Multi-domain Adaptation for Sentiment Classification: using Multiple Classifier Combining Methods

Shoushan LI and Chengqing ZONG

National Laboratory of Pattern Recognition,

Institute of Automation, Chinese Academy of Sciences, Beijing, 100190

{sshanli, cqzong }@nlpr.ia.ac.cn

Abstract:

Sentiment classification is very domain-specific and good domain adaptation methods, when the training and testing data are drawn from different domains, are sorely needed. In this paper, we address a new approach to domain adaptation for sentiment classification in which classifiers are adapted for a specific domain with training data from multiple source domains. We call this new approach 'multi-domain adaptation' and present a multiple classifier system (MCS) framework to describe and understand it. Under this framework, we propose a new combining method, called Multi-label Consensus Training (MCT), to combine the base classifiers for selecting 'automatically-labeled' samples from unlabeled data in the target domain. The experimental results for sentiment classification show that multi-domain adaptation using this method improves adaptation performance.

Keywords:

Sentiment classification; domain adaptation; multiple classifier combining.

1. Introduction

Sentiment classification is the task of classifying documents according to their opinion, or sentiment, regarding a given subject matter. Many studies have focused on this task [1]. Most previous studies, however, are based on the assumption that the training data is representative of the testing data. This assumption is routinely incorrect when the application domain changes. Here, 'domain adaptation' is proposed. This usually involves teaching a classifier using both plentiful labeled data from the source domain and unlabeled data from the target domain. Blitzer et al. [2] discuss precisely this problem where a structural correspondence learning (SCL) algorithm is applied to sentiment classification and shown to be successful when an appropriate domain is chosen to match the target domain.

Sentiment classification is a very domain-specific problem [3], and adaptation algorithms successful in one domain to another are extremely difficult. Using learning data from only a single domain to teach adaptation for other domains is not feasible because adaptation fails totally when the data distribution in the target domain is very different from that in the source domain [2]. We, therefore, suggest improving adaptation results by using multiple domains to source learning data. In the rest of this paper this new approach of domain adaptation is called multi-domain adaptation, while adaptation from one domain to another is referred to as one-to-one adaptation. For multi-domain adaptation, we can firstly apply an adaptability measurement to select the domain with the best adaptability and then use one-to-one adaptation algorithms. Consider, for example, labeled product reviews from three different domains: books, DVDs, and electronics, and we want to teach a classifier with documents discussing kitchens. Since the electronics domain shows the best computed adaptability [2], labeled data from the electronics domain is selected to train a classifier. One major drawback of this method for multi-domain adaptation is that it drops the labeled data from the other domains that may contain helpful information. We believe there are superior methods that can make full use of all the available labeled data.

In this paper, we present a multiple classifier system (MCS) framework to describe and understand the approach of multi-domain adaptation. Constructing an MCS generally consists of two steps: training a number of component classifiers then using combination rules to combine them. In the first step, classifiers can be trained in different ways including using different learning algorithms, training data, and feature sets. These component classifiers are usually called the base classifiers. In multi-domain adaptation, they can be naturally obtained using labeled data from different domains. There are various rules presented in the MCS literature to combine the base classifiers in the second step. The architecture of the MCS framework for multi-domain adaptation is shown in Figure 1.

978-1-4244-2780-2/08/$25.00 ©2008 IEEE

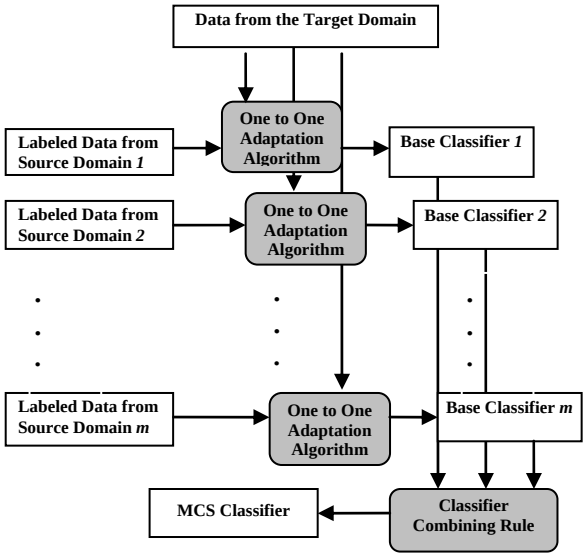

Figure 1: MCS framework for multi-domain adaptation

2. Our Framework

In a standard classification problem, a classifier is the predicting function that maps an input vector to the corresponding output. Usually, the classifier is trained using training samples and then used to predict other label-unknown samples. In domain adaptation, the training samples and the label-unknown samples do not come from the same domain. In other words, the testing (target) data and the training (source) data are drawn from different statistical distributions. As far as multi-domain adaptation is concerned, the training samples themselves come from different domains while the testing samples are from another domain which is different from all the multiple source domains.

