Bi-Weighting Domain Adaptation for Cross-Language Text Classification

Chang Wan, Rong Pan* and Jiefei Li
School of Information Science and Technology
Sun Yat-sen University, Guangzhou, China
{wanchang@mail2, panr@mail, lijiefei@mail2}.sysu.edu.cn

Abstract

Text classification is widely used in many real-world applications. To obtain satisfied classification performance, most traditional data mining methods require lots of labeled data, which can be costly in terms of both time and human efforts. In reality, there are plenty of such resources in English since it has the largest population in the Internet world, which is not true in many other languages. In this paper, we present a novel transfer learning approach to tackle the cross-language text classification problems. We first align the feature spaces in both domains utilizing some on-line translation service, which makes the two feature spaces under the same coordinate. Although the feature sets in both domains are the same, the distributions of the instances in both domains are different, which violates the i.i.d. assumption in most traditional machine learning methods. For this issue, we propose an iterative feature and instance weighting (Bi-Weighting) method for domain adaptation. We empirically evaluate the effectiveness and efficiency of our approach. The experimental results show that our approach outperforms some baselines including four transfer learning algorithms.

1 Introduction

Over the past few decades, a considerable number of studies have been made in data mining and machine learning. However, many machine learning methods work well only under the case where the training and test data are drawn from the same feature space and the same distribution. A few attempts have been made for leaning under different distributions. When the distribution changes, it could be costly in terms of time and human efforts to re-collect the needed training data and rebuild the models. As a result, transfer learning has been proposed to address this problem.

Recently, transfer learning was found to be useful in many real-world applications. One important application is web mining, where the goal is to classify a given web document into several predefined categories. If the training documents and the test documents are in the same distribution, we can use the traditional machine learning methods to solve it. However, in the real world, this assumption cannot always be satisfied. In reality, there are plenty of such resources in English since it has the largest population in the Internet world, which is not true in many other languages. From the statistics in ODP[1], it has sorted English web pages up to 1,429,760. However the number of web pages in other languages are much smaller (e.g. 13,293 classified Chinese web pages and 189,323 sorted Japanese web pages). It is well known that, classification requires a large number of labeled training data. Generally, the more labeled training data we get, the better the classification accuracy is. Fortunately, there exist many Web pages in English with class labels. Thus we should consider how to make use of the information got from the Web pages in English to classify the Web pages in other languages. This problem is called cross-language text classification, which we address in this paper. In order to utilize Web pages in English to classify Web pages in other languages, we can use a translation tool to translate the target data sets into English language. In this way, a classifier trained on English Web pages can be applied. Unfortunately, this directly application of this method may lead to some serious problems due to the following reasons:

- First, due to the difference in language and culture, there exists a word drift. This means that a word which frequently appears in English Web pages may hardly appear in other languages Web pages. In machine learning we call this different distribution in feature between training data and test data. This problem needs to be overcome in our study.
- Second, due to the errors introduced in the translation process, there may be different kinds of errors in the translated text. This noise problem produced in translation procedure should also be addressed effectively for the purpose of improving the accuracy of classification.
- Due to the word drift, some features functional for classification on source domain may be useless for classifying data in target domain. Thus, we have to make the distributions of source domain and target domain as sim-

*Corresponding author

[1]According to the statistics reported from ODP on August 1, 2010

ilar as possible to let the features selected from source domain also work for the target domain.

To solve the above problems, we develop a novel algorithm for cross-language text classification. We introduce the weights of features and training instances for domain adaptation. An objective function on these two type weights is built to take the distance between source domain and target domain and the inner distance in domain into consideration. We try to optimize the objective function through Bi-Weighting. The detail of our algorithm will be described later.

The rest of our paper is organized as follows. In section 2, we discuss the related work. In section 3, we give the mathematical formulation of the problem we focus on in the paper. Section 4 describes our proposed algorithm in details, including feature weighting and instance weighting. The experimental results are presented and analyzed in section 5. Finally, we conclude this paper with future works in section 6.

