

Using Unlabeled Data to Handle Domain-transfer Problem of Semantic Detection

Songbo Tan¹, Yuefen Wang², Gaowei Wu¹ and Xueqi Cheng¹

¹Information Security Center, Institute of Computing Technology, Chinese Academy of Sciences, China

²Information Center, Chinese Academy of Geological Sciences, China

tansongbo@software.ict.ac.cn, tansongbo@gmail.com

ABSTRACT

Due to highly domain-specific nature, supervised sentiment classifiers typically require a large number of new labeled training data when transferred to another domain. This is so-called domain-transfer problem. In this work, we attempt to tackle this problem by combining old-domain labeled examples with new-domain unlabeled ones. The basic idea is to use old-domain-trained classifier to label some informative unlabeled examples in new domain, and train the base classifier again. The experimental results demonstrate that proposed method dramatically boosts the accuracy of the base sentiment classifier on new domain.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval-search process; I.5 [Pattern Recognition]: Applications

General Terms

Algorithms, Performance, Experimentation

Keywords

Sentiment Classification; Opinion Mining; Information Retrieval

1. INTRODUCTION

With the rapid growth of semantic web page (such as product reviews, movie reviews and book reviews) in Internet, the detection and analysis of opinions, feelings, or attitudes expressed in a text has received more and more attention in the community of information retrieval and natural language processing. A key problem in this area is sentiment classification [1][3][4][5][8][10][12][14][15], in which a document is labeled as a positive or negative evaluation of a target object (book, product, etc.). The classification of sentiment constitutes a problem that is orthogonal to the traditional task of text classification: whereas in typical text classification the focus is on topic identification, in sentiment classification the focus is on the assessment of the writer's sentiment toward the topic.

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 Classification (ME), and Support Vector Machines (SVM) [1][3][5][10][12]. Due to highly

domain-specific nature, however, supervised sentiment classifier typically requires a large amount of new labeled training data when moving from one domain to another [1]. As a result, when transferred to another domain without any labeled examples, a sentiment classifier often performs extremely bad. This is so-called domain-transfer problem [1][17].

There are many factors that contribute to this problem. Firstly, the word space changes with the domain. For example, the word "portable" or "faster" occurs frequently in notebook review while hardly occurs in house review. Secondly, the polarity of sentiment words may change with the domain too. For instance, the word "small" in house review may be negative (e.g. "The bedroom is very small."), while in cell-phone review may be positive (e.g. "The Nokia N3100 is so small as to be put in any pockets."). Thirdly, sentiment in different domains can be expressed in very different ways [4]. Such factors may also compound, and the larger the variation, the poorer the classification performance will be.

A simple solution to this problem is to manually label a large number of examples for each new domain. Unfortunately, however, this method is unfeasible in practice because the acquisition of these labeled data would be time-consuming and expensive. Consequently, it is an important and urgent job to investigate an ideal and practicable method for domain-transfer problem.

To the best of our knowledge, no previous work has been conducted on exactly this kind of problem, where there are a large amount of labeled data in old domain but scarcely any labeled data in new domain. The most related research is conducted by Aue and Gamon [1]. But their work still needs to manually label a small amount of training data for a new domain.

In this work, we attempt to tackle domain-transfer problem by combining old-domain labeled examples with new-domain unlabeled ones. The basic idea is to use old-domain-trained classifier ("old classifier" for brevity) to label top n most informative unlabeled examples in new domain and learn a new classifier based on these selected examples (n is a pre-defined number indicating how many examples in new domain shall be picked out as informative ones). Without loss of generality, we employ centroid classifier [6] as the base classifier.

To investigate the effectiveness and robustness of this technique, we conduct an extensive experiment on three domain-specific sentiment corpora, including education reviews, house reviews, and computer reviews. The experimental results indicate that proposed approach can dramatically improve the performance of the base sentiment classifier on new domain.

The rest of this paper is constructed as follows: Next section presents related work. Centroid learning method is described in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

Copyright 2008 ACM 978-1-59593-753-7/08/0003...\$5.00.

section 3. Section 4 presents the proposed method. Experimental results are given in section 5. Section 6 discusses some issues about proposed scheme. Finally section concludes this paper.

2. RELATED WORK

Related work to ours comes from three perspectives: sentiment classification, unlabeled examples aided learning and combination of the former two.

2.1 Sentiment Classification

Most researches about sentiment classification up to this point have focused on training machine learning algorithms to classify reviews.

Pang et al. [12] conducted an extensive experiment on movie reviews using three traditional supervised machine-learning methods (i.e., Naive Bayes, maximum entropy classification, and support vector machines). His results indicate that standard machine learning techniques definitively outperform human-produced baselines. However, he argued that machine learning methods couldn't perform as well on sentiment classification as on traditional topic-based categorization.

Mullen et al. [10] employed support vector machines to bring together diverse sources of potentially pertinent information, including several favorability measures for phrases and adjectives and, where available, knowledge of the topic of the text. Models using the features introduced are further combined with unigram models and lemmatized versions of the unigram models.

