Cross-Domain Sentiment Classification Using a Two-Stage Method

Kang Liu, Jun Zhao
Institute of Automation, Chinese Academy of Sciences
HaiDian District, Beijing, China
+86 10 8261 4468
{kliu, jzhao}@nlpr.ia.ac.cn

ABSTRACT
In this paper, we give out a two-stage approach for domain adaptation problem in sentiment classification. In the first stage, based on our observation that customers often use different words to comment on the similar topics in the different domains, we regard these common topics as the bridge to link the different domain-specific features. We propose a novel topic model named Transfer-PLSA to extract the topic knowledge between different domains. Through these common topics, the features in the source domain are corresponded to the target features, so that those domain-specific knowledge can be transferred across different domains. In the second step, we use the classifier trained on the labeled examples in the source domain to pick up some informative examples in the target domain. Then we retrain the classifier on these selected examples, so that the classifier is adapted for the target domain. Experimental results on sentiment classification in four different domains indicate that our method outperforms other traditional methods.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural language processing

General Terms
algorithms, experimentation

Keywords
Domain adaptation, Sentiment Classification, Feature translation.

1. INTRODUCTION
In this paper, we investigate the domain adaptation problem in sentiment classification, where only the instances in source domain are labeled and no labeled data in target domain are available. We expect the sentiment classifier trained only by labeled data in source domain has good performance on the data in the target domain. Thus, for sentiment classification, we mainly consider two characteristics of domain adaptation as follows.

1) ***The instances in the source and the target domain are represented using different features.*** In different domains, people use different words to express their opinions, which makes their feature space to be not same. For example,. when expressing positive sentiment in car domain, people often use "faster, powerful, safe, …" frequently, while in the movie domain, they often use "impressive, fun, predictable, …". Those domain-specific features, occurring only in the source domain, are not useful for the testing data in the target domain because they aren't observed in the target domain.

2) ***The distribution of data from the different domain is different.*** Even though two datasets from different domains share the same feature space, the frequencies of the same feature in different domains are often different. That causes the distribution difference between different domains. Therefore, that will make the decision hyper-plane trained on the source domain is not correct for classify the data in the target domain.

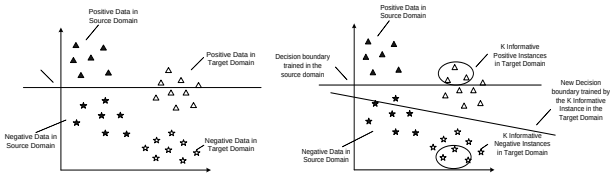

Figure 1. The Negative Effect of the Different Feature Distribution on the Classification Performance

For example, in the left graph in figure 1, the dark triangles and the dark stars denote the positive and negative instances in the source domain, respectively. And the blank triangles and the blank stars denote the positive and negative instances in the target domain. As our observation, the instances from different domains share the same feature space but the different distribution. If we use the source dataset to train a sentiment classifier, we can obtain a decision hyper-plane denoted as a line, which cannot correctly separate the negative instances and the positive instances in the target domain.

In this paper, when performing cross-domain sentiment classification, we propose a two-stage method, where each stage gives a solution for each characteristic, respectively.

The aim of the first stage in our method is to construct a unique feature space that the different domains can share. We construct a feature translator $\phi(w_s, w_t)$ like [3], which can translate a features w_s in the source domain to a w_t in the target domain. In this way, the learned classifier treat those domain-specific features similarly, so that those domain-specific knowledge can be transferred across different domains. To construct this feature translator, Our intuitive idea is to regard the common topic knowledge as the bridge between the features from different domains. Although the topics that customers comment on may be different, there are

CIKM'09, November 2-6, 2009, Hong Kong, China.

some similar topics that are shared between different domains. If two different features describe the similar topics, they can be linked together through these topics. For example:

"The appearance of this phone is very exquisite."

"BMW's appearance looks very noble."

