Incorporating Domain and Sentiment Supervision in Representation Learning for Domain Adaptation

Biao Liu, Minlie Huang
Tsinghua University
Beijing, China
liubiao2638@gmail.com
aihuang@tsinghua.edu.cn

Jiashen Sun, Xuan Zhu
Samsung R&D Institute
Beijing, China
jiashens.sun@samsung.com
xuan.zhu@samsung.com

Abstract

Domain adaptation aims at learning robust classifiers across domains using labeled data from a source domain. Representation learning methods, which project the original features to a new feature space, have been proved to be quite effective for this task. However, these unsupervised methods neglect the domain information of the input and are not specialized for the classification task. In this work, we address two key factors to guide the representation learning process for domain adaptation of sentiment classification — one is domain supervision, enforcing the learned representation to better predict the domain of an input, and the other is sentiment supervision which utilizes the source domain sentiment labels to learn sentiment-favorable representations. Experimental results show that these two factors significantly improve the proposed models as expected.

1 Introduction

Domain adaptation [Daumé III and Marcu, 2006] aims at learning a robust classifier with labeled data on a source domain, and at the same time attemps to enforce the classifier to work well on a target domain. The demand for quick transfer between domains are commonly seen in text classification tasks, particularly urgent in sentiment analysis.

The key challenge in domain adaptation is that the data distribution or feature representation may be quite different across domains. For instance, we may use quite different vocabularies in writing reviews, say *attractive, boring* in *books* reviews while *reliable, affordable* in *electronics* reviews. The simplest solution is to train a classifier for each domain, unfortunately, this requires heavy annotation on every single domain. Another solution may be to train a robust classifier with labeled samples from a source domain, and to enforce the classifier to work well on a target domain.

Several models had been proposed for domain adaptation such as instance reweighting [Huang *et al.*, 2006; Mansour *et al.*, 2009], joint feature representation learning [Blitzer *et al.*, 2006; Xue *et al.*, 2008; Glorot *et al.*, 2011b; Chen *et al.*, 2012] , and feature projection [Bollegala *et al.*, 2014]. Among these methods, Stacked Denoising Autoencoders (SDA) for domain adaptation [Glorot *et al.*, 2011b] , one of the joint feature representation learning methods, has been reported to be quite effective. The SDA model learns a neural network using reviews from all domains, and then trains a SVM sentiment classifier on the source domain using the learned representations. The SVM classifier is then applied to other domains and gains remarkable improvements over other methods.

Intuitively, if the learned representations have a good separation for domain-specific and sentiment-specific constituents, a sentiment classifier will be less susceptible to those domain-specific features and more favorable for domain adaptation. Furthermore, if the learned representations are specialized for sentiment classification during training, more sentiment information can be encoded and thus improve the performance of sentiment classification.

However, despite the success of SDA, the model simply mixed all samples from different domains during the training process. Such an oversimplified strategy does not encourage the model to discriminate domain-specific and sentiment-specific features. There is also no supervision for training sentiment-favorable representations which are more related to the classification goal.

This is what we do in this paper. We guide the representation learning process with two types of supervision — one is domain supervision, endowing the proposed model with the capability of predicting the domain of the input. We believe this factor will encourage the model to better discriminate domain-specific and sentiment-specific features. The other is sentiment supervision which utilizes the source domain sentiment labels. In this way, the model will be able to learn sentiment-favorable representations.

More specifically, on top of SDA, our learning objective consists of two terms: the reconstruction loss between the input and decoded output; the prediction loss for predicting the domain or/and sentiment label in the output layer. Therefore, the domain supervision and sentiment supervision can be easily introduced by adding the "domain loss" and "sentiment loss" respectively. Once the representation is constructed, we use the learned representations to train a classifier on a source domain, and predict the labels of the data on a target domain. Experimental results show that the two factors significantly improve the performance of domain adaptation.

