Multi-domain Sentiment Classification

Shoushan Li and **Chengqing Zong**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{sshanli,cqzong}@nlpr.ia.ac.cn

Abstract

This paper addresses a new task in sentiment classification, called multi-domain sentiment classification, that aims to improve performance through fusing training data from multiple domains. To achieve this, we propose two approaches of fusion, feature-level and classifier-level, to use training data from multiple domains simultaneously. Experimental studies show that multi-domain sentiment classification using the classifier-level approach performs much better than single domain classification (using the training data individually).

1 Introduction

Sentiment classification is a special task of text categorization that aims to classify documents according to their opinion of, or sentiment toward a given subject (e.g., if an opinion is supported or not) (Pang et al., 2002). This task has created a considerable interest due to its wide applications.

Sentiment classification is a very domain-specific problem; training a classifier using the data from one domain may fail when testing against data from another. As a result, real application systems usually require some labeled data from multiple domains, guaranteeing an acceptable performance for different domains. However, each domain has a very limited amount of training data due to the fact that creating large-scale high-quality labeled corpora is difficult and time-consuming. Given the limited multi-domain training data, an interesting task arises, how to best make full use of all training data to improve sentiment classification performance. We name this new task, 'multi-domain sentiment classification'.

In this paper, we propose two approaches to multi-domain sentiment classification. In the first, called feature-level fusion, we combine the feature sets from all the domains into one feature set. Using the unified feature set, we train a classifier using all the training data regardless of domain. In the second approach, classifier-level fusion, we train a base classifier using the training data from each domain and then apply combination methods to combine the base classifiers.

2 Related Work

Sentiment classification has become a hot topic since the publication work that discusses classification of movie reviews by Pang et al. (2002). This was followed by a great many studies into sentiment classification focusing on many domains besides that of movie.

Research into sentiment classification over multiple domains remains sparse. It is worth noting that Blitzer et al. (2007) deal with the domain adaptation problem for sentiment classification where labeled data from one domain is used to train a classifier for classifying data from a different domain. Our work focuses on the problem of how to make multiple domains 'help each other' when all contain some labeled samples. These two problems are both important for real applications of sentiment classification.

3 Our Approaches

3.1 Problem Statement

In a standard supervised classification problem, we seek a predictor f (also called a classifier) that

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 257–260,
Columbus, Ohio, USA, June 2008.

maps an input vector x to the corresponding class label y. The predictor is trained on a finite set of labeled examples $\{ (X_i, Y_i) \}$ (i=1,…,n) and its objective is to minimize expected error, i.e.,

$$\hat{f} = \arg\min_{f \in \mathbf{H}} \sum_{i}^{n} L(f(X_i), Y_i)$$

Where L is a prescribed loss function and **H** is a set of functions called the hypothesis space, which consists of functions from x to y. In sentiment classification, the input vector of one document is constructed from weights of terms. The terms $(t_1,...,t_N)$ are possibly words, word n-grams, or even phrases extracted from the training data, with N being the number of terms. The output label y has a value of 1 or -1 representing a positive or negative sentiment classification.

In multi-domain classification, m different domains are indexed by k={1,…,m}, each with n_k training samples (X_{i_k}, Y_{i_k}) $i_k = \{1,...,n_k\}$. A straightforward approach is to train a predictor f_k for the k-th domain only using the training data $\{(X_{i_k}, Y_{i_k})\}$. We call this approach single domain classification and show its architecture in Figure 1.

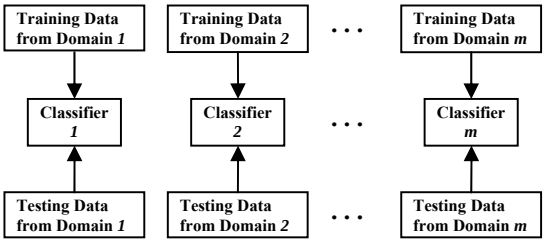

Figure 1: The architecture of single domain classification.

3.2 Feature-level Fusion Approach

Although terms are extracted from multiple domains, some occur in all domains and convey the same sentiment (this can be called global sentiment information). For example, some terms like 'excellent' and 'perfect' express positive sentiment information independent of domain. To learn the global sentiment information more correctly, we can pool the training data from all domains for training. Our first approach is using a common set of terms $(t'_1,...,t'_{N_{all}})$ to construct a uniform feature vector x' and then train a predictor using all training data:

$$\widehat{f_{all}} = \arg\min_{f_{all} \in \mathbf{H}_{all}} \sum_{k=1}^{m} \sum_{i_k=1}^{n_k} L(f(X'_{i_k}), Y_{i_k})$$

We call this approach feature-level fusion and show its architecture in Figure 2. The common set of terms is the union of the term sets from multiple domains.

