A Text Classifier with Domain Adaptation for Sentiment Classification

Wei Chen and Jingyu Zhou

School of Software
MOE-MS Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240, P.R. China
sagitchen@sjtu.edu.cn,
zhou-jy@cs.sjtu.edu.cn

Abstract. In sentiment classification, traditional classification algorithms cannot perform well when the number of labeled data is limited. EM-based Naïve Bayes algorithm is often employed to argument the labeled data with the unlabeled ones. However, such an approach assumes the distributions of these two sets of data are identical, which may not hold in practice and often results in inferior performance.

We propose a semi-supervised algorithm, called Ratio-Adjusted EM-based Naïve Bayes (RAEMNB), for sentiment classification, which combines knowledge from a source domain and limited training instances from a target domain. In RAEMNB, the initial Bayes model is trained from labeled instances from both domains. During each EM iteration, we add an extra R-step to adjust the ratio of predicted positive instances to negative ones, which is approximated with labeled instances of target domain. Experimental results show that our RAEMNB approach outperforms the traditional supervised, semi-supervised classifiers.

Keywords: domain adaptation, sentiment classification, Naïve Bayes, EM, semi-supervised.

1 Introduction

Sentiment classification, or polarity classification, is the binary classification task of labeling a document as expressing an overall positive or negative opinion. In recent years, sentiment classification has been widely adopted in many applications, such as analyzing results of political debate [14] and customer reviews [10,16].

Previous research [10,9] has applied various text-categorization algorithms for sentiment classification, which requires a large number of training instances to be effective. For domains that have little or no labeled instances, transfer learning algorithms [1,9,6,12,2] can be applied. Among them, EM-based Naïve Bayes (EMNB) [6] and its extensions have received a lot of attentions. However, the performance of EMNB could degrade during each EM iteration [15].

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 61–72, 2010.

An important reason for this phenomenon is that the distributions of source domain and target domain are different, which has an adverse effect on the prediction accuracy for Naïve Bayes classifier.

We propose our Ratio-Adjusted EM-based Naïve Bayes (RAEMNB) algorithm for sentiment classification. RAEMNB is a semi-supervised classifier that utilizes knowledge from a source domain with rich labeled instances and limited training items from the target domain. The limited labeled data from the target domain serves two purposes. First, these data are used as training instances for initial Naïve Bayes model. The second more important usage of these data is for estimating the real distribution of the target domain. Specifically, RAEMNB first trains an initial Naïve Bayes model with labeled data from both source and target domains. Inside each EM iteration step, RAEMNB introduces an extra Ratio-adjustment step (R-step) between E and M step to keep the ratio of the predicted positive and negative instances during EM iteration consistent with the ratio of the target domain. To measure the distance of a predicted document and positive (or negative) labeled instances, we use the Kullback-Leibler divergence [5].

We crawled more than 130,000 online reviews of four categories from Amazon.com to construct 12 domain adaptation tasks for evaluation. In our experiments, we compare our proposed algorithm RAEMNB with traditional supervised, semi-supervised algorithms and ANB [13]. Experimental results show that RAEMNB outperforms all other algorithms with our Amazon online review dataset.

The rest of the paper is organized as follows. Section 2 presents our RAEMNB algorithm. Section 3 evaluates the performance of RAEMNB by comparing with previous supervised, semi-supervised and ANB. Section 4 discusses related work. Finally, Section 5 concludes the paper with future work.

2 Ratio-Adjusted EM-Based Naïve Bayes Algorithm

This section presents our algorithm for sentiment classification. Due to limited labeled instances from the target domain, our approach builds an initial Naïve Bayes classifier from labeled instances of both the source and the target domains. Then EM algorithm is employed to improve the initial classifier. During each EM iteration, we introduce an extra R-step to adjust the ratio for predicted instances during the expectation step, where the ratio is estimated with labeled data in the target domain.

Before describing our algorithm in detail, we define the notations used in this paper in Table 1.