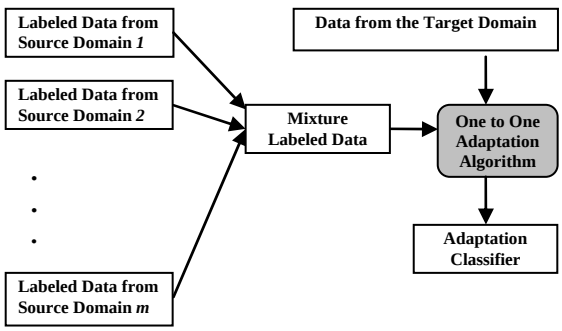

Figure 2: *Add-all* framework for multi-domain adaptation

Multi-domain adaptation can be simplified to one-to-one adaptation in two straightforward ways. The first is to simply mix the training samples from all domains and then to train a classifier with the mixed data. This method has been pursued by Aue and Gamon [3] as a baseline, and is named *all-data*. It is outside the scope of our MCS framework since it only uses one classifier; and belongs to a framework we call *add-all* as shown in Figure 2. The second way is to use the samples from the domain with the best adaptability where the adaptability is evaluated by some measurement (e.g., to compute *A*-distance between two domains using unlabeled data) [2].

We approach this problem from a different perspective. In multi-domain adaptation, multiple base classifiers can be naturally obtained using labeled data from different domains. Given these multiple classifiers, the problem can be treated as a typical MCS problem. Note that the second method mentioned above can be seen as an instance of the MCS framework where the classifier combination rule is select-best. There are some notable advantages to treating this problem as an MCS problem:

1. The idea that one can benefit from combining multiple classifiers has been widely proposed in various research areas including: pattern recognition, machine learning and natural language processing. In particular, combining classifiers trained naturally from different sources is extremely effective for the overall performance improvement.
2. MCS, as an important research field, has been studied intensively over the last decade [4]. Many problems in this field that could help us better understand multi-domain adaptation have been addressed and solved. For example, an important issue in MCS is the classifier selection problem that states that good classifiers should be selected for combination rather than all [5]. The same issue can be found in multi-domain adaptation in terms of whether we need to select good source domains for adaptation rather than using all.
3. MCS possesses a great many well-designed combination rules. Once a suitable MCS mode is built, the corresponding rules can be directly borrowed or slightly modified to deal with the current problem.

Generally speaking, domain adaptation can be divided into two main categories [6]. In the first, called fully supervised, a few labeled samples are used in the learning process. In the second, called semi-supervised, only unlabeled samples are available for the learning process. In multi-domain adaptation, different categories demand different base classifiers and methods of combination. In accordance with the MCS framework, let us give our methods for multi-domain adaptation in two steps: training base classifiers and combining base classifiers; these are described in Sections 3 and 4 respectively.

3. Training Base Classifiers

The first step in constructing an MCS is to train multiple base classifiers. In the fully supervised case, each base classifier can be directly trained with the labeled data from both each source domain and the target domain. In the semi-supervised case, each base classifier is trained using labeled data from each source domain and unlabeled data from the target domain simultaneously. To do this, a structural correspondence learning (SCL) algorithm is used to train multiple one-to-one adaptation classifiers. The SCL algorithm represents the-state-of-the-art in domain adaptation and has been shown to be very effective in sentiment classification [2]. We provide a basic description of this algorithm below. For a detailed description, please see Blitzer et al. [7].

The input data for SCL is labeled data from the source domain and unlabeled data from the target domain or both domains. Generating a classifier more suitable for classification of data in the target domain consists of three main steps. First, pivot features that occur frequently in both domains are selected. Second, correlations are modeled between the pivot features and all other features by training with linear classifiers to classify occurrences of each pivot in all unlabeled data. Finally, with the help of the correlations, the classifier is trained on the labeled source domain data that also provides the influence of the unseen features in the training data.

