Leveraging Auxiliary Tasks for Document-Level Cross-Domain Sentiment Classification

Jianfei Yu
School of Information Systems
Singapore Management University
jfyu.2014@phdis.smu.edu.sg

Jing Jiang
School of Information Systems
Singapore Management University
jingjiang@smu.edu.sg

Abstract

In this paper, we study domain adaptation with a state-of-the-art hierarchical neural network for document-level sentiment classification. We first design a new auxiliary task based on sentiment scores of domain-independent words. We then propose two neural network architectures to respectively induce document embeddings and sentence embeddings that work well for different domains. When these document and sentence embeddings are used for sentiment classification, we find that with both pseudo and external sentiment lexicons, our proposed methods can perform similarly to or better than several highly competitive domain adaptation methods on a benchmark dataset of product reviews.

1 Introduction

Sentiment classification is a fundamental task in opinion mining (Pang et al., 2002; Hu and Liu, 2004; Choi and Cardie, 2008; Nakagawa et al., 2010). Recently, with the advances of deep learning techniques for many NLP applications, various kinds of neural network (NN)-based models have been proposed for this task (Socher et al., 2013; Lei et al., 2015; Yang et al., 2016).

As with any supervised learning method, the NN-based models also suffer from the *domain adaptation* problem, where training data and test data come from different domains. The reason for this is that sentiments are often expressed with domain-specific words and expressions. For example, in the **Book** domain, expressions like *an insider's look* and *a must read* are usually positive, but they may not be useful for the **Kitchen** domain. Similarly, words such as *sharp* and *clean*, which are positive in the **Kitchen** domain, can rarely be seen in the **Book** domain. Due to the high cost of obtaining labeled data, it would be very attractive if we can adapt a model trained on a *source domain* to a *target domain*.

A number of different models have been proposed for cross-domain sentiment classification, and the core idea of them is to learn a shared latent representation that is general across domains. Most of these studies can be categorized into two lines. The first line of work focuses on carefully designing some auxiliary prediction tasks to induce a robust cross-domain representation (Blitzer et al., 2007; Pan et al., 2010; Bollegala et al., 2015, 2016). With the trend of deep learning, another line of work centers on employing denoising autoencoders to learn hidden representations across domains in a purely unsupervised learning manner (Glorot et al., 2011; Chen et al., 2012; Zhou et al., 2016).

However, most of the two lines of research are based on traditional discrete feature representations, and the induced shared representations are not necessarily specific to sentiment classification. In our recent work, we designed two simple auxiliary tasks, which are closely related to the actual end task, for sentence-level cross-domain sentiment classification (Yu and Jiang, 2016). Furthermore, we proposed to jointly learn domain-independent sentence embeddings based on the two auxiliary tasks together with the classifier for the end task in a unified NN framework. Although our joint learning model has been shown to outperform previous domain adaptation methods in sentence-level sentiment classification, it is unclear how to extend this to document-level sentiment classification since the two auxiliary tasks will become much less useful for documents.

In this paper, we aim to propose a domain adaptation method for document-level sentiment clas-

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 654–663,
Taipei, Taiwan, November 27 – December 1, 2017

sification based on our earlier joint model (Yu and Jiang, 2016). Specifically, instead of predicting the occurrence of pivot words as in previous work (Blitzer et al., 2007; Yu and Jiang, 2016), we introduce a new auxiliary task based on sentiment scores of pivot words. Moreover, we propose two different architectures to incorporate the auxiliary task into a state-of-the-art hierarchical NN model for document-level sentiment classification, in which we respectively induce a shared document embedding for each document in both domains and a shared sentence embedding for each sentence in all documents. Evaluation on a widely used dataset about product reviews from four different domains shows that our methods can significantly outperform a number of baselines and are able to achieve comparable or even better results compared with a strong baseline proposed by us.

2 Related Work

Domain Adaptation: Domain adaptation has been extensively studied in recent years (Pan and Yang, 2010). In NLP, it has also attracted much attention, where most domain adaptation methods can be categorized into two groups: instance reweighting (Jiang and Zhai, 2007; Xia et al., 2014) and shared representation learning (Blitzer et al., 2006; Daumé III, 2007; Titov, 2011). In this work, we follow the latter line of work, and focus on inducing a domain-independent feature space based on a recently proposed NN architecture.

Neural Networks for Sentiment Classification: With the recent trend of deep learning, a large amount of NN models, including Convolutional Neural Network (Kim, 2014), Recursive Neural Network (Irsoy and Cardie, 2014) and Recurrent Neural Network (Tai et al., 2015), have been proposed for sentiment classification. Although these models have achieved highly competitive results on different benchmarks, most of them are targeted at sentence-level sentiment classification. Considering that the relations between sentences are important for predicting the sentiment polarity of any document, Tang et al. (2015) proposed a hieararchical NN model to encode the relations between sentences for document-level sentiment classification. Since it has been shown to significantly outperform standard non-hierarchical models on several benchmarks, we try to apply this model to domain adaptation settings in this work.

