Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis

Songbo Tan[1], Xueqi Cheng[1], Yuefen Wang[2], and Hongbo Xu[1]

[1] Key Laboratory of Network, Institute of Computing Technology, China
[2] Information Center, Chinese Academy of Geological Sciences, China
tansongbo@software.ict.ac.cn, tansongbo@gmail.com

Abstract. In the community of sentiment analysis, supervised learning techniques have been shown to perform very well. When transferred to another domain, however, a supervised sentiment classifier often performs extremely bad. This is so-called domain-transfer problem. In this work, we attempt to attack this problem by making the maximum use of both the old-domain data and the unlabeled new-domain data. To leverage knowledge from the old-domain data, we proposed an effective measure, i.e., Frequently Co-occurring Entropy (FCE), to pick out generalizable features that occur frequently in both domains and have similar occurring probability. To gain knowledge from the new-domain data, we proposed Adapted Naïve Bayes (ANB), a weighted transfer version of Naive Bayes Classifier. The experimental results indicate that proposed approach could improve the performance of base classifier dramatically, and even provide much better performance than the transfer-learning baseline, i.e. the Naïve Bayes Transfer Classifier (NTBC).

Keywords: Sentiment Classification, Opinion Mining, Information Retrieval.

1 Introduction

Recent years have seen a rapid growth in non-topical text analysis, in which characterizations are sought of the opinions, feelings, and attitudes expressed in a text, rather than just the subjects [1]. A key problem in this area is sentiment classification [2-4][13], where a document is labeled as a positive or negative evaluation of a target object (film, book, product, etc.).

In most cases, the use of statistical or machine learning techniques has proven to be successful in this context, such as Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM) [2-4]. Pang's researches [2] indicate that standard machine learning methods perform very well, even definitively outperform human classifiers.

When transferred to another domain, however, a sentiment classifier often performs extremely bad. This is so-called domain-transfer problem [3-4]. A simple and intuitive explanation is that sentiment is often expressed differently in different domains. For example, "rise" or "rebound" is often used to express positive sentiment for stock review; while "luxury" or "classical" is often employed to convey positive sentiment for house review.

M. Boughanem et al. (Eds.): ECIR 2009, LNCS 5478, pp. 337–349, 2009.

From the perspective of machine learning, the reason can be explained as: some high-frequency domain-specific features that have high correlations with certain class labels in the old domain, do not have high correlations with the same class labels any more in the new domain, and vice versa [10]. In another word, a few high-frequency domain-specific features carry classification knowledge for the old domain but not carry any classification knowledge for the new domain.

On the other hand, although the old-domain data is relatively out-of-date, there are still some portions of the data that can be used. For example, the word “good” or “excellent” occurs frequently in stock review as well as in house review. In the rest of this paper, we call this kind of features as “generalizable features” [10].

With this inspiration, we attempt to attack domain-transfer problem by leveraging useful knowledge from both the old-domain data and the unlabeled new-domain data. To acquire classification knowledge from the old-domain data, we proposed an effective measure, i.e., Frequently Co-occurring Entropy (FCE), to pick out generalizable features that occur frequently in both domains and have similar occurring probability. When training the classifier on the old domain, we only use the generalizable features, which leads to a generalizable classifier: it may perform a little bad in the old domain but will perform acceptably well in the new domain. In another word, the generalizable features serve as a bridge, which builds a road from one domain to another domain.

To gain knowledge from the new-domain data, we proposed Adapted Naïve Bayes (ANB), a weighted transfer version of Naive Bayes Classifier. ANB employs a weighted Expectation-Maximization (EM) algorithm to train a transfer model using the old-domain data as well as the new-domain data. The main difference from the traditional EM-based classifier is that, with the iteration, ANB gradually enlarges the weight for the new-domain data while decreases the weight for the old-domain data, and in the same time uses all features for the new-domain data while only uses generalizable features for the old-domain data. In a word, we use the old-domain data corresponding to generalizable features as a bridge, and adapt the classifier model to the new-domain data step by step. Therefore, our method could be regarded as an extension of traditional EM-based Naïve Bayes Classifier [5].