In sentiment classification, pivot features with high mutual information (MI) are especially helpful for the adaptation [2]. Therefore, we select the pivot features from the top-10% MI features in our experiments.

4. Classifier Combining Methods

4.1. Fully supervised methods

In the supervised case, various trained combination rules, such as the weighted sum rule, Dempster-Shafer and the weighted voting rule can be 'borrowed' [8] [9]. Among these trained methods, a class of methods called meta-learning has been shown to be very effective. The key idea behind this class is to train a meta-classifier with input attributes that are the output of the base classifiers [10].

Aue and Gamon [3] have considered precisely this case for multi-domain adaptation and use the meta-learning combination rule. Their experimental results using small labeled samples in the target domain show significant improvement over the all-data method in some, but not all, domains. Since the number of labeled samples can not be large (otherwise, the adaptation is not necessary), meta-learning becomes difficult. As a result, making use of a large amount of unlabeled samples in the target domain seems to be very promising as acquiring unlabeled samples is much easier.

4.2. Semi-supervised methods

Unlike the supervised case, there have been very few methods proposed for semi-supervised MCS [11]. Didaci and Roli [12] propose two general methods for semi-supervised MCS using co-training and self-training, called the extended co-training algorithm and the ensemble-driven self-training algorithm. With the help of their work, we propose the ensemble-driven self-training algorithm for multi-domain adaptation shown in Figure 3. The basic idea of this algorithm is to label some of the unlabeled data with high classification confidence with an MCS, and then put the 'automatically-labeled' data into the training data of each source domain.

Ensemble-driven Self-training for Multi-domain Adaptation

Input:

(a) m sets L_k $(k = 1, ..., m)$ of labeled training samples from m different source domains

(b) a set U of unlabeled samples from the target domain

Algorithm:

Step 1. Create a set U' of samples by choosing u samples at random from U

Step 2. Loop for N iterations:

(1) Build m classifiers c_k using both L_k and U' with the one-to-one adaptation algorithm

(2) Create a MCS using $\{c_k\}$

(3) Run MCS on U'

(4) For each class j

- Select the n_j samples assigned to the class j by the MCS with the highest confidence

(5) Add these $m \times \sum_j n_j$ self-labeled samples to each L_k

(6) Randomly choose $m \times \sum_j n_j$ samples from U to replenish U'

Output:

New base classifiers $\{c_k\}$ trained on the new L_k and U with the one-to-one adaptation algorithm

Figure 3: The ensemble-driven self-training algorithm applied to multi-domain adaptation

This semi-supervised algorithm is a general algorithm. Two operations in the algorithm need to be specified for any given task: the one-to-one adaptation

algorithm and the combining rule to construct the MCS. For sentiment classification, the one-to-one adaptation algorithm applied is the SCL algorithm introduced above. With respect to the combination rule here, it must satisfy two basic requirements: (1) it can select unlabeled samples from the target domain; (2) these selected samples are classified with a very high accuracy. Accordingly, we define two functions whose input is one sample *x* and corresponding output represents whether an event related to *x* happens:

$$E_1(x) = \begin{cases} 1 & x \text{ is selected by MCS} \\ 0 & x \text{ is not selected by MCS} \end{cases}$$

and

$$E_2(x) = \begin{cases} 1 & \text{The selected } x \text{ is correctly classified} \\ 0 & \text{The selected } x \text{ is not correctly classified} \end{cases}$$

According to the definition, we have

$$P(E_2(x)=1) = P(l_{MCS}(x) = real(x) \mid E_1(x)=1) \tag{1}$$

where l_{MCS} is the class label of the sample assigned by the MCS and the function $real(x)$ returns the real class label. To obtain some high-precision 'automatically-labeled' samples from the target domain, we should guarantee a high probability of $P(E_2(x)=1)$ and also not a too low probability of $P(E_1(x)=1)$.