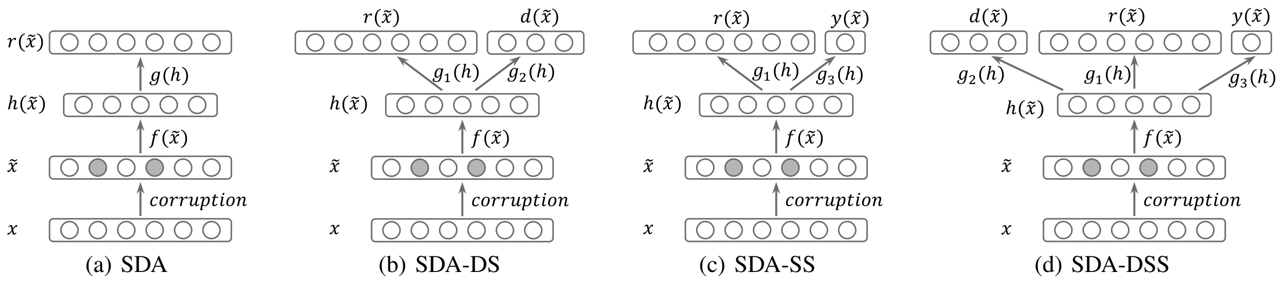

Figure 1: Architectures of SDA and the proposed extensions.

To summarize, our main contributions in this work are as follows:

- We reveal two key factors in representation learning for domain adaptation — domain supervision to better disentangle domain-specific and sentiment-specific features, and sentiment supervision which guides the model to learn sentiment-favorable representations.
- We proposed three modified SDAs for representation learning — SDA-DS, SDA-SS and SDA-DSS. Experimental results show that our models can learn better representations and significantly improve the performance of domain adaptation.

The rest of the paper is structured as follows: We survey related work in section 2, introduce our method in detail in section 3, and present the experiments in section 4. We discuss our work in section 5 and summarize the work in section 6.

2 Related Work

Domain adaptation, first introduced by Daumé III and Marcu [2006], aims to generalize classifiers across domains. Our work is mainly related to the following approaches for domain adaptation. These works fall into three types.

The first type is instance reweighting in which instances are assigned different weights so that these weighted instances have similar feature distributions. Huang et al. [2006] proposed *kernel mean matching* (KMM), a kernel-based method, to assign weights to instances. Mansour et al. [2009] investigated domain adaptation with multiple sources. They reweighted the patterns in different data sources by estimating the data distribution.

The second type is to transform the data representations of the source and target domains to a joint learned feature space. The idea is to share feature distributions across all domains so that a classifier trained on one domain is applicable for other domains. Blitzer et al. [2006] proposed Structural Corresponding Learning (SCL) to learn a low-rank representation of the data. They heuristically selected those features that is frequent in both source and target domains. Ben et al. [2007] analysed the effect of representation transforming. Pan et al.[2010] proposed spectral feature alignment (SFA) which used domain independent words to align domain-specific words into unified clusters and discover robust representations across different domains. Glorot et al. [2011b] proposed to use a deep learning approach (SDA), which can encode complex nonlinear feature mapping, to learn representations for domain adaptation and obtained remarkable improvements. A large amount of data across all domains are mixed together to train the SDA model. As an extension to SDA, mSDA is proposed to accelerate the training of SDA by hundreds of times and has comparable performance [Chen *et al.*, 2012]. A vital property of mSDA is that it has a closed form solution. A particularly interesting conjecture is that the success of SDA may derive from the fact that domain-specific and sentiment-specific features are separated in the learned representations. Along this line, we find that much improvements can be obtained by employing domain or sentiment supervision. By mixing the source and target data in different proportion as training set, Chopra et al.[2013] combined several SDAs together to improve performance of domain adaptation.

Bollegala et al. [2014] proposed a third type called feature projection. Different from feature representation learning which projects different domains to a shared space, they project the source domain feature space to the target domain using Partial Least Squares Regression (PLSR).