To investigate the effectiveness and robustness of proposed approach, we conduct an extensive experiment on three Chinese domain-specific tasks, including education reviews, stock reviews and computer reviews. The experimental results indicate that proposed approach could improve the performance of base classifier dramatically, and even provide much better performance than the transfer-learning baseline, i.e. the Naïve Bayes Transfer Classifier (NTBC) [11].

2 Related Work

In practical text categorization, labeled documents are often very sparse while there are often abundant unlabeled documents. As a result, exploiting these unlabeled data has become an active research problem in text classification recently.

Nigam et al. [5] introduced an EM-like approach that combines Expectation Maximization (EM) algorithm with Naive Bayes classifier. The result of combining these two is an algorithm that extends conventional text learning algorithms by using EM to dynamically derive pseudo labels for unlabeled documents during learning thereby providing a way to incorporate unlabeled data into supervised learning.

Joachims [6] modified SVM to exploit the unlabeled data (often called TSVM). TSVM expects to find a low-density area of data and constructs a linear separator in this area so that the margin over both the labeled data and the unlabeled data can be maximized.

Recently, transfer learning has been recognized as an important topic in machine learning research. Several researchers have proposed new approaches to solve the problem of transfer learning [9-11].

DaumeIII and Marcu [9] studied the domain-transfer problem in statistical natural language processing. They consider the common case in which labeled out-of-domain data is plentiful, but labeled in-domain data is scarce. Then they introduce a statistical formulation of this problem in terms of a simple mixture model and present an instantiation of this framework to Maximum Entropy classifiers and their linear chain counterparts.

Wenyuan Dai [11] proposed an EM-based Naive Bayes classifier for domain-transfer problem. In order to transfer the model from one old domain to another new domain, he used KL-divergence to decide the trade-off parameters between the old-domain data and the new-domain data. In fact, KL-divergence only serves as a constant parameter that impacts a much larger weight on the new-domain data when estimating the class probability and word probability. Therefore, his method does not consider the adverse influence of high-frequency domain-specific features.

3 Methodology

In this section, we present our transfer approach. In proposed approach, we first use Frequently Co-occurring Entropy (FCE) to pick out generalizable features that occur frequently in both domains and have similar occurring probability, then employ Adapted Naïve Bayes (ANB) to train a classification model for the new domain.

3.1 Frequently Co-occurring Entropy

Generally speaking, the features of one domain can be divided into two categories: domain-specific and nondomain-specific. The sentiment features occurs in both categories,

$$F = \begin{cases} \textit{domain-specific} & \begin{cases} \textit{sentiment (ds)} \\ \textit{nonsentiment (dns)} \end{cases} \\ \textit{nondomain-specific} & \begin{cases} \textit{sentiment (nds)} \\ \textit{nonsentiment (ndns)} \end{cases} \end{cases} .$$

Our goal is to employ domain-specific sentiment features (*ds*) as well as nondomain-specific sentiment features (*nds*) to train a sentiment classifier for one given domain. Unfortunately, in the new domain, it is often a difficult job to obtain a large amount of labeled examples for training. The only feasible way is to make use of one old-domain labeled data. In this time the useful knowledge is the data corresponding to *nds* features, because the *ds* features is not appropriate for the new-domain data. As a result a good idea is to pick out nondomain-specific features. In another word, the nondomain-specific features serve as a bridge from the old domain to the new domain. When we arrive the new domain, we can train a new classifier, using nondomain-specific features as well as domain-specific features. For the sake of convenience, we call nondomain-specific features as “generalizable features”.