Cross-Domain Sentiment Classification: For sentiment classification, most existing domain adaptation methods focus on inducing shared representations across domains. One line of work tries to leverage the co-occurrences of domain-specific and domain-independent features to learn a general low-dimensional cross-domain representation (Blitzer et al., 2007; Pan et al., 2010; He et al., 2011; Bollegala et al., 2015; Bhatt et al., 2015). Another line of work is based on a purely unsupervised learning method, denoising autoencoders, where the hidden layers in multi-layer neural networks are believed to be robust against domain shift (Glorot et al., 2011; Chen et al., 2012; Zhou et al., 2016). However, all these methods are still based on traditional discrete representations, and the shared representations are learned separately from the final classifier and therefore not directly related to sentiment classification. More recently, we proposed a unified neural model to jointly learn the shared sentence embeddings and the final classifier together for sentence-level sentiment domain adaptation (Yu and Jiang, 2016). But the auxiliary task in this earlier work is only designed for sentences; it will be less useful for documents. Moreover, the neural model is based on CNNs, which fail to achieve satisfactory results in document-level sentiment classification. Hence in this work, we focus on proposing a new auxiliary task for documents, followed by incorporating it into a state-of-the-art hierarchical NN model for document-level sentiment domain adaptation.

3 Methodology

In this section we present our domain adaptation method for document-level sentiment classification.

3.1 Problem Definition and Notation

Our task is sentiment classification at the document level. We assume that each input d is a document containing n sentences, and the i^{th} sentence contains a sequence of m_i words. Let us use $w_{i,j} \in \mathcal{V}$ to denote the j^{th} word of the i^{th} sentence, where $\mathcal{V}$ is the vocabulary. Let $y \in \{+, -\}$ denote the sentiment label of input d, where $+$ and $-$ denote the positive sentiment and the negative sentiment, respectively.

We consider a cross-domain setting, in which we assume that we have a set of labeled training documents from a source domain, denoted by $\mathcal{D}^{\text{s}}$. In addition, we have a set of unlabeled documents

from a target domain, denoted by $\mathcal{D}^{t,u}$. Our goal is to train a good document-level sentiment classifier using $\mathcal{D}^{s}$ and $\mathcal{D}^{t,u}$ so that the classifier can generally work well in the target domain. To evaluate the trained classifier, we test its performance on a set of labeled documents from the target domain, denoted by $\mathcal{D}^{t,l}$.

3.2 Method Overview

The core of our domain adaptation method is to use a domain-independent auxiliary task to help induce a cross-domain hidden representation that is useful for both source and target domains. The idea of learning cross-domain hidden representations by leveraging auxiliary tasks for domain adaptation is not new (Blitzer et al., 2006, 2007; Yu and Jiang, 2016; Ding et al., 2017). These previous studies essentially follow the multi-task learning framework (Ando and Zhang, 2005). The rationale behind them is that if there are some auxiliary tasks related to the actual prediction task and the labels of the auxiliary tasks can be easily obtained for both source and target domains, the induced low-dimensional feature space is a good representation for domain adaptation.

Our work follows this line of research and aims to extend our recently proposed domain adaptation method for sentence-level sentiment classification (Yu and Jiang, 2016). Our method is based on an existing hierarchical neural network (HNN) model for document-level sentiment classification proposed by Tang et al. (2015), which encodes each sentence in the input document into a sentence embedding vector through a CNN, followed by combining all sentence embeddings into a document embedding vector with a gated RNN. Different from Tang et al. (2015), however, we use the sentence embeddings or document embeddings for predicting not only the actual sentiment labels but also the labels of a carefully designed auxiliary task. Since the auxiliary task is domain-independent, we expect the sentence embeddings and document embeddings learned by our method to work well in both domains.

3.3 A Hierarchical Neural Network for Document-level Sentiment Classification

We first describe our baseline method for document-level sentiment classification. This is a HNN model proposed by Tang et al. (2015) that has been shown to significantly outperform simpler, non-hierarchical models. We re-implement this model with some minor modifications.

Recall that an input document d is represented by a sequence of sentences, each containing a sequence of words, and $w_{i,j} \in \mathcal{V}$ is the j^{th} word of the i^{th} sentence in d. We use $\mathbf{x}_{i,j} \in \mathbb{R}^l$ to denote an l-dimensional dense embedding vector for word $w_{i,j}$, which is retrieved from a lookup table $\mathbf{X} \in \mathbb{R}^{l \times |\mathcal{V}|}$ for all words. We first apply a one-layer CNN (Kim, 2014) to obtain an embedding vector $\mathbf{z}_i \in \mathbb{R}^p$ for the i^{th} sentence: $\mathbf{z}_i = \mathit{CNN}_{\Theta_1}(\mathbf{x}_i)$, where Θ_1 denotes all the parameters in this CNN.