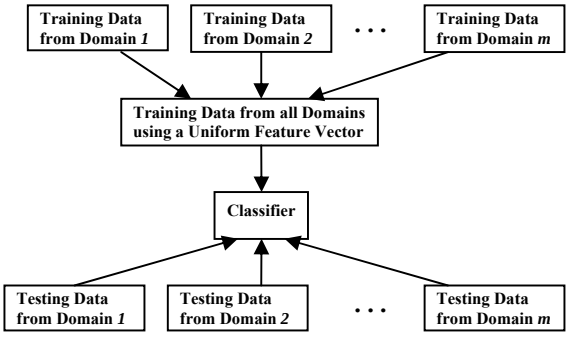

Figure 2: The architecture of the feature-level fusion approach

Feature-level fusion approach is simple to implement and needs no extra labeled data. Note that training data from different domains contribute differently to the learning process for a specific domain. For example, given data from three domains, books, DVDs and kitchen, we decide to train a classifier for classifying reviews from books. As the training data from DVDs is much more similar to books than that from kitchen (Blitzer et al., 2007), we should give the data from DVDs a higher weight. Unfortunately, the feature-level fusion approach lacks the capacity to do this. A more qualified approach is required to deal with the differences among the classification abilities of training data from different domains.

3.3 Classifier-level Fusion Approach

As mentioned in sub-Section 2.1, single domain classification is used to train a single classifier for each domain using the training data in the corresponding domain. As all these single classifiers aim to determine the sentiment orientation of a document, a single classifier can certainly be used to classify documents from other domains. Given multiple single classifiers, our second approach is to combine them to be a multiple classifier system for sentiment classification. We call this approach classifier-level fusion and show its architecture in Figure 3. This approach consists of two main steps:

(1) train multiple base classifiers (2) combine the base classifiers. In the first step, the base classifiers are multiple single classifiers f_k (k=1,…,m) from all domains. In the second step, many combination methods can be applied to combine the base classifiers. A well-known method called meta-learning (ML) has been shown to be very effective (Vilalta and Drissi, 2002). The key idea behind this method is to train a meta-classifier with input attributes that are the output of the base classifiers.

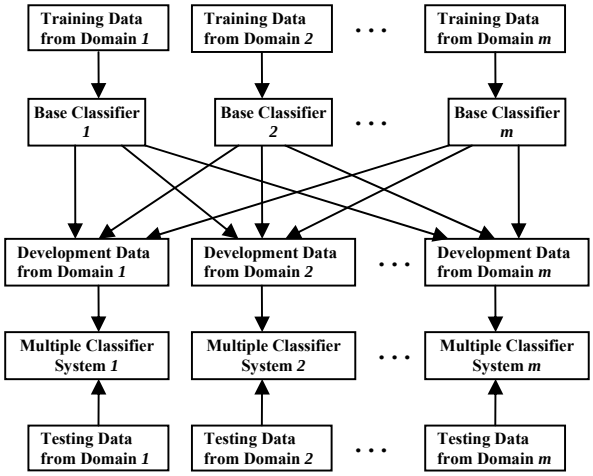

Figure 3: The architecture of the classifier-level fusion approach

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

$$p_k(X_{k'}) = < p_k(c_1 \mid X_{k'}), ..., p_k(c_n \mid X_{k'}) >$$

For the k'-th domain, we train a meta-classifier $f_{k'}$ ($k'=1,...,m$) using the development data from the k'-th domain with the meta-level feature vector $X_{k'}^{meta} \in R^{m \cdot n}$

$$X_{k'}^{meta} = < p_1(X_{k'}), ..., p_k(X_{k'}), ..., p_m(X_{k'}) >$$

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

Different from the feature-level approach, the classifier-level approach treats the training data from different domains individually and thus has the ability to take the differences in classification abilities into account.

4 Experiments

Data Set: We carry out our experiments on the labeled product reviews from four domains: books, DVDs, electronics, and kitchen appliances[1]. Each domain contains 1,000 positive and 1,000 negative reviews.

Experiment Implementation: We apply SVM algorithm to construct our classifiers which has been shown to perform better than many other classification algorithms (Pang et al., 2002). Here, we use LIBSVM[2] with a linear kernel function for training and testing. In our experiments, the data in each domain 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 meta-classifier.