In order to pick out generalizable features, we proposed Frequently Co-occurring Entropy (FCE). Our measure includes two criteria: a) occur frequently in both domains; b) has similar occurring probability. To satisfy these requirements, we proposed the following formula:

$$f_w = \log\left(\frac{P_o(w)\cdot P_n(w)}{\left|P_o(w)-P_n(w)\right|}\right) \tag{1}$$

where $P_o(w)$ and $P_n(w)$ indicate the probability of word w in the old domain and the new domain respectively:

$$P_o(w) = \frac{\left(N_w^o + \alpha\right)}{\left(\left|D^o\right| + 2\cdot\alpha\right)} \tag{2}$$

$$P_n(w) = \frac{\left(N_w^n + \alpha\right)}{\left(\left|D^n\right| + 2\cdot\alpha\right)} \tag{3}$$

where $N^o{}_w$ and $N^n{}_w$ is the number of examples with word w in the old domain and the new domain respectively; $|D^o|$ and $|D^n|$ is the number of examples in the old domain and the new domain respectively. In order to overcome overflow, we set α=0.0001 in our experiment reported in section 4.

Table 1. A simple example for FCE measure

No.	$N_o(w)$	$N_n(w)$	*FCE*	
			Score	*Rank*
1	**100**	**100**	4.6052	**1**
2	**100**	**90**	-0.1153	3
3	100	45	-2.5051	6
4	100	4	-5.4817	10
5	**50**	**50**	3.2189	**2**
6	**50**	**45**	-0.8183	4
7	50	23	-3.1598	7
8	50	2	-6.1759	12
9	4	4	-1.8326	**5**
10	4	3	-4.5182	8
11	4	2	-5.5703	11
12	1	1	-4.6052	**9**

In the extreme case when $P_o(w)$=$P_n(w)$, above formula does not work. As a result, we modify the formula as following,

$$f_w = \log\left(\frac{P_o(w)\cdot P_n(w)}{\left|P_o(w)-P_n(w)\right|+\beta}\right) \tag{4}$$

where β is set as 0.0001 in our experiment.

In above formula, "$P_o(w)P_n(w)$" embodies the first requirement, and "$|P_o(w)$-$P_n(w)|$" embodies the second requirement. Obviously, the two factors have the same weight in calculating the FCE score. For the sake of being easy to adjust the strength of the two factors, we introduce a trade-off parameter π,

$$f_w = \log\left(\frac{\left(P_o(w)\cdot P_n(w)\right)^{\pi}}{\left(\left|P_o(w)-P_n(w)\right|+\beta\right)^{(1-\pi)}}\right) \tag{5}$$

From above formula, if π>0.5, it indicates that "$P_o(w)P_n(w)$" plays a more important role in calculating the FCE score and vice versa.

To better understand this measure, let's take a simple example (see Table 1). Given an old-domain dataset with 1000 documents and a new-domain dataset with 1000 documents, 12 candidate features, and a task to pick out 4 most generalizable features. According to our understanding, the best choice is to pick out w_1, w_2, w_5, and w_6.

According to formula (4), fortunately, we successfully pick out w_1, w_2, w_5, and w_6. This simple example validates the effectiveness of proposed FCE formula.

With the aim to further validate the effectiveness of FCE measure, we provide the top 40 generalizable features between stock reviews and computer reviews (See Table 2).

Table 2. The top 40 generalizable features between stock reviews and computer reviews

不错(not bad)	寒(cold)	慢(slow)	吸引(attractive)
不好(not good)	好(good)	美(beautiful)	虚(void)
不及(inferior to)	好处(benefit)	难(difficult)	庸(stodgy)
不利于(adverse)	喜(pleased)	难关(challenge)	优(excellent)
不明(unclear)	特色(characteristic)	能够(can)	优势(superority)
不能(cannot)	困难(difficulty)	骗(deceive)	优秀(excellence)
不足(deficient)	快(fast)	伤(injure)	有效(effective)
差(bad)	快速(quick)	失(lose)	正常(natural)
独立(independent)	垄断(monopolize)	凸显(outstanding)	正面(positive)
精彩(wonderful)	落后(behindhand)	无法(incapable)	阻(obstruct)

3.2 Naïve Bayes Classifier

The Naïve Bayes algorithm is a widely used algorithm for document classification. There are two commonly used models (i.e., multinomial model and multi-variate Bernoulli model) for text categorization. Without loss of generality, in this paper, we run the multinomial model adopted by numerous authors [12].

