Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 553-560
doi:10.4028/www.scientific.net/AEF.6-7.553

A Multi-level Naïve Bayes Classifier for Sentiment Classification

Junzheng Shi [a], Lei Guo[b] and Shimin Wei[c]

Automation School, Beijing University of Posts and

Telecommunications, Beijing 100876, People's Republic of China

[a]shijunzheng@bupt.edu.cn, [b] guolei@bupt.edu.cn, [c] wsmly@bupt.edu.cn

Keywords: sentiment classification; naïve Bayes Classifier; multi-mixture model

Abstract. A great demand of sentiment classification comes with the rapid development of the internet. At present, the methods about sentiment classification based on machine learning have been widely used. The sentiment classification is a more difficult task, which needs more in-depth study than the traditional topic-based classification method [1]. Naïve Bayesian classifier is widely used in text classification. However, it requires two basic assumptions as its prerequisite and the performance would have been poor if these two were dissatisfied. We propose a multi-level naïve Bayes classifier to make up the deficiency of the traditional naïve Bayes classifier. The research below shows that the multi-level naïve Bayes classifier gets better performance than the traditional naïve Bayes classifier on the sentiment classification of movie reviews.

Introduction

Compared to traditional text classification based on topic, text sentiment classification refers to the classification of viewpoint and attitude. That is, classifying the text into positive Category and negative Category.

[2] introduced the methods for text sentiment classification based on machine learning, we use the corpus of [2]. In [3], the authors extracted the subjective sentences from the text, as materials for classification. With the same corpus as [2], the classification accuracy was significantly improved in [3] Nonetheless, the corpora containing subjective sentences and objective sentences are required. The experiment of [4] shown the sentiment classification got poor performance when the training set and testing set comes from different categories, It is also suggested the need of specific subjective/objective sentences corpus for sentiment classification. [5] proposed a two-level classifier, we benefited from their ideas about multi-level classification.

There are two basic assumptions in the naïve Bayesian classifier. Assumptions 1, feature independent assumption, which means the features are independent with each other, when there are interdependencies between features, this assumption is false. Assumptions 2: every category is generated from one specific mixture model. This assumption, however, is often not satisfied in practice. [6] has improved the classification accuracy of EM algorithms through breaking limitations of the assumption 2. There are two cases when the category and mixture model are not completely corresponding. Case 1: the same category of training set and testing set corresponds with different mixture model, e.g. the training set is about movie reviews and the testing set is about product reviews. In this case, if the classifier trained by the former were used to classify the latter, the classification results would be bad. Case 2: every category of training and testing set is respectively generated by the same two or more mixture models, e.g. both the training and testing set contain movie reviews and product reviews. In this case, the positive and negative categories of comments are generated by different mixture models, the corresponding relations between categories and mixture models cannot be established. A way to get better performance is to train and testing for different category separately in case 2, Above all, the classification accuracy is reduced in both cases.

In the sentiment classification, the distributions of features in reviews with explicit attitudes and those with implicit attitudes are different, thus we introduce an assumption that the reviews with different modes of expression are generated by different mixture model. This paper studies this assumption and puts forward a new multi-level naïve Bayes classifier (MNB) to improve the performance of classification in this assumption.

The article is organized as follows. Section II introducing the theory of naïve Bayes classifier briefly. In Section II describes the multi-level naïve Bayes classifier. Experimental settings and results are commented in Section III .Section IV concludes the paper and indicates the activities for future work.

The naïve Bayes classifier

The basic idea of the naïve Bayes classifier (NB) is to make use of product of the probability of category and the probability of feature to get the probability of category which the text belongs. According to the formula of naïve Bayes, the probability of text D belongs to the category C_i is

$$P(C_i|D) = \frac{P(D|C_i) \times P(C_i)}{P(D)}. \tag{1}$$

The predicting category C_i of text D can be expressed as

$$C_i = argmax(P(C_i|D)). \tag{2}$$

The object of training phase is to get parameter set θ which can be expressed as

$$\theta = \{\theta_{t_i|C_j} : \theta_{t_i|C_j} = P(t_i|C_j), t_i \in V; \theta_{C_j} : \theta_{C_j} = P(C_j), C_j \in C\}. \tag{3}$$

The multi-level naïve Bayes classifier

The introduction of the multi-level naïve Bayes classifier. According to the assumption of the naïve Bayes classifier, the training and testing set are generated by the same mixture model. However, the assumption is usually not satisfied. We try to get better performance through the improvement of naïve Bayes classifier when the categories generated by different mixture models are linearly separable.