2 RELATED WORK

2.1 Cross-Language Text Classification

Many attempts have been made on addressing cross-language classification problems. [Bel *et al.*, 2003] study English-Spanish cross-language classification problem. [Rigutini *et al.*, 2005] proposes an EM-based learning method to address English-Italian cross-language classification and acquires good empirical results. It applies feature selection before each iteration. [Ling *et al.*, 2008] proposes an approach based on information bottleneck theory [Tishby *et al.*, 2000]. The method allows all the information to be put through a "bottleneck". Then, the approach maintains most of the common information and disregards the irrelevant information.

2.2 Transfer Learning

[Pan and Yang, 2009] summarize the relationship between traditional machine learning and transfer learning and give us the categorization under three sub-settings, inductive transfer learning, transductive transfer learning and unsupervised transfer learning. Instance transfer tries to re-weight the source domain data or reduce the effect of the "bad" ones encouraging the "good" source data to contribute more for the classification. Based on this definition, [Dai *et al.*, 2007] proposed a booting algorithm, TrAdaBoost, which is an extension of the AdaBoost algorithm. TrAdaBoost assumes that the source and target domain data share the same set of features and labels, but the distributions of the data in the two domains are different. In addition, [Jiang and Zhai, 2007] propose a heuristic method to remove "misleading" training examples from the source data. Feature transfer focuses on finding "good" feature space to reduce the gap among domains and minimize classification or regression model error [Pan *et al.*, 2009; Chen *et al.*, 2009]. Supervised feature learning is similar to methods used in multi-task learning. [Argyriou *et al.*, 2006] propose a sparse feature learning method for multi-task learning. In a follow-up work [Argyriou *et al.*, 2008] propose a spectral regularization framework on matrices for multi-task structure learning. [Raina *et al.*, 2007] propose applying sparse coding [Lee *et al.*, 2006],which is an unsupervised feature construction method in order to learn higher level features for transfer learning. Recently, manifold learning has been an alternative method for transfer learning. [Wang and Mahadevan, 2008] propose a Procrustes analysis based approach to manifold alignment without correspondences, which can be used to transfer the knowledge across domains via the aligned manifolds.

3 Problem Statement

In this section, we describe the problem we focus on in this paper and introduce some notations and definitions used in this paper.

A *domain* $\mathcal{D}$ includes two components: a feature space $\mathcal{X}$ and a joint probability distribution $P(\vec{x})$, where $\vec{x} \in \mathcal{X}$. Given a specific domain $\mathcal{D} = \{\mathcal{X}, P(\cdot)\}$, a *task* $\mathcal{T}$ associated with domain $\mathcal{D}$ consists of two parts: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (represented by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which can be learned from a training sample with data pairs $\{\vec{x}_i, y_i\}$, where $\vec{x}_i \in \mathcal{X}$ and $y_i = f(\vec{x}_i) \in \mathcal{Y}$.

Consider the case where there are one source domain $\mathcal{D}_S$ with its corresponding task $\mathcal{T}_S$ and one target domain $\mathcal{D}_T$ with its corresponding task $\mathcal{T}_T$. Here $\mathcal{D}_S = \{\vec{x}_{S_1}, \ldots, \vec{x}_{S_{n_S}}\}$, where $\vec{x}_{S_i} \in \mathcal{X}_S$ is an instance, $\mathcal{X}_S$ is the feature space in the source domain, and n_S is the number of instances in the source domain; the task in the source domain $\mathcal{T}_S = \{\mathcal{Y}_S, f_S(\cdot)\}$, where $\mathcal{Y}_S$ is the label space in the source domain and f_S is the corresponding predictive function; $\mathcal{D}_T = \{\vec{x}_{T_1}, \ldots, \vec{x}_{T_{n_T}}\}$, where $\vec{x}_{T_i} \in \mathcal{X}_T$ is an instance, $\mathcal{X}_T$ is the feature space in the target domain, and n_T is the number of instances in the target domain; and the task in the target domain $\mathcal{T}_T = \{\mathcal{Y}_T, f_T(\cdot)\}$, where $\mathcal{Y}_T$ is the label space in the target domain and f_T is the corresponding predictive function. In addition, we set $\vec{x} = \{x_1, \cdots, x_{n_f}\}$, where x_i stands for the value of the *i-th* feature A_i in instance $\vec{x}$ and n_f is the number of features.