Our work is also related to neural network for sentiment classification. Glorot et al. [2011a] showed that rectifier units (hard ($max(0,t)$) or soft ($\log(1+e^t)$)) outperform other nonlinear functions for sentiment classification. We use the hard version rectifier since it can produce sparse coding (exactly zero).

3 Method

3.1 Background

Representation learning attempts to project the original features to a new feature space, using shallow or deep learning architectures (usually neural networks). Several forms of representation learning models, like Restricted Boltzmann Machines (RBMs) [Hinton *et al.*, 2006] and auto-encoders [Bengio *et al.*, 2007], have been developed for deep learning. We mainly discuss auto-encoder here. A simplest auto-encoder is composed of two steps: an encoder h maps the input x to a hidden representation $h(x)$, and a decoder g tries to reconstruct x from $h(x)$, i.e. $r(x) = g(h(x))$. The auto-encoder is trained to minimize the reconstruction loss between $r(x)$ and x. Square error and KL divergence are two commonly used loss functions. The hidden layer $h(x)$ is regarded as a new representation of x after training, and can be concatenated with x to form a new input to task-specific classifiers.

Denoising auto-encoder (Figure 1(a)), referred as denoiser, is a slightly modified version of auto-encoder. It introduces a corruption process before the hidden layer, and tries to reconstruct x from this corrupted input $\tilde{x}$, i.e. $r(\tilde{x}) = g(h(\tilde{x}))$. The corruption process could be a masking noise (each element of x has a probability of p to be set to 0). A denoiser is also trained to minimize the loss between x and its reconstruction form $r(\tilde{x})$. A typical architecture of denoiser is fully connected from the corrupted input $\tilde{x}$ to the hidden layer $h(\tilde{x})$ and from the hidden layer $h(\tilde{x})$ to the output layer $r(\tilde{x})$, and different nonlinear activation functions can be used. Several denoisers can be stacked together to form a stacked denoising auto-encoder (SDA) [Vincent *et al.*, 2008].

3.2 SDA for Domain Adaptation

Glorot et al. [2011b] apply SDA to domain adaptation for sentiment classification. The input x is a zero-one vector indicating the absence or presence of a word in a review (bag-of-words representation), thus the dimension of x is the same as the vocabulary size. They use hard rectifier $rec(t) = max\{0, t\}$ as nonlinear function for the hidden layer $h(\tilde{x})$ and sigmoid $\sigma(t) = \frac{1}{1+e^{-t}}$ for the output layer $r(\tilde{x})$. The objective function is an element-wise KL divergence between $r(x)$ and x:

$$\begin{aligned} \mathcal{O}_{SDA} &= \sum_x loss(x, r(\tilde{x})) \\ &= \sum_x \sum_i x_i \log \frac{x_i}{r(\tilde{x})_i} + (1 - x_i) \log \frac{1 - x_i}{1 - r(\tilde{x})_i} \end{aligned} \tag{1}$$

where x_i and $r(\tilde{x})_i$ are the i-th element of x and $r(\tilde{x})$. A L_1 norm of the hidden layer $||h(\tilde{x})||_1$ can also be added to the objective to encourage sparsity of the representation.

The training process is to minimize the objective function on a large amount of data across all domains. After the SDA is constructed, new representations are obtained by simply concatenating the input x and the hidden layer $h(\tilde{x})$. A standard sentiment classifier (SVM, logistic regression, etc.) is trained on the source domain using the new representations, and then applied to other target domains.

Glorot et al. [2011b] studied both single-layer and multi-layer SDAs ("layer" here refers to "denoiser"), and reported that a single-layer SDA was good enough for domain adaptation. We'll mainly discuss the single-layer SDA here.