In 2006, Cui et al. [3] conducted experiments on large-scale online product reviews with an average length of over 800 bytes crawled from the Web. He employed three algorithms: Passive-Aggressive Algorithm Based Classifier, Language Modeling Based Classifier, and Winnow Classifier. His experimental results show that Passive-Aggressive model significantly outperforms the other two kinds of models.

2.2 Unlabeled Examples Aided Learning

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. [11] 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.

Following this direction, Lanquillon [9] described a general framework for extending any text-learning algorithm to utilize unlabeled documents in addition to labeled document using an Expectation-Maximization-like scheme. In his work, he used three traditional methods, i.e., Naive Bayes classifier, Single Prototype (or Centroid) classifier, and SVM, as base classifier.

Blum et al. [2] proposed Co-Training method that splits the original feature set into two conditional independent feature sets. The algorithm initially trains two classifiers separately based on labeled data, and then each algorithm's predictions on new unlabeled examples are used to enlarge the training set of the other. He also provided a PAC-style framework for the general problem of learning from both labeled and unlabeled data.

Joachims [7] 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.

2.3 Combination of the Former Two

Aue and Gamon [1] combined little amount of labeled data with large amount of unlabeled data to customize supervised classifier for a new domain. They use an algorithm introduced by Nigam et al. [11], which learns from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and a naive Bayes classifier. The algorithm first trains a classifier using the available labeled documents, and probabilistically labels the unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates to convergence.

Nevertheless, if there are large amount of labeled data in old domain but no labeled data in new domain, it does not work in most cases. In fact, this method is based on a severe assumption that the data is generated by a mixture model, and that there is a correspondence between mixture components and classes. When these assumptions are not satisfied, EM may actually degrade rather than improve classifier accuracy. Obviously, as is often the case, the data in old domain and the data in new domain cannot share the same generative model.

3. BASE CLASSIFIER

The idea behind the centroid classification algorithm [6] is extremely simple and straightforward. First we compute the weighted representation of each training document; second, we calculate the prototype vector or centroid vector C_i for each training class c_i ; then, we compute the similarity between a testing document d to all centroids; finally, based on these similarities, we assign d the class label corresponding to the most similar centroid. In the following, we will elaborate these steps in detail.

In this work, the documents are represented using vector space model. In this model, each document d is considered to be a vector in the term-space. For term weight we employ TFIDF [13]:

$$w(t, d) = tf(t, d) \times \log\left(\frac{N}{n_t}\right) \quad (1)$$

where N is the total number of training documents, and n_t is the number of documents containing the word t . $tf(t, d)$ indicates the occurrences of word t in document d .

After giving the representation of documents, centroid can be computed as following:

$$C_i = \frac{1}{|c_i|} \sum_{d \in c_i} d \quad (2)$$

where $|z|$ indicates the cardinality of set z .

Then we calculate the similarity of one document d to each centroid by cosine measure,

$$Sim(d, C_i) = \frac{d \cdot C_i}{\|d\|_2 \|C_i\|_2} \quad (3)$$

where $\|\cdot\|_2$ denotes the 2-norm of one vector, and " \cdot " denotes the dot-product of two vectors.

Lastly, based on these similarities, we assign d the class label corresponding to the most similar centroid:

$$c = \arg \max_{c_i} (Sim(d, C_i)) \quad (4)$$

4. PROPOSED METHOD

4.1 Rationale

In this subsection, we first illustrate that a sentiment classifier trained on old domain often performs extremely poorly on new domain; then demonstrate why new-domain unlabeled data may help to learn a new classifier which performs very well in new domain; finally we present the detailed algorithm using a base classifier. Without loss of generality, we employ centroid classifier as the base classifier in this work.

For the sake of being easy to explain, we take a simple example (see Figure 1(A)). The old-domain examples are represented by two circles: negative example is denoted by grey and positive example is denoted by white; the new-domain examples are represented by two ellipses: negative example is denoted by grey and positive example is denoted by white. C_{ON} and C_{OP} are the centroids of negative and positive class in old domain respectively. Middle Line is the perpendicular bisector of the line between C_{ON} and C_{OP} . From another perspective, Middle Line serves as a decision hyper-plane that separate negative class and positive class.

As a result, according the Middle Line, we can observe that all examples of old domain can be correctly classified. However, this Middle Line does not work well in new domain: from this figure, these new-domain negative examples under the Middle Line will be misclassified into positive class. This is the mechanism why domain-transfer degrades the performance of old classifier.

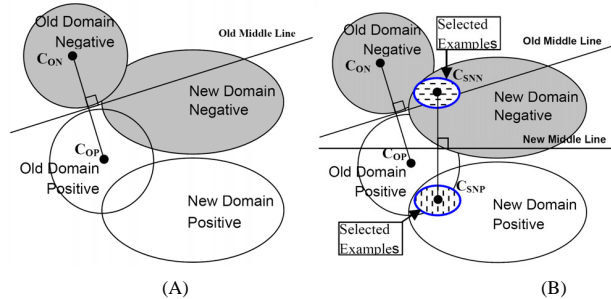


Figure 1: Performance of old classifier when transferred to new domain

An intuitive method to address this issue is to pick out some informative examples for new-domain and to train the base classifier again (see Figure 1(B)). We use “-” to represent selected examples from new-domain negative class, and use “+” to denote selected ones from new-domain positive class. Then we calculate two centroids, i.e., C_{SNN} and C_{SNP} , for the two new classes. As such, New Middle Line can be drawn. In this time, we can observe that most examples in new domain can be correctly categorized. This is the rationale why unlabeled data can be used to address domain-transfer problem.