These two sentences come from different domains. They both express positive emotions on the product's appearance. We can see that the word "*exquisite*" is a key feature for identifying the polarity of the first sentence. But in the second sentence, customers use "*noble*" rather than "*exquisite*" to appraise the product's appearance. Therefore, these domain-specific features make the feature spaces of each domain to be different. We use the common topic (the product's appearance) as a bridge to correspond these two domain-specific features, so the classifier will treat them similarly. In other words, knowledge is transferred across domains. Figure 2 shows our idea for identifying the correspondence between different features. In this figure, triangles denote the common topics between two domains, and circles denote the features. We can see that the domain-specific features ("exquisite" in the cell-phone domain, "noble" in the car domain) are connected by a common topic ("appearance").

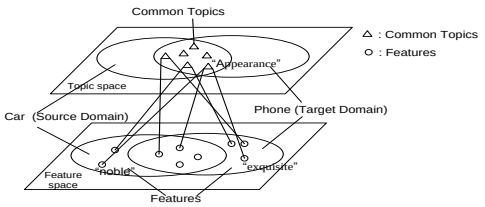

Figure 2. Connecting Different Features through Common Topics

In the second stage, the aim of our method is to resolve the second characteristic. We use the source instances to train a classifier and classify the unlabeled instances in the target domain. Then we select some informative target instances,. and use them to retrain a new classifier, which can revise the original decision hyper-plain. As shown in the right graph in figure 1, our classifier retrained by the informative instances in the target domain can more accurately classify the testing data than the classifier trained only in the source domain.

2. THE PROPOSED APPROACH

In our problem, each instance x_s in the source domain D_s can be represented by a feature vector $(w_s^1, \cdots, w_s^{N_s})$ with the probabilistic distribution $p_s(\cdot)$, where $w_s^{N_i} \in W_s$ and W_s is the source feature space. Each instance x_t in the target domain can be represented by a feature vector $(w_t^1, \cdots, w_t^{N_t})$ with the probabilistic distribution $p_t(\cdot)$, where $w_t^{N_i} \in W_t$ and W_t is the target feature space. Both datasets have the same label space Y. Only the instances in D_s are labeled, which have different feature representation with the testing instances in D_t.

2.1 The First Stage: Finding Common Feature Representation Based on Feature Translation

As mentioned in the first section, to identify the correspondence between the features of two different domain, a feature translator $\phi(w_s, w_t)$ is constructed. We believe that $\phi(w_s, w_t) \propto p(w_t \mid w_s)$ like [3], where w_s denotes the features in the source domain and w_t denotes the features in the target domain.

To compute $p(w_t \mid w_s)$, we regard the topics as the bridge to connect different features from different domains. So that

$$p(w_t \mid w_s) = \sum_z p(w_t \mid z)\, p(z \mid w_s) \propto \sum_z p(w_t \mid z)\, p(w_s \mid z)\, p(z) \quad (1)$$

where z is the topic of the document. This process is similar to the process of machine translation. The features in the source domain are translated into the features in the target domain through the topic knowledge.

2.1.1 Extracting Topic Knowledge across Different Domains using Transfer-PLSA

In our algorithm, we use the topic model PLSA [5] to compute $p(z)$ and $p(w \mid z)$ in formula (1). Since the aim of us is to find the common topics between the documents from the different domains, we must use a joint probabilistic model rather than two separated PLSA models. Furthermore, since the distribution of the datasets from the different domains are different, if we perform standard PLSA on it, the performance will drop. Thus, we proposed a new topic model named Transfer-PLSA to find the topic distribution between two different datasets. Since the sentiment classification is performed in the target domain, so the likelihood must be estimated under the target distribution $p_t(\cdot)$. The likelihood L of the standard PLSA can be modified as the following formula:

$$L = \sum_{d \in D_s} \sum_{w \in W_s} n(d,w) \log p_t(d,w) + \sum_{d \in D_t} \sum_{w \in W_t} n(d,w) \log p_t(d,w) \quad (2)$$

where $n(d,w)$ denotes the term frequency in document d. And $P_t(d,w)$ can be obtained using the following formula:

$$p_t(d,w) = \sum_z p_t(d \mid z)\, p_t(w \mid z)\, p_t(z) \quad (3)$$

To find the optimal solution of the object L, EM algorithm is used for a local optimal solution of the object L, which contains two steps as follows

