Li SS, Huang CR, Zong CQ. Multi-domain sentiment classification with classifier combination. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(1): 25–33 Jan. 2011. DOI 10.1007/s11390-011-1108-9

Multi-Domain Sentiment Classification with Classifier Combination

Shou-Shan Li[1,2] (李寿山), Chu-Ren Huang[2] (黄居仁), and Cheng-Qing Zong[3] (宗成庆)

[1] *NLP Lab, School of Computer Science and Technology, Soochow University, Suzhou 215006, China*

[2] *Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University, Hong Kong, China*

[3] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

E-mail: {shoushan, churenhuang}@gmail.com; cqzong@nlpr.ia.ac.cn

Received December 24, 2009; revised November 4, 2010.

Abstract State-of-the-arts studies on sentiment classification are typically domain-dependent and domain-restricted. In this paper, we aim to reduce domain dependency and improve overall performance simultaneously by proposing an efficient multi-domain sentiment classification algorithm. Our method employs the approach of multiple classifier combination. In this approach, we first train single domain classifiers separately with domain specific data, and then combine the classifiers for the final decision. Our experiments show that this approach performs much better than both single domain classification approach (using the training data individually) and mixed domain classification approach (simply combining all the training data). In particular, classifier combination with weighted sum rule obtains an average error reduction of 27.6% over single domain classification.

Keywords sentiment classification, multiple classifier system, multi-domain learning

1 Introduction

Sentiment classification is the task of classifying text according to sentiment information[1]. It can be considered as a special case of text categorization, where the criterion of classification is the attitude expressed in the text (e.g., recommended or not recommended, positive or negative) rather than some facts (e.g., sport or education). Recently, this task has received considerable attention in the communities of natural language processing and information retrieval due to its many existing and potential applications such as online product review classification[2], question answering[3], and automated summarization[4].

Note that almost all existing studies conduct the sentiment classification tasks for single domains separately without interactions among different domains. In a real application system, however, multiple domains are often involved. For example, when designing an online product review classification system, we cannot merely collect labeled review data on one product, e.g., book, to train the classifier because this classifier may perform very badly on some other products, e.g., electronics, due to the domain-specific character of sentiment classification[5]. As a result, we need to collect some training data from several domains. Given the multi-domain training data, a new task arises, called multi-domain sentiment classification, which aims to classify the reviews from different domains.

In this paper, we employ classifier combination approach to multi-domain sentiment classification which involves two main steps: generating single domain classifiers (called member classifiers) by using training data from each domain and combining them with some combining rules. Experiments are performed on a dataset consisting of four product reviews, and the results demonstrate the effectiveness of this approach.

The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 states the task of multi-domain sentiment classification and gives two straightforward approaches. The details of the classifier combination approach are described in Section 4. Section 5 presents and discusses the evaluation results. Lastly we conclude our paper in Section 6.

2 Related Work

The methods for sentiment classification can be roughly categorized into either unsupervised or supervised. Unsupervised methods usually first calculate the semantic orientations of phrases with their associations with some human-selected seed words (e.g., "poor" and

Short Paper

Supported by the National Natural Science Foundation of China under Grant No. 61003155 and Start-Up Grant for Newly Appointed Professors under Grant No. 1-BBZM in The Hong Kong Polytechnic University.

"excellent"). The sentiment of a document is then predicted using the average semantic orientations of all the phrases it contains[6-7]. In contrast, supervised methods first annotate some data in each category and then train a classifier with the annotated data with some machine learning classification algorithms[1]. In this study, we focus on supervised methods.

Pang *et al.*[1] first employ machine learning methods to sentiment classification where three popular classification algorithms are tested on classifying movie reviews. They find that machine learning methods definitely outperform human-produced baselines and the classification algorithm of support vector machine (SVM) achieves the best performance. Subsequently, many other studies make efforts to improve the performance of machine-learning based classifiers by some special means such as using subjectivity summarization[8], seeking new superior textual features[9], and using document subcomponent information[10]. Their efforts do improve the classification performances to a certain extent. Moreover, their studies have extended their evaluations from "movie" domain to many other domains.

However, research on sentiment classification over multiple domains remains sparse. Aue and Gamon[11] and Blitzer *et al.*[5] discuss domain adaptation problems which involve two domains: source domain and target domain. Specifically, labeled data from the source domain is used to train a classifier for classifying data from the target domain (where none or very few labeled data are available). Our work focuses on the problem of how to make multiple domains "help each other" when all contain some labeled samples. Previously, Li and Zong[12] firstly address the concept of multi-domain sentiment classification and also employ classifier combination to deal with this new task. Apparently, their work is elementary and only focuses on one combining rule of meta-learning. In this study, we will check more combining rules and give much more detailed experimental results. Moreover, we will show that two simple fixed combining rules of sum and product rule sometimes give comparative performances to meta-learning rule and a trained rule of weighted sum rule performs significantly better than meta-learning rule.