3.3 SDA with Domain Supervision (SDA-DS)

As depicted in Figure 1(b), in addition to reconstructing the input in the output layer, we add a fully connected layer and use softmax as nonlinear function to predict the domain distribution $d(\tilde{x})$ of the input. The hidden layer $h(\tilde{x})$ is required not only to reconstruct the input, but also to predict the domain label. In this way, the proposed model is encouraged to extract domain-specific constituents better in the learned representation. This is what we expected, and several papers have reported that domain adaptation may benefit from such a property [Glorot *et al.*, 2011b; Chen *et al.*, 2012].

The reference domain distribution $\hat{d}(x)$ is an indicating vector with 1 corresponding to the domain that x comes from and 0 elsewhere. Thus, the domain supervision can be formulated as the KL divergence between $\hat{d}(x)$ and $d(\tilde{x})$,

$$loss(\hat{d}(x), d(\tilde{x})) = KL(\hat{d}(x)||d(\tilde{x})) = \sum_i \hat{d}(x)_i \log \frac{\hat{d}(x)_i}{d(\tilde{x})_i}$$

where $\hat{d}(x)_i$ and $d(\tilde{x})_i$ are the i-th element of $\hat{d}(x)$ and $d(\tilde{x})$, respectively. By combining the reconstruction error and domain supervision, we obtain the objective of SDA-DS as follows:

$$\mathcal{O}_{SDA-DS} = \sum_x \lambda \cdot loss(x, r(\tilde{x})) + loss(\hat{d}(x), d(\tilde{x})) \tag{2}$$

where $\lambda \in (0, 1)$ is used to control the balance of the two terms. Similar to SDA, a L_1 norm of the hidden layer $||h(\tilde{x})||_1$ can be added to the objective function.

The training process is to minimize the objective function across all domains. For domain adaptation, a standard sentiment classifier is trained using the new representations (concatenate x and $h(\tilde{x})$) on the source domain and then applied to other target domains.

3.4 SDA with Sentiment Supervision (SDA-SS)

Since our final task is to discover new features for sentiment classification, a simple idea is to guide the representation learning process with sentiment supervision so that the learned representations are sentiment-specialized and can improve performance of sentiment classification.

Figure 1(c) shows how we incorporate sentiment supervision into SDA. Using a fully connected layer and sigmoid nonlinear function, we output $y(\tilde{x})$ indicating the probability of how likely the sentiment of x is positive. Since not all samples have sentiment labels, we construct the sentiment reference $\hat{y}(x)$ as follows: for those labeled samples in the source domain, the reference $\hat{y}(x)$ is just its sentiment label (1 for positive, 0 for negative), for those unlabeled or target domain samples, we just set $\hat{y}(x) = 0.5$ as a reference distribution [1]. Thus, the sentiment supervision can be formulated as:

$$\begin{aligned} loss(\hat{y}(x), y(\tilde{x})) &= KL(\hat{y}(x)||y(\tilde{x})) \\ &= \hat{y}(x) \log \frac{\hat{y}(x)}{y(\tilde{x})} + (1 - \hat{y}(x)) \log \frac{1 - \hat{y}(x)}{1 - y(\tilde{x})} \end{aligned}$$

The objective function of SDA-SS is a combination of the reconstruction error $loss(x, r(\tilde{x}))$ and sentiment supervision $loss(\hat{y}(x), y(\tilde{x}))$:

$$\mathcal{O}_{SDA-SS} = \sum_x \lambda \cdot loss(x, r(\tilde{x})) + loss(\hat{y}(x), y(\tilde{x})) \tag{3}$$

The training process is the same as SDA-DS.

[1]This is equivalent to the maximum entropy principle.