In this paper, we address the cross-domain text classification problems, which imply that $\mathcal{D}_S \neq \mathcal{D}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$. We aim at searching for a predictive function $f_P(\cdot)$ approaching the target predictive function $f_T(\cdot) \in \mathcal{T}_T$ as much as possible with aid of the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$.

4 Our Approach

In this section, we present a novel domain adaptation approach to tackle the cross-domain classification problems by means of both feature and instance selection. The general assumption in domain adaptation is that marginal densities, $\mathcal{P}(\vec{x}_S)$ and $\mathcal{P}(\vec{x}_T)$, are very different. It is the key reason leading to low classification performance because a classifier having good performance in $\mathcal{D}_S$ may lead to poor results on $\mathcal{D}_T$. Here, we try to measure the distance of $\mathcal{D}_S$ and $\mathcal{D}_T$ and make it as small as possible. On the other hand, a feature has major difference of probability distribution among different classes is more likely to help us classify the test data. Thus, we also take the difference of a feature's distribution among class labels into consideration. The main idea here is to select featurs which have distinguished utility for classification from $\mathcal{D}_S$ and make distributions of $\mathcal{D}_S$ and $\mathcal{D}_T$ as similar as possible. In this way, we can assume that features useful for classifying instances in $\mathcal{D}_S$ could also be functional for

classification on $\mathcal{D}_T$. This can be achieved by feature and instance weighting. $W_F \in \mathbb{R}^{n_f}$ and $W_I \in \mathbb{R}^{n_S}$ are the weights of features and training instances respectively and used to get the weighted $\mathcal{D}_S$ and $\mathcal{D}_T$ to realize the previous goals. For $\forall i, j, W_{F_i}$ and W_{I_j} range from 0 to 1. From the above discussion, we can define the objective function as follows:

$$\mathcal{J}(W_F, W_I) = \|D_B\| - \|D_I\| + \lambda_{W_F}\|\mathbf{1_F} - W_F\|^2 + \lambda_{W_I}\|\mathbf{1_I} - W_I\|^2, \quad (1)$$

where $D_B,\ D_I \in \mathbb{R}^{n_f}$ are column vectors. D_B is used to estimate the distance between $\mathcal{D}_S$ and $\mathcal{D}_T$ (the smaller, the better). Besides, we let D_I to estimate the difference of A_i among different classes, which can be also considered as the inner distance of a domain. $\mathbf{1_F} \in \mathbb{R}^{n_f}$ and $\mathbf{1_I} \in \mathbb{R}^{n_S}$ are column vectors with all ones. $\|\mathbf{1_F} - W_F\|^2$ and $\|\mathbf{1_I} - W_I\|^2$ are to control the change of two domains. λ_{W_F} and λ_{W_I} are trade-off factors. Here, we use KL divergence[Kullback and Leibler, 1951] to calculate the distance of features. Thus Eq.(1) can be written as:

$$\begin{aligned}\mathcal{J}(W_F, W_I) =& \sum W_{F_i} * D_{KL}(p(A_i)\|q(A_i)) \\ &- \sum W_{F_i} * (\sum_{y_j, y_k \in \mathcal{Y}} D_{KL}(p_j(A_i)\|p_k(A_i)) \\ &+ \lambda_{W_F}\|\mathbf{1_F} - W_F\|^2 + \lambda_{W_I}\|\mathbf{1_I} - W_I\|^2,\end{aligned} \quad (2)$$

where $p(A_i)$ and $q(A_i)$ are probability distributions of feature A_i in $\mathcal{D}_S$ and $\mathcal{D}_T$ respectively. $p_j(A_i)$ and $p_k(A_i)$ reprsent the distributions of A_i on class j and k in $\mathcal{D}_S$ respectively. D_I only consider the inner distance of $\mathcal{D}_S$ due to the absence of class labels in $\mathcal{D}_T$. $D_{KL}(p(x)\|q(x))$ is the KL-divergence defined as follows:

$$\begin{aligned}D_{\mathrm{KL}}(p(x)\|q(x)) &= \int_{-\infty}^{\infty} p(x)\log\frac{p(x)}{q(x)} \\ &= \sum_x p(x)\log\frac{p(x)}{q(x)},\end{aligned} \quad (3)$$