In the multinomial model, a document is an ordered sequence of word events, drawn from the same vocabulary V. We assume that the lengths of documents are independent of class. We then make a naive Bayes assumption: that the probability of each word event in a document is independent of the word's context and position in the document. Thus, each document d_i is drawn from a multinomial distribution words with as many independent trials as the length d_i. This yields the familiar "bag of words" representation for documents.

We can estimate the probability of a word given its class as following,

$$P(w_t|c_k) = \frac{\sum_{i=1}^{|D|} N_{t,i} \cdot P(c_k|d_i) + 1}{\sum_{t=1}^{|V|}\sum_{i=1}^{|D|} N_{t,i} \cdot P(c_k|d_i) + |V|} \tag{6}$$

where $N_{t,i}$ is the number of appearances of word w_t in document d_i, $|V|$ and $|D|$ refer to the vocabulary size and the dataset size respectively.

The class prior probability can be estimated as the Maximum Likelihood Estimate:

$$P(c_k) = \frac{\sum_{i=1}^{|D|} P(c_k|d_i)}{|D|} \tag{7}$$

As a result the conditional probability $P(c_k|d_i)$ can be estimated as following,

$$P(c_k|d_i) \propto P(c_k) \prod_{t \in |V|} \left(P(w_t|c_k)\right)^{N_{t,i}} \tag{8}$$

At last, we can write down the Bayes' rule for classification decision as following,

$$c = \arg\max_{c_k} \left(P(c_k) \prod_{t \in |V|} \left(P(w_t|c_k)\right)^{N_{t,i}} \right) \tag{9}$$

3.3 Adapted Naïve Bayes Classifier

In this section, we adapt EM-based Naïve Bayes (EMNB) [5] for transfer learning. In principle, EMNB requires both labeled data and unlabeled data obey the same distribution: all the data is produced by a mixture model; and there is a one-to-one correspondence between generative mixture components and classes.

Obviously, our transfer-learning setting does not satisfy this requirement. Fortunately, however, the problem can be solved, if we use FCE measure to pick out some generalizable features and only use these features to initialize a Naïve Bayes model for EM iteration.

Another problem arises: only using generalizable features is not enough to accurately predict the examples in the new domain. To solve this problem, we proposed a new weighted EMNB classifier: with the iteration it gradually enlarges the weight for the new-domain data while decreases the weight for the old-domain data, and in the same time use all features for the new-domain data which can greatly enhance the prediction ability of classifier for the new domain.

Note that because old-domain specific features don't have any useful information for the new-domain classifier, only generalizable features are used for the old-domain data throughout the iteration.

The EM algorithm finds a local maximum likelihood parameterization for more data: both the old-domain and the new-domain data. But in transfer-learning setting, our goal is to find a local maximum likelihood parameterization only for the new-domain data. For the sake of being easy to understand, we first give its log likelihood of the parameters as following,

$$l(\theta|D,z)=\sum_{i=1}^{|D|}\sum_{k=1}^{|C|} z_{ik}\log\left(P(c_k|\theta)P(d_i|c_k;\theta_k)\right) \tag{10}$$

where θ_k indicates the parameter of mixture model for class c_k; D includes the old-domain and the new-domain data; C includes the class labels ("0" for positive and "1" for negative); z_i is a binary vector, z_i=<z_{i1}, z_{i2}, …, $z_{i|C|}$> where z_{ik}=1 iff y_i=k else z_{ik}=0.

EM algorithm iterates two steps to find a local maximum parameterization for $l(\theta|D)$,i.e., E-step and M-step. The E-step corresponds to calculating the conditional probability $P(c_k|d_i)$ for every example by using current estimate θ; the M-step corresponds to computing a new maximum likelihood estimate for θ using current estimates for examples labels, i.e., $P(c_k|d_i)$.