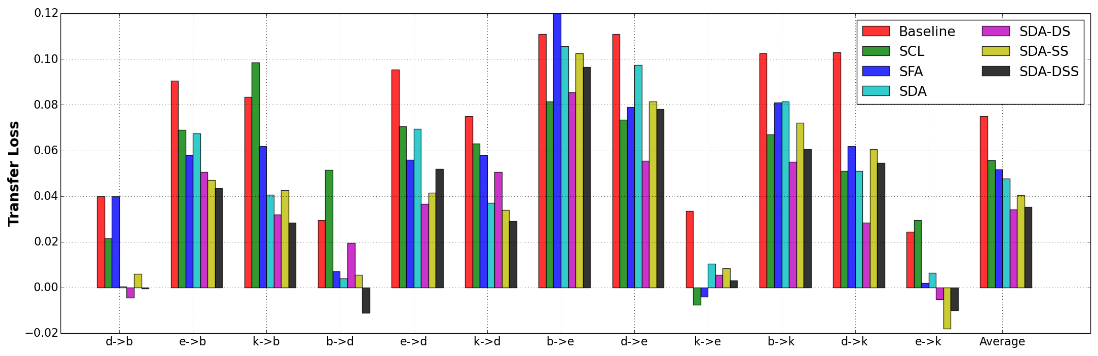

Figure 2: Transfer losses of the 12 domain pairs on 4 domains: *books (b)*, *dvds (d)*, *electronics (e)* and *kitchen (k)*. The four SDAs outperform baseline, and the three SDAs with extra supervisions (SDA-DS, SDA-SS, SDA-DSS) perform much better compared with SDA.

3.5 SDA with both Domain and Sentiment Supervision (SDA-DSS)

Since we have introduced domain supervision and sentiment supervision in previous sections, a natural idea is to combine them together, enforcing the learned representation domain-distinguishable and sentiment-favorable.This simple combination of SDA-DS and SDA-SS is depicted in Figure 1(d), and we name it as SDA-DSS.

The objective function of SDA-DSS is thus a combination of the reconstruction error $loss(x, r(\tilde{x}))$, domain supervision $loss(\hat{d}(x), d(\tilde{x}))$ and sentiment supervision $loss(\hat{y}(x), y(\tilde{x}))$:

$$\mathcal{O}_{SDA-DSS} = \lambda \cdot loss(x, r(\tilde{x})) + \eta \cdot loss(\hat{d}(x), d(\tilde{x})) + loss(\hat{y}(x), y(\tilde{x})) \quad (4)$$

where λ and η control the weights of the three terms in the objective function. The training process is the same as SDA.

4 Experiments

4.1 Data Preparation

We experiment on a benchmark dataset — products reviews from Amazon, first collected by Blitzer et al. [2007][2]. This dataset contains about 340,000 reviews of 22 different products. Each review is either unlabeled or labeled as positive or negative. We conduct domain adaptation experiments on reviews of four products (12 pairs for domain adaptation) — *book (b)*, *dvds (d)*, *electronics (e)* and *kitchen (k)*. Table 1 shows the number of positive and negative reviews, and unlabeled reviews from these four domains.

domain	positive	negative	unlabeled
books	1,000	1,000	4,465
dvds	1,000	1,000	3,586
electronics	1,000	1,000	5,681
kitchen	1,000	1,000	5,945

Table 1: The statistics of the corpus.

4.2 Experiment Settings

Parameter Settings

The preprocessing stage of the reviews is the same as [Blitzer *et al.*, 2007]: we use bag-of-words schema, namely a zero-one vector indicating the absence or presence of a word, to represent a review. Only the top 5,000 frequent unigrams and bigrams are kept here. The hidden layer size is 5,000 for all the four networks.

We use the standard backward propagation and batch gradient descent method to train all the four neural networks. There are some hyper-parameters to be tuned and we attempt the following settings: masking noise probability $p \in \{0.3, 0.5, 0.8\}$, L_1 regularization penalty $l_1 \in \{0, 10^{-4}, 10^{-3}\}$, batch size $bs \in \{10, 20\}$, epoch number $ep \in \{1, 2, 3, 4, 5\}$, step length $sl \in \{0.1, 0.2, 0.01, 0.02\}$, weight control coefficient in SDA-DS, SDA-SS, SDA-DSS $\lambda \in \{0.01, 0.1\}$ and $\eta = \{1, 0.5\}$. All these parameters are selected by minimizing the cross validation error (5-fold) on the source domain training data.