Note that KL-divergence is always non-negative due to the Gibbls' inequality[Cover and Thomas, 1991]. In addition, $p(A_i)$, $p_j(A_i)$ and $p_k(A_i)$ are the distributions of A_i consisting of the weighted training instances. Each A_i has the internals $V_i = \{v_{i1}, \cdots, v_{il}\}$. Eq.(2) can be rewritten as:

$$\begin{aligned}\mathcal{J}(W_F, W_I) =& \sum_i W_{F_i} * D_{KL}(p(W_{I_m} \cdot \mathcal{D}_{S_{mi}})\|q(\mathcal{D}_{T_{ri}})) \\ &- \sum_i W_{F_i} * (\sum_{y_j, y_k \in \mathcal{Y}} D_{KL}(p_j(W_{I_m} \cdot \mathcal{D}_{S_{mi}}) \\ &\|p_k(W_{I_m} \cdot \mathcal{D}_{S_{mi}}))) \\ &+ \lambda_{W_F}\|\mathbf{1_F} - W_F\|^2 + \lambda_{W_I}\|\mathbf{1_I} - W_I\|^2 \\ =& \sum_i W_{F_i} * \sum_{u=i1}^{u=il}(p(v_u)\log\frac{p(v_u)}{q(v_u)}) \\ &- \sum_i W_{F_i} * (\sum_{y_j, y_k \in \mathcal{Y}} \sum_{u=i1}^{u=il}(p_j(v_u)\log\frac{p_j(v_u)}{p_k(v_u)}) \\ &+ \lambda_{W_F}\|\mathbf{1_F} - W_F\|^2 + \lambda_{W_I}\|\mathbf{1_I} - W_I\|^2,\end{aligned} \quad (4)$$

where $\mathcal{D}_{S_{mi}}$ is the ith feature of instance $\vec{x}_{S_m} \in \mathcal{D}_S$. Accordingly, $\mathcal{D}_{T_{ri}}$ is the ith feature of instance $\vec{x}_{T_r} \in \mathcal{D}_T$. Our objective is to acquire W_F and W_I making $\mathcal{J}(W_F, W_I)$ as small as possible.

Now, we introduce how to iteratively update W_F and W_I. In the first step, we fix W_I and update W_F to optimize $\mathcal{J}(W_F, W_I)$ by computing the derivate of $\mathcal{J}(W_F, W_I)$ with parameter W_F:

$$\begin{aligned}\frac{\partial\mathcal{J}(W_F, W_I)}{\partial W_{F_i}} =& \sum_{u=i1}^{u=il}(p(v_u)\log\frac{p(v_u)}{q(v_u)}) \\ &- (\sum_{y_j, y_k \in \mathcal{Y}} \sum_{u=i1}^{u=il}(p_j(v_u)\log\frac{p_j(v_u)}{p_k(v_u)}) \\ &- 2 * \lambda_{W_F} * (1 - W_{F_i}).\end{aligned} \quad (5)$$