After obtaining the sentence embeddings for all the n sentences in d, we then apply an LSTM to sequentially combine all sentences together: $\mathbf{h}_i = \mathit{LSTM}_{\Theta_2}(\mathbf{h}_{i-1}, \mathbf{z}_i)$, where $\mathbf{h}_i \in \mathbb{R}^q$ is the i^{th} hidden state, and Θ_2 denotes all the parameters in the LSTM[1]. Note that Tang et al. (2015) used bi-directional gated RNN to chain the sentences into a document embedding, but we did not observe any significant gain over LSTM based on our preliminary experiments.

Finally, a softmax classifier is learned to map the document representation $\mathbf{h}_n$ to a label y:

$$p(y \mid \mathbf{h}_n) = \text{softmax}(\mathbf{W}\mathbf{h}_n + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{2 \times q}$ is a weight matrix and $\mathbf{b} \in \mathbb{R}^2$ is a bias vector.

In the following sections, we will present two NN architectures built on top of the baseline method that leverages an auxiliary task for domain adaptation of document-level sentiment classification. The first architecture uses a document-level auxiliary task to help induce a document-level hidden representation, while the second architecture uses a sentence-level auxiliary task to help induce a sentence-level hidden representation.

3.4 Document-level Shared Representation Learning for Domain Adaptation

Document-level Auxiliary Task

We first introduce an auxiliary task that is closely related to the original task of document-level sentiment classification. Our auxiliary task is inspired by our recent work for sentence-level cross-domain sentiment classification (Yu and Jiang, 2016). In this recently proposed method, we used two auxiliary tasks to induce shared sentence embeddings across domains. Considering

[1] To simplify the discussion, we will not give the details of CNN and LSTM here. Interested readers can refer to Kim (2014) and Hochreiter and Schmidhuber (1997).

that an input sentence containing a positive (or negative) domain-independent sentiment word is more likely to express an overall positive (or negative) sentiment, the two auxiliary tasks are about whether an input sentence contains at least one positive or one negative domain-independent sentiment word, respectively.

Although the two auxiliary tasks have been shown to benefit sentence-level cross-domain sentiment classification, they may not work well in document-level sentiment classification. The reason is the following. In sentence-level sentiment classification, since sentences are short, a sentence is more likely about only one aspect of the topic being discussed, and it tends to express a consistent sentiment polarity towards that aspect. However, in document-level sentiment classification, a document may contain mixed opinions towards different aspects of the topic, and the sentiment polarities towards different aspects may differ. Moreover, at document level, there may also be comparison and contrast among different topics, and the sentiment polarities towards them could be different. In summary, it is highly possible for a document to contain both positive and negative domain-independent sentiment words. In this case, the two auxiliary tasks would not be of much use because most documents would have the same labels for both these two tasks[2].

To address this limitation, we propose an alternative auxiliary task based on sentiment scores of the domain-independent sentiment words. The intuition is as follows. Assume that we have an external sentiment lexicon, where each word is assigned a general sentiment score. For an input document, if it contains more domain-independent words with high positive sentiment scores, the document is more likely to express an overall positive sentiment, regardless of the domain the document is from. More importantly, the remainder of the document without domain-independent words may also contain domain-specific positive words or expressions.

Take the following review as an example.

> One of the *best*! You will go **wrong** if you read this as an intro to deep learning. *Truly* an insider's look. A must read for everyone who *loves* neural networks.

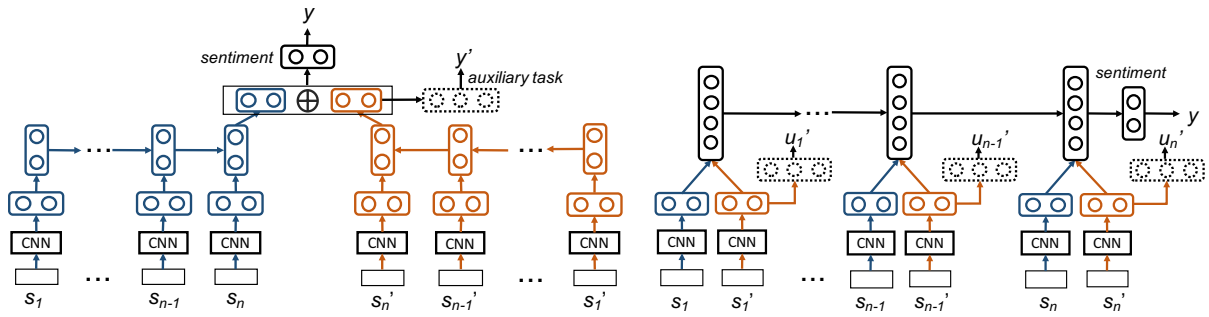
We can see that the document contains three words with high positive sentiment scores (shown in italic), and one word with a high negative sentiment score (shown in bold). But overall, its sentiment polarity is positive, which correlates with the sum of all the domain-independent words' sentiment scores. Then, if we hide all the domain-independent sentiment words and use the remaining domain-specific words to predict the overall sentiment score of the domain-independent sentiment words, it should be helpful for identifying some important domain-specific sentiment expressions such as *an insiders' look* and *a must read* in the example above.