In the following we give the detailed formulas under transfer-learning setting using the adapted E-step and M-step.

E-step:

$$P(c_k|d_i)\propto P(c_k)\prod_{t\in|V|}\left(P(w_t|c_k)\right)^{N_{t,i}} \tag{11}$$

M-step:

$$P(c_k)=\frac{(1-\lambda)\cdot\sum_{i\in D^o}P(c_k|d_i)+\lambda\cdot\sum_{i\in D^n}P(c_k|d_i)}{(1-\lambda)\cdot\left|D^o\right|+\lambda\cdot\left|D^n\right|} \tag{12}$$

$$P(w_t|c_k)=\frac{(1-\lambda)\cdot\left(\eta_t^o\cdot N_{t,k}^o\right)+\lambda\cdot\left(N_{t,k}^n\right)+1}{(1-\lambda)\cdot\sum_{t=1}^{|V|}\left(\eta_t^o\cdot N_{t,k}^o\right)+\lambda\cdot\sum_{i=1}^{|V|}\left(N_{t,k}^n\right)+|V|} \tag{13}$$

$$N_{t,k}^o=\sum_{i\in D^o}\left(N_{t,i}^o\cdot P(c_k|d_i)\right) \tag{14}$$

$$N_{t,k}^n=\sum_{i\in D^n}\left(N_{t,i}^n\cdot P(c_k|d_i)\right) \tag{15}$$

where $N^o_{t,k}$ and $N^n_{t,k}$ is the number of appearances of word w_t in old-domain and new-domain class c_k respectively; λ is a parameter that controls the impact between the old-domain and the new-domain data. It changes with the iteration as following manner,

$$\lambda=\min\{\delta\cdot\tau,1\} \tag{16}$$

where τ indicates the iteration step, $\tau\in\{1, 2, ,,, \}$, and δ is a constant which controls the strength to update the parameter λ, $\delta\in(0,1)$. η_t^o is a constant,

$$\eta_t^o=\begin{cases}0 & if \quad w_t\notin V_{FCE}\\ 1 & if \quad w_t\in V_{FCE}\end{cases} \tag{17}$$

```
1 Load the old-domain data ($D^o$), the new-domain unlabeled data ($D^n$), and parameters, $\lambda$, $\delta$, $\eta_t^o$, $\eta_t^n$;
2 Pick out $N_{FCE}$ generalizable features using formula (4);
3 Calculate $P(w_t|c_k)$ and $P(c_k)$ over the old-domain data using formula (6-7)
4 Iterate following two steps until convergence;
  4.1 E-step: Calculate $P(c_k|d_i)$ for $D^n$ using formula (11);
  4.2 M-step: Calculate $P(w_t|c_k)$ and $P(c_k)$ for $D^o+D^n$ using formula (12-13);
5 Use the new classifier to classify examples on the new
```

Fig. 1. The outline of Adapted Naïve Bayes Algorithm

where V_{FCE} includes the generalizable features. The above formulas (13)(17) indicate that for the old-domain data, throughout all iteration, only generalizable features are used; while for the new-domain data, all features including domain-specific features are used.

Detailed description for proposed method is outlined in Figure 1. First we pick our N_{FCE} generalizable features using formula (4), then based on these generalizable features, calculate $P(w_t|c_k)$ and $P(c_k)$ over the old-domain data using formula (6-7). In substance, the generalizable features serve as a bridge that builds a road from one old domain to another new domain.

Since the labels of the old-domain data are given at first, what we only need to do is to estimate the label of the new-domain examples, or the probability $P(c_k|d_i)$. Step 4 can do this job: it iterates E-step and M-step until convergence. With the iteration, the parameter λ keeps being enlarged. Obviously, this strategy attempts to find a local maximum likelihood parameterization for the new-domain data.