Baseline: The baseline uses the single domain classification approach mentioned in sub-Section 2.1. We test four different feature sets to construct our feature vector. First, we use unigrams (e.g., 'happy') as features and perform the standard feature selection process to find the optimal feature set of unigrams (1Gram). The selection method is Bi-Normal Separation (BNS) that is reported to be excellent in many text categorization tasks (Forman, 2003). The criterion of the optimization is to find the set of unigrams with the best performance on the development data through selecting the features with high BNS scores. Then, we get the optimal word bi-gram (e.g., 'very happy') (2Gram) and mixed feature set (1+2Gram) in the same way. The fourth feature set (1Gram+2Gram) also consists of unigrams and bi-grams just like the third one. The difference between them lies in their selection strategy. The third feature set is obtained through selecting the unigrams and bi-grams with high BNS scores while the fourth one is obtained through simply uniting the two optimal sets of 1Gram and 2Gram.

From Table 1, we see that 1Gram+2Gram features perform much better than other types of features, which implies that we need to select good unigram and bi-gram features separately before combine them. Although the size of our training data are smaller than that reported in Blitzer et al.

[1] This data set is collected by Blitzer et al. (2007): http://www.seas.upenn.edu/~mdredze/datasets/sentiment/

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

(2007) (70% vs. 80%), the classification performance is comparative to theirs.

Features	Books	DVDs	Electronic	Kitchen
1Gram	0.75	0.84	0.8	0.825
2Gram	0.75	0.73	0.815	0.785
1+2Gram	0.765	0.81	0.825	0.80
1Gram+2Gram	**0.79**	**0.845**	**0.85**	**0.845**

Table 1: Accuracy results on the testing data of single domain classification using different feature sets.

We implement the fusion using 1+2Gram and 1Gram+2Gram respectively. From Figure 4, we see that both the two fusion approaches generally outperform single domain classification when using 1+2Gram features. They increase the average accuracy from 0.8 to 0.82375 and 0.83875, a significant relative error reduction of 11.87% and 19.38% over baseline.

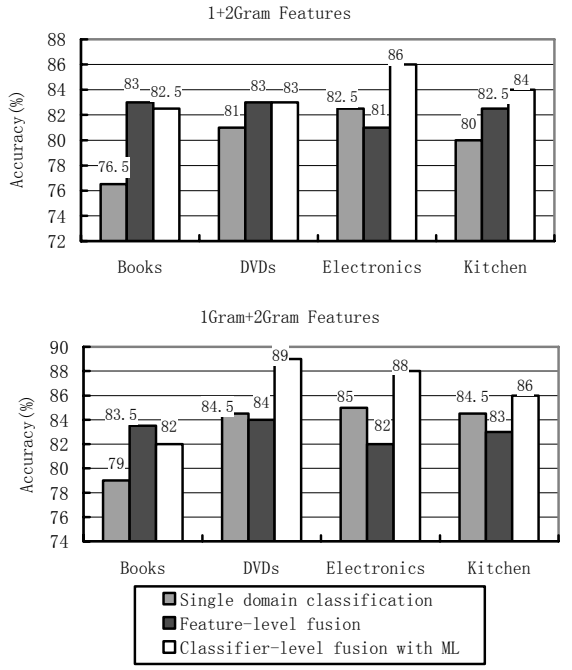

Figure 4: Accuracy results on the testing data using multi-domain classification with different approaches.

However, when the performance of baseline increases, the feature level approach fails to help the performance improvement in three domains. This is mainly because the base classifiers perform extremely unbalanced on the testing data of these domains. For example, the four base classifiers from Books, DVDs, Electronics, and Kitchen achieve the accuracies of 0.675, 0.62, 0.85, and 0.79 on the testing data from Electronics respectively. Dealing with such an unbalanced performance, we definitely need to put enough high weight on the training data from Electronics. However, the feature-level fusion approach simply pools all training data from different domains and treats them equally. Thus it can not capture the unbalanced information. In contrast, meta-learning is able to learn the unbalance automatically through training the meta-classifier using the development data. Therefore, it can still increase the average accuracy from 0.8325 to 0.8625, an impressive relative error reduction of 17.91% over baseline.

5 Conclusion

In this paper, we propose two approaches to multi-domain classification task on sentiment classification. Empirical studies show that the classifier-level approach generally outperforms the feature approach. Compared to single domain classification, multi-domain classification with the classifier-level approach can consistently achieve much better results.

Acknowledgments

The research work described in this paper has been partially supported by the Natural Science Foundation of China under Grant No. 60575043, and 60121302, National High-Tech Research and Development Program of China under Grant No. 2006AA01Z194, National Key Technologies R&D Program of China under Grant No. 2006BAH03B02, and Nokia (China) Co. Ltd as well.

References

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*.

G. Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3: 1533-7928.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*.

R. Vilalta and Y. Drissi. 2002. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2): 77–95.