification. Glorot et al. (2011) first employed stacked Denoising Auto-encoders (SDA) to extract meaningful representation for domain adaptation. Chen et al. (2012) proposed marginalized SDA (mSDA) that addressed the high computational cost and lack of scalability to high-dimensional features. Zhuang et al. (2015) proposed a state-of-the-art method called transfer learning with deep auto-encoders (TLDA).

For SCL, PJNMF, SDA, mSDA and TLDA, we use the source codes provided by the authors. For SFA and MCT, we re-implement them based on the original papers. The above methods serve as comparisons in our empirical evaluation. For fair comparison, all hyper-parameters are set by 5-fold cross validation on the training set from the source domain.[3] For our proposed BTDNNs, the number

[3]We keep the default value of some of the parameters in SCL and SFA, e.g., the number of stop-words removed and stemming parameters – as they were already tuned for this

of hidden neurons is set as 1000, the regularization parameter γ is tuned via 5-fold cross-validation.

For SDA, mSDA, TLDA and BTDNNs, we can construct the classifiers for the target domain in two ways. The first way is directly to use the stacking SVM on top of the output of the hidden layer. The second way is to apply the standard SVM to train a classifier for source domain in the embedding space. Then the classifiers is applied to predict sentiment labels for target domain data. For fair comparison with the shallow models, we choose the second way in this paper.

Figure 2 shows the accuracy of classification results for all methods and for all source-target domain pairs. We can check that all compared methods achieve the similar performance with the results reported in the original papers. From Figure 2, we can see that our proposed approach BTDNNs outperforms all other eight comparison methods in general. The `baseline` performs poorly on all the 12 tasks, while the other seven domain adaptation methods, SCL, MCT, SFA, PJNMF, SDA, mSDA and TLDA, consistently outperform the `baseline` method across all the 12 tasks, which demonstrates that the transferred knowledge from the source domain to the target domain is useful for sentiment classification. Nevertheless, the improvements achieved by these seven methods over the `baseline` are much smaller than the proposed approach BTDNNs.

Surprisingly, we note that the deep learning based methods (SDA, mSDA and TLDA) perform worse than our approach, the reason may be that SDA, mSDA and TLDA learn the unified domain-invariable feature representations by combining the source domain data and that of the target domain data together, which cannot well characterize the domain-specific features as well as the commonality of domains. On the contrary, our proposed BTDNNs ensures the feasibility of transferring between domains, and the distribution consistency between the transferred domain and the desirable domain is constrained with a linear data reconstruction manner.

We also conduct significance tests for our proposed approach BTDNNs and the state-of-the-art method (TLDA) using a McNemar paired test for labeling disagreements (Gillick and Cox, 1989). In general, the average result on the 12 source-target domain pairs indicates that the difference between BTDNNs and TLDA is mildly significant with $p < 0.08$. Furthermore, we also conduct the experiments on a much larger industrial-strength data set of 22 domains (Glorot et al., 2011). The preliminary results show that BTDNNs significantly outperforms TLDA ($p < 0.05$). Therefore, we will report our detailed results and discussions in our future work.

benchmark set by the authors.

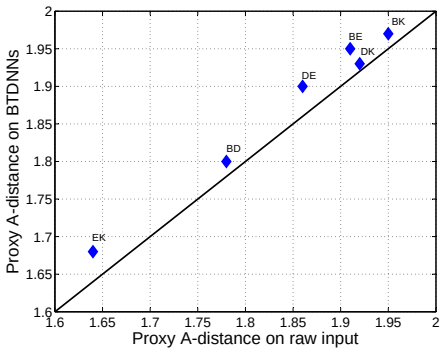

Figure 3: Proxy A-distance between domains of the Amazon benchmark for the 6 different pairs.

4.3 Domain Divergence

In this subsection, we look into how similar two domains are to each other. Ben-David et al. (2006) showed that the A-distance as a measure of how different between the two domains. They hypothesized that it should be difficult to discriminate between the source and target domains in order to have a good transfer between them. In practice, computing the exact A-distance is impossible and one has to compute a proxy. Similar to (Glorot et al., 2011), the proxy for the A-distance is then defined as $2(1 - 2\epsilon)$, where ϵ is the generalization error of a linear SVM classifier trained on the binary classification problem to distinguish inputs between the two domains.

Figure 3 presents the results for each pair of domains. Surprisingly, the distance is increased with the help of new feature representations, e.g., distinguishing between domains becomes easier with the BTDNNs features. We explain this effect through the fact that BTDNNs can ensure the feasibility of transferring between domains, and the distribution consistency between the transferred domain and the desirable domain is constrained with a linear data reconstruction manner, which can learn a generally better representations for the input data. This helps both tasks, distinguishing between domains and sentiment classification

(e.g., in the book domain BTDNNs might interpolate the feature "exciting" from "boring", both are not particularly relevant for sentiment classification but might help distinguish the review from the Electronic domain.).

5 Conclusions and Future Work

In this paper, we propose a novel Bi-Transferring Deep Neural Networks (BTDNNs) for cross-domain sentiment classification. The proposed BTDNNs attempts to transfer the source domain examples to the target domain, and also transfer the target domain examples to the source domain. The linear transformation of BTDNNs ensures the feasibility of transferring between domains, and the distribution consistency between the transferred domain and the desirable domain is constrained with a linear data reconstruction manner. Experimental results show that BTDNNs significantly outperforms the several baselines, and achieves an accuracy which is competitive with the state-of-the-art method for sentiment classification adaptation.