4 Experimental Results

4.1 Datasets

To validate the effectiveness and robustness of proposed method, we collected three Chinese domain-specific datasets: Education Reviews (Edu, from http://blog.sohu.com/learning/), Stock Reviews (Sto, from http://blog.sohu.com/stock/) and Computer Reviews (Comp, from http://detail.zol.com.cn/). All of these datasets are annotated by three linguists. We use Chinese text POS tool ICTCLAS [8] to parse and tag these Chinese reviews.

Education Reviews. There are 1,012 negative reviews and 254 positive reviews in this corpus. The average size of reviews is about 600 words, and the cardinality of vocabulary is 19,150.

Stock Reviews. This collection consists of 683 negative reviews and 364 positive reviews. The average length of reviews is about 460 terms and the different terms amount to 12,674.

Computer Reviews. This dataset contains 390 negative reviews and 544 positive reviews about computer. The average length of reviews is about 120 words. This dataset comprises a very small vocabulary-only 4,725 different words.

4.2 Comparison Methods

In our experiments, we run one supervised baseline, i.e., Naïve Bayes (NB) [12], which only uses one old-domain labeled data as training data.

For semi-supervised learning baseline, we use EM-based Naïve Bayes classifier (EMNB) [5]. It makes use of the old-domain labeled data as well as the given-domain unlabeled data.

For transfer-learning baseline, we use Naive Bayes Transfer Classifier (NBTC) [11]. Like EMNB, it makes use of the old-domain labeled data as well as the given-domain unlabeled data.

4.3 Does Proposed Approach Work?

To evaluate a sentiment classification system, we use Micro and Macro F1 measure [7], which emphasize the performance of the system on common and rare categories respectively. To conduct our experiments, we use the old-domain data as labeled training set, and use the new-domain data as unlabeled set and testing set. Table 3 shows the results of experiments comparing proposed approach with supervised learning, semi-supervised learning and transfer learning. For proposed approach, the N_{FCE} is set to 500, and δ is set to 0.2.

As expected, proposed approach does indeed improve the performance of base classifier dramatically, as well as provide much better performance than semi-supervised and transfer-learning techniques. For example, the averaged MicroF1 of proposed approach beats the base classifier by about 22 percents, beats other classifiers by at least 10 percents. This trend is even more pronounced for averaged MacroF1. Accordingly, the result is very encouraging and of enormous value in sentiment-analysis applications that require high-precision classification but hardly have any labeled training data.

EM-like method indeed boosts the base classifier on two domain-transfer problems, i.e., “Sto->Edu” and “Sto->Comp”. However, on other problems it degrades the performance of base classifier. This seems to be counter-intuition as the EM-like method is trained on the old-domain examples as well as on the new-domain examples while the Naïve Bayes baseline only uses the old-domain examples.

In substance, the EM algorithm requires an assumption: both the old-domain data and the new-domain data should share the same generative model. However, in transfer-learning setting, this assumption does not hold. This is the reason why the EM-like method is much worse than the Naïve Bayes method in other four cases, such as “Edu->sto”, “Edu->Comp”, “Comp->sto”, and “Comp->Edu”.

NBTC [11] attempts to learn a more compatible model for the new-domain data. Its main idea is to use KL-divergence to design a larger constant weight for the new-domain data when training the classifier model.

As can be observed from Table 3, NBTC indeed produces better results than EMNB. But this improvement is very limited. This result indicates that NBTC does not essentially solve the problem, i.e., word-distribution difference between the old domain and the new domain. In sentiment-transfer community, the distribution difference is often so large that most of the predicted labels for the new domain are wrong. Keep this fact in mind, larger weight for the new-domain data may degrade the performance rather than upgrade the performance.