After the networks are constructed, the new representations (concatenation of x and $h(\tilde{x})$) are used to train a logistic regression classifier[3] on the source domain and this classifier is then applied to the target domains.

Compared Methods

The baseline is a logistic regression classifier trained on the original raw bag-of-words features using the labeled data from the source domain.

The other methods to be compared are including: Structural Correspondence Learning (SCL) which was used in domain adaptation of sentiment classification by Blitzer et al. [2007]; Spectral Feature Alignment (SFA) proposed by Pan et al. [2010]; SDA proposed by Glorot et al. [2011b].

[2]http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

[3]Note that we did not choose SVM here as previous work just because SVM training is much slower than LR.

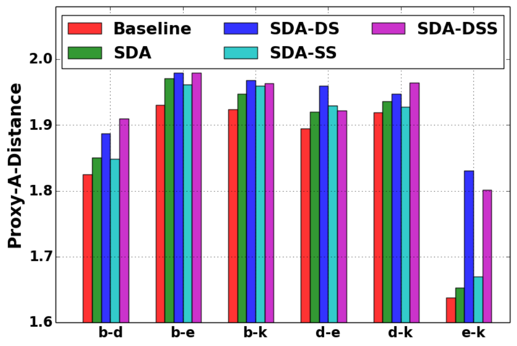

Figure 3: PAD Measures

Evaluation Metrics
As described in [Glorot *et al.*, 2011b], *transfer error* $e(S, T)$ is defined as the test error on the target domain T of a model which is trained on the source domain S. Therefore, $e(T, T)$ which is called *in-domain error* is the cross validation error on domain T, and $e_b(T, T)$ refers to *in-domain error* of the baseline (the logistic regression model trained with bag-of-words feature). We compare different models (including the baseline) using *transfer loss* which is defined as $t(S, T) = e(S, T) - e_b(T, T)$.

4.3 System Comparison

The domain adaptation results of the 12 domain pairs and the mean value are shown in Figure 2. As can be seen, the three proposed models significantly outperform other methods. The mean transfer losses are: Baseline (0.0750), SCL (0.0558), SFA (0.0518), SDA (0.0476), SDA-DS (0.0341) SDA-SS (0.0403) and SDA-DSS (0.0353). SDA-DS performs better than other baseline methods in 10 of the 12 tasks, SDA-SS is better in 9 tasks and SDA-DSS is better in 11 tasks.

Our hypothesis is justified by the results — supervising the representation learning process with domain and sentiment information helps to learn better representations and thus significantly improve the performance of domain adaptation.

The results also show that domain supervision is more effective than sentiment supervision, the reason may be that the domain label of all samples are available while only those source domain labeled data have sentiment labels (note that we have much more unlabeled data than labeled data, see Table 1).

4.4 Proxy-A-Distance

Proxy-A-Distance (PAD) [Ben-David *et al.*, 2007] measures the difference between two domains. It is defined as $2(1-2\epsilon)$, where ϵ is the cross validation error when distinguishing the two domains. Ben-David et al. [2007] argued that low PAD measure is favorable for domain adaptation (low PAD $\Rightarrow$ high $\epsilon \Rightarrow$ the two domains are similar and difficult to distinguish $\Rightarrow$ favorable for domain adaptation). The PAD measures of SDA and the proposed models are shown in Figure 3 (5-fold cross validation is used). For each domain pair $d_1 - d_2$, we report the mean PAD of the two representations — transferring from d_1 to d_2 and reverse.

Source	*b*	*d*	e	*k*	*mean*
Baseline	–	–	–	–	836
SDA	609	748	703	932	748.00
SDA-DS	428	557	416	521	480.50
SDA-SS	782	911	627	721	760.25
SDA-DSS	500	639	508	454	525.25

Table 2: Number of Overlapping Features in the Learned Representations (smaller values mean less overlapping, or better disentangled features). *b (books)*, *d (dvds)*, *e (electronics)*, *k (kitchen)* refers to which domain is used as source domain.