Outline of proposed technique is presented in Figure 2. In this algorithm, the parameter “Ratio” indicates what percentage of new-domain data shall be picked out as informative examples.

Obviously, the most important and difficult job in this scheme is to label some informative ones for a new domain, because the old classifier performs poorly in new domain. In the following subsection we attempt to solve this problem.

-
- 1 Load old-domain labeled data (OL), new-domain unlabeled data (NU), and parameter “Ratio”;
 - 2 Train base classifier using labeled data in old domain;
 - 3 Label some informative unlabeled ones in new domain;
 - 4 Learn a new classifier using these selected examples;
 - 5 Classify examples in new domain using new classifier.
-

Figure 2: The Outline of Proposed Scheme

4.2 Analysis and Method

In this subsection, we first explain why the old classifier often performs poorly in new domain in most cases; then justify that discriminant function based ranking can pick out some informative examples in new domain.

Assuming X_1 denotes the word space of old domain, X_2 denotes the word space of new domain, and $X=X_1 \cup X_2$ denotes the total word space. We make following presumptions:

- (i): Words in X are independent each other;
- (ii): $X_\cap = X_1 \cap X_2 \neq \Phi$;
- (iii): X_\cap in different domains accords with the same probability distribution.

According to Bayesian formula, with respect to one example d in old domain, the Bayesian discriminant function is

$$\begin{aligned} p(c|d) &= \frac{p(d|c)p(c)}{p(d)} \\ &= \frac{p(d_{X_1 \setminus X_\cap}, d_{X_\cap} | c)p(c)}{p(d)} \\ &= \frac{p(d_{X_1 \setminus X_\cap} | c)p(d_{X_\cap} | c)p(c)}{p(d)} \end{aligned}$$

$$g_1(d) = \ln(p(d_{X_1 \setminus X_\cap} | c)) + \ln(p(d_{X_\cap} | c)) + D_1 \quad (5)$$

where c denotes the class label, i.e., positive or negative, and D_1 indicates a constant.

As such, with respect to one example e in new domain, the Bayesian discriminant function is

$$p(c|e) = \frac{p(e_{X_2 \setminus X_\cap} | c)p(e_{X_\cap} | c)p(c)}{p(e)}$$

$$g_2(e) = \ln(p(e_{X_2 \setminus X_\cap} | c)) + \ln(p(e_{X_\cap} | c)) + D_2 \quad (6)$$

where D_2 indicates a constant.

Consequently, if we directly apply the discriminant rule trained on old domain to new domain, the Bayesian discriminant function is

$$g_1(e) = \ln(p(e_{X_\cap} | c)) + D_1 \quad (7)$$

Obviously, there is a noticeable difference between above discriminant function and new-domain discriminant function. In most cases, accordingly, direct application of old-domain discriminant function (5) to classify new-domain examples is unfeasible. This is the mechanism why old-domain-trained classifier often performs extremely badly in new domain (as illustrated in Figure 1(A)).

However, X_{\cap} and $X_2 \setminus X_{\cap}$ are independent each other. According to formulas (6) and (7), for every example e on new domain, we can make a conclusion that the larger $g_1(e)$ is, the larger $g_2(e)$ is.

Assuming the data in $X_2 \setminus X_{\cap}$ is in accord with a kind of distribution. As a result, with respect to examples e_1, e_2 in new domain, we obtain,

$$g_1(e_1) > g_1(e_2) \Rightarrow p(g_2(e_1) > g_2(e_2)) > p(g_2(e_1) < g_2(e_2)) \quad (8)$$

Above formula indicates that, if $g_1(e_1) > g_1(e_2)$, then the probability of $g_2(e_1) > g_2(e_2)$ is bigger than the probability of $g_2(e_1) < g_2(e_2)$.

With respect to centroid classifier, we can train a classifier in old domain,

$$C_1^P = (C_{X_1 \setminus X_{\cap}}^P, C_{X_{\cap}}^P),$$

$$C_1^N = (C_{X_1 \setminus X_{\cap}}^N, C_{X_{\cap}}^N)$$

where C_1^P and C_1^N are positive and negative centroids respectively. For one example x in new domain, we can calculate its positive similarity (S^P) and negative similarity (S^N) as following,

$$S_1^P = C_{X_1 \setminus X_{\cap}}^P \cdot x_{X_1 \setminus X_{\cap}} + C_{X_{\cap}}^P \cdot x_{X_{\cap}}$$

$$S_1^N = C_{X_1 \setminus X_{\cap}}^N \cdot x_{X_1 \setminus X_{\cap}} + C_{X_{\cap}}^N \cdot x_{X_{\cap}}$$

Under this scenario, formula (8) leads to a conclusion: for one example, the larger the S^N , the more likely it is drawn from negative class; the larger the S^P , the more likely it is taken from positive class. Based on this conclusion, we propose Similarity Ranking method (SR): we first rank S^N of all examples, and assign top $n/2$ largest examples as negative; then rank S^P , and label top $n/2$ largest ones as positive.