Table 3. The performance of different methods

	NB		EMNB		NBTC		Proposed Approach	
	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1
Edu->Sto	0.6704	0.4553	0.6628	0.4266	0.6743	0.4659	0.7669	0.7109
Edu->Comp	0.5085	0.4696	0.4175	0.3118	0.6059	0.5918	0.8854	0.8814
Sto->Edu	0.6824	0.5867	0.6962	0.6056	0.8303	0.8080	0.9171	0.9119
Sto->Comp	0.5053	0.5025	0.5192	0.5169	0.5128	0.5103	0.7901	0.7652
Comp->Sto	0.6580	0.4148	0.6552	0.4036	0.6580	0.4148	0.6962	0.5942
Comp->Edu	0.6114	0.4105	0.6074	0.4003	0.6114	0.4105	0.9013	0.8920
Average	0.6060	0.4732	0.5930	0.4441	0.6488	0.5336	0.8262	0.7926

4.4 How Many Generalizable Features Are Sufficient to Train a Transfer Model?

Generalizable features play a very important role in proposed approach; namely, they build a bridge from one old domain to another new domain. So our question is how many generalizable features are sufficient enough to transfer a sentiment classifier to another domain. We vary the number of generalizable features from 50 to 1200, run the proposed approach on six transfer tasks, and finally draw two curves of the averaged performance on the six tasks (Figure 2). Note that for proposed approach, δ is set to 0.2.

As can be observed from Figure 2, with the increase of the number of generalizable features, the performance first grows and then descends. Between 400 and 600, proposed approach shows a robust and excellent performance. This observation validates

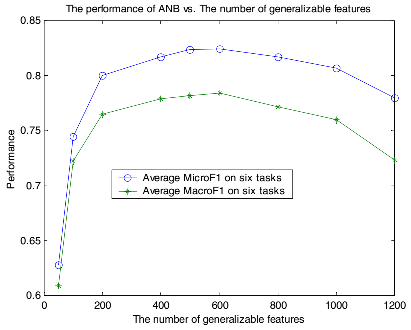

Fig. 2. The performance curves of ANB vs. the number of generalizable features

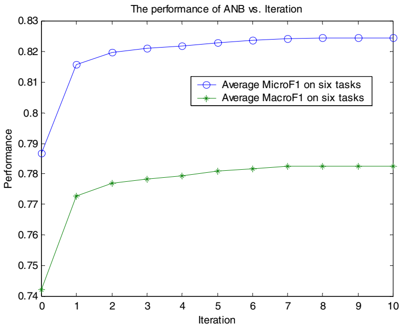

Fig. 3. The performance curves of ANB vs. Iteration

our intuitive estimation. Obviously, the number of generalizable features is limited. Hence if the predefined number is bigger than the latent number of generalizable features, many domain-specific features are picked out so that the quality of "bridge" is inevitably degraded.

On the contrary, if the predefined number is smaller than the latent number of generalizable features, that is to say, only a part of generalizable features are used to build a "bridge", which is not strong enough for transfer learning.

4.5 How Does the Iteration Step Affect the Performance of Proposed Approach?

To examine whether proposed approach can convergence to its local maximal point, we run it on six transfer tasks, and draw two curves of the averaged performance (Figure 3). Note that for proposed approach, the N_{FCE} is set to 500, and δ is set to 0.2.

It can be seen that proposed approach always converges at or close to the local maximal point and the convergence speed is very fast.

When the Iteration is equal to 0, propose approach is simplified as a traditional Naïve Bayes classifier which only uses knowledge from the old-domain data with respective to generalizable features. From this figure, we can observe that the first round provides the maximal improvement for Naïve Bayes classifier. This observation indicates that combining the old-domain data and the new-domain data can greatly boost the performance of transfer learning.

4.6 How Does the Parameter λ Affect the Accuracy of Proposed Approach?

The labels of the old-domain data are annotated by linguists so the old-domain data is more accurate than the new-domain data but it contains too many domain-specific features; on the contrary, the new-domain data has a lot of useful information for classification but its labels are unknown. So an intuitive question is how to weight the examples from the old domain and the new domain respectively.