The results are shown in Figure 3. On one hand, results show that low PAD measures correlated with better domain adaptation performance. As can be seen, domain pair $b - d$ and $e - k$ have the lowest PAD measures among the 6 domain pairs and we can also see from Figure 2 that the transfer losses of $b \to d$, $d \to b$, $e \to k$ and $k \to e$ are particularly low in the 12 domain transfer pairs. This observation is consistent with the observation of Ben-David et al. [2007].

On the other hand, the proposed models produce not only high PAD values but also better domain adaption performance (SDA-DS and SDA-DSS against SDA and baselines). A reasonable explanation is that those domain-specific and sentiment-specific constituents are better disentangled in the learned representations, and thus improve both domain prediction and sentiment classification.

The PAD measure of SDA is slightly higher than the baseline while SDA-DS and SDA-DSS are much higher especially in the *e-k* setting. This is mainly because we introduce domain supervision in SDA-DS and SDA-DSS, and thus their learned representations are well trained for domain classification. This is exactly what we expected at the beginning. We enforce SDA-DS and SDA-DSS to predict the domain of the input so that those domain-specific and sentiment-specific features are better disentangled, and thus can improve the performance of domain adaptation.

4.5 Feature Overlapping Test

The motivation that we incorporate domain supervision into SDA is to guide the representation learning process to better disentangle those domain-specific and sentiment-specific features. In this section, we want to directly analyse the overlap of active features for domain prediction and sentiment prediction.

A feature is called active if it can provide "sufficient" information for classification. Given a classification task (either domain prediction or sentiment prediction), in order to identify these active features, we can train a logistic regression model with L_1 sparse regularization (most of the coefficients are driven to 0). After training, each feature f_i is associated with several coefficients $w_i^{(1)}, w_i^{(2)}, ..., w_i^{(n)}$ (n is 1 in binary classification and is the number of classes in multi-class classification). If one of these n coefficients is nonzero, we regard f_i as an active feature.

Source	*b*	*d*	*e*	*k*	*mean*
Baseline	0.150	0.156	0.168	0.174	0.162
SDA	0.139	0.143	0.153	0.163	0.150
SDA-DS	**0.133**	**0.132**	**0.144**	0.160	**0.142**
SDA-SS	**0.130**	**0.134**	**0.142**	**0.154**	**0.140**
SDA-DSS	0.138	**0.134**	0.148	**0.151**	**0.143**

Table 3: Sentiment Cross Validation Error (bold text means statistically significant).

We apply the original input (bag-of-words representation) and the learned representations (only the hidden layer $h(\tilde{x})$) to two tasks: sentiment classification (binary) and domain classification (4-class). If a feature f_i in the representation is active in both tasks, it is called an overlapping feature. Obviously, if those domain-specific and sentiment-specific features are disentangled in the learned representations, the number of overlapping features should be small.

For each method in each source domain, there is an optimal parameter setting which is obtained by minimizing the cross validation error on source domain. The results are shown in Table 2. The number of overlapping features in the learned representations is much smaller than the baseline (original bag-of-words representation). SDA-DS and SDA-DSS also have a much smaller number of overlapping features compared with the original SDA. Without domain supervision, SDA-SS does not reduce the number of overlapping features.

This overlapping test strongly supports our claim that incorporating domain supervision in representation learning helps to learn better disentangled representations for domain adaptation. The domain supervision significantly reduces the number of overlapping features.

4.6 Cross-validation of Sentiment Classification

We propose to learn sentiment-favorable representations by incorporating sentiment supervision. These learned representations are expected to improve sentiment classification performance. We verify this claim in this section.

Using the original input or the learned representations (only the hidden layer), we test the sentiment cross validation error (5-fold) of the logistic regression model on the target domains.