2.1 Train the Initial Naïve Bayes Classifier

Our EM-based Naïve Bayes algorithm first trains the initial Naïve Bayes classifier from labeled instances in both source and target domains. Here we assume the number of labeled instances in the source domain is much larger than the one of the target domain, i.e.,

Table 1. Notations used in the paper

Notations	Description
D_{sl}	Source domain labeled data
D_{tl}	Target domain labeled data
D_{tl}^{+}	Positive instances in D_{tl}
D_{tl}^{-}	Negative instances in D_{tl}
D_{tu}	Target domain unlabeled training data
D_{tu}^{test}	Target domain unlabeled data for testing (different from D_{tu})
D_l	Labeled data from both source and target domain (=$D_{sl} \cup D_{tl}$)
D_t	Target domain data (=$D_{tl} \cup D_{tu} \cup D_{tu}^{test}$)
N_{pos}	numbers of predicted positive instances in D_{tu}
N_{neg}	numbers of predicted negative instances in D_{tu}

Algorithm 1. Ratio-Adjusted EM-based Naïve Bayes

Input: Training set D_{sl}, D_{tl}, and D_{tu}
Output: Naïve Bayes model $\hat{\theta} = \{P_D(C), P_D(W \mid C)\}$

Build initial Naïve Bayes model: $\theta = \{P_{D_l}(C), P_{D_l}(W \mid C)\}$
while the performance improves with estimated $\hat{\theta}$ **do**
  (E Step) Label the document d in the D_{tu} with the model $\hat{\theta}$
  (R Step) Calculate $\gamma = \frac{N_{pos}}{N_{neg}}$
  Let $\hat{\gamma} = \frac{|D_{tl}^{+}|}{|D_{tl}^{-}|}$
  if $\gamma > \hat{\gamma}$ **then**
    for each predicted positive instance d_i **do**
      $LM(d_i) = KL(D_{tl}^{+} \mid\mid d_i) - KL(D_{tl}^{-} \mid\mid d_i)$
    end for
    Sort the sequence $LM(d_i)$ $(1 \leq i \leq N_{pos})$ in decreasing order
    Change the label of first $\triangle N_{pos} = \frac{N_{pos} - \hat{\gamma} \cdot N_{neg}}{1+\hat{\gamma}}$ instances from positive to negative
  else $\{\gamma < \hat{\gamma}\}$
    for each labeled negative instance d_i **do**
      $LM(d_i) = KL(D_{tl}^{-} \mid\mid d_i) - KL(D_{tl}^{+} \mid\mid d_i)$
    end for
    Sort the sequence $LM(d_i)$ $(1 \leq i \leq N_{neg})$ in decreasing order
    Change the label of first $\triangle N_{neg} = \frac{\hat{\gamma} \cdot N_{neg} - N_{pos}}{1+\hat{\gamma}}$ instances from negative to positive
  end if
  (M Step) Re-train the Naïve Bayes classifier to acquire new $\hat{\theta}$
end while

$$\lambda = \frac{|D_{sl}|}{|D_{tl}|} > 1. \tag{1}$$

The initial Naïve Bayes model is calculated with the following formulas:

$$P(c_k) = \frac{\sum_{d_i \in D_l} P(c_k \mid d_i)}{|D_{sl}| + |D_{tl}|}, c_k \in C \tag{2}$$

and

$$P(w_i \mid c_k) = \frac{1 + n(w_i, c_k)}{|W| + n(c_k)}, c_k \in C, w_i \in W \tag{3}$$

where $\mathrm{P}(c_k \mid d_i)$ is 1 if d_i is in category c_k, otherwise 0. Here W is the word set and C is the set of categories.

2.2 Ratio-Adjusted EM Steps

Traditional EMNB algorithm often assumes the distributions of labeled and unlabeled data are identical, which leads to classification errors. For instance, assume the actual ratio of the positive instances to negative ones is 1:1 and the Bayes model is trained from data with a distribution ratio of 2:1. Thus, more instances will be predicted as positive during each EM iteration, resulting in low accuracy [15].