E-Step:

$$p_t(z \mid d,w) = \frac{p_t(z)\, p_t(d \mid z)\, p_t(w \mid z)}{\sum_z p_t(z)\, p_t(d \mid z)\, p_t(w \mid z)} \quad (4)$$

M-Step:

$$\begin{aligned} p_t(w \mid z) &= \sum_{D_i, i \in \{s,t\}} p_t(w, D_i \mid z) \\ &\propto \sum_{D_i, i \in \{s,t\}} p_t(w \mid D_i, z)\, p_t(z \mid D_i)\, p_t(D_i) \end{aligned} \quad (5)$$

$$p(d \mid z) = \frac{\sum_w n(d,w)\, p(z \mid d,w)}{\sum_{d \in D_s \cup D_t, w} n(d,w)\, p(z \mid d,w)} \quad (6)$$

$$p_t(z) = \sum_{D_i, i \in \{s,t\}} p_t(z, D_i) = \sum_{D_i, i \in \{s,t\}} p_t(z \mid D_i)\, p_t(D_i) \quad (7)$$

In formula (5), $p_t(w \mid D_i, z)$ denotes the probability of generating a word w by a topic z in domain D_i, and $p_t(z \mid D_i)$ denotes the

probability distribution of the topic z in domain D_i. We simply use $p(w|D_i,z)$ and $p(z|D_i)$ to approximate $p_t(w|D_i,z)$ and $p_t(z|D_i)$ separately. They can be estimated as follows:

$$p(w|D_i,z)=\frac{\sum_{d\in D_i} n(d,w)p(z|d,w)}{\sum_{d\in D_i,w} n(d,w)p(z|d,w)} \quad (8)$$

$$p(z|D_i)=\frac{\sum_{d\in D_i,w} n(d,w)p(z|d,w)}{\sum_{d\in D_i,w} n(d,w)} \quad (9)$$

It's worth noting that $p_t(D_i)$ is a tradeoff parameter, which can be understood as a kind of relevance metric between dataset D_i and the target domain. We set $p_t(D_s)+p_t(D_t)=1$. It's difficult to estimate $p_t(D_i)$ for each domain, so we set $p_t(D_s)=\lambda$ and $p_t(D_t)=1-\lambda$ in our method. When $0<\lambda<0.5$, it means that our algorithm will put more weight on the topic distribution in the target domain. When $0.5\le\lambda<1$, it means that it will put more weight on the source domain.

2.1.2 Finding Common Topics between Different Domains and Constructing Feature Translator

Our intuitive idea is that the common topics are the bridge between two different domains Therefore, the common topics must be extracted first after $p_t(w|z)$ is obtained .

For each topic z, the KL distance between different domains is used as the criterion for extracting common topics as follows.

$$\begin{aligned} Score(z) &= D_{kl}(p(W|z,D_s)\,\|\,p(W|z,D_t)) \\ &= \sum_{w\in W_s\cup W_t} p(w|z,D_s)\log_2\frac{p(w|z,D_s)}{p(w|z,D_t)} \end{aligned} \quad (10)$$

We rank topic z by $Score(z)$ and choose the top k topics with smaller $Score(z)$ as the common topics.

Although these common topics are regarded as the bridge between the different feature spaces, we could not ignore other uncommon topics completely. We set different weights on the common topics and the uncommon topics to compute $p(w_t|w_s)$, so that

$$p(w_t|w_s)\propto\sum_{z_{common}} p(w_t|z)p(z|w_s)+\alpha\sum_{z_{uncommon}} p(w_t|z)p(z|w_s) \quad (11)$$

where $\alpha\in[0,1]$.

In this way, different features are linked through the common topics and we obtain the translating probability from a source feature w_s to a target feature w_t. If two different features are correlated with the same common topics in a similar way, they will have higher degree of correspondence. Based on the matrix of $p(w_t|w_s)$, the feature translator $\phi(w_s,w_t)$ can be constructed. Then we can construct a linear projection θ from the source feature space to the target feature space, which is proportional to $\phi(w_s,w_t)$. In the source feature space, for the common feature $w_s\in W_s\cap W_t$, feature mapping will not be performed. For the domain-specific features $w_s\notin W_s\cap W_t$, they will be translated to w_t according to $p(w_t|w_s)$. Each source instance will be represented as an augmented feature vector which contains all the common features and the mapped domain-specific features θw_s, so that the domain-specific features, whatever in source domain or target domain can be useful for the deciding the sentiment of the documents in the target domain.