Given W_I and solving

$$\frac{\partial\mathcal{J}(W_F, W_I)}{\partial W_{F_i}} = 0, \quad (6)$$

we can get the new value of W_{F_i}. Next, we update W_I when W_F is fixed. We calculate the derivate of $\mathcal{J}(W_F, W_I)$ with parameter W_I:

$$\begin{aligned}\frac{\partial\mathcal{J}(W_F, W_I)}{\partial W_{I_m}} =& \sum_i (W_{F_i} * \partial(p(v_u)\log\frac{p(v_u)}{q(v_u)}) \\ &- \sum_i W_{F_i} * \sum_{y_j, y_k \in \mathcal{Y}} \partial(p_j(v_u)\log\frac{p_j(v_u)}{p_k(v_u)}) \\ &- 2 * \lambda_{W_I} * (1 - W_{I_i}),\end{aligned} \quad (7)$$

where v_u equals to $\mathcal{D}_{S_{mi}}$. It is apparent that $p(v_u)$, $p_j(v_u)$, $p_k(v_u)$ are the first-order function on W_{I_m}'s reciprocal. Let

$$\frac{\partial\mathcal{J}(W_F, W_I)}{\partial W_{I_m}} = 0 \quad (8)$$

and we obtain the new value of W_{I_m}. We can use feature weighting and instance weighting iteratively until convergence or for the specific times. We summarize the above process in **Algorithm 1** .

Algorithm 1 BI-Weighting (BIW) Domain Adaptation for Cross-Domain Text Classification

INPUT $\mathcal{D}_S$: source domain; $\mathcal{D}_T$: target domain; λ_{W_F}: smooth factor for feature weight; λ_{W_I}: smooth factor for source domain instance weight; C_B: base classifier trainer (e.g. SVM).

OUTPUT the predictive function f_P.

1: Initialize W_F and W_I;
2: **repeat**
3: Fixing W_I, update W_F in light of Eqs. (5 & 6);
4: Fixing W_F, update W_I in light of Eqs. (7 & 8);
5: **until** convergence (or achieving iterNum times)
6: Revise $\mathcal{X}_S$ to $\mathcal{X}_S^R$ with W_F and W_I;
7: Build a classifier f_P with $\mathcal{X}_S^R$ and C_B;
8: **RETURN** f_P

5 Experiments

In this section, we empirically evaluate the effectiveness and efficiency of the algorithm proposed in Section 4.

5.1 Experimental Setting

Our evaluation uses the Web pages crawled from the Open Directory Project(ODP)[2] during May 2010, including categories of Arts, Computers, Games, Health, Home, News, Recreation, Reference, Science and Shopping. Each Web page in ODP was classified by human experts. We preprocess the raw data as follows. First, all the Chinese Web pages are translated into English by Google Translator[3]. Then, we transform all the letters to lowercase, and stem the words using the Porter's stemmer [Porter, 1980]. Afterwards, stop words are removed.

In order to evaluate our algorithm, we set up six cross-language classification tasks. Five of them are binary classification tasks, and the other one is for three-class classification. We randomly resample 50000 instances from English Web pages as the training set due to the computational issue.

In the following experiments, we choose two traditional classifiers: Naive Bayes and Support Vector Machines (SVM) [Chang and Lin, 2001], and four transfer learning approaches: Transductive SVM (TSVM) [Joachims, 1999], Information Bottleneck (IB) [Ling *et al.*, 2008], Transfer Component Analysis (TCA) [Pan *et al.*, 2009] and domain adaptation with Extracting Discriminative Concepts (EDC) [Chen *et al.*, 2009] for the purpose of comparisons. λ_{W_F} and λ_{W_I} are set to 0.05.

With the help of the knowledge from the source domain, transfer learning aims at predicting labels for instances in the target domain with classification performance as close as possible to the traditional classification scenarios, training a classifier with instances in the target domain and applying it to the instances in the same domain. In particular, for the purpose of comparisons, we implement a "upper bound" algorithm by committing 5-fold cross-validations with NB, SVM and TSVM individually over the data in the target domain (the Chinese text), which are called NB-CN, SVM-CN and TSVM-CN respectively. Note that, these classifiers are virtual "enemies" against the transfer learning algorithms. If the gap between a specific transfer learning algorithm and the virtual "enemies" is narrow, the transfer learning algorithm is close to the "limit"; otherwise, there is still space to improve. Precision, recall and F_1-measure are calculated in each experiment in this paper, which are widely used as evaluation metrics in text classification.