However, this method doesn't hold when the length difference among different reviews is very large, because it is often the case that the larger the length of one review, the larger the S^N or S^P . What's worse, when transfer the old classifier to another domain, even if the actual length of reviews is nearly the same, the word-space difference between old domain and new domain can make a large difference on S^N or S^P .

To tackle this problem, we normalize (or divide) the original similarity so that the adverse effect of length difference and word-space variation can be offset to a high degree. This is the basic idea of relative similarity. Formally, we define Negative Relative Similarity (S^{RN}) and Positive Relative Similarity (S^{RP}) as following,

$$S^{RP} = \frac{S^P}{(S^N + S^P)/2} \quad (9)$$

$$S^{RN} = \frac{S^N}{(S^N + S^P)/2} \quad (10)$$

Up to this point, we can make a refined **conclusion** that, for one example, the larger the S^{RN} , the more likely it is drawn from negative class; the larger the S^{RP} , the more likely it is taken from positive class. According to this supposition, we propose Relative Similarity Ranking method (RSR): we first rank S^{RN} of all examples, and assign top $n/2$ largest examples as negative; then rank S^{RP} , and label top $n/2$ largest ones as positive.

In conclusion, we present the detailed algorithm (Figure 3) for proposed scheme using RSR. In this figure, Sizeof(NU) indicates the cardinality of unlabeled example set in new domain.

-
- 1 Load old-domain labeled data (OL), new-domain unlabeled data (NU), and parameter "Ratio";
 - 2 Train base classifier using labeled data in old domain;
 - 3 Label informative unlabeled ones:
 - 3.1 Calculate Relative Similarity using formulas (9-10);
 - 3.2 Rank S^{RN} and S^{RP} respectively;
 - 3.3 Label top $n/2$ examples as Negative in S^{RN} list, and label top $n/2$ as positive in S^{RP} list, where $n = \text{Ratio} \cdot \text{Sizeof(NU)}$.
 - 4 Learn a new classifier using these selected examples;
 - 5 Classify examples in new domain using new classifier.
-

Figure 3: The Detailed Outline of Proposed Scheme

4.3 A Case Study

Let's see a domain-transfer example. Given Computer review as old domain and House review as new domain. For Computer review we select a subset (CompSet for brevity) that contains 195 negative reviews and 272 positive reviews; for House review we use a subset (HouSet for brevity) consisting of 53 negative reviews and 67 positive reviews.

In Figure 4, examples (No.0-52) are from negative class, the other (No.53-119) are from positive class. For each example, we calculate its S^N and S^P . For the sake of being easy to observe, negative examples and positive examples are sorted by S^N respectively.

Figure 4(A) shows the similarities of base classifier when trained and tested using the same corpus "HouSet". In this case, most examples can be classified correctly: for most negative examples, their S^N is bigger than S^P ; for most positive ones, their S^N is smaller than S^P .

However, when we train the classifier on CompSet and test it on HouSet (see Figure 4(B)), the situation changes over completely. In this time, most examples will be classified into negative class but scarcely any ones can be assigned into positive class: in most cases, whether the example is from negative or positive class, its S^N is bigger than its S^P . This observation explains why domain-transfer often degrades the performance of base classifier. As a result, it is very difficult to pick out informative examples for positive class.

For the sake of being easy to explain, we take out six examples ($d_{15}, d_{20}, d_{49}, d_{63}, d_{76}, d_{114}$) randomly (refer to Table 1) from HouSet. The former three examples (d_{15}, d_{20}, d_{49}) are drawn from negative class, and the latter three (d_{63}, d_{76}, d_{114}) are coming from positive class. For each example, we calculate S^N and S^P . In this time, centroid decision rule doesn't work at all: it classifies the six examples into negative class.

In accordance with SR, we first rank S^N and label examples (d_{15}, d_{20}, d_{63}) as negative; then rank S^P and assign (d_{49}, d_{76}, d_{114}) as positive (assuming $n=6$). Obviously, d_{63} and d_{49} are misclassified. This observation indicates that SR suffers from word-space difference incurred by domain-transfer.

To overcome this shortcoming of SR, we proposed RSR method in preceding subsection. Let's turn to Table 1 again. First we calculate relative similarity for six examples using formulas (9-10), then rank S^{RN} and label examples (d_{15}, d_{20}, d_{49}) as negative; rank S^{RP} and label examples (d_{63}, d_{76}, d_{114}) as positive (assuming $n=6$). In this time, all examples are correctly classified.