Most recently, Dredze and Crammer[13] propose an online algorithm for multi-domain learning which can be directly applied to multi-domain sentiment classification. This algorithm is mainly based on the parameter combination of multiple classifiers. Different from their algorithm, our classifier-combination method is used to combine the results (outputs) from the member classifiers rather than the parameters. For comparison, we will test another parameter-combination based approach called feature augmentation[14] which is reported to exhibit comparable performance to their algorithm but easier to implement.

3 Problem Statement and Two Straightforward Approaches

Standard sentiment classification tasks aim to seek a predictor f (also called a classifier) that can classify a document (represented as a vector $\boldsymbol{x}$) into one of the defined n categories (denoted as $\{c_1, \ldots, c_n\}$). In sentiment classification, the categories usually include two kinds of sentiment orientation: positive and negative. To train the classifier f, a set of labeled samples called training data need to be collected. Most existing studies assume that these training samples are all coming from one single domain.

Multi-domain sentiment classification aims to seek a predictor which can classify documents from multiple domains. Formally, there are m different domains which are indexed by $k = \{1, \ldots, m\}$ and a sample from the k-th domain is denoted as x_k.

To handle this new task, two straightforward approaches can easily be proposed. The first one, called single domain classification (SDC), makes use of the training data drawn from the l-th domain to train a single domain classifier f_l $(l = 1, 2, \ldots, m)$ that is used to predict the reviews from the same domain. That is to say, the classifiers are individually trained and tested using the training and testing data from each domain. The second one, called all-mixed classification (AMC), simply mixes all the training data from all domains to train a common classifier $f_{\rm common}$ with the mixed training data. The classifier $f_{\rm common}$ is used to classify the reviews regardless of their domains. Note that this approach is called feature-level fusion in [12]. The architectures of these two approaches are shown in Fig.1 and Fig.2 respectively.

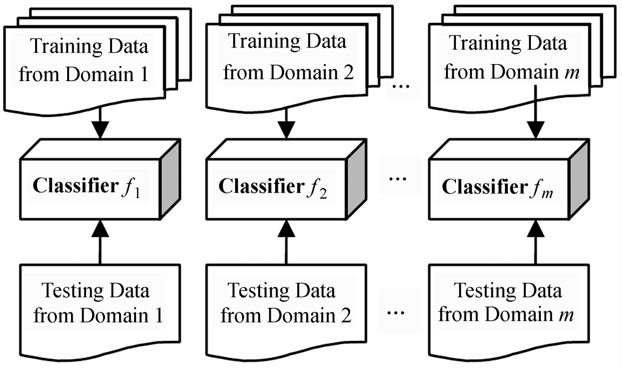

Training Data from Domain 1
Training Data from Domain 2
Training Data from Domain m
...
Classifier f_1
Classifier f_2
Classifier f_m
Testing Data from Domain 1
Testing Data from Domain 2
Testing Data from Domain m

Fig.1. Architecture of single domain classification (SDC).

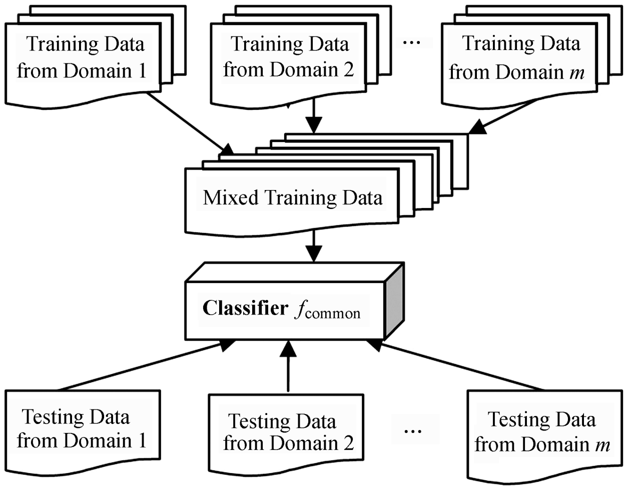

Training Data from Domain 1
Training Data from Domain 2
···
Training Data from Domain m
Mixed Training Data
Classifier $f_{\rm common}$
Testing Data from Domain 1
Testing Data from Domain 2
···
Testing Data from Domain m

Fig.2. Architecture of all-mixed classification (AMC).

We believe that both these two approaches are not very competent and the reasons are as follows. On one side, general sentiment text often shares similar expressions independent of domains and thus reviewers sometimes use similar words or even sentences to deliver their positive or negative opinions on reviews cross different domains. For example, the word of "worthless" can always be used for negatively reviewing any products while the word of "perfect" is usually found in the positive reviews of many domains. Given the training data from multiple domains, we can certainly use all the data to better learn global sentiment classification information. Apparently, SDC approach is often difficult to capture adequate information for sentiment classification due to the limited scale training data in one single domain. Using the data from other domains is possibly to get more global sentiment classification information which might be missing in the in-domain data. On the other side, each domain has its own character on expressing sentiment information. For example, the term "quiet" is very informative in the kitchen product reviews but provides no sentiment information in some other domains like books or DVDs. Unfortunately, AMC approach neglects or weakens local information for sentiment classification on each domain. In brief, we believe that a better approach to multi-domain sentiment classification must take into account both global and local sentiment classification information from the multi-domain training data.