Our approach addresses the above problem by introducing an extra R-step between the E and M steps. The extra R-step, i.e., Ratio-adjustment Step, adjusts the labels predicted in the E step so that the distribution ratio is consistent with its real value. We define

$$\hat{\gamma} = \frac{|D_{tl}^{+}|}{|D_{tl}^{-}|} \tag{4}$$

$$\gamma = \frac{N_{pos}}{N_{neg}} \tag{5}$$

The predicted distribution γ is adjusted to be close to the actual distribution of D_t. Since the actual distribution for D_t is unknown, our approach is to use the distribution ratio in D_{tl}, i.e., $\hat{\gamma}$, as an approximation. Assuming labeled instances of D_{tl} are randomly sampled from D_t, such an approximation is acceptable and our experiments in Section 3.6 confirm this. Specifically, if $\gamma > \hat{\gamma}$, which indicates that some predicted positive instances are actually negative, then we should adjust some predicted positive instances to be negative. Conversely, if $\gamma < \hat{\gamma}$, we should adjust some predicted negative instances to be positive.

We use Likelihood Measure (LM) to estimate if a document d is more close to positive instances or negative instances:

$$LM(d) = sgn(\gamma - \hat{\gamma}) \cdot (KL(D_{tl}^{+} \mid\mid d) - KL(D_{tl}^{-} \mid\mid d)) \tag{6}$$

Recall that D_{tl}^{+} and D_{tl}^{-} represent positive and negative instances of D_{tl}, respectively. It is natural to estimate the relevance between d and D_{tl}^{+} (or D_{tl}^{-}) with a relevance metric. Kullback-Leibler (KL) divergence [5], is a measure of distance between two probability functions and is used as relevance metric in our algorithm. KL(·) is defined as follows,

$$KL(D_{tl}^{+} \mid\mid d) = \sum_{w \in d} P(w \mid D_{tl}^{+}) \cdot \log \frac{P(w \mid D_{tl}^{+})}{P(w \mid d)} \tag{7}$$

$$KL(D_{tl}^{-} \mid\mid d) = \sum_{w \in d} P(w \mid D_{tl}^{-}) \cdot \log \frac{P(w \mid D_{tl}^{-})}{P(w \mid d)} \tag{8}$$

The complete algorithm is described in Algorithm 1. To determine whether the current model improves, we use the same metric as Nigam et al. [6]:

$$l(\hat{\theta} \mid \hat{D}_l) = \sum_{j=1}^{|C|} \log P(c_j \mid \hat{\theta}) \prod_{k=1}^{|W|} P(w_k \mid c_j; \hat{\theta}) + \sum_{d_i \in \hat{D}_l} \sum_{j=1}^{|C|} z_{ij} \log P(c_j \mid \hat{\theta}) P(d_i \mid c_j; \hat{\theta}) \tag{9}$$

where $\hat{D}_l$ is labeled positive and predicted positive instances in target domain, $z_{ij} = 1$ if the class label of d_i is c_j, otherwise $z_{ij} = 0$.

3 Evaluation

3.1 Dataset

We crawled more than 130,000 product reviews from Amazon.com within four categories: Books, Grocery, Movie and Sports Instruments. These reviews are scored within the range of 1 to 5 by users, with higher scores representing more positive feedbacks. In the experiments, we assume reviews with scores greater than three are positive and ones whose scores are less than three are negative. Table 2 illustrates the distributions of our crawled data. For domain adaptation tasks, each of the four category can be the source domain and the other three are target domains. Thus, we have a total of 12 domain adaptation problems.

The training and testing sets of each category are generated as follows. We randomly sample 10% instances as labeled data (D_{tl}) and randomly selected another 20% as testing data (D_{tu}^{test}). The rest 70% data is used as unlabeled training data (D_{tu}). Labeled data from source domain is also randomly selected, subjecting to the following limit, i.e., $|D_{sl}| = \min\{\lambda \cdot |D_{tl}|, |D_{sl}|\}$.

We preprocessed the crawled data before applying learning algorithms. Specifically, words are stemmed with the Porter Stemmer [11] and stop words are filtered from texts. Then feature selection method is applied — we employ Document Frequency (DF). As suggested by [18], DF is a simple feature selection method and has a comparable performance with Information Gain and CHI. In the experiments, we keep terms whose DF value is greater than three. In the end, we have 7,248 unigram terms.