First we propose the definition of posteriori probability ratio (*PPR*).

$$PPR = \frac{P(C_i|D)}{P(C_j|D)}, i \neq j, C = \{C_i, C_j\}. \tag{4}$$

If $PPR > 1$, the text is classified as category C_i, else, if $PPR < 1$, the text is classified as category C_i, $PPR = 1$ is the demarcation point between C_i and C_j.

Then we propose a definition of distinction of orderly posterior probability (DOPP).

Definition 3.1 If we map all the possible samples which are generated by two different mixture models to the X axis by their *PPR* and they could be distinguished in a certain extent, that is to say the two mixture models meet the definition of DOPP. If mixture model A and mixture model B meet DOPP and the absolute value of samples' expected values generated by A are greater than those generated by B, it means mixture model A is outside mixture model B and mixture model B is inside mixture model A. The category demarcation point is taken as original point in this definition.

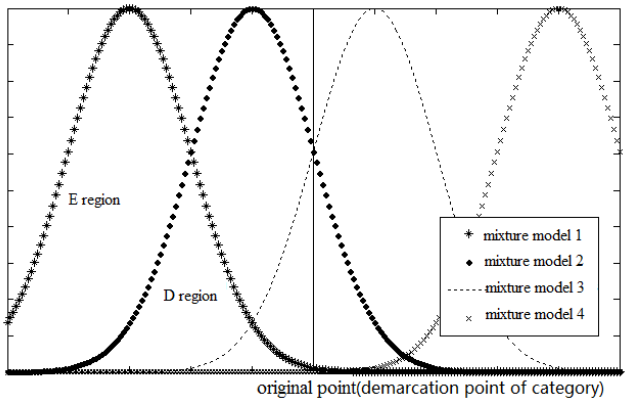

Fig. 1. An example of the mixture models meet DOPP.

Fig. 1 shows an example of 4 Gaussian mixture models meet DOPP. The different mixture models on both sides of the origin are considered as the same one based on the assumption of naïve Bayes classifier; as a result, the outboard mixture model may affect the inboard mixture model in the classification. We try to find a pair of demarcation points of which the outboard individuals are excluded when classifying the individuals generated by the inboard mixture models.

The following definition and algorithm are based on this assumption which is unproven: the naïve Bayes classifier gets better performance when the training and testing set are generated by the same mixture model than those are not. This assumption has been confirmed by experiment in many literatures, e.g. in [4,7,8].

Discussion about the demarcation point. Base discussion above, to remove the impact of outboard mixture model by removing the texts outside a specific pair of demarcation points, the new training set should reduce the number of individuals that generated by outboard mixture model (out-individual) as much as possible. At the same time, the number of individuals (co-individual) that generated both by the outboard mixture model and inboard mixture model shouldn't be reduced seriously. Otherwise, the classification of co-individuals will perform worse due to lacking of training. As shown in the Fig. 1 D region. On the other hand, we must consider the balance problem of the new training set inside the specific pair of demarcation points, thus, the two demarcation points are mutually restrained. The two demarcation points are called as 'training demarcation point'.

Due to the absence of the training of outboard mixture model, the new classifier doesn't work well in the testing of classifying the out-individuals. We should reduce the testing of out-individuals to reduce the error. As shown in the Fig. 1 E region. Because the primary purpose is to improve the classification accuracy of whole testing suite, we can't decrease the number of training set that given to the new classifier seriously. A solution is to find a demarcation point in the testing set region that mapped to number axis. The individuals in the inner side of the demarcation point meets requires of ours. The demarcation points are called as 'testing demarcation point'.

The definition of training demarcation point and testing demarcation point are shown as following.