Hence, we propose a new auxiliary task by predicting whether the sum of all the domain-independent sentiment words' sentiment scores is larger than, equal to or less than 0. It is worth noting that (1) given any sentiment lexicon, we can automatically derive the label of the auxiliary task[3], and (2) the auxiliary task is closely related to the main binary sentiment classification task.

Formally, let us assume that we have a sentiment lexicon, which can be either directly taken from an external resource or derived from the labeled source domain data. Details of how the sentiment lexicon is obtained will be given in Section 3.7.1. Following SCL (Blitzer et al., 2007), we choose the words which frequently occur in both domains and have a high (positive or negative) sentiment score as the domain-independent sentiment words, and refer to them as pivot words. For each input document d, we use a special token *UNK* to substitute these pivot words, which follows the practice in our earlier work (Yu and Jiang, 2016). To be consistent with the notation before, let us use d' to denote the new document with *UNK* tokens and $w'_{i,j}$ the j^{th} token in the i^{th} sentence in d'. Let $\mathbf{x}'_{i,j} \in \mathbb{R}^l$ denote the embedding vector of $w'_{i,j}$. $\mathbf{x}'_{i,j}$ is the same as $\mathbf{x}_{i,j}$ when $w'_{i,j}$ is not *UNK*. When $w'_{i,j}$ is *UNK*, $\mathbf{x}'_{i,j}$ is set to be a special embedding vector for *UNK*. We then introduce an auxiliary label y' for d', which indicates whether the sum of the sentiment scores of the pivot words in the original document d is larger than, equal to or less than 0. We further use $\mathcal{D}^{\text{a,d}}$ to denote documents with the document-level auxiliary labels derived from both $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t,u}}$.

[2]Based on our observation on a benchmark dataset collected by Blitzer et al. (2007), for almost all the 12 source/target pairs, over 90% of the reviews contained both positive and negative domain-independent sentiment words.

[3]For any sentiment lexicon, we can rescale its original sentiment scores to [-K, K], where K can be any positive integer, and -K and K respectively denote the most negative and the most positive sentiments. In this paper, we set K to 2.



(a) Document-level Shared Representation (b) Sentence-level Shared Representation

Figure 1: Overview of our proposed methods.

Document-level Shared Representation (DSR)

Now that we have defined the auxiliary task, we can present the NN architecture. Figure 1(a) gives the outline of this method, which essentially tries to directly learn an auxiliary hidden layer for each input document. For the i^{th} sentence in d', we first use a CNN to obtain its auxiliary embedding vector $\mathbf{z}'_i = \mathit{CNN}_{\Theta'_1}(\mathbf{x}'_i)$. These sentence embeddings are further combined together with an LSTM parameterized by Θ'_2, and the final hidden state $\mathbf{h}'_n$ is fed to predict the auxiliary label y':

$$p(y' \mid \mathbf{h}'_n) = \text{softmax}(\mathbf{W}'\mathbf{h}'_n + \mathbf{b}').$$

Besides, we also apply another CNN and LSTM to obtain the standard document representation $\mathbf{h}_n$, and concatenate it with the auxiliary hidden vector $\mathbf{h}'_n$ to predict the sentiment label y:

$$p(y \mid \mathbf{h}_n, \mathbf{h}'_n) = \text{softmax}\Big(\mathbf{W}(\mathbf{h}_n \oplus \mathbf{h}'_n) + \mathbf{b}\Big).$$

3.5 Sentence-level Shared Representation Learning for Domain Adaptation

Unlike the first architecture, our second proposal focuses on learning an auxiliary hidden layer for each sentence in a given document. As illustrated in Figure 1(b), instead of using an overall auxiliary label for the whole document, we will have an auxiliary label for each sentence in the document.

Sentence-level Auxiliary Task

Although the auxiliary task in Section 3.4 is designed for documents, it is also suitable for sentences since if a sentence contains more domain-independent words with high positive sentiment scores, the rest of the sentence excluding these words may still express a positive sentiment.