4 Classifier Combination Approach

4.1 Overview

Combining multiple classifiers is a learning mechanism where a collection of classifiers is trained for the same classification task. The combining operation might be beneficial since different classifiers would offer complementary information of the patterns for classification. Over the past twenty years, multiple classifier system (MCS) has often been considered as a more practical and effective solution for many recognition tasks than using single classifier[15-16].

Generally speaking, the construction of an MCS consists of two main steps, i.e., training a number of component classifiers (also called member classifiers) and using combining rules to integrate the results from the member classifiers for the final decision.

In the first stage, a number of member classifiers might be generated where the individual member classifiers can differ in applying different learning algorithms, using different training data, or employing multiple feature sets. For multi-domain sentiment classification, multiple member classifiers are naturally obtained through training different labeled data from each domain. Note that these member classifiers are also employed as single domain classifiers f_l $(l = 1, 2, \ldots, m)$ in SDC approach. Apparently, each member classifier contains local information.

In the second stage, combining rules integrate the outputs of first-stage decisions and make the final classification decision. This combining process makes global information be impliedly considered in the combination classifier. Generally, the combining rules are categorized into two types, i.e., fixed rules and trained rules. They will both be applied to multi-domain sentiment classification.

4.2 Fixed Rules

Given the member classifiers, fixed rules are conducted to combine their outputs in a fixed way. For multi-domain sentiment classification, the constructed MCS (denoted as $f_{\rm MCS}$) with fixed rules will be used to classify testing data regardless of the domains they belong to. The corresponding architecture is shown in Fig.3.

Note that each member classifier usually provides not only the class label outputs but also some kind of confidence information, e.g., posterior probabilities of the testing sample belonging to each class. Then sum and product rules will be conducted to combine these probabilities. Formally, each member classifier f_l $(l = 1, \ldots, m)$ assigns a posterior probability vector $\boldsymbol{P}_l(x_k)$ to a test sample (denoted as x_k) from the k-th domain

$$\boldsymbol{P}_l(x_k) = [p_l(c_l|x_k), \ldots, p_l(c_n|x_k)]^t$$

where $p_l(c_l|x_k)$ denotes the probability that the l-th member classifier considers that x_k belongs to c_i.

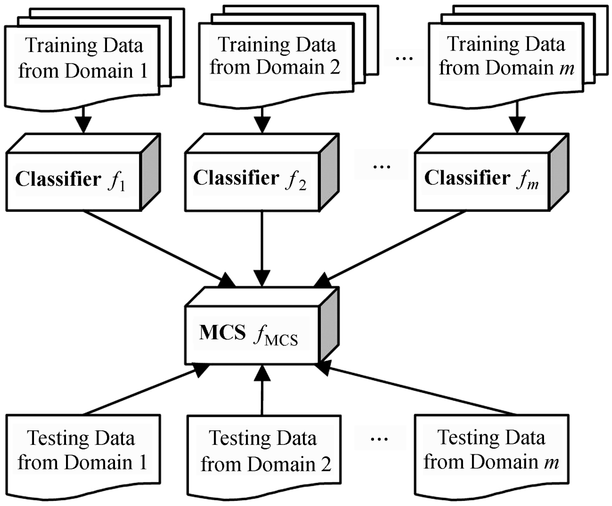

Training Data from Domain 1
Training Data from Domain 2
...
Training Data from Domain *m*
Classifier f_1
Classifier f_2
...
Classifier f_m
MCS $f_{\rm MCS}$
Testing Data from Domain 1
Testing Data from Domain 2
...
Testing Data from Domain *m*

Fig.3. Architecture of the classifier combination approach with fixed combining rules.

Product rule combines member classifiers by multiplying the posterior possibilities and use the multiplied possibility for decision, i.e.,

$$\text{assign } y \to c_j$$

$$j = \arg\max_i \prod_{l=1}^{m} p_l(c_i|x_k).$$

Sum rule combines member classifiers by summing the posterior possibilities and use the summation for decision, i.e.,

$$\text{assign } y \to c_j$$

$$j = \arg\max_i \sum_{l=1}^{m} p_l(c_i|x_k).$$

4.3 Trained Rules

Trained rules are conducted to combine the outputs in a trained way and multiple MCSs will be constructed. The corresponding architecture is shown in Fig.4.

Two popular trained rules, namely, weighted sum rule and meta-learning rule, will be applied respectively.