Table 1: The relative similarities of six randomly selected examples

Similarity Examples	Original Similarity		Relative Similarity	
	S^N	S^P	S^{RN}	S^{RP}
d15	0.6467(1)	0.5844(1)	1.0505	0.9493
d20	0.6178(3)	0.5549(3)	1.0537	0.9464
d49	0.4252(5)	0.3832(5)	1.0520	0.9480
d63	0.6206(2)	0.5815(2)	1.0324	0.9674
d76	0.5939(4)	0.5453(4)	1.0427	0.9573
d114	0.2888(6)	0.2622(6)	1.0483	0.9517

Figure 4(C) displays the relative similarities of base classifier when trained on CompSet and tested on HouSet. In this time, for

most examples, its S^{RN} is still bigger than S^{RP} . But the S^{RN} of most negative examples is bigger than the S^{RN} of positive ones; the S^{RP} of most negative examples is smaller than the S^{RP} of positive ones. In this case, although it is difficult to pick out informative examples by centroid decision rule, Relative Similarity Ranking can help us to pick out informative examples for each class.

Figure 4(D) depicts error rate curve when using RSR method. The ‘‘Ratio’’ indicates what percentage of unlabeled examples in new domain shall be selected as informative ones. The error rate is computed using formula (11). From this figure, we can observe that RSR performs very well when Ratio is smaller than 0.5.

$$Error\ Rate = \frac{misclassified\ examples}{ratio \times number\ of\ unlabelled\ examples} \quad (11)$$

$$= \frac{misclassified\ examples}{total\ classified\ examples}$$

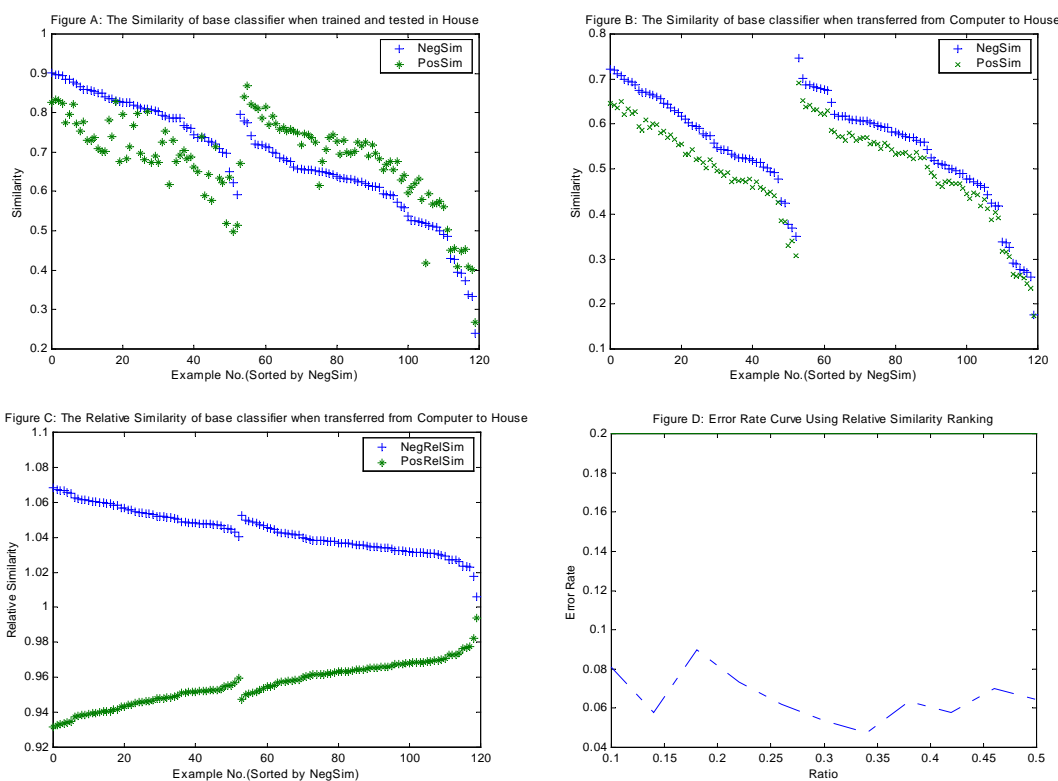


Figure 4: The similarity and relative similarity of old classifier

5. EXPERIMENT RESULTS

5.1 Datasets

To validate the effectiveness and robustness of proposed method, we collected three domain-specific datasets: Computer Reviews (Comp), Education Reviews (Edu) and House Reviews (Hou).

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 4725 different words.

Education Reviews There are 1012 negative reviews and 254 positive reviews in this corpus. Much larger than Computer

Reviews, the average size of reviews is about 600 words, and the cardinality of vocabulary is 19150.

House Reviews This collection consists of 445 negative reviews and 555 positive reviews. Larger than Computer Reviews and smaller than Education Reviews, the average length of reviews is about 300 terms and the different terms amount to 12674.

5.2 Performance Measure

To evaluate a semantic classification system, we use F1 measure. This measure combines recall and precision in the following way:

$$Recall = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}}$$

$$\text{Precision} = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}}$$

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})}$$

For ease of comparison, we summarize the F1 scores over the different categories using the Micro- and Macro-averages of F1 scores [16]:

Micro - F1 = F1 over categories and documents

Macro - F1 = average of within - category F1 values

The MicroF1 and MacroF1 emphasize the performance of the system on common and rare categories respectively. Using these averages, we can observe the effect of different kinds of data on a classification system.