To facilitate the discussion, let us use s_i to denote the i^{th} sentence in the original document d and s'_i the i^{th} sentence in the modified document d'. We then introduce an auxiliary label u'_i for s'_i, which indicates whether the sum of the sentiment scores of the pivot words in s_i is larger than, equal to or less than 0. We further use $\mathbf{u}' \in \mathbb{R}^n$ to denote the auxiliary labels for all the n sentences in d'. Let $\mathcal{D}^{\text{a,s}}$ denote documents with the sentence-level auxiliary labels derived from both $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t,u}}$.

Sentence-level Shared Representation (SSR)

Based on the sentence-level auxiliary task, we use two CNNs to obtain sentence embeddings $\mathbf{z}_i$ and $\mathbf{z}'_i$, respectively for s_i and s'_i. Next, the auxiliary hidden layer $\mathbf{z}'_i$ will be used for predicting the auxiliary label u'_i:

$$p(u'_i \mid \mathbf{z}'_i) = \text{softmax}(\mathbf{W}'\mathbf{z}'_i + \mathbf{b}').$$

Besides, we also concatenate $\mathbf{z}_i$ and $\mathbf{z}'_i$ together as a combined sentence embedding for the i-th sentence. Then, all the n combined sentence embeddings are further combined together via an LSTM:

$$\mathbf{h}_i = \mathit{LSTM}_{\Theta_2}\Big(\mathbf{h}_{i-1}, (\mathbf{z}_i \oplus \mathbf{z}'_i)\Big).$$

Finally, we feed the last hidden representation $\mathbf{h}_n$ to predict the label of our main task:

$$p(y \mid \mathbf{h}_n) = \text{softmax}(\mathbf{W}\mathbf{h}_n + \mathbf{b}).$$

3.6 Parameter Learning

Since our two NN architectures consist of the main task and the auxiliary task, we jointly optimize them in a single loss function. For space limitation, here we only show the objective function for the first model, and the objective function for the latter one can be derived similarly. Using cross-entropy loss, we can learn Θ_1, Θ_2, Θ'_1, Θ'_2, $\mathbf{W}$, $\mathbf{b}$,

$\mathbf{W}'$ and $\mathbf{b}'$ by minimizing the following function:

$$\begin{aligned} & J(\Theta_1, \Theta_2, \Theta_1', \Theta_2', \mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}') \\ = & -\Big(\sum_{(d,y)\in\mathcal{D}^{\mathrm{s}}} \log p(y \mid d) \\ & + \sum_{(d',y')\in\mathcal{D}^{\mathrm{a,d}}} \log p(y' \mid d') \Big). \end{aligned}$$

3.7 Implementation Details

3.7.1 Sentiment Lexicons

We use two kinds of sentiment lexicons, which we refer to as WN and MI.

WN is extracted from a well-known sentiment lexicon called SentiWordNet (Baccianella et al., 2010). Since the original sentiment scores in SentiWordNet are probabilities for each word being positive or negative, we rescale them to $[-2, 2]$. Also, for the same words with different part-of-speech tags, we only keep the sentiment score with the highest absolute value.

To reduce the reliance on external resources, we also experiment with another method based on mutual information (MI) on the source labeled data $\mathcal{D}^{\mathrm{s}}$ to automatically derive a pseudo sentiment lexicon. Specifically, we first extract only adjectives, adverbs and verbs from the documents in $\mathcal{D}^{\mathrm{s}}$, and measure each remaining word's MI with the positive and the negative classes:

$$r(w_i, y) = \log \frac{\tilde{p}(w_i, y)}{\tilde{p}(w_i)\tilde{p}(y)},$$

where w_i denotes the i^{th} word in $\mathcal{V}$, $y \in \{+, -\}$ is a sentiment label, and $\tilde{p}(w_i, y)$ is the empirical probability of observing w_i and y together. Then, we only keep those words with positive MI, i.e., $r(w_i, y) > 0$, and obtain two lists $\mathcal{R}_+$ and $\mathcal{R}_-$. Moreover, for each word $w \in \mathcal{R}_+$, we use its MI score as its sentiment score, while for each word $w \in \mathcal{R}_-$, we reverse its MI score as its sentiment score. Finally, we merge the two word lists to form the pseudo sentiment lexicon, and rescale the sentiment scores into $[-2, 2]$.

3.7.2 Pivot Words Selection

Recall that pivot words should frequently occur in both domains and be sentiment sensitive. Hence, we first choose those words occurring at least 10 times in $\mathcal{D}^{\mathrm{s}}$ and at least 30 times in $\mathcal{D}^{\mathrm{t,u}}$ as pivot candidates[4], and remove negation and stop words. Then, we only keep those candidates with high sentiment scores ($[-2, -1]$ and $[1, 2]$ for WN or $[-2, -0.9]$ and $[0.9, 2]$ for MI) as the pivot words.

[4]The ratio between $|\mathcal{D}^{\mathrm{s}}|$ and $|\mathcal{D}^{\mathrm{t,u}}|$ is 1:3 in our dataset.