Weighted sum rule combines member classifiers by summing the posterior possibilities with different weights. Formally, weighted sum rule is

$$\text{assign } y \to c_j$$

$$j = \arg\max_i \sum_{l=1}^{m} \omega_l^k \cdot p_l(c_i|x_k)$$

where ω_l^k ($l = 1, \ldots, m$; $k = 1, \ldots, m$, $0 \leqslant \omega_l^k \leqslant 1$) are the weights assigned to each member classifier while the combination classifier is used to testing the k-th domain. In real applications, an essential work is to train the values of ω_l^k. The training process can be considered as an optimization problem which is solved on an extra set of labeled data (called development data). Suppose the development data from the k-th domain contains N_k samples denoted as $\{x_1^k, \ldots, x_q^k, \ldots, x_{N_k}^k\}$. The optimal weights in $f_{\text{MCS-}k}$ are thus the solution of the following optimization problem:

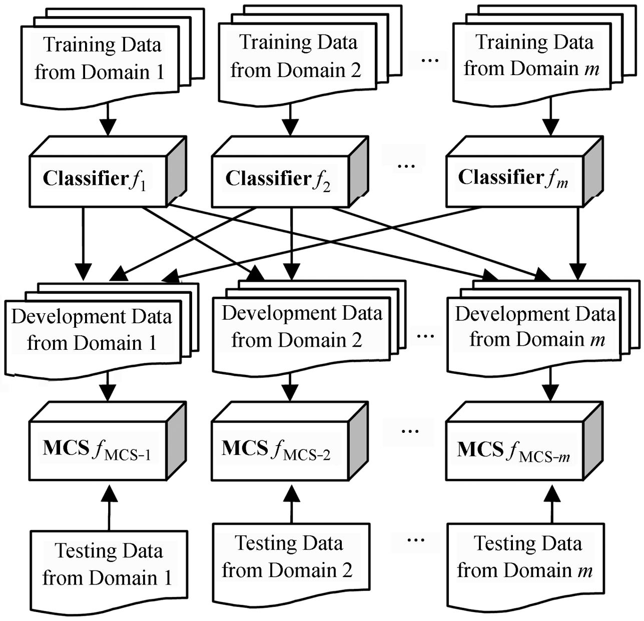

Training Data from Domain 1
Training Data from Domain 2
...
Training Data from Domain *m*
Classifierf_1
Classifierf_2
...
Classifierf_m
Development Data from Domain 1
Development Data from Domain 2
...
Development Data from Domain *m*
MCS $f_{\text{MCS-1}}$
MCS $f_{\text{MCS-2}}$
...
MCS $f_{\text{MCS-}m}$
Testing Data from Domain 1
Testing Data from Domain 2
...
Testing Data from Domain *m*

Fig.4. Architecture of the classifier combination approach with trained combining rules.

$$\arg\min_{w_1^k,\ldots,\omega_m^k} \sum_{q=1}^{N_k} Z\Big(\arg\max_i \sum_{l=1}^{m} (\omega_l^k \cdot p_l(c_i|x_q^k)), l_{\text{real}}(x_q^k)\Big)$$

where $Z(l_{\text{estimated}}, l_{\text{real}})$ is a zero-one loss function which is defined as

$$Z(l_{\text{estimated}}, l_{\text{real}}) = \begin{cases} 0, & \text{when } l_{\text{estimated}} = l_{\text{real}}, \\ 1, & \text{when } l_{\text{estimated}} \neq l_{\text{real}}, \end{cases}$$

and $l_{\text{real}}(x_q^k)$ is the function which returns the true label of the sample x_q^k. One simple way to solve this problem is to perform an exhaustive search for the optimal weights on the development data. But it becomes intractable when the number of member classifiers is large. Therefore, special optimization approaches need to be adopted such as Powell method, multi-dimensional downhill simplex method and some stochastic optimization approaches[17]. In our study, we use the multi-dimensional downhill simplex optimization algorithm which uses geometric relationships to aid in finding function minimums. This method can be easily implemented but performs well in practice[17]. The initial value of each weight is assigned to 0.5.

Another well-known trained rule called meta-learning which has been shown to be very effective in many applications[18]. The key idea behind this rule is to train a meta-classifier with input attributes that are the output of the member classifiers.

Formally, let $\boldsymbol{x}_{k'}$ denote a feature vector of a sample from the development data of the k'-th domain ($k' = 1, \ldots, m$). The output of the l-th member classifier f_l on this sample is the probability distribution over the set of classes $\{c_1, c_2, \ldots, c_n\}$, i.e.,

$$\boldsymbol{P}_l(\boldsymbol{x}_{k'}) = \langle p_l(c_1|\boldsymbol{x}_{k'}), p_l(c_2|\boldsymbol{x}_{k'}), \ldots, p_l(c_n|\boldsymbol{x}_{k'})\rangle.$$

For the k'-th domain, we train a meta-classifier $f_{\text{MCS-}k'}$ ($k' = 1, , 2, \ldots, m$) using the development data from the k'-th domain with the meta-level feature vector $\boldsymbol{x}_{k'}^{\text{meta}} \in \mathbb{R}^{m \cdot n}$

$$\boldsymbol{x}_{k'}^{\text{meta}} = \langle \boldsymbol{P}_1(\boldsymbol{x}_{k'}), \ldots, \boldsymbol{P}_l(\boldsymbol{x}_{k'}), \ldots, \boldsymbol{P}_m(\boldsymbol{x}_{k'})\rangle.$$

Each meta-classifier is then used to test the testing data from the same domain.