5.2 Impact of Iteration Times

Since our algorithm BIW is an iterative algorithm, an important factor of BIW is the number of iterations ($iterNum$) or the convergence speed. We run a few tests to observe the convergence speed. Figure 1 shows the impact of iteration times on BIW in one of these tests. F1-measure at zero point indicates the value of the algorithm without any feature and instance weighting process. BIW usually converges at the two or third iteration. In the following experiments, we set $iterNum$ to be 3 to make sure that BIW is to converge.

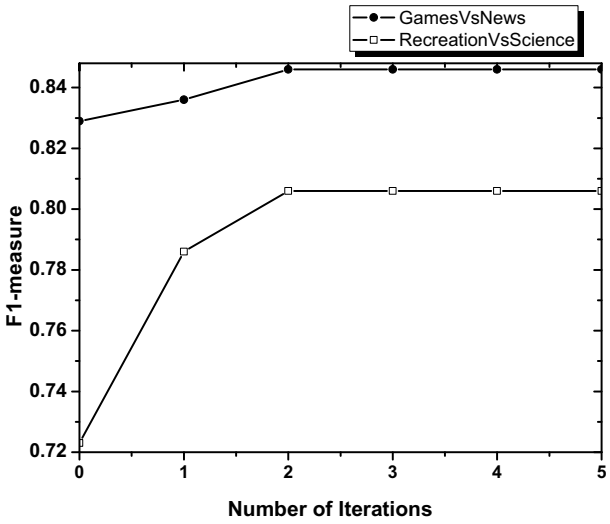

Figure 1: Impact of Iteration Times

5.3 Comparison with Instance Weighting Only

Since a few instance weighting methods have been applied on transfer learning, in this subsection, we compare our algorithm BIW with the methods using only instance weighting(Uni-Weighting) to see whether it is necessary to integrate instance weighting and feature weighting. We will compare BIW with feature weighting (or extracting) only methods in the next subsection. Figure 2 shows the classification performance of NB, Uni-NB and BIW-NB on two cross-language classification tasks. From the figure, it is clear that Uni-NB outperforms NB but does worse than BIW-NB. The results show the benefit of combining feature weighting and instance weighting.

5.4 Comparison with Baselines

In this subsection, we conduct experiments on all the six datasets. Five of the tasks are binary class classification while the other one is a 3-class classification. We use naive Bayes, LibSVM and IB as base classifiers in our BIW algorithm, which are named BIW-NB, BIW-SVM and BIW-IB respectively. Table 2 shows the experiments results of the comparisons with different baseline methods as well as the "upper bound" methods. From these tables, we see that the BIW algorithms are consistently better than their base classifiers. Further more, the BIW algorithms in some tasks perform as good as or even outperform the "upper bound" methods.

We compare the BIW algorithm with TCA [Pan *et al.*, 2009] and EDC [Chen *et al.*, 2009] in smaller datasets (randomly select around 1000 instances in each task with about 10000 features) for the computational issue including memory usage of EDC[4]. Table 1 shows the result of the comparison. Apparently, BIW outperforms TCA and EDC by 5.2% and 7.3% in the overall F_1-measure.

[2]http://www.dmoz.com/.

[3]http://www.google.com/language_tools.

[4]EDC needs $O(m^2)$ of memory, where m is the number of features.