2.2 The Second Stage: Retrain the Classifier by Selected Informative Instances

After the first stage, all data can be represented in a unique feature space through feature translation. In the second stage of our method, we add the unlabeled instance into the training process to make our classifier to be closer to the target domain.

Our method is similar to [6]. At first we use the classifier trained on the transformed source labeled data to select l informative instances in the target domain. Then, we will use these informative instances to retrain a new classifier. We believe this classifier will be more closer to the target domain than the original classifier. We use SVM as our classifier and select the top l instances which is the farthest to the decision hyper-plane as the informative examples, because the example is more far to the hyper-plain, it has more confidence to belong to some class.

3. Experiments

3.1 Experimental Design

The dataset which we used is as same as [1], which contains four domains. For the evaluation of our method, we select naïve Bayes (NB) and Support Vector Machine (SVM) as the baselines, we also select some semi-supervised approaches as the baselines, such as EM based naïve Bayes (EMNB), Transductive Support Vector Machine (TSVM) [4]. Furthermore, we design some baselines for comparison. We list them as follows:

ONLY-FIRST: We only perform the first stage for cross-domain sentiment classification. And the second stage is not performed. In ONLY-FIRST we set $k=4$, $\lambda=0.3$, $\alpha=0.2$ and the number of the topics be 20.

ONLY-SECOND: In this method, we refer to the method [6], which is also a transfer learning method. ONLY-SECOND only performs the second stage of our method when training the classifier, but doesn't perform the first stage.

TWO-STAGE: In this method, we use the proposed two-stage method to train our classifier. Both stages are used to transfer knowledge across different domains. The setting of each stage is the same as ONLY-FIRST and ONLY-SECOND, respectively.

SPLSA: SPLSA is similar to the first stage of our algorithm. The difference between them is that SPLSA use standard PLSA to associate words with topics but not transfer PLSA. Other setting is as same as ONLY-FIRST.

All the methods mentioned above use unigrams as features. And we use 2,000 instances in one domain (source domain) as the training set and 2,000 instances in the other domain (target domain) as the testing set.

3.2 Experiment Results and Analysis

The sentiment classification results in four domains are presented in table 1. The first column in Table 1 shows that the dataset which we use, where the first letter indicates the source domain and the second letter indicates the target domain. For example, "D

-> B" indicates that we train the classifier on the "dvd" domain and test it on the "book" domain.

From the results, we can see that the ONLY-FIRST outperforms not only supervised method, but also semi-supervised method in all datasets. It proves the effectiveness of our first stage, which is more adaptive to the target domain than the traditional learning methods. We believe the reason is that our method can connect different domain-specific features between different domains according their semantic relationship, so that the domain-specific knowledge can be transferred across different domains. Comparing the results of ONLY-SECOND with that of other baselines, we can find the strategy in the second stage in our method can effectively improve the performance, which also prove the validity of our method. We can obtain the conclusion that the new classifier trained by the selected informative examples can be more adaptive to the target domain than the original classifier trained on the source domain.

Furthermore, we can see TWO-STAGE further increase the classification performance of ONLY-FIRST and ONLY-SECOND. We believe the reason is that the domain adaptation in the first stage of our approach can obtain better performance on the target domain than traditional methods, so that it is helpful for selecting more informative examples to retrain a new classifier. At the same time, those selected informative examples in the target domain can effective revise the original classifier because their distribution is similar to the distribution of the target domain.