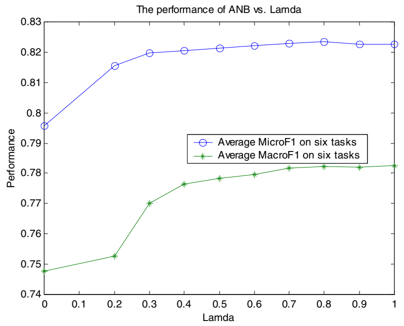

Fig. 4. The performance curves of ANB vs. Lamda (λ)

In this section we attempt to answer this question. We vary the parameter λ from 0.0 to 1.0, run the proposed approach on six transfer tasks, and finally draw two curves of the averaged performance on the six tasks (Figure 4). Note that for proposed approach, the N_{FCE} is set to 500.

As expected, with the increase of the parameter λ, the performance first climbs up and then levels off. From this figure, we can see that the ideal values of the parameter λ fall into [0.6, 1.0]. This experiment verifies our analysis that both the old-domain data and the new-domain data play a key role in domain adaptation.

Another important observation is that the peak value locates close to 0.8, which indicates that the weight of the new-domain data is about four times larger than the weight of the old-domain data. This observation is also in accordance with our intuition: obviously, the new-domain data should be given much larger weight than the old-domain data because the new-domain data is just the goal of transfer learning. This experiment indicates that proposed ANB algorithm performs much better than traditional Naïve Bayes, in making the maximum use of the old-domain data as well as the new-domain data.

5 Conclusion Remarks

In this paper, we proposed a novel approach for domain adaptation in the context of sentiment analysis. The main contributions are three-folds:

First, in order to make the maximum use of the old-domain data, we proposed an effective method, i.e., Frequently Co-occurring Entropy (FCE), to pick out generalizable features and use these figure as a bridge linking one old domain to another new domain. Throughout the iteration of proposed approach, we only use generalizable features for the old-domain data.

Second, in order to acquire knowledge from the old domain as well as the new domain, we proposed Adapted Naïve Bayes classifier (ANB). Its main idea is to employ a weighted EM algorithm to combine the old-domain data with the new-domain data, and gradually enlarge the weight for the new-domain data while decrease the weight for the old-domain data with the iteration, with the hope to fit the new-domain data as well as possible.

Thirdly, we conducted extensive experiments on six domain adaptation tasks. As expected, proposed approach improves the performance of base classifier dramatically, as well as provides much better performance than semi-supervised and transfer-learning techniques.

Although proposed approach indeed improves the classification accuracy, there is a lot of room for improvement. For example, FCE is not the best strategy to pick out generalizable features; can other classifiers work under this approach? All these questions are waiting for our future efforts.

Acknowledgments

This work was mainly supported by two funds, i.e., 0704021000 and 60803085, and one another project, i.e., 2004CB318109.

References

[1] Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: CIKM (2005)

[2] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: EMNLP 2002 (2002)

[3] Aue, A., Gamon, M.: Customizing Sentiment Classifiers to New Domains: a Case Study. In: RANLP 2005 (2005)

[4] Tan, S., Wu, G., Tang, H., Cheng, X.: A novel scheme for domain-transfer problem in the context of sentiment analysis. In: CIKM 2007 (2007)

[5] Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to classify text from labeled and unlabeled documents. In: AAAI 1998 (1998)

[6] Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML (1999)

[7] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)

[8] Zhang, H.: Chinese Lexical Analysis Using Hierarchical Hidden Markov Model. In: The Second SIGHAN workshop affiliated with 41st ACL (2003)

[9] DaumeIII, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research 26, 101–126 (2006)

[10] Jiang, J., Zhai, C.: A Two-Stage Approach to Domain Adaptation for Statistical Classifiers. In: CIKM 2007 (2007)

[11] Dai, W., Xue, G., Yang, Q., Yu, Y.: Transferring Naive Bayes Classifiers for Text Classification. In: AAAI 2007 (2007)

[12] McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI/ICML Workshop on Learning for Text Categorization (1998)

[13] Wilson, T., Wiebe, J., Hwa, R.: Recognizing Strong and Weak Opinion Clauses. Computational Intelligence 22(2), 73–99 (2006)