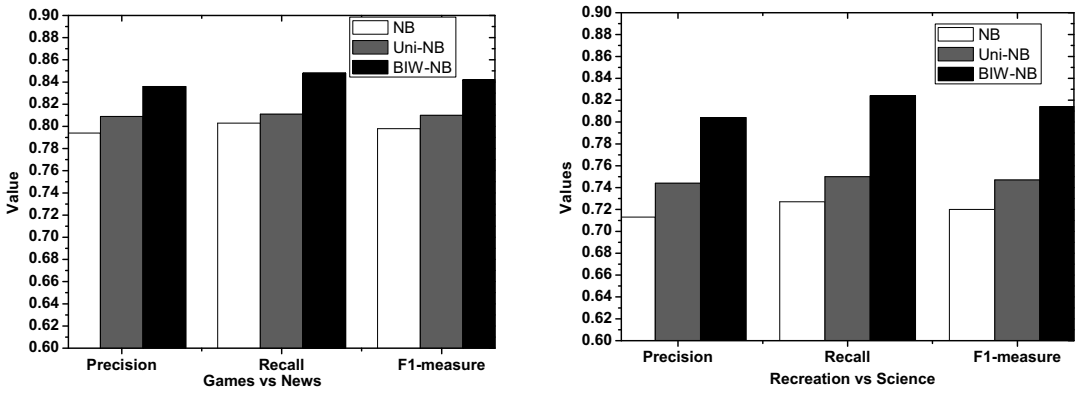

Figure 2: Comparison with Instance Weighting Only

Table 1: Comparisons with other Transfer Learning methods 1:Games vs. News, 2:Health vs. Home, 3:News vs. Recreation, 4:News vs. Recreation, 5:Recreation vs. Science and 6:Recreation vs. Reference vs. Shopping.

Data set	Precision			Recall			F1-measure		
	TCA	EDC	BIW	TCA	EDC	BIW	TCA	EDC	BIW
1	0.833	0.808	0.835	0.759	0.813	0.823	0.794	0.810	0.829
2	0.803	0.703	0.848	0.607	0.712	0.686	0.691	0.706	0.758
3	0.77	0.783	0.889	0.846	0.802	0.849	0.806	0.792	0.868
4	0.815	0.775	0.856	0.720	0.694	0.701	0.765	0.732	0.771
5	0.845	0.830	0.953	0.876	0.840	0.831	0.860	0.835	0.888
6	0.533	0.561	0.614	0.666	0.524	0.633	0.592	0.542	0.623
Average	0.767	0.743	**0.833**	0.746	0.731	**0.754**	0.751	0.736	**0.790**

6 Conclusion and Future Works

In this paper, we present a novel transfer learning approach to tackle the cross-domain text classification problems. We first align the feature spaces in both domains utilizing some online translation service, which makes the two feature spaces under the same coordinate. Then we propose an alternated method for domain adaptation. We empirically evaluate the effectiveness and efficiency of our approach. The experimental results show that our approach outperforms some baselines including supervised, semi-supervised and transfer learning algorithms.

In the future, we plan to study other potentially better algorithms for the transductive transfer learning problems. We will apply our approach to other domains. We also plan to extend our method to the regression scenario.

Acknowledgments

We thank the anonymous reviewers for helpful comments. This work was supported by National Natural Science Foundation of China (61003140, 61033010) and the Fundamental Research Funds for the Central Universities (09lgpy62).

References

[Argyriou *et al.*, 2006] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

[Argyriou *et al.*, 2008] Andreas Argyriou, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. A spectral regularization framework for multi-task structure learning. In *NIPS*, pages 25–32, Cambridge, MA, 2008. MIT Press.

[Bel *et al.*, 2003] Núria Bel, Cornelis H. A. Koster, and Marta Villegas. Cross-lingual text categorization. In Traugott Koch and Ingeborg Sølvberg, editors, *ECDL*, volume 2769 of *Lecture Notes in Computer Science*, pages 126–139. Springer, 2003.

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chen *et al.*, 2009] Bo Chen, Wai Lam, Ivor W. Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *KDD*, pages 179–188. ACM, 2009.

[Cover and Thomas, 1991] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[Dai *et al.*, 2007] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In Ghahramani [2007], pages 193–200.

[Ghahramani, 2007] Zoubin Ghahramani, editor. *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvalis, Oregon, USA, June*

Table 2: The Precision, Recall and F1-measure on Six Datasets 1:Games vs. News, 2:Health vs. Home, 3:News vs. Recreation, 4:News vs. Recreation, 5:Recreation vs. Science and 6:Recreation vs. Reference vs. Shopping.