Table 1. Experimental Results

Data Set	NB	EMNB	SVM	TSVM	SPLSA	ONLY-FIRST	ONLY-SECOND	TWO-STAGE
D -> B	0.711	0.6925	0.69	0.7175	0.725	0.747	0.774	**0.78**
E -> B	0.6675	0.644	0.675	0.6775	0.68	0.683	0.715	**0.72**
K -> B	0.6655	0.63	0.6665	0.67	0.672	0.685	0.7055	**0.7175**
B -> D	0.74	0.74	0.7425	0.73	0.7415	0.75	0.7675	**0.768**
E -> D	0.6775	0.68	0.6745	0.677	0.685	0.712	**0.7215**	0.72
K -> D	0.6975	0.7025	0.695	0.694	0.702	0.71	0.7215	**0.7275**
B -> K	0.6925	0.56	0.702	0.7375	0.708	0.742	0.752	**0.765**
D -> K	0.649	0.5475	0.7125	0.71	0.7225	0.736	**0.7885**	0.77
E -> K	0.7575	0.7845	0.7715	0.79	0.799	0.83	0.851	**0.86**
B -> E	0.619	0.5380	0.6775	0.6775	0.71	0.72	0.7745	**0.79**
D -> E	0.6095	0.534	0.692	0.712	0.722	0.73	0.7765	**0.785**
K -> E	0.7585	0.7095	0.7735	0.788	0.8	0.817	**0.823**	0.82
Mean	0.687	0.6468	0.7065	0.7185	0.7225	0.7384	0.7642	0.7696

In Table 1, we can see that the performance of ONLY-FIRST is better than that of SPLSA methods. That indicates that Transfer-PLSA proposed in this paper can more accurately extract the topic knowledge across different domains. Through these topics, the semantic relationship between features across different domains can be constructed more accurately. The standard PLSA doesn't distinguish the distribution difference between domains. In contrast, our algorithm can set different weights to the instances in the different domains through λ. If we can find the appropriate λ, the performance of the sentimental classifier for domain adaptation can be promoted.

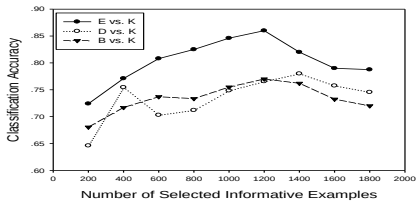

Figure 3. Performance for Different Number of the Selected Informative Examples

In the second stage of our method, the number of the selected informative examples has great influence on the final performance. In this experiment, we conduct the experiment on "kitchen" domain to empirically analyze how classification accuracy evolves when the number of the selected informative examples changes from 200 to 1800. The experimental results of TWO-STAGE are shown in figure 3.

From results, we can see that the best performance is obtained when the number of the informative examples is set between 1200 to 1600 .When the size of selected examples is too small, we have no sufficient knowledge to train the classifier. So the performance decreases, even lower than the original classifier trained on the labeled data in the source domain. However, if we select appropriate the number of the informative examples, the performance of TWO-STAGE will be improved too much. Therefore, the results prove that the second stage in our method is useful and effective for improving the performance.

4. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a novel method for domain adaptation in sentiment classification, which contains two stages. In the first stage, we proposed a new approach based on feature translation, which can find a unique feature representation between different domains. In the second stage of our method, we use transformed labeled instances in the source domain to select some informative examples in the target domain and use them to retrain a new classifier, so that the hyper-plane can be revised to be more closer to the distribution of the target domain.

5. ACKNOWLEDGMENTS

The work is supported by the National High Technology Development 863 Program of China under Grants no. 2006AA01Z144, and the National Natural Science Foundation of China under Grants no. 60673042 and 60875041.

6. REFERENCES

[1] John Blitzer, Mark Dredze and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proc. of the 45th Annual Meeting of the Association of Computational Linguistics (ACL), pages 440-447.

[2] Hal Daumé III and Daniel Marcu. 2006. Domain Adaptation for Statistical Classifiers. Journal of Artificial Intelligence Research 26, pages 101-126.

[3] Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang and Yong Yu. 2008. Translated Learning: Transfer Learning across different feature space. In Proc. of NIPS.

[4] T. Joachims, 1999, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press.

[5] Thomas Hofmann. Probabilistic Latent Semantic Indexing.1999. In Proc. of SIGIR.

[6] Songbo Tan, Gaowei Wu, Huifeng Tang and Zueqi Cheng. 2007. A Novel Scheme for Domain-transfer Problem in the context of Sentiment Analysis. In Proc. of CIKM 2007

[7] Gui-Rong Xue, Wenyuan Dai Qiang Yang and Yong Yu. 2008. Topic-bridged PLSA for Text Classification. In Proc. of SIGIR.