5.3 Experimental Design

To ensure the rate of training set and test set is close to 1:1, we only use 50% of one dataset as training set when it is used as old domain. There is no doubt that feature selection may remove some important features in new domain, so we don't delete any features when training the base classifier in old domain.

For EM-like method [1], we use centroid classifier as its base classifier. We use 50% of old-domain data as labeled training examples and use 50% of new-domain data as unlabeled ones.

Joachims's SVM-light package can be used for TSVM classification. (<http://svmlight.joachims.org/>). We use a linear kernel and leave all parameters as default. Like EM-like method, we use 50% of old-domain data as labeled training examples and use 50% of new-domain data as unlabeled ones.

5.4 Comparison and Analysis

Table 2 reports the performance of different methods when transferred to another domain. Under the proposed scheme, we use two methods to pick out new-domain informative examples: Similarity Ranking (SR), and Relative Similarity Ranking (RSR). Both SR and RSR pick out some informative examples rather than label all examples in new domain. We split the new-domain data evenly into unlabeled set and test set; the Ratio is set to 0.4 for SR and RSR respectively.

As we can observe from this table, RSR dramatically improves the performance of base classifier in new domain. The MacroF1 of proposed method beats the base classifier by about 37 percents on problem "Comp->Edu", by about 52 percents on "Comp->Hou", by about 36 percents on "Edu->Comp", by about 28 percents on "Edu->Hou", by about 21 percents on "Hou->Comp" and by about 22 percents on "Hou->Edu". The wide margin improvement indicates that proposed scheme combined with Relative Similarity Ranking method performs very effectively and robustly.

Despite of simplicity and straightforwardness, SR performs quite well. Its average accuracy is about 5% lower than RSR but about 6% higher than TSVM (baseline). For problem "Edu->Hou", SR even achieves better result than RSR. This performance provides convincing proof for the effectiveness of ranking based method.

EM-like method indeed boosts the base classifier on three domain-transfer problems, i.e., Comp->Hou, Edu->Comp, and Edu->Hou. However, on other problems it degrades the performance of base classifier. This phenomenon demonstrates that EM-like method cannot achieve robust results for domain-

transfer problems. This observation validates the analysis in section 2, that is, when old-domain data and new-domain data don't share the same generative model, EM may actually degrade rather than improve classifier accuracy.

A question may arise: why the EM-like method doesn't even outperform the baseline method. This seems to be counter-intuition as the EM-like method is trained on old-domain examples as well as on new-domain examples while the centroid baseline only uses the old-domain examples.

In fact, when the iteration step is set to zero, the EM-like method is equal to the centroid baseline. Accordingly, the centroid baseline can be regards as a special case of the EM-like method. However, with the increase of iteration step, more and more misclassified new-domain examples are employed to train the EM-like method, thus the performance of the EM-like method is degraded step by step. This is the reason why the EM-like method is much worse than the centroid method in some cases, such as "Comp->Edu" and "Hou->Edu".

Table 2(A): MicroF1 of different methods when transferred to another domain

	Centroid	TSVM (Baseline)	EM Scheme (Baseline)	Proposed Scheme	
				SR	RSR
Comp->Edu	0.7993	0.6887	0.2006	0.6966	0.8530
Comp->Hou	0.4540	0.8960	0.5540	0.8320	0.8440
Edu->Comp	0.5053	0.6509	0.8543	0.7751	0.8051
Edu->Hou	0.5120	0.6100	0.5900	0.8280	0.7200
Hou->Comp	0.7387	0.7815	0.6916	0.8094	0.8993
Hou->Edu	0.5781	0.6840	0.2006	0.7109	0.8214
Average	0.5979	0.7185	0.5152	0.7753	0.8238

Table 2(B): MacroF1 of different methods when transferred to another domain

	Centroid	TSVM (Baseline)	EM Scheme (Baseline)	Proposed Scheme	
				SR	RSR
Comp->Edu	0.4442	0.6572	0.1671	0.6595	0.8119
Comp->Hou	0.3272	0.8953	0.3564	0.8318	0.8429
Edu->Comp	0.4484	0.6392	0.8442	0.7702	0.8050
Edu->Hou	0.4374	0.5800	0.4835	0.8267	0.7197
Hou->Comp	0.6875	0.7764	0.6067	0.8078	0.8982
Hou->Edu	0.5624	0.6531	0.1671	0.6773	0.7795
Average	0.4845	0.7002	0.4375	0.7622	0.8095

In contrast to EM-like scheme, TSVM performs very well for domain-transfer problems. Apart from "Comp->Edu", TSVM beats Centroid classifier by a wide margin. The average MicroF1 of TSVM is about 12 percents higher than Centroid classifier, and 20 percents higher than EM-like scheme. On other hand, however, TSVM is still outperformed by proposed scheme. For examples, apart from "Comp->Hou", TSVM is outperformed by SR or RSR scheme with wide margin. This observation indicates that proposed straightforward scheme can produce much better results than theoretically-more-sound TSVM.

Figure 5 displays the error rate curves when using SR or RSR to pick out informative examples in new domain. The parameter

“Ratio” indicates what percentage of new-domain unlabeled data

shall be picked out as informative ones.