3.7.3 Training Details

In our domain adaptation setting, for the labeled data $\mathcal{D}^{\mathrm{s}}$ from the source domain, we have labels for both the main task and the auxiliary task, while for the unlabeled data $\mathcal{D}^{\mathrm{t,u}}$ from the target domain, we only have labels for the auxiliary task. Hence, we adopted an alternating training approach, where in each epoch we first optimize all the model parameters given $\mathcal{D}^{\mathrm{s}}$, and then switch to only optimizing the parameters corresponding to the auxiliary task (including Θ_1', Θ_2', $\mathbf{W}'$ and $\mathbf{b}'$) given $\mathcal{D}^{\mathrm{t,u}}$. During the training stage, we share the word embeddings of the actual task and our auxiliary task, and never update the word embedding of *UNK* by setting it as a zero vector.

4 Experiments

4.1 Experiment Settings

Datasets: To evaluate our proposed method, we conduct experiments on a benchmark dataset released by Blitzer et al. (2007). This dataset consists of Amazon product reviews from four different domains: Book, DVD, Electronics and Kitchen. Each domain has 1000 positive and 1000 negative reviews as well as 17547 unlabeled reviews on average. Since the number of unlabeled reviews in each domain is different, we randomly choose 6000 unlabeled reviews for each domain.

Following previous studies (Pan et al., 2010; Zhou et al., 2016), we consider 12 pairs of source-target domain pairs. For each pair, all the 2000 labeled reviews from the source domain are treated as training data. We randomly choose 200 positive and 200 negative reviews from the target domain as development data, and the remainder (i.e., 800 + 800 reviews) from the target domain as test data. Moreover, for domain adaptation methods, we also use the 6000 unlabeled reviews from the target domain during the training stage.

Methods for comparison:

- **Naive** is a non-domain-adaptive baseline based on traditional discrete representations.
- **SCL** is the Structural Correspondence Learning method, which uses all the non-pivot features to predict the occurrence of each pivot feature, and employs SVD on the learned weight vectors.

- **mDA** is one of the state-of-the-art methods, marginalized denoising auto-encoders (Chen et al., 2012), which learns a shared hidden representation by reconstructing pivot features.
- **HNN** is another non-domain-adaptive NN-based baseline as detailed in Section 3.3.
- **H-WN** simply combines **HNN** with our auxiliary task, which represents each label derived from our auxiliary task using WN as a three-dimensional one-hot vector and appends it to the document embedding $\mathbf{h}_n$ in **HNN**, followed by a softmax classifier.
- **H-SCL** is a naive combination of **SCL** with **HNN**, which appends the induced representation from **SCL** to the document embedding in **HNN**, followed by a softmax classifier.
- **H-mDA** is similar to **H-SCL** but uses the hidden representation from **mDA**, and this can be considered as a strong baseline.

Meanwhile, to show the effectiveness of the proposed auxiliary tasks, we also use the two auxiliary tasks in our earlier work as the auxiliary task of our two architectures for comparison, respectively denoted by **DSRE** and **SSRE** (Yu and Jiang, 2016).

Besides, we consider four variants of our proposed methods, where the auxiliary tasks in **DSR** and **SSR** are derived from the pseudo sentiment lexicon, while the auxiliary tasks in **DSRW** and **SSRW** are based on SentiWordNet.

- **DSRE**, **DSR** and **DSRW** are based on our first NN architecture to learn document-level shared representations, as introduced in 3.4.
- **SSRE**, **SSR** and **SSRW** are based on our second NN architecture to induce sentence-level shared representations, as introduced in 3.5.

Hyperparameters: For **Naive**, we train linear classifiers with LibLinear[5] by using unigrams and bigrams with a frequency of at least 5 as features. For **SCL** and **mDA**, we use mutual information to select pivot features, and the number of chosen pivots is tuned from $\{500, 1000, 1500, 2000\}$ on the development set. In **SCL**, we tune the number of induced features K in $\{25, 50, 100\}$, and also use normalization and rescaling. In **mDA**, we employ the dropout noise strategy used by Yang and Eisenstein (2014) without any parameter. In all the neural network models, we set the dimension of word embeddings l to 300, and initialize the lookup table $\mathbf{X}$ with *word2vec*[6]. We set the non-linear activation function in CNN as ReLU, and set the sizes of hidden layers in both CNN and LSTM as 150, i.e., $p = q = 150$. All the models were trained using AdaGrad with a learning rate of 0.05 and a minibatch size of 5. Also, the dropout rate α equals 0.5, and all the model parameters are regularized with a L2 regularization strength of 10^{-4}.