Different from fixed rules and the weighted sum rule, the classification algorithms are employed in both the two steps: member classifier generation and the result fusion to construct a multiple classifier system with meta-learning approach. In the first step of member classifier construction, the classification algorithm is used to train the multiple in-domain classifiers where the input vectors are the Boolean weights of all words (e.g., unigrams, bigrams). In the second step of result fusion, the classification algorithm is used to train the meta-classifier where the input vectors are the classification results from member classifiers (e.g., posterior probabilities).

4.4 Discussion

In the first step, each member classifier is obtained by training the data from one domain. But each is also adaptive for classifying data from any other domain. Their adaptation capability to another domain is mainly due to the shared features which appear in both domains and the shared features convey the sentiment classification information between two different domains. Imagine that two domains contain entirely different features and then the corresponding member classifiers will not help each other at all. Therefore, we emphasize that sharing some features is a precondition for applying classifier combination approach for multi-domain sentiment classification. We will check the statistic information about the shared features in the first part of our experimental results.

In the second step, the combination classifier combines the outputs from all member classifiers and combining rules often highly influence the performances. Fixed rules are simple to implement and need no development data. In contrast, trained rules are more complex and need extra labeled data to train some parameters. On the basis of experimental and theoretical results, researchers agree that fixed rules usually perform well for classifiers exhibiting similar accuracy, and zero or similar negative correlation among their outputs (balanced classifiers). Trained rules are instead claimed to outperform fixed rules for performance-imbalanced classifiers where some classifiers performs much better than some others[19]. We will empirically compare their performances in the second part of our experimental results.

5 Experimentation

We carry out our experiments on classifying reviews into two categories: positive or negative. The data are the labeled product reviews from four domains: books, DVDs, electronics, and kitchen appliances[①]. For short, they are referred to as B, D, E and K respectively. Each domain contains 1000 positive and 1000 negative reviews and they are partitioned randomly into training data, development data, and testing data with the proportion of 70%, 20%, and 10% respectively. The development data are used to train the MCS (e.g., meta-classifier in meta-learning rule) when using the trained rules.

5.1 Experimental Results of Shared Features

We use unigrams (single word, e.g., "good") as features and perform a standard feature selection process with Bi-Normal Separation (BNS) method that is reported to be excellent in many text categorization tasks[20]. Table 1 shows the numbers of the shared features between every two domains when only 100 top features are selected from each domain. From the table, we see that those most informative features according to BNS scores are very different from each domain. For example, many words like "dull", "flat", "endless", and "boring" are very informative features in the B domain but become much less informative and even missing in the domains of E and K. These specific features maintain local classification information. But there still exist a few features which are shared by two domains. Specially, the domains E and K share the features with the most quantity of 26. These shared features are believed to convey global sentiment classification information.

[①] This dataset is available at: http://www.seas.upenn.edu/∼mdredze/datasets/sentiment/.

Table 1. Number of Shared Features Between Every Two Domains When Only 100 Features Are Selected from Each Domain

	No. Shared Features	Examples
B & D	11	"stupid", "pathetic"
B & E	7	"crap", "surprisingly"
B & K	10	"beautifully", "excellent"
D & E	10	"worse", "sucks", "highly"
D & K	11	"sucks", "highly"
E & K	26	"refund", "awesome"

More specifically, we change the numbers of the selected features and calculate the shared features' corresponding proportions which are shown in Fig.5. We can see that the proportions of shared features increase when selecting more features. These features ensure different member classifiers can also be used to classify data from other domains. In addition, the proportions of shared features also can reflect the similarity between two domains. For example, E and K apparently share more features than any other two domains which means they take more similarities. Thus we believe that the member classifiers f_E will perform well in the testing data in K, i.e., not too much worse compared to in-domain classifier, i.e., f_K. Similarly, f_K will perform well in the testing data in E. The similarities reflected from Fig.5 are almost consistent with the results in [5].