Specifically, given a method m and a source domain d, we report the mean sentiment cross validation error on the other three domains (except d). For example, the cross validation error of SDA-DS on b, is averaged from the cross validation errors of that model on d, e, and k. Thus, m is associated with four errors, one for each domain.

The results are shown in Table 3. As can be seen, sentiment supervision (SDA-SS, SDA-DSS) reduces the cross validation error on the target domains even though it only utilizes labeled data from source domain. The results demonstrate that incorporating sentiment supervision is able to learn sentiment-favorable representations. The cross validation error of SDA-DS is also reduced. This means that disentangling domain-specific and sentiment-specific features can also improve sentiment classification performance.

Source	*b*	*d*	*e*	*k*	*mean*
Baseline	–	–	–	–	1629
SDA	1,963	1,868	1,905	2,400	2,034.00
SDA-DS	1,303	1,472	1,285	1,550	1,402.50
SDA-SS	2,103	2,245	1,708	1,910	1,991.50
SDA-DSS	1,455	1,593	1,430	1,368	1,461.50

Table 4: Domain Active Features.

5 Discussion

We list the number of domain active features in table 4. The results show that with domain supervision, the number of domain active features in our models (SDA-DS, SDA-DSS) is largely reduced, demonstrating that those domain-specific constituents are more centralized with less feature overlapping. We also note that the number of active features in SDA and SDA-SS is increased. The reason may be that when we project the original discrete feature space to a continuous feature space without supervision, domain information will disperse in more dimensions. Unlike SDA which disentangles domain and sentiment features implicitly, our proposed models give more specific and clear signals to the presentation learning process.

The training of SDA and SDA-DS is irrelevant to which domain is the source and which are the targets. In other words, only one SDA or SDA-DS need to be trained to transfer across all domains. But SDA-SS and SDA-DSS do not have such a property since different source domains have different annotated data. However, the source domain is always given in reality, so there is no much difference in the applicability of these three methods.

Glorot et al. [2011b] reported that stacking several denoisers together could learn even better representations and improve domain adaptation performance. But training such a network is extremely time-consuming and further improvement is relatively small, so we just test the one-layer architecture in our work.

6 Conclusion

In this paper, we investigate two key factors (domain/sentiment supervision) for domain adaptation and propose three models (SDA-DS, SDA-SS, SDA-DSS) to learn better representations for this task. We find that domain supervision helps to disentangle domain-specific and sentiment-specific constituents in the learned representations, and that sentiment supervision helps to learn sentiment-favorable representations. Experimental results show that the proposed models obtain significant improvements over some competitive baselines such as SCL, SFA, SDA, etc.

Though we have supplied clear signals to disentangle domain-specific and sentiment-specific constituents in the learned representation, as future work, we are planning to design neural network structures that are able to explicitly represent the two kinds of information.

7 Acknowledgments

This work was partly supported by the National Basic Research Program (973 Program) under grant No.

2012CB316301/2013CB329403, the National Science Foundation of China under grant No. 61272227/61332007, and the Beijing Higher Education Young Elite Teacher Project. The work was also supported by Tsinghua University – Beijing Samsung Telecom R&D Center Joint Laboratory for Intelligent Media Computing.

References

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[Bengio *et al.*, 2007] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[Blitzer *et al.*, 2006] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

[Blitzer *et al.*, 2007] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447. Citeseer, 2007.

[Bollegala *et al.*, 2014] Danushka Bollegala, David Weir, and John Carroll. Learning to predict distributions of words across domains. pages 613–623, 2014.

[Chen *et al.*, 2012] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[Chopra *et al.*, 2013] Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, volume 2, page 5, 2013.

[Daumé III and Marcu, 2006] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.(JAIR)*, 26:101–126, 2006.

[Glorot *et al.*, 2011a] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, volume 15, pages 315–323, 2011.

[Glorot *et al.*, 2011b] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

[Hinton *et al.*, 2006] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[Huang *et al.*, 2006] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

[Pan *et al.*, 2010] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[Xue *et al.*, 2008] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634. ACM, 2008.