4.2 Results

In Table 1, we report the results of all the methods. It is easy to see that the performance of **Naive** is very limited. **SCL** and **mDA** can outperform the baseline model respectively by 2.7 and 3.7 percentage points on average, which shows that these two methods are useful for domain adaptation based on discrete representations. However, we can also see that the performance of these domain adaptation methods is much lower than the hierarchical neural network model (**HNN**) based on continuous representations. This demonstrates that **HNN** is more robust against domain shift. But comparing the performance of **HNN** in standard in-domain and our cross-domain settings, we find that the in-domain performance still outperforms the cross-domain performance by 6.2 percentage points on average. This indicates that it will be more challenging and useful to develop domain adaptation methods based on such a competitive baseline.

Moreover, we can easily see that the performance of simply appending three-dimensional one-hot vector from the auxiliary task to **HNN** (i.e., **H-WN**) is close to the performance of **HNN** in most cases. In addition, although **SCL** can outperform **Naive** with a large margin on almost all the data set pairs, the performance of **H-SCL** is not satisfactory, which can only improve the baseline by 0.5 percentage point on average. But for **H-mDA**, although the shared hidden representations are also derived from discrete representations, it can improve the performance of **HNN** on all data set pairs except one. This implies that the derived shared hidden representations by **mDA** can generalize better across domains, and are generally useful for domain adaptation. Furthermore, it is easy to observe that by simply incorporating the

[5] http://www.csie.ntu.edu.tw/cjlin/liblinear/

[6] https://code.google.com/p/word2vec/

Task	In-D	Compared Methods (Cross-Domain)									Proposed Methods (Cross-Domain)			
	HNN	Naive	SCL	mDA	HNN	H-WN	H-SCL	H-mDA	DSRE	SSRE	DSR	SSR	DSRW	SSRW
E2D		0.680	0.700	0.727	0.805	0.806	0.820	0.811	0.798	0.814	0.810	**0.823**†	0.821	0.816
B2D	0.845	0.773	0.771	0.806	0.814	0.832	0.813	0.829	0.823	0.832	0.832	0.822	**0.840**†	0.837†
K2D		0.698	0.721	0.741	0.791	0.796	0.799	0.796	0.788	0.803	0.798	**0.808**†	0.805†	0.801
E2B		0.693	0.704	0.728	0.786	0.790	0.780	0.789	0.789	0.781	0.790	0.792	0.790	**0.794**†
D2B	0.843	0.751	0.780	0.802	0.805	0.810	0.796	0.818	0.809	0.826	**0.835**†	0.822	0.825†	0.822
K2B		0.690	0.740	0.725	0.766	0.773	0.772	0.774	0.781	0.773	0.774	**0.784**†	0.781	0.776†
B2E		0.701	0.746	0.753	0.755	0.751	0.751	**0.786**	0.758	0.758	0.760	0.773	0.771	**0.786**
D2E	0.858	0.706	0.743	0.746	0.772	0.768	0.771	0.786	0.774	0.782	0.787	0.790†	**0.810**†	0.799†
K2E		0.799	0.818	0.830	0.836	0.837	0.847	0.839	0.837	0.843	0.830	0.835	**0.850**†	0.846
E2K		0.828	0.829	0.833	0.852	0.848	0.865	0.859	0.864	0.862	0.859	**0.874**†	0.867	0.858
B2K	0.883	0.724	0.763	0.754	0.780	0.788	0.785	**0.798**	0.784	0.791	0.785	0.783	0.796	0.794
D2K		0.716	0.758	0.742	0.778	0.774	0.786	0.773	0.793	0.809	**0.809**†	0.806	0.800†	0.803
AVG	0.857	0.729	0.756	0.766	0.795	0.798	0.799	0.805	0.800	0.806	0.806	0.809	**0.813**	0.811

Table 1: Comparison of classification accuracies of different methods. † indicates that DSR and DSRW (or SSR and SSRW) are significantly better than HNN, H-WN, H-SCL and H-mDA, DSRE (or SSRE) with $p < 0.05$ based on McNemar's paired significance test. **In-D** denotes the in-domain setting by splitting each target domain's labeled reviews into 1400/200/400 as training, development and test sets.

two auxiliary tasks in our earlier work into our two architectures, the performance of **DSRE** is not satisfactory, but **SSRE** can perform the best on average among all the compared systems. This demonstrates that the two auxiliary tasks are more suitable in sentence level, but become less useful in document level, which agrees with the intuition behind our auxiliary task.