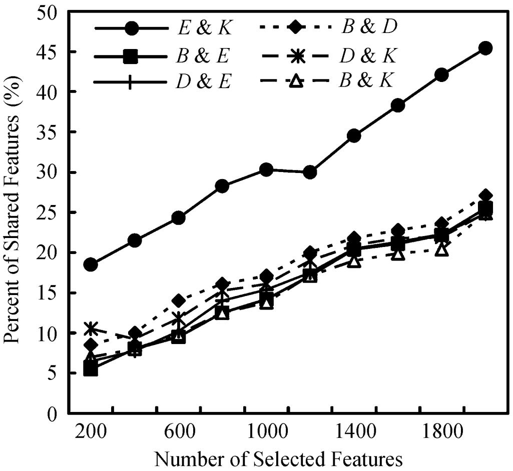


Fig.5. Numbers of selected features vs. shared features' proportions.

Among the shared features, there is one type of features which would hurt the classification performance because those features take different polarities in different domains. For example, "*read-the-book*" is more likely to express a positive opinion in the Book domain but is possibly to be a negative expression in the DVD domain. In order to check the number of the shared features of this type, we take the category information into account: if one feature appears more frequently in positive (or negative) category, it will be considered as a positive (or negative) feature. Then, the number of shared features between every two domains but with different polarities is checked. Our experiment shows that there exist none shared features with different polarities when 100 top features are selected from each domain with BNS method. Even when the number of the selected features increases to 1000 and the features include word bigrams, the shared features with different polarities are very few (less than 3) and they are all bigrams, e.g., "*just_so*", "*house_and*", "*until_it*". This finding is same as the one which has been reported that features which are positive in one domain but negative in another appear very infrequently in [5].

5.2 Experimental Results of Classifier Combination

To construct the member classifiers, we apply SVM algorithm and use LIBSVM[①] with a linear kernel function. This tool provides probability outputs with the good estimation method proposed by [21].

Beside the unigram features, we also use word bigrams (e.g., "very happy") as features. BNS is conducted to select the best sub-sets of unigrams and bigrams with the development data and then their mixture set is used to train member classifiers. The number of the best-performed features are 4400, 1150, 650, and 800 in the four domains of B, D, E and K respectively.

Our baseline uses the single domain classification approach mentioned in Subsection 3.1. Four single domain classifiers are obtained using the training data from the four domains individually and they are also used as the member classifiers in classifier combination. For short, these member classifiers are referred to as f_B, f_D, f_E, and f_K respectively.

Table 2 shows the performances of these member classifiers when testing on each domain. In particular, the performance of baseline is shown in the diagonal results. From this table, we can see that the performances of the member classifiers are very imbalanced when

Table 2. Accuracy Performances of the Member Classifiers When Testing on Each Domain

	→ B	→ D	→ E	→ K
f_B	**0.790**	0.782	0.745	0.715
f_D	0.700	**0.845**	0.755	0.650
f_E	0.675	0.620	**0.850**	0.790
f_K	0.735	0.735	0.785	**0.845**

[①] LIBSVM is an integrated software for SVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

testing each domain. Also, we confirm the above guess that f_K and f_E perform well (0.79 and 0.785) when testing the domains of E and K respectively.

In order to better compare the results of the combining rules, we compute significance levels using two different methods. One is called paired bootstrap resampling[22]. Specifically, we resample the testing data: repeatedly (1000 times) bootstrap new 200 (2000 × 10%) testing samples from the fixed testing data. The reported results are the average of all the 1000 times' accuracies. The other method for evaluating significance is t-test[23]. The conclusion is drawn that rule A is significantly better than rule B only when A performs better than B in more than 95% of the times and the p-value in t-test is less than 0.05.

Table 3 shows the comparison results over four combining rules including sum, product, and meta-learning rules when combining every two member classifiers for testing each domain. From this table, we see that two fixed rules of sum and product rules achieve exactly the same performance. This is because they become equivalent when the applied task involves two categories and two member classifiers[24]. Their performances are comparative to those of one trained rule of meta-learning but are nearly always worse than the results of weighted sum rule.

Table 3. Accuracy Performances of the Four Combining Rules When Combining Every Two Member Classifiers and Testing on Each Domain

	Sum	Product	Weighted Sum	Meta-Learning
$f_B, f_D \rightarrow B$	0.785	0.785	0.800	**0.805**
$f_B, f_E \rightarrow B$	0.790	0.790	**0.800**	0.780
$f_B, f_K \rightarrow B$	**0.815**	**0.815**	**0.815**	0.800
$f_D, f_B \rightarrow D$	**0.885**	**0.885**	0.880	0.875
$f_D, f_E \rightarrow D$	0.825	0.825	0.825	**0.840**
$f_D, f_K \rightarrow D$	0.855	0.855	**0.865**	0.855
$f_E, f_B \rightarrow E$	0.840	0.840	**0.855**	0.850
$f_E, f_D \rightarrow E$	0.865	0.865	**0.880**	0.870
$f_E, f_K \rightarrow E$	0.85	0.850	**0.870**	0.865
$f_K, f_B \rightarrow K$	0.830	0.830	**0.835**	0.830
$f_K, f_D \rightarrow K$	0.830	0.830	**0.850**	0.840
$f_K, f_E \rightarrow K$	0.880	0.880	**0.885**	0.875