Data set	Precision									
	NB	BIW-NB	SVM	BIW-SVM	TSVM	IB	BIW-IB	NB-CN	SVM-CN	TSVM-CN
1	0.823	0.829	0.866	0.882	0.876	0.846	0.893	0.767	0.903	0.960
2	0.794	0.836	0.812	0.841	0.976	0.840	0.888	0.937	0.905	0.914
3	0.774	0.793	0.856	0.890	0.864	0.752	0.846	0.922	0.894	0.903
4	0.653	0.717	0.610	0.661	0.764	0.750	0.774	0.784	0.846	0.764
5	0.713	0.804	0.752	0.889	0.820	0.833	0.842	0.857	0.928	0.822
6	0.673	0.725	0.654	0.711	-	0.712	0.785	0.839	0.889	-
Average	0.738	0.784	0.758	0.812	-	0.789	0.838	0.851	0.894	-
Data set	Recall									
	NB	BIW-NB	SVM	BIW-SVM	TSVM	IB	BIW-IB	NB-CN	SVM-CN	TSVM-CN
1	0.815	0.820	0.871	0.888	0.788	0.862	0.902	0.946	0.956	0.912
2	0.803	0.848	0.835	0.859	0.719	0.842	0.900	0.808	0.904	0.811
3	0.799	0.819	0.875	0.923	0.669	0.813	0.827	0.793	0.989	0.816
4	0.662	0.730	0.583	0.685	0.569	0.722	0.779	0.773	0.852	0.877
5	0.727	0.824	0.768	0.901	0.749	0.773	0.865	0.854	0.916	0.932
6	0.659	0.714	0.662	0.726	-	0.736	0.802	0.83	0.891	-
Average	0.744	0.793	0.766	0.830	-	0.791	0.846	0.834	0.918	-
Data set	F1-measure									
	NB	BIW-NB	SVM	BIW-SVM	TSVM	IB	BIW-IB	NB-CN	SVM-CN	TSVM-CN
1	0.819	0.824	0.869	0.885	0.830	0.854	0.897	0.847	0.929	0.935
2	0.798	0.842	0.823	0.850	0.828	0.841	0.892	0.868	0.904	0.859
3	0.786	0.805	0.865	0.906	0.754	0.782	0.836	0.853	0.939	0.857
4	0.657	0.723	0.596	0.673	0.652	0.736	0.777	0.778	0.846	0.817
5	0.720	0.814	0.760	0.895	0.783	0.802	0.853	0.855	0.922	0.874
6	0.666	0.719	0.658	0.718	-	0.724	0.793	0.834	0.890	-
Average	0.741	0.788	0.772	0.821	-	0.790	0.841	0.839	0.905	-

20-24, 2007, volume 227 of *ACM International Conference Proceeding Series*. ACM, 2007.

[Jiang and Zhai, 2007] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*. The Association for Computer Linguistics, 2007.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In Ivan Bratko and Saso Dzeroski, editors, *ICML*, pages 200–209. Morgan Kaufmann, 1999.

[Kullback and Leibler, 1951] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[Lee *et al.*, 2006] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *NIPS*, pages 801–808. MIT Press, 2006.

[Ling *et al.*, 2008] Xiao Ling, Gui-Rong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu. Can chinese web page be classified with english data source? In *Proceedings of the Seventeenth World Wide Web Conference*, 2008.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[Pan *et al.*, 2009] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. In Craig Boutilier, editor, *IJCAI*, pages 1187–1192, 2009.

[Porter, 1980] M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

[Raina *et al.*, 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: transfer learning from unlabeled data. In Ghahramani [2007], pages 759–766.

[Rigutini *et al.*, 2005] Leonardo Rigutini, Marco Maggini, and Bing Liu. An em based training algorithm for cross-language text categorization. In Andrzej Skowron, Rakesh Agrawal, Michael Luck, Takahira Yamaguchi, Pierre Morizet-Mahoudeaux, Jiming Liu, and Ning Zhong, editors, *Web Intelligence*, pages 529–535. IEEE Computer Society, 2005.

[Tishby *et al.*, 2000] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *CoRR*, physics/0004057, 2000.

[Wang and Mahadevan, 2008] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1120–1127. ACM, 2008.