Finally, we observe that (1) all of our proposed methods can significantly outperform the baseline **HNN** in almost all the data set pairs, and perform better than **H-WN** and **H-SCL** in most cases, which shows that the idea of learning a hidden representation using our proposed auxiliary tasks is generally effective; (2) even compared with **H-mDA**, **DSR** can achieve comparable results while **SSR**, **DSRW** and **SSRW** can still achieve significantly better performance in most cases. We conjecture that the gains of our methods may come from the sharing between two word embedding lookup tables and joint learning of our auxiliary task and the actual task; (3) in comparison with **DSRE** and **SSRE**, **DSR** and **SSR** can bring improvements in both sentence level and document level on average, which shows the effectiveness of the proposed auxiliary task; (4) among our proposed models, we find that the performance of **DSR** and **SSR** is not stable: sometimes they can achieve the best result, but sometimes they perform even worse than or similar to the baseline **HNN**. In contrast, **DSRW** and **SSRW** can always outperform **HNN**, and perform better than **DSR** and **SSR** on average. This is intuitive since the sentiment scores in **MI**, derived from source labeled data, are specific to the source domain, while the sentiment scores in SentiWordNet are general across domains. Besides, we can also see that the gap between **SSR** and **SSRW** is much smaller than the gap between **DSR** and **DSRW**. This suggests that our document-level shared representation learning method is more sensitive to the quality of sentiment lexicons, and with a high quality sentiment lexicon, it can perform best on average.

4.3 Case Study

Finally, to explore how our proposed models help to improve the performance of **HNN** in the test data set, we conduct a case study on **B2D** to get a deeper insight of our model **DSR**.

Specifically, we sample several samples from the test data set, i.e., *DVD*. As shown in Table 2, **HNN** only correctly predicts the sentiment of the first document but gives wrong predictions on another two documents, since ***worth watching*** only occurs once in the source *Book* domain. However, our model **DSR** can make correct predictions for all of them. The reason is as follows. In Table 2, we can observe that in the unlabeled data from the the *DVD* domain, ***worth watching*** often co-occur with some general positive sentiment words like *good*, *great*, *wonderful* and *fantastic*. Based on these unlabeled documents, **DSR** can implicitly learn that ***worth watching*** are highly correlated with the positive sentiment via our auxiliary task, and ultimately make correct predictions for the two test samples. This further indicates that compared with **HNN**, our models can identify more domain-specific sentiment words, and there-

B2D	Review	HNN	DSR
Test	*Definitely a great movie, rules ..... It is a movie definitely* ***worth watching*** *..... Strongly recommended along with black hawk down, a few good men, and courage under fire .*	1	1
	If your not already hooked on the story of these interns you have some catching up to do. Its not your typical medical drama and certainly ***worth watching****.*	0	1
	This film gets 4 stars because the child actors shine so much in it. The plot is not very intriguing, and ..... So no wonder you can not compete with him. Anyway, this film is certainly ***worth watching*** *as a family entertainment!*	0	1
Unlabel	*Very* **good** *film with a* **great** *cast. Reese and wahlberg are* **wonderful** *in their roles and play them to perfection, walhberg especially. Very much* ***worth watching*** */ owning.*	-	-
	A **great** *movie! What an all star cast! This movie is* ***worth watching*** *over and over again.*	-	-
	I found this dvd to be well produced and **engaging** *to go along with the powerful content.* **Fantastic***. Loads of deleted scenes that are very* ***worth watching****.*	-	-

Table 2: Examples drawn from **B2D** whose sentiment labels are incorrectly predicted by the baseline model (**HNN**) but correctly inferred by our model (**DSR**). The sentiment words specific to the target domain are in **bold** and *italic*, and the pivot sentiment words are only in **bold**. 0 and 1 denote the negative and positive sentiments respectively.

fore improve the performance.

5 Conclusions

We presented a domain adaptation method for document-level sentiment classification. We first devised a new auxiliary task based on sentiment scores of pivot words. Then, we proposed two neural network architectures to respectively induce shared document embeddings and sentence embeddings across domains. Experiment results show that with a pseudo sentiment lexicon, our methods can achieve comparable results compared with several highly competitive domain adaptation methods; and with an external sentiment lexicon, we can further boost the performance of both architectures to achieve the state-of-the-art result.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research* 6:1817–1853.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*.

Himanshu Sharad Bhatt, Deepali Semwal, and Shourya Roy. 2015. An iterative similarity based adaptation technique for cross-domain text classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Danushka Bollegala, Tingting Mu, and John Goulermas. 2016. Cross-domain sentiment classification using sentiment sensitive embeddings. *IEEE TKDE* 6(2):398–410.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In Proceedings of the Twenty-eight International Conference on Machine Learning*.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association of Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* pages 1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding CNNs for text: non-linear, non-consecutive convolutions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of of the 19th international conference on World Wide Web*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE TKDE* 22(10):1345–1359.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ivan Titov. 2011. Domain adaptation by constraining inter-domain variability of latent feature representation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Rui Xia, Jianfei Yu, Feng Xu, and Shumei Wang. 2014. Instance-based domain adaptation in NLP via in-target-domain logistic approximation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. 2016. Bi-transferring deep neural networks for domain adaptation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.