Table 2. Positive instance and negative instance distribution of four product types

Product Type	Positive #	Negative #	Ratio of positive to negative
Books	56,377	10,396	5.42
Grocery	13,659	1,818	7.51
Movie	25,463	3,006	8.47
Sports	18,185	2,186	8.32

3.2 Evaluation Metric

The evaluation metric for experiments is accuracy, which is defined as:

$$Accuracy = \frac{TruePositives + TrueNegatives}{TotalNumberOfInstances}. \tag{10}$$

3.3 Overall Performance

This experiment compares the performance of our RAEMNB algorithm with other classifiers. For supervised baseline, we selected Naïve Bayes and SVM. Both classifiers only use D_{tl} as training data. For the SVM algorithm, we employ LibSVM [3] and all parameters are set to the default values. For semi-supervised baseline, EMNB [6] and ANB [13] are chosen for comparison. EMNB use both labeled data D_{tl} and unlabeled data D_{tu} for training. For ANB, the labeled training data is from D_{sl} and D_{tl} while the unlabeled data is from D_{tu}; parameter N_{fce} and δ are 500 and 0.2, respectively, as suggested in [13]. For our RAEMNB, λ is set to 10. All these classifiers are tested with the data from D_{tu}^{test}.

Table 3 shows the results of supervised and semi-supervised baseline algorithms. Table 4 shows the ANB and RAEMNB algorithms results. Naïve Bayes and SVM perform poorly even though both their training and testing data are from the same domain. This is mainly because the number of training data is very limited. The accuracy for Books category of Naïve Bayes is much higher

Table 3. Accuracies of Naïve Bayes, SVM, and EMNB over four product types with limited training instances

Product Type	Naïve Bayes	SVM	EMNB
Books	72.90%	69.55%	67.41%
Grocery	49.35%	68.15%	67.88%
Movie	50.28%	68.53%	67.37%
Sports	50.18%	68.76%	67.58%
Average	55.68%	68.75%	67.56%

Table 4. Accuracies of ANB & RAEMNB over 12 domain adaptation problems

Domain Adaptation Problems	ANB	RAEMNB
Books → Grocery	45.51%	74.13%
Books → Movie	79.70%	84.11%
Books → Sports	36.44%	81.96%
Grocery → Books	76.34%	80.33%
Grocery → Movie	58.43%	82.07%
Grocery → Sports	64.10%	82.31%
Movie → Books	78.10%	79.74%
Movie → Grocery	30.23%	76.23%
Movie → Sports	25.91%	83.36%
Sports → Books	77.70%	79.04%
Sports → Grocery	42.93%	75.55%
Sports → Movie	71.00%	84.49%
Average	57.20%	80.28%

compared with other categories. The reason is that the average length of texts of Books category is much longer than others. As the training data contains more vocabularies than others, Naïve Bayes classifier is trained better for the Books category. Our RAEMNB has an average accuracy of 80.28%, outperforming the semi-supervised EMNB and ANB by about 13% and 23%, respectively. The reason for the inferior performance of the semi-supervised algorithm, EMNB, is that the labeled data is very limited in our experiments so it cannot be fully trained. For ANB, it cannot achieve a high accuracy because the distributions of source domain and target domain are different. While in our RAEMNB, this problem is addressed by the ratio-adjustment step during EM iterations.

3.4 Study on the Effectiveness of R-Step and Sensitivity of λ

This experiment studies the effectiveness of ratio-adjustment step of our RAEMNB algorithm. We compare the performance of RAEMNB initial model, initial model with standard EM iterations, and RAEMNB whose R-step is with LM and Naïve Bayes (NB) ranking (i.e., the probability values predicted by NB). For each scheme, we vary the number of instances in D_{sl} to change λ and the average accuracies of all 12 domain adaptation tasks are illustrated in Figure 1.