There are some ways in which this research could be continued. First, since deep learning may obtain better generalization on large-scale data sets (Bengio, 2009), a straightforward path of the future research is to apply the proposed BTDNNs for domain adaptation on a much larger industrial-strength data set of 22 domains (Glorot et al., 2011). Second, we will try to investigate the use of the proposed approach for other kinds of data set, such as 20 newsgroups and Reuters-21578 (Li et al., 2012; Zhuang et al., 2013).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61303180 and No. 61573163), the Fundamental Research Funds for the Central Universities (No. CCNU15ZD003 and No. CCNU16A02024), and also supported by a Discovery grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada and an NSERC CREATE award. We thank the anonymous reviewers for their insightful comments.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *NIPS*, pages 137–144.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, Universite De Montreal, and Montreal Quebec. 2007. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160.

Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*, pages 120–128.

John Blitzer, M. Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In *EMNLP*, pages 120–128.

Danushka Bollegala, David Weir, and John Carroll. 2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *ACL*, pages 132–141.

Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *NIPS*, pages 1–9.

Minmin Chen, John Blitzer, and Kilian Weinberger. 2011a. Co-training for domain adaptation. In *NIPS*, pages 1–9.

Minmin Chen, Kilian Weinberger, and Yixin Chen. 2011b. Automatic feature decomposition for single view co-training. In *ICML*, pages 953–960.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *ICML*, pages 767–774.

W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu. 2008. Translated learning: transfer learning across different feature spaces. In *NIPS*, pages 353–360.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*, pages 256–263.

Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(12):123–149.

R. Fan, Chang K., Hsieh C., Wang X., and Lin C. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

L. Gillick and S. Cox. 1989. Some statistical issues in the comparison of speech recoginition algorithms. In *ICASSP*, pages 532–535.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *ACL*, pages 123–131.

Zhanying He, Chun Chen, Bu, Jiajun, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document summarization based on data reconstruction. In *AAAI*, pages 620–626.

J. Huang, A. Smola, A. Gretton, K. Bordwardt, and B. Scholkopf. 2007. Correcting samples selection bias by unlabeled data. In *NIPS*, pages 601–608.

J. Jiang and C. Zhai. 2007. Instance weighting for domain adaptation in nlp. In *ACL*, pages 264–271.

Meina Kan, Shiguang Shan, and Xilin Chen. 2015. Bi-shifting auto-encoder for unsupervised domain adaptation. In *ICCV*, pages 3846–3854.

Abhishek Kumar, Avishek Saha, and Hal Daumé III. 2010. A co-regularization based semi-supervised domain adaptation. In *NIPS*, pages 478–486.

Shoushan Li and Chengqing Zong. 2008. Multi-domain adaption for sentiment classification: Using multiple classifier combining classification. In *NLPKE*.

Tao Li, Vikas Sindhwani, Chris H. Q. Ding, and Yi Zhang 0005. 2009. Knowledge transformation for cross-domain sentiment classification. In *SIGIR*, pages 716–717.

Lianghao Li, Xiaoming Jin, and Mingsheng Long. 2012. Topic correlation analysis for cross-domain text classification. In *AAAI*, pages 998–1004.

Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *IJCAI*, pages 2127–2133.

B. Liu. 2012. Sentiment analysis and opinion mining. *Morgan & Claypool Publishers*.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105.

T. Mansour, M. Mohri, and A. Rostamizadeh. 2009. Domain adatation with multiple sources. In *NIPS*, pages 264–271.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *WWW*, pages 751–760.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(12):1–135.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86.

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *NAACL*, pages 300–307.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*, pages 151–161.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, pages 1555–1565.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *EMNLP*, pages 327–335.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103.

Rui Xia, , and Chengqing Zong. 2010. A pos-based ensemble model for cross-domain sentiment classification. In *IJCNLP*, pages 614–622.

Rui Xia, Xuelei Hu, Jianfeng Lu, and Chengqing Zong. 2013a. Instance selection and instance weighting for cross-domain sentiment classification via pu learning. In *IJCAI*, pages 2276–2182.

Rui Xia, Chengqing Zong, Xuelei Hu, and Cambria Erik. 2013b. Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18.

Min Xiao and Yuhong Guo. 2013. Online active learning for cost-sensitive domain adaptation. In *CoNLL*, pages 1–9.

Min Xiao and Yuhong Guo. 2015. Learning hidden markov models with distributed state representations for domain adaptation. In *ACL*, pages 524–529.

Min Xiao, Feipeng Zhao, and Yuhong Guo. 2013. Learning latent word representations for domain adaptation using supervised word clustering. In *EMNLP*, pages 152–162.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *ACL*, pages 538–544.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *NAACL*, pages 672–682.

Shuangfei Zhai and Zhongfei (Mark) Zhang. 2016. Semi-supervised autoencoder for sentiment analysis. In *AAAI*, pages 1394–1400.

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. 2011. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*, pages 653–662.

Joey Tianyi Zhou, Sinno Jialin Pan, IvorW. Tsang, and Yan Yan. 2014. Hybrid heterogeneous transfer learning through deep learning. In *AAAI*, pages 2213–2219.

Guangyou Zhou, Tingting He, Wensheng Wu, and Xiaohua Hu. 2015a. Linking heterogeneous input features with pivots for domain adaptation. In *IJCAI*, pages 1419–1425.

Guangyou Zhou, Yin Zhou, Xiyue Guo, Xinhui Tu, and Tingting He. 2015b. Cross-domain sentiment classification via topical correspondence transfer. *Neurocomputing*, 159:298–305.

Fuzhen Zhuang, Ping Luo, Peifeng Yin, Qing He, and Zhongzhi Shi. 2013. Concept learning for cross-domain text classification: A general probabilistic framework. In *IJCAI*, pages 1960–1966.

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. 2015. Supervised representation learning: Transfer learning with deep autoencoders. In *IJCAI*, pages 4119–4125.