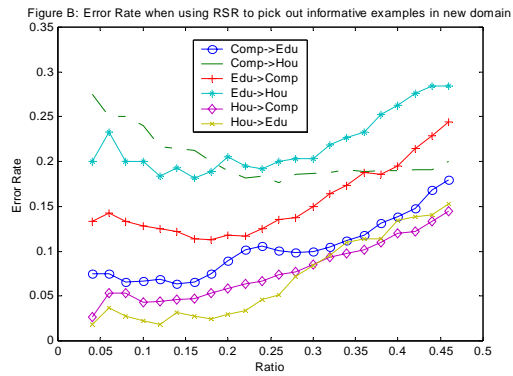
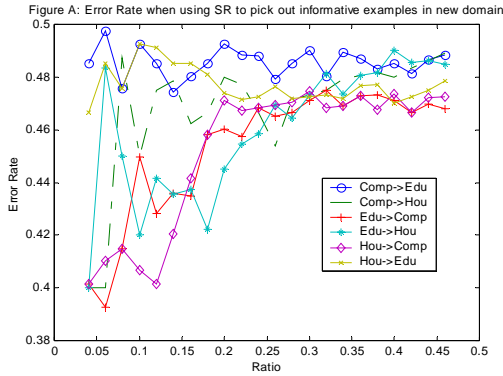


Figure 5: Error Rate Curves when using SR or RSR to pick out informative examples in new domain

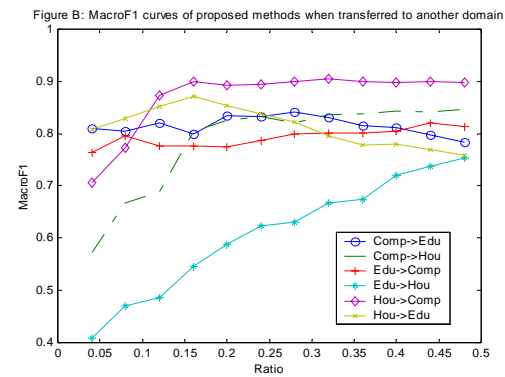
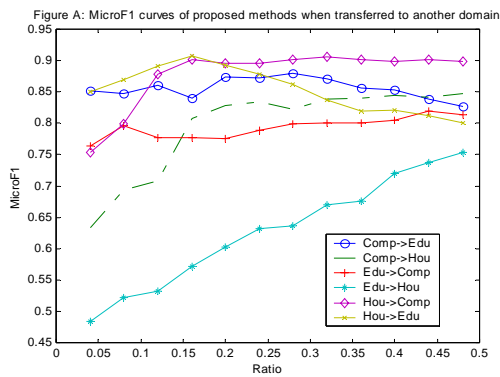


Figure 6: performance curves of proposed method vs. the Ratio

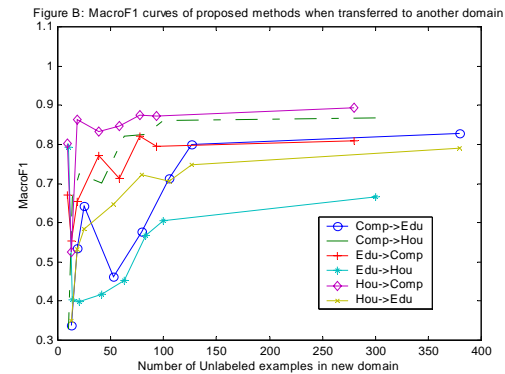
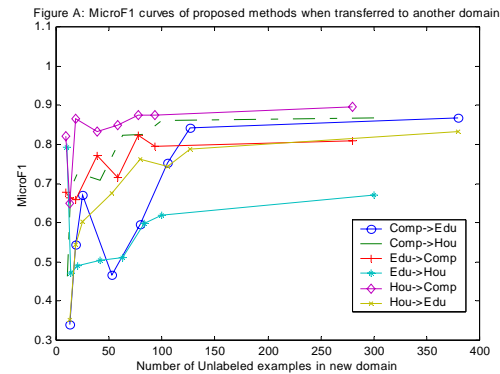


Figure 7: performance curves of proposed method vs. the number of unlabeled examples in new domain

As we can observe, RSR achieves much lower error rate than SR. On all problems, RSR beats SR by at least 10%. Especially when using “Hou” as old domain, the error rate of RSR is 30% lower than that of SR. This figure indicates that both SR and RSR can pick out informative examples but SR takes in more error ones. This fact explains why SR can perform quite well under the proposed scheme but still be outperformed by RSR by a wide margin.

Figure 6 shows the performance curves of proposed method vs. the Ratio. It is worth noticing that we only use RSR to pick out informative examples. We split the new-domain data evenly into unlabeled set and test set.

We can clearly observe that increasing the Ratio increases the classification accuracy in new domain. However, the increase in accuracy is not directly proportional to the increase in the Ratio. As the Ratio gets larger, the accuracies start leveling off as we can observe from this figure apart from the problem “Edu->Hou”.

The second observation is that, when the Ratio exceeds 0.15, proposed method achieves persistent results except “Edu->Hou” problem. This fact validates that RSR can pick out informative examples in new domain.