We can observe that the RAEMNB scheme performs significantly better than the approach using standard EM iterations, which is in turn better than the RAEMNB initial model. The RAEMNB perform the best because they can adjust the distribution of predicted instances, thus avoiding the drawback of traditional EM algorithm. Additionally, the performance of RAEMNB (with

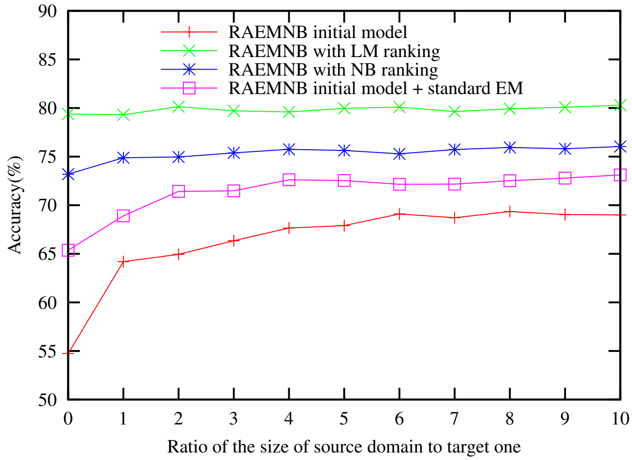

Fig. 1. The accuracies of RAEMNB with LM ranking, RAEMNB with Naïve Bayes ranking, RAEMNB initial model, and initial model with standard EM when λ is changing

either LM or NB ranking) remains stable with varying number of labeled data in the target domain, even when λ is 10. This indicates that our RAEMNB are effective with relatively lower number of labeled instances in the target domain.

This experiment shows that LM ranking outperforms NB ranking with different λ values. The lower performance of NB ranking can be mainly attributed to the imperfect quality of the current Naïve Bayes classifier. In comparison, LM ranking uses relevance between a document and labeled datasets, thus to some extent calibrates the wrong prediction of Naïve Bayes. As a result, our RAEMNB algorithm chooses LM as the ranking method in the R-step.

3.5 Study on the Convergence of RAEMNB

This experiment studies the convergence of our RAEMNB algorithm. Figure 2 illustrates the accuracies of all domain adaptation problems for different iterations. It can observed that our RAEMNB converges in less than five iterations, which indicates the extra R-step does not break the convergence of the traditional EMNB. In particular, we can observe that the first iteration usually has the most significant performance improvement. The reason is that the initial model was established with little target domain training data, thus is not very accurate. On the other hand, this shows that the extra R-step is effective for performance improvement.

3.6 Study on Sensitivity of $\hat{\gamma}$

In RAEMNB, parameter $\hat{\gamma}$ from D_{tl} is estimated as the distribution for the target domain (D_t). Because D_{tl} is only a fraction of target domain data, such

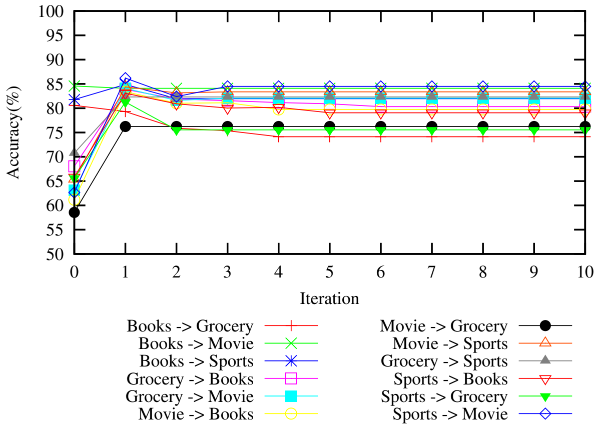

Fig. 2. The accuracies of twelve domain adaptation problems for different iterations ($\lambda = 10$)

an approximation may have some impact on the performance of RAEMNB. Thus, this experiment is designed to study the sensitivity of $\hat{\gamma}$.

To obtain a reasonable range for parameter $\hat{\gamma}$, we perform the following experiment: for a given sampling ratio of target domain data, we random sample D_t 100 times and calculated the $\hat{\gamma}$ value. The results are illustrated in Figure 3. From the figure, we can observe that when sampling ratio is large, the range of $\hat{\gamma}$ becomes smaller. The upper and lower bound for $\hat{\gamma}$ always happens when sampling ratio is smallest (0.02), because samples are more biased.