*Definition 3.1.*Assume each individual in the overall $d_i \in D$ that only belonging to one of the category sets $C = \{C_j, C_k \mid j \neq k\}$. Set D is set partitioning by its proper subsets $D_i = \{CLASS(d_i) = C_j\}$, $D_j = \{CLASS(d_i) = C_k\}$. Map each individual d_i in D to X axis according to their *PPR* value between $[min(PPR_D), max(PPR_D)]$, assume the original point's *PPR* value is PPR_c. There are some mixture models in C_j of which every two satisfy the DOPP, which are $HM_{j_1}, HM_{j_2} \ldots HM_{j_m}$ from outside to inside. In C_k there are also some mixture models like in C_j that satisfy the DOPP, which are $HM_{k_1}, HM_{k_2} \ldots HM_{k_n}$ from outside to inside.

Definition 3.2. Take set D as the training set. Find two points x_{train_l}, x_{train_r} during the region $[min(PPR_D), max(PPR_D)]$, take all individuals in $[x_{train_l}, x_{train_r}]$ as training set of the new naïve Bayes classifier to make the classifier work best in all samples with the same prior probability

that generated by mixture models $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ alone or together. The individual set in $[x_{train_l}, x_{train_r}]$ is called as 'optimized training set' of mixture models $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ about training set D, x_{train_l} and x_{train_r} are called as 'training demarcation point' of mixture models $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ about training set D. $[x_{train_l}, x_{train_r}]$ is the corresponding training set region.

Definition 3.3. Take set D as the testing set. Find two points $x_{testing_l}, x_{testing_r}$ during the region $[x_{testing_l}, x_{testing_r}]$, make the correct classified ones in $[x_{testing_l}, x_{testing_r}]$ which were classified by the classifier trained by all the possible individuals generated by the mixture models $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ and the correct classified ones out of $[x_{testing_l}, x_{testing_r}]$ which were classified by the same method achieve the highest number, then the set of all individuals in the region $[x_{testing_l}, x_{testing_r}]$ is called as 'optimized testing set' of the mixture model $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ about set D, $x_{testing_l}$ and $x_{testing_r}$ are called as 'testing demarcation point' of the mixture model $HM_{j_l}, HM_{j_{l+1}} \ldots HM_{j_m}$; $HM_{k_h}, HM_{k_{h+1}} \ldots HM_{k_n}$ about set D, $[x_{testing_l}, x_{testing_r}]$ is the corresponding testing set region.

General multi-level naïve Bayes text classification algorithm. Base on the analysis above, we propose a multi-level naïve Bayes classification algorithm about text classification problem.

Algorithm 3.1. Training algorithm. In the text classification problem, the training text sets D_{train} are generated by some mixture models that fit the definition of DOPP. Take $D = D_{train}$, $i = 0$.

Step 1. Train the parameter set θ_i of D, and record them. Classify D by taking θ_i as parameter using Eq. 1. Calculate the *PPR* value of all $d_i \in D$, the demarcation point between categories is PPR_c.

Step 2. Sort all the text in D according to their *PPR* value, find training demarcation points x_{train_l}, x_{train_r} separately in $[min(PPR), PPR_c]$ and $[min(PPR), PPR_c]$. If anyone of the training demarcation point cannot be found, classify and mark all the texts in the region $[min(PPR), max(PPR)]$;otherwise, only classify and mark the ones in $[min(PPR), x_{train_l}]$ and $[x_{train_r}, max(PPR)]$, then find and record the testing demarcation points $x_{\theta_{il}}$ and $x_{\theta_{ir}}$ in $[x_{train_1}, x_{train_2}]$.

Step 3. Remove all texts that have been marked. If there is no text left, quit the algorithm; otherwise, all the left texts form a text set D_{left}, let $D = D_{left}$, $i = i + 1$, and return to step 1.

Algorithm 3.2. Classification algorithm. In the text classification problem, the training text set $D_{testing}$ are generated by some mixture models that fit the definition of DOPP. Take $D = D_{testing}$, $i = 0$.

Step 1. Take θ_i as the parameter set. Use Eq. 2 to classify D and calculate the *PPR* value of all $d_i \in D$.

Step 2. Sort all the texts in D according to their *PPR* value. Set j increasing from i+1 one by one to find the first parameter set θ_j in $[x_{\theta_{jl}}, x_{\theta_{jr}}]$ that fulfills $min(PPR) < x_{\theta_{jl}}, x_{\theta_{ir}} < max(x_{\theta_{ir}})$. Classify and mark texts in region $[min(PPR), x_{\theta_{il}}]$ and $[x_{\theta_{ir}}, max(PPR)]$ if set θ_j exist. Otherwise classify and mark all the texts.

Step 3. Remove all texts that have been marked. If there is no text left, quit the algorithm. Otherwise all the left texts form a text set D_{left}, let $D = D_{left}$, $i = j + 1$, and return to step 1.

The algorithm proposed in this paper is an approximate algorithm. The error of training set will increase with the iteration which causes the deviation of parameter set θ_i. The actual number of iteration is determined by the model.

The approximate algorithm of finding the demarcation point. It's hard to get the concrete parameters of mixture model. Fortunately, what the algorithm concerns about is the training demarcation point and testing demarcation point. We can get these points by the approximate algorithm.

Take D_{train} as the training text set, set the fold of cross-validation to N, do cross-validation M times, the texts in D_{train} which is for the training of the classifier is called D_{train_1}, the corresponding testing texts is called D_{train_2},both the numbers of the training and testing sets are K. Take the average classification accuracy of D_{train_2} in M times cross-validation as AvgAcc. The optimized objective function is expressed as.

$$O\begin{pmatrix} algorithm3.1, D_{train_1}, algorithm3.2, D_{train_2} \\ param = \{(x_{train_{il}}, x_{train_{ir}}), (x_{\theta_{il}}, x_{\theta_{ir}}) \end{pmatrix} = Max(AvgAcc). \quad (5)$$

It means getting training and testing demarcation point sets with maximal AvgAcc by using the optimized algorithm. The demarcation parameter of the set is expressed as $\{(x_{train_{il}}, x_{train_{ir}}), (x_{\theta_{il}}, x_{\theta_{ir}}) \mid 2 < i < K, K \in N\}$.

Algorithm 3.3. The approximate algorithm of the demarcation point. We introduce the simulated annealing algorithms (SA) to handle approaching to the demarcation point. In algorithm 3.3, K_{high} is the utmost number of demarcation point region. Let $j = 2$, $j < K_{high}$, $AvgAcc = 0$, param be null.

Step 1. If $j > K_{high}$, quit the algorithm.

Step 2. Set the initial temperature T.

Under the restriction of definition 3.2 and 3.3, generate randomly a parameter set $param_j$ which consist of 2*j demarcation point pairs. The parameter set $param_j$ is shown as $\{(x_{train_{il}}, x_{train_{ir}}), (x_{\theta_{il}}, x_{\theta_{ir}}) \mid 0 < i < j\}$.

Step 2.1. Do M times N-fold cross-validation according to the algorithm 3.1 and 3.2 in D_{train} with $param_j$. Get the classification accuracy $AvgAcc_j$.

Step 2.2. If $T = 0$ or the algorithm has converged, quit step 2.

Step 2.3. Randomly choose a testing demarcation point pair $(x_{\theta_{il}}, x_{\theta_{ir}})$ or a training demarcation point pair $(x_{train_{il}}, x_{train_{ir}})$, make the both side of the chosen point shrink or expand 1 individual, each individual represents a text. Update $param_j$ to $param_j^{'}$, and use $param_j^{'}$ as parameter set to do the step 2.1 to get the corresponding classification accuracy $AvgAcc_j^{'}$.

If $AvgAcc_j^{'} > AvgAcc_j$, update the $param_j^{'}$ as the new parameter set, i.e. $param_j = param_j^{'}$ and $AvgAcc_j = AvgAcc_j^{'}$.

If $AvgAcc_j^{'} < AvgAcc_i$, the probability of taking $param_j^{'}$ as the new parameter is expressed as

$$p = e^{(-(AvgAcc_j - AvgAcc_j^{'})*\mu/T)}. \quad (6)$$

Make temperature T down with cool rate T_{cool}, return to step 2.2.

Step 3.Let j=j+1.If $AvgAcc_j > AvgAcc$, let $AvgAcc = AvgAcc_j$, $param = param_j$. Return to step 1.

The time complexity of algorithm 3.3 is high. It can be simplified by the followed method. Divide the training set and testing set region into equal W part according to the *PPR* value. In step 2.3, make both sides of the demarcation point shrink or expand 1 part which is $(Max(PPR) - Min(PPR)) / W$.

To accelerate the speed of convergence, the parameter μ in Eq. 6 could be adjusted to increase tolerance of poor solution. The algorithm 3.3 turns to be hill-climbing algorithm when p always equal to 0.

The multi-level naïve Bayes classifier in the text sentiment classification

The baseline naïve Bayes classifier. We choose the Polarity Dataset v2.0, which was used by B.Pang as the corpus of the experiment in [2,3]. We only focus on the features based on unigrams with which the naïve Bayes classifier get the best performance [2]. Polarity Dataset v2.0 contains 1000 positive movie reviews and 1000 negative movie reviews, we set the prior probability of each category to 50%, adopt 3-fold cross-validation and consider the expected value of 10 executions as the result. To get the baseline naïve Bayes classifier, we try different feature extraction methods, different weighting calculation method of the feature and determine whether to stem the features or not.

In Fig. 2, CHI stands for Chi-Square feature extraction method, ALL stands for full features extraction method (Take any term as feature, including punctuation.); TF stands for adopting term frequency as the weight of feature and IDF stands for adopting term frequency-inverse document frequency as the weight of feature. STEM stands for stemming the features.

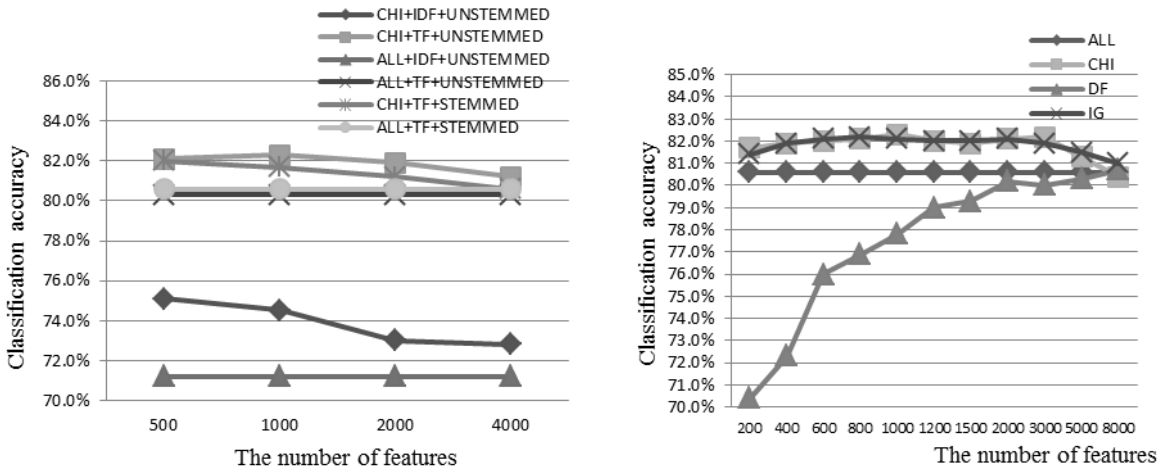

Fig. 2. The left figure shows the performance of the NB in different feature extraction methods and different pretreatment methods, the right figure shows the performance of the NB with different weighting methods of feature.

The left figure in Fig. 2 shows the classifier gets better performance when adopts TF as the weight of feature than that adopts IDF and gets better performance when does not stem the feature. All The experiments in the right figure in Fig. 2 adopts TF as the weight of feature, obviously, the classifiers which adopt CHI feature extraction method and information gain (IG) feature extraction method get better performance than those adopt full (ALL) feature method and document frequency (DF) feature extraction method. The maximum value of the accuracy rate reaches up to 82.3% when the classifier keeps the 1000 highest score features selected by CHI and adopts TF as the weight of feature. In the following experiment we set the NB with CHI as the feature extraction method and TF as the weight of feature as the baseline classifier, which keeps the 1000 highest score features.

Simulations of the MNB. The language model is complicated, so we find the approximate demarcation point by algorithm 3.3. Since the high time complexity of algorithm 3.3, experiment in this paper employs the degraded version of it, i.e. the hill-climbing algorithm. Divide the testing and training set into equal 20 parts according to *PPR* value. The utmost number of demarcation point region is 4. Do 1 time 3-fold cross-validation to get every $AvgAcc_j$ value. Quit the algorithm until it is convergent.