Figure 7 plots the performance curves of proposed method vs. the number of unlabeled examples. It is worth noticing that we only use RSR to pick out informative examples. To conduct this

experiment, we split the new-domain data evenly into original unlabeled set (O_U for brevity) and test set, then pick out new unlabeled set (N_U for brevity) randomly from O_U , and finally use N_U rather than O_U to pick out informative examples. The Ratio is set to 0.4.

As we can see, when the size of N_U is bigger than 100, proposed method achieves persistent performance on all domain-transfer problems. Furthermore, except for "Edu->Hou" problem, the performance of proposed method keeps nearly unchanged when the cardinality of N_U exceeds 100.

6. DISCUSSION

At the first glance, it seems that proposed scheme is nearly the same as EM-like unlabeled ones aided learning algorithm, but there are a number of details that are different. First proposed method doesn't need to iterate training base classifier until convergence. In fact it trains base classifier only twice: one is over training data; second is over selected examples in new domain. Second, proposed method labels a part of informative unlabeled examples rather than all unlabeled ones. This difference protects our method from adverse effect of mislabeled ones. Thirdly, initial labeled examples are used only once by proposed method.

There are a few limitations with this work. First, although RSR can pick out "informative" examples in new domain, we still cannot guarantee that the selected "informative" ones are representative to the new domain. The second problem is that the chosen old domain may be far different from the new one, which makes it difficult to select really "informative" examples. Thirdly, when the negative examples are severely overlapped with the positive ones, RSR may label one same example into negative class as well as into positive class. Lastly, for unbalanced classification problems where one class is dominant, RSR that takes an equal number of examples for both positive and negative classes might degrade the performance of proposed scheme.

7. CONCLUSION REMARKS

Sentiment classification is a very domain-specific problem, that is to say, classifiers trained in one domain do not perform well in others in most cases. So if we transfer the classifier to new domain, we require a large amount of labeled data of new domain to train the base classifier again. In practical applications, unfortunately, we are often confronted with sentiment classification problems where there are scarcely any labeled examples.

In this work, we attempt to address this issue by making use of unlabeled data in new domain to aid in training the base classifier. The basic idea is to use old-domain-trained classifier to label some informative unlabeled examples in new domain, and train the base classifier again. In order to effectively pick out informative ones, we proposed Relative Similarity Ranking method. The main idea is to counteract the adverse affect of domain-transfer by altering the original similarities.

An empirical evaluation conducted on three domains indicates that proposed method dramatically enhances the accuracy of the base sentiment classifier on new domain. In addition, when the size of unlabeled examples is bigger than 100, proposed method achieves persistent performance in most cases.

It is no doubt that the results reported here are by no means the best that can be obtained. Our future effort is to investigate how to

improve the performance of this scheme. Another job is to study how well this scheme performs when using other learning method, such as SVM or Naïve Bayes, as base classifier.

8. ACKNOWLEDGMENTS

This work was mainly supported by special fund of Chinese Academy of Sciences, "Research on Opinion Mining of Web Text", under grant number 0704021000 and two projects, i.e., 2007CB311100 and 2007AA01Z441.

9. REFERENCES

- [1] Aue, A. and Gamon, M. Customizing Sentiment Classifiers to New Domains: a Case Study. RANLP. 2005.
- [2] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with Co-Training. COLT. 1998, 92-100.
- [3] Cui, H., Mittal, V., Datar, M. Comparative Experiments on Sentiment Classification for Online Product Reviews. AAAI. 2006.
- [4] Engström, C. Topic Dependence in sentiment classification. Unpublished M.Sc. thesis, University of Cambridge, 2004.
- [5] Finn, A., and Kushmerick, N. 2003. Learning to classify documents according to genre. In IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis
- [6] Han, E. and Karypis, G. Centroid-Based Document Classification Analysis & Experimental Result. PKDD. 2000.
- [7] Joachims, T. Transductive inference for text classification using support vector machines. ICML. 1999, 200-209.
- [8] Kennedy, A. and Inkpen, D. Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. FINEXIN. 2005.
- [9] Lanquillon, C. Learning from Labeled and Unlabeled Documents: A Comparative Study on Semi-Supervised Text Classification. PKDD. 2000, 490-497
- [10] Mullen, T. and Collier, N. Sentiment analysis using support vector machines with diverse information sources. EMNLP. 2004, 412-418
- [11] Nigam, K., McCallum, A., Thrun, S. and Mitchell, T. Learning to classify text from labeled and unlabeled documents. AAAI. 1998, 792-799.
- [12] Pang, P., Lee, L., and Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. EMNLP. 2002.
- [13] Salton, G., McGill, M. Introduction to Modern Information Retrieval. McGraw-Hill Book Company, New York. 1983.
- [14] Turney, P. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. ACL. 2002, 417-427
- [15] Whitelaw, C., Garg, N., Argamon, S. Using appraisal groups for sentiment analysis. CIKM. 2005, 625-631.
- [16] Yang, Y. A study on thresholding strategies for text categorization. SIGIR. 2001, 137-145
- [17] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. ACL 2007.