Then, we use the upper and lower bounds of $\hat{\gamma}$ obtained above, along with the actual distribution value, and study accuracies of RAEMNB for 12 domain adaptation tasks. Table 5 illustrates the average accuracy values with different λ values. For both lower bound and upper bound, the performance is comparable to the one using actual distribution.

In summary, this experiment demonstrates that our RAEMNB is insensitive to parameter $\hat{\gamma}$. In other words, RAEMNB effectively only needs to sample a small amount data from the target domain.

4 Related Work

Previous research [4,10,16] has applied traditional supervised or semi-supervised algorithms for sentiment classification. The domain adaptation problem of has often been studied. ANB [13] is an extension of EMNB [6] for sentiment classification, where co-occurring features from both source and target domains are used to build an initial Naïve Bayes model. Domain adaptation is achieved by increasing of the knowledge from target domain during EM iterations. ANB assumes the distributions of two domains are identical, thus limiting its performance. Structural Correspondence Learning (SCL) [2] employs pivot features to

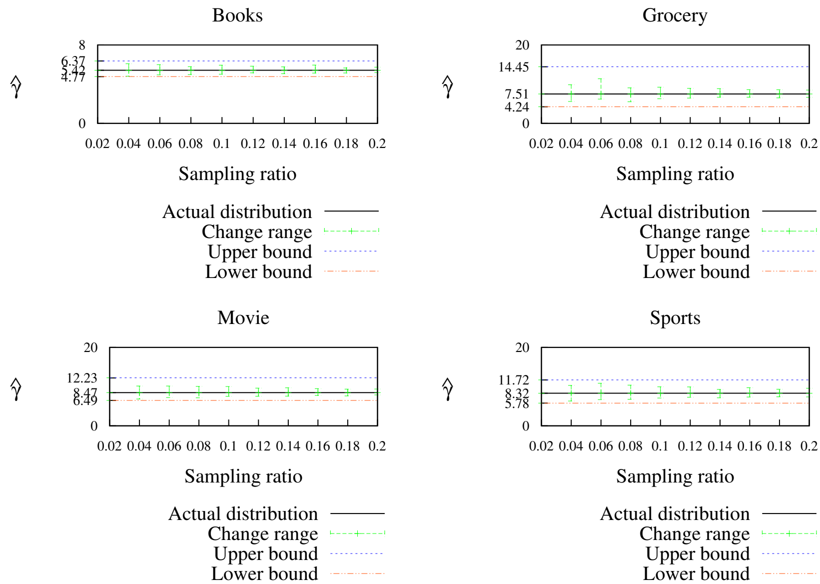

Fig. 3. The range of $\hat{\gamma}$ with respect to different sampling ratios. Each range is obtained by sampling D_{tl} 100 times.

Table 5. The impact of the deviation of $\hat{\gamma}$ to the average accuracy over 12 domain adaptation problems with different λ

λ	**Upper bound**	**Lower bound**	**Actual distribution**
0	79.27%	79.18%	80.86%
1	80.37%	80.01%	79.39%
2	79.95%	79.80%	80.31%
3	79.39%	79.75%	80.96%
4	79.99%	80.09%	80.15%
5	79.85%	80.18%	80.30%
6	80.43%	79.80%	80.42%
7	80.44%	79.39%	80.05%
8	80.92%	79.08%	80.00%
9	79.99%	80.17%	80.45%
10	80.08%	80.20%	80.28%

find the correspondences of features from source and target domains and trains a domain adaptation classifier with pivot and non-pivot features. W-SCL [12] improves SCL by assigning smaller weights to high-frequency domain-specific features and larger weights to the instances whose label is the same as the one of involved pivot features. Spectral Feature Alignment (SFA) [7] employs the domain-independent features as a bridge to align domain-specific features from different domains into the unified cluster. In this way, SFA minimizes the gap between different domains. These approaches perform domain adaptation with consideration of domain-specific and domain-independent features, while our RAEMNB employs a R-step to adjust distributions. Transfer learning [8,17] is another way to solve the domain adaptation problem, which often studies how to classify texts into multiple topics. Our work focuses on sentiment classification, which is often considered to be more challenging [9].