In addition to the four combination rules, we implement another straightforward approach AMC and a parameter-combination based approach called feature augmentation. In the method of feature augmentation, the vector of each document consists of m parts (m is the number of domains and equals four here). Each part has the same feature set and represents each domain: the document in the l-th domain will generate one feature vector which becomes the l-th part of the augmentation vector and the other parts are zero vectors. The difference between AMC and feature augmentation lies in their feature spaces: AMC utilizes the mixed feature set from all domains to build feature vector $\boldsymbol{x} \in \mathbb{R}^F$ while feature augmentation approach use the same feature set but augment the feature vector $\boldsymbol{x}$ to a new one $\boldsymbol{x}' \in \mathbb{R}^{m \cdot F}$. For more detailed instruction about the implementation of this approach, please refer to [14].

Fig.6 reports the performances of different approaches when using the data from all four domains. From this figure, we see that the two straightforward approaches along with the feature augmentation approach perform similarly but are apparently worse than combination approaches. In the domains of E and K, all combining rules perform significantly better than both SDC and AMC approaches. We also find that the two fixed rules give comparative performances to the trained rule of meta-learning except for the domain D.

Weighted rule is shown to be the best one which performs significantly better than SDC and meta-learning rule. Averagely, the four rules of sum, product, weighted, and meta-learning give relative error reductions of 11.94%, 11.94%, 17.91%, and 27.61% compared to the baseline of SDC method respectively. These results are consistent with the agreement that trained rules usually outperform fixed rules for performance-imbalanced classifiers.

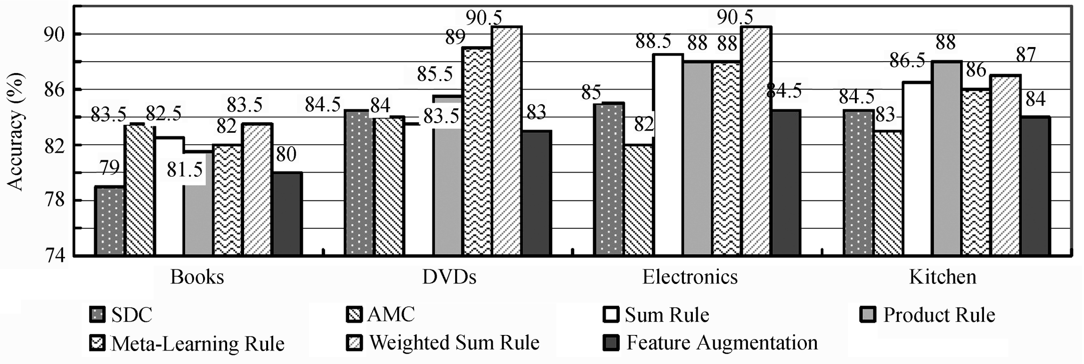


Fig.6. Accuracy performances of different approaches when using the data from all four domains.

6 Conclusion and Future Work

In this paper, we employ classifier combination to the task of multi-domain sentiment classification. Experimental results show that multi-domain classification with classifier combination significantly improves the overall performance compared to single domain classification. In particular, classifier combination with weighted sum rule is shown to optimize classification performance for all domains.

In our experimental results, we found that even though combination of classifiers improves performance, the combination of more classifiers do not necessarily improve the performance. For example, when testing the K domain, weighted sum rule can get an accuracy of 0.885 by combining two member classifiers f_K and f_E but get a lower accuracy of 0.88 when all four member classifiers are combined. This result suggests that improvements may depend on selecting classifiers from suitable domains for combination, especially when the number of domains is larger. In our future work, we will consider the domain selection issue and apply more domains for empirical studies.

References

[1] Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques, In *Proc. EMNLP 2002*, Philadelphia, USA, Jul. 7-12, 2002, pp.79-86.

[2] Cui H, Mittal V, Datar M. Comparative experiments on sentiment classification for online product reviews. In *Proc. AAAI 2006*, Boston, USA, Jul. 16-20, 2006, pp.1265-1270.

[3] Kim S, Hovy E. Identifying opinion holders for question answering in opinion texts. In *Proc. Workshop on Question Answering in Restricted Domains* (*AAAI 2005*), Pittsburgh, USA, Jul. 9-13, 2005, pp.100-107.

[4] Ku L, Liang Y, Chen H. Opinion extraction, summarization and tracking in news and blog corpora. In *Proc. the Spring Symposia on Computational Approaches to Analyzing Weblogs* (*AAAI-CAAW 2006*), Stanford University, USA, Mar. 27-29, 2006, pp.100-107.

[5] Blitzer J, Dredze M, Pereira F. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. ACL 2007*, Prague, Czech, Jun. 23-30, 2007, pp.440-447.

[6] Turney P. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. ACL 2002*, Philadelphia, USA, Jul. 7-12, 2002, pp.417-424.