From previous studies, we find that the best base classifier often achieves much higher accuracy than other base classifiers and even the MCS. One example is a supervised case that combines all the base classifiers using 50 training samples from the target domain. All the adaptation results using meta-learning are worse than using the select-best rule (comparing Table 1 with Table 5 in Aue and Gamon [3]). Another example is a semi-supervised case in which it was reported that, for any fixed amount of labeled data, it is always better to draw from the nearest domain as a source than using some combination of all available domains [2]. Given these findings, our combination rule needs to select samples with a higher precision than the best base classifier when using MCS. Thus we select samples whose labels assigned by the base classifiers are the same. That is to say, an sample is selected and assigned to class j when the class labels assigned by all the base classifiers are consistent, i.e.,

$$\begin{aligned} &E_1(x)=1 \text{ and } l_{mcs} = j \text{ if and only if} \\ &l_1(x) = \ldots = l_i(x) = \ldots = l_m(x) = j \end{aligned} \tag{2}$$

Where $l_i(x)$ is the class label assigned by the *i*-th base classifier. For simplicity, we call this rule Multi-label consensus and the ensemble-driven self-training algorithm using this combining rule Multi-label Consensus Training (MCT). Note that the outputs of this algorithm are still multiple one-to-one adaptation classifiers. These new classifiers are called MCT classifiers.

The rule of Multi-label consensus has the advantage that it guarantees that the MCS selects samples with a higher accuracy than the best classifier. We prove this advantage below.

Each base classifier c_i achieves accuracy A_i which can also be seen as the estimation of the probability $P(l_i(x) = real(x))$ that a test sample is correctly classified by this classifier. For simplicity, it is briefly denoted as P_i. We assume:

(I) $P_i > 1 - P_i$, i.e., $P_i > 0.5$, which means each base classifier achieves accuracy higher than a random classification of a binary classification problem. This assumption is easily satisfied in real applications, and;

(II) The probabilities of each P_i are independent of each other. Although this assumption is hard to make hold in real applications, it can be approximately satisfied in multi-domain adaptation where the base classifiers are trained using different features and also different data sets.

Given the condition (2) and the assumption (I), we find

$$P(l_{MCS}(x) = real(x)) = P_1 \cdot P_2 \cdot \ldots P_m \tag{3}$$

According to the definition of $E_1(x)$ and the condition (2), we have

$$\begin{aligned} P(E_1(x)=1) = {} & P(l_1(x) = \ldots = l_m(x) = real(x)) \\ & + P(l_1(x) = \ldots = l_m(x) \neq real(x)) \end{aligned} \tag{4}$$

Since sentiment classification we care about here is a binary classification problem, we have

$$\begin{aligned} & P(l_1(x) = \ldots = l_m(x) \neq real(x)) \\ & = P(l_1(x) \neq real(x), \ldots, l_m(x) \neq real(x)) \end{aligned} \tag{5}$$

Consider the conclusion that if two events are independent from each other then their complement events are also independent, we have

$$\begin{aligned} & P(l_1(x) = \ldots = l_m(x) \neq real(x)) \\ & = (1-P_1)(1-P_2)\ldots(1-P_m) \end{aligned} \tag{6}$$

According to (3), (4), (6), and (1), we get

$$P(E_2(x)=1) = \frac{P_1 \cdot P_2 \cdot \ldots P_m}{P_1 \cdot P_2 \cdot \ldots P_m + (1-P_1)(1-P_2)\ldots(1-P_m)} \tag{7}$$

To guarantee that $P(E_2(x)=1)$ is larger than P_{best}, the probability that one test sample is correctly classified by the best base classifier, the following formula must be satisfied:

$$P(E_2(x)=1) / P_{best} > 1 \tag{8}$$

which can be rewritten as:

$$\frac{P_1 \cdot ... P_{best-1} \cdot P_{best+1} \cdot ... P_m}{P_1 \cdot ... P_{best} \cdot ... P_m + (1-P_1)...(1-P_{best})...(1-P_m)} > 1 \quad (9)$$

The above condition is equivalent to:

$$P_1 \cdot ... P_{best-1} \cdot (1-P_{best}) \cdot P_{best+1} \cdot ... P_m - (1-P_1)...(1-P_{best})...(1-P_m) > 0 \quad (10)$$

This condition is satisfied using the assumption (I). Therefore, given the two assumptions, the Multi-consensus rule guarantees that MCS selects samples with a higher accuracy than the best classifier. It is worth pointing out that this rule is designed for multi-domain sentiment classification when the number of source domains is small. When the number increases, this selecting strategy is too strict to select enough 'automatically-labeled' data (as $P(E_1(x)=1)$ becomes very small). Thus, the selecting strategy needs to be modified for large numbers of source domains. For example, we can demand that a fixed number of output labels are consistent and also includes the label from the best base classifier.