5 Conclusion

In this paper, we have proposed a new semi-supervised classifier, RAEMNB, for sentiment classification with domain adaptation. RAEMNB enhances traditional EMNB [6] algorithm with an additional ratio-adjustment step during each EM iteration so that the distribution of predicted instances does not deviate from real distribution much. We have compared RAEMNB with traditional supervised, semi-supervised classifiers. Our experiments on a dataset of 12 domain adaptation tasks demonstrate that our RAEMNB algorithm performs better than other algorithms. Particularly, even though our estimation of real distribution of target domain data is from a small randomly sampled fraction, experiments show that our algorithm is robust with estimation errors.

Currently, RAEMNB converges over 12 domain adaptation tasks, which indicates the convergence from an empirical perspective. In future work, we plan to study the convergence of RAEMNB from a theoretical perspective. Another direction is to apply ration adjustment for other classifiers and traditional learning applications.

Acknowledgment

We thank anonymous reviews for their comments. This work is supported in part by the National Natural Science Foundation of China (Grant No. 60811130528 and 60725208). Jingyu Zhou is the corresponding author.

References

1. Aue, A., Gamon, M.: Customizing Sentiment Classifiers to New Domains: a Case Study. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing, pp. 207–218 (2005)

2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification. In: ACL. pp. 187–205 (2007)
3. Chang, C.C., Lin, J.: LIBSVM: A Library for Support Vector Machines. Software (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Kennedy, A., Inkpen, D.: Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. In: Computational Intelligence, vol. 22, pp. 110–125 (2006)
5. Kullback, S., Leibler, R.A.: On Information and Sufficiency. In: The Annals of Mathematical Statistics, vol. 22, pp. 79–86 (1951)
6. Nigam, K., Mccallum, A.K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents Using EM. In: Machine Learning, vol. 39, pp. 103–134 (2000)
7. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain Sentiment Classification via Spectral Feature Alignment. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 751–760 (2010)
8. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. Technical report. (2009), http://doi.ieeecomputersociety.org/10.1109/TKDE.2009.191 (in press,2009)
9. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. In: Foundations and Trends in Information Retrieval, vol. 2, pp. 1–135 (2008)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment Classification Using Machine Learning Techniques. In: EMNLP 2002: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86 (2002)
11. Porter, M.F.: An Algorithm for Suffix Stripping. In: Program, vol. 14, pp. 130–137 (1980)
12. Tan, S., Cheng, X.: Improving SCL Model for Sentiment-Transfer Learning. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 181–184 (2009)
13. Tan, S., Cheng, X., Wang, Y., Xu, H.: Adapting Naïve Bayes to Domain Adaptation for Sentiment Analysis. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 337–349. Springer, Heidelberg (2009)
14. Thomas, M., Pang, B., Lee, L.: Get out the Vote: Determining Support or Opposition from Congressional Floor-debate Transcripts. In: EMNLP 2006: Proceedings of the ACL 2006 Conference on Empirical Methods in Natural Language Processing,Association for Computational Linguistics. pp. 327–335 (2006)
15. Tsuruoka, Y., Tsujii, J.: Training a Naïve Bayes Classifier via the EM Algorithm with a Class Distribution Constraint. In: Proceedings of the Seventh Conference on Natural language Learning at HLT-NAACL, pp. 127–134 Association for Computational Linguistics. Morristown (2003)
16. Turney, P.D.: Thumbs up or Thumbs down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, pp. 417–424 (2002)
17. Xue, G.R., Dai, W., Yang, Q., Yu, Y.: Topic-bridged PLSA for Cross-domain Text Classification. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 627–634 (2008)
18. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of ICML 1997, 14th International Conference on Machine Learning, Nashville, US, pp. 412–420. Morgan Kaufmann Publishers, San Francisco (1997)