[7] Zagibalov T, Carroll J. Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *Proc. COLING 2008*, Manchester, UK, Aug. 18-22, 2008, pp.1073-1080.

[8] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. ACL 2004,* Barcelona, Spain, Jul. 21-26, 2004, pp.271-278.

[9] Riloff E, Patwardhan S, Wiebe J. Feature subsumption for opinion analysis. In *Proc. EMNLP 2006*, Sydney, Australia, Jul. 22-23, 2006, pp.440-448.

[10] McDonald R, Hannan K, Neylon T, Wells M, Reynar J. Structured models for fine-to-coarse sentiment analysis. In *Proc. ACL 2007*, Prague, Czech, Jun. 23-30, 2007, pp.432-439.

[11] Aue A, Gamon M. Customizing sentiment classifiers to new domains: A case study. In *Proc. RANLP 2005*, Borovets, Bulgaria, Sept. 21-23, 2005.

[12] Li S, Zong C. Multi-domain sentiment classification (short paper). In *Proc. ACL 2008*, Columbus, USA, Jun. 15-20, 2008, pp.257-260.

[13] Dredze M, Crammer K. Online methods for multi-domain learning and adaptation. In *Proc. EMNLP 2008*, Hawaii, USA, Oct. 25-27, 2008, pp.689-697.

[14] Daumé III H. Frustratingly easy domain adaptation. In *Proc. ACL 2007*, Prague, Czech, Jun. 23-30, 2007, pp.256-263.

[15] Kittler J, Roli F. Multiple classifier systems. In *the First International Workshop on MCS*, Cagliari, Italy, Jun. 21-23, 2000.

[16] Ranawana R, Palade V. Multi-classifier systems: Review and a roadmap for developers. *International Journal of Hybrid Intelligent Systems*, 2006, 3(1): 35-61.

[17] Press W, Teukolsky S, Vetterling W, Flannery B. Numerical Recipes in C++: The Art of Scientific Computing, Second Edition. Cambridge University Press, 2002.

[18] Vilalta R, Drissi Y. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2): 77-95.

[19] Roli F, Fumera G. Analysis of linear and order statistics combiners for fusion of imbalanced classifiers. In *Proc. MCS 2002*, Cagliari, Italy, Jun. 24-26, 2002, pp.252-261.

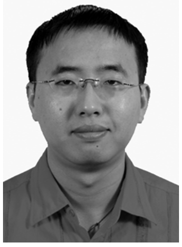
[20] Forman G. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3: 1533-7928.

[21] Wu T, Lin C, Weng R. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, 5: 975-1005.

[22] Koehn P. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP 2004*, Barcelona, Spain, Jul. 25-26, 2004, pp.388-295.

[23] Yang Y, Liu X. A re-examination of text categorization methods. In *Proc. SIGIR 1999*, Berkeley, USA, Aug. 15-19, 1999, pp.42-49.

[24] Li S, Zong C. Classifier combining rules under independence assumptions. In *Proc. MCS 2007*, Prague, Czech, May 23-25, 2007, pp.322-332.



Shou-Shan Li received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (IACAS) in 2008. He was a postdoctoral fellow at The Hong Kong Polytechnic University from 2008 to 2010. Currently, he is an associate professor at the School of Computer Science and Technology, Soochow University, Suzhou, China. His research interests are focused on pattern recognition and natural language processing.

Chu-Ren Huang is the dean of Faculty of Humanities and chair professor of Applied Chinese Language Studies at The Hong Kong Polytechnic University and a research fellow at the Institute of Linguistics, Academia Sinica. He received his Ph.D. degree in linguistics from Cornell University in 1987 and has since played a central role in developing

Chinese language resources and in leading the fields of Chinese corpus and computational linguistics. He is a permanent member of the International Committee on Computational Linguistics and currently serves on the editorial boards of Cambridge Studies in Natural Language Processing, Computational Linguistics and Chinese Language Processing, Journal of Chinese Information Processing, Journal of Chinese Linguistics, Language and Linguistics, Language Resources and Evaluation, and Taiwan Journal of Linguistics. He has published over 70 journal and book articles and over 280 conference papers on different aspects of Chinese linguistics. He has also edited over 14 books or journal special issues, including the just completed volume entitled Ontology and the Lexicon, to be published by Cambridge University Press in the Cambridge Studies in Natural Language Processing Series.

Cheng-Qing Zong is a professor in natural-language technology at the National Laboratory of Pattern Recognition and the deputy director of the National Laboratory of Pattern Recognition, which is part of the Institute of Automation, Chinese Academy of Sciences. He is also a guest professor at Tsinghua University and the Graduate University of the Chinese Academy of Sciences. His research interests include machine translation, document classification, and human-computer dialogue systems. He received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 1998. He is a director of the Chinese Association of Artificial Intelligence and the Society of Chinese Information Processing, and is an executive member of the Asian Federation of Natural Language Processing.