Let us also consider the iteration number *N* in MCT. Since our base classifiers are trained using different data sets, the diversity among these classifiers becomes smaller when more 'automatically-labeled' samples are added. As a result, the 'automatically-labeled' data gradually takes the dominant place in the base classifiers. As these are not true labeled data, classification errors accumulate quickly as the iteration time grows. Moreover, using the same 'automatically-labeled' samples in each new base classifier makes the assumption (II) for the Multi-label consensus rule harder to satisfy. Therefore, the selecting process is performed only once across all unlabeled data *U*.

Note that there is another semi-supervised method, the extended co-training algorithm, in Didaci and Roli [12] which can also be applied to multi-adaptation. It shares the same basic idea as the ensemble-driven self-training algorithm, except in selecting the samples using each base classifier rather than MCS. This algorithm is not suitable, however, for sentiment classification because sentiment classification is very domain-specific and classifiers trained in one domain may perform very badly on the target domain even given the adaptation process. For example, to classify reviews of books, an adaptation classifier using labeled reviews of kitchen appliances performs much worse than one using DVDs. Clearly, adding the 'automatically-labeled' samples from these bad classifiers only adds noisy information.

5. Experiments

As the supervised case has been experimentally studied using meta-learning in Aue and Gamon [3], we do not consider this case in our experiments. Our experimental studies concentrate on multi-domain adaptation of sentiment classification in the semi-supervised case.

Data Set[2]: our experiments are carried out on a data set of product reviews from four domains: books, DVDs, electronics and kitchen appliances, B, D, E and K, for short, in the tables and figures. This data was assembled by Blitzer et al. [2] and used to evaluate the SCL algorithm in sentiment classification. Each domain has 1,000 positive and 1,000 negative samples. Besides the labeled data, large amounts of unlabeled samples are available in this data set. The unlabeled samples are disregarded here. Instead, the labeled samples in one domain are used as unlabeled samples when this domain is the target domain in the adaptation process.

Classification Algorithm: Following Blitzer et al. [2], we use linear predictors on unigram and bi-gram features, trained to minimize Huber loss with stochastic gradient descent [13]. To implement adaptation for one-to-one domains in the semi-supervised case, we apply the SCL algorithm which has been introduced in Section 3.

Experiment Implement: For each domain, we first split the labeled data into 5 folds. Then, we select 4 folds for training and use the remaining one fold for testing. As a result, 5 different distributions are obtained for each domain. Similarly, we get 5 different distributions for the whole data set by selecting the distribution from each domain sequentially: selecting the first distribution of each domain, then the second, the third and so on. All the experiments below are run on these 5 different distributions of the complete data set and the reported result is the mean value of the accuracies of the 5 runs.

Baseline and Gold Standard: The baseline is a linear classifier trained without adaptation, while the gold standard is an in-domain classifier trained on the same domain as it is tested. Classification accuracies are reported in Table 1 where the first column represents the source domains and the values in the diagonal are the results of the gold standard domain classifiers.

Source Domain	B	D	E	K
B→	*0.81*	**0.765**	0.704	0.733
D→	**0.747**	*0.837*	0.696	0.713
E→	0.690	0.683	*0.842*	**0.795**
K→	0.672	0.686	**0.787**	*0.840*

Table 1: Mean accuracy results using the baseline and gold standard.

First, let us consider the iteration number *N* in MCT. The quality of the adaptation decreases slightly as the

[2] The data set is available at http://www.seas.upenn.edu/~mdredze/datasets/sentiment/

iteration number N increases. The reason for this decrease was proposed in the last section. Due to a limited space, we only give the results when N=1.

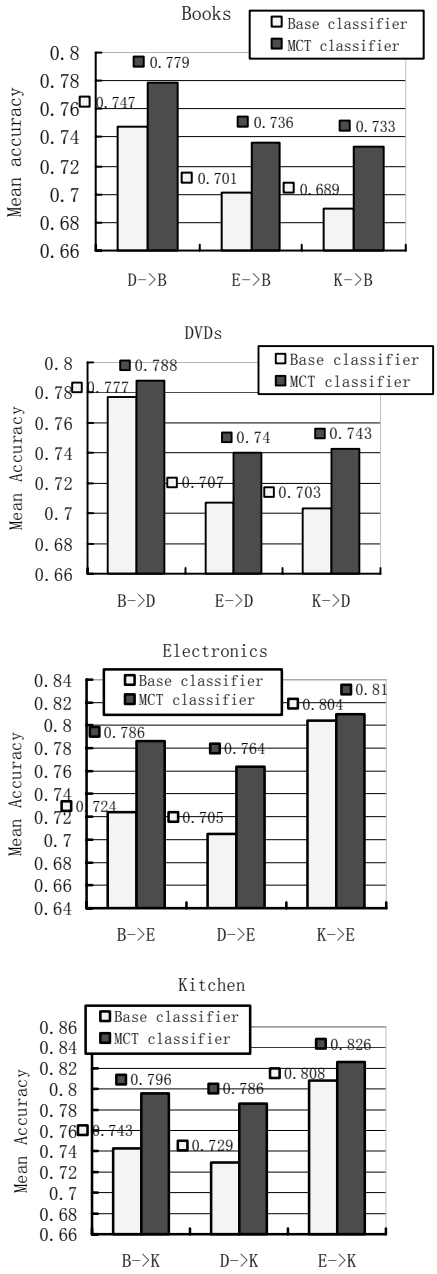

Figure 4: Mean accuracy results for domain adaptation between all pairs using base classifiers and MCT classifiers.

Figure 4 presents the results for all pairs of domain adaptation. The main difference between base and MCT classifiers is that MCT classifiers use 'automatically-labeled' samples as additional training data in the target domain while base classifiers do not. From the results, we can see that these 'automatically-labeled' samples play a very important role in the adaptation process, especially when the source domain is very different from the target domain. Even when the domains are similar (and the best performance is achieved), the MCT classifier consistently shows superior performance to the base classifiers. More specifically, the best MCT classifiers on the target domains B, D, E, and K reduce the error by 12.6%, 5.4%, 3.1%, and 9.3% using source domains D, B, K and E respectively. Note that the performance from domain D to domain B shows the most impressive error reduction of 12.6%. We think this is mainly due to differences in the performances of the base classifiers not being too large, e.g., the performance of the best base classifier is 0.747 which is only 0.046 better than 0.701 for the second best classifier. The distances are much larger when the target domains are D, E, and K, which are 0.07, 0.08, and 0.065 respectively.

These results are encouraging because they suggest a better adaptation approach: when training data from multiple domains is available, combining them produces better results than selecting only the best, especially when the multiple domains show similar adaptabilities.

Table 2 shows the two probabilities of $P(E_1(x)=1)$ and $P(E_2(x)=1)$ which are estimated as

$$P(E_1(x)=1)=\frac{\textit{the number of selected unlabeled samples}}{\textit{the number of all unlabeled samples}}$$

$$P(E_2(x)=1)=\frac{\textit{the number of correctly classified samples}}{\textit{the number of selected unlabeled samples}}$$

Probability	B	D	E	K
$P(E_1(x)=1)$	0.581	0.595	0.600	0.597
$P(E_2(x)=1)$	0.872	0.864	0.876	0.908

Table 2: The two probabilities estimated through testing on unlabeled samples in the target domains: B, D, E, and K by the multi-consensus MCS respectively.

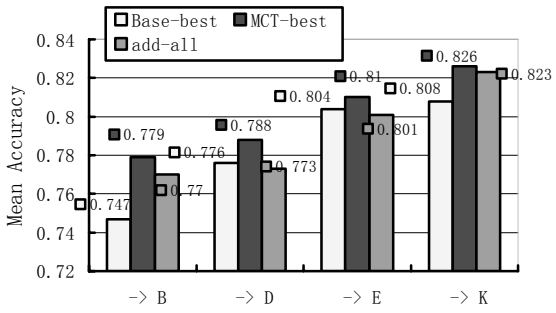

Figure 5: Comparison of selecting the best base classifier, selecting the best MCT classifier, and the *add-all* method.

From Table 2, we observe that MCS achieves a

much higher precision than any single base classifier, which can explain why our method does work.

Besides our MCS framework, there exist other models that can be used to describe the task of multi-domain adaptation. One straightforward model is to mix labeled data from all available domains, (mentioned in Section 2 as *add-all*). We provide a comparison result in Figure 5 where *add-all* denotes an SCL classifier trained using both labeled samples from all source domains and un-labeled samples from the target domain. The result demonstrates that although this model can sometimes outperform selecting the best base classifiers, it is consistently worse than selecting the best MCT classifier.

6. Related Work

To the best of our knowledge, there have been very few previous studies that directly investigate the problem of multi-domain adaptation. There are two related studies that have discussed solutions for this task for the fully supervised case. The first is reported by Aue and Gamon [3] who apply meta-learning to combine classifiers as they are applied here. The second, from Daumé III [6], is the feature augmentation method which is proposed for one-to-one domain adaptation and is easily extended to the multi-domain case. This method can be seen as a special case of the all-all method that mixes all data with augmented input features. However, neither of these two methods is suitable for the semi-supervised case as discussed in this paper since they cannot deal with unlabeled data.

7. Conclusion and Future Work

In summary, the contribution of this paper is twofold. First, we provide an MCS framework to describe multi-domain adaptation. Under this framework, we seek methods for this problem in two cases: 'fully supervised' and 'semi-supervised'. Second, we propose an MCS driven self-training combination method called MCT for the semi-supervised case and theoretically prove that this method selects 'automatically labeled' samples with a higher accuracy than using the best base classifier. Experimental results demonstrate that 'automatically labeled' samples from MCT can greatly improve adaptation. This good performance confirms that when data from multiple source domains is available, its combination is an effective way to improve adaptation performance.

It is worth pointing out that although this work is motivated by and evaluated for adaptation of sentiment classification, the general framework is suitable for other classification problems. It will be interesting to investigate this new task for other NLP problems, e.g., parsing, name entity recognition and so on; this will be the subject of future work.

Acknowledgements

The research work in this paper is partially funded by the Natural Science Foundation of China under Grant No. 60575043 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech Research and Development Program (863) of China under Grant No. 2006AA01Z194 and 2006AA010108-4, and Nokia (China) Co. Ltd as well.

References

[1] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques", In Proceedings of EMNLP, 2002.

[2] John Blitzer, Mark Dredze, and Fernando Pereira. "Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for sentiment classification", In Proceedings of ACL, 2007.

[3] Anthony Aue and Michael Gamon. "Customizing sentiment classifiers to new domains: a case study", In Proceedings of RANLP, 2005.

[4] Romesh Ranawana, and Vasile Palade. "Multi-Classifier Systems: Review and a roadmap for developers", International Journal of Hybrid Intelligent Systems, 3(1): 35-61, 2006.

[5] Zhi H. Zhou, Jianxin Wu, and Wei Tang. "Ensembling neural networks: Many could be better than all", Artificial Intelligence, 137(1-2): 239-263, 2002.

[6] Hal Daumé III. "Frustratingly easy domain adaptation", In Proceedings of ACL, 2007.

[7] John Blitzer, Ryan McDonald, and Fernando Pereira. "Domain adaptation with structural correspondence learning", In Proceedings of EMNLP, 2006.

[8] Giorgio Fumera, and Fabio Roli. "A theoretical and experimental analysis of linear combiners for multiple classifier systems", PAMI, 27: 942 – 956, 2005.

[9] Lei Xu, Adam Krzyzak, and Ching Y. Suen. "Methods of combining multiple classifiers and their applications to handwriting recognition", IEEE Tran. Systems, Man and Cybernetics, 22(3):418-435, 1992.

[10] Saso Dzeroski, Bernard Zenko. "Is combining classifiers with stacking better than selecting the best one?" Machine Learning, 54(3): 255-273, 2004.

[11] Fabio Roli. "Semi-supervised multiple classifier systems: background and research directions", In

Proceedings of Multiple Classifier Systems (MCS), 2005.
[12] Luca Didaci, and Fabio Roli. “Using co-training and self-training in semi-supervised multiple classifier systems”, In Proceedings of SSPR/SPR, 2006.
[13] Tong Zhang. “Solving large scale linear prediction problems using stochastic gradient descent algorithms”, In Proceedings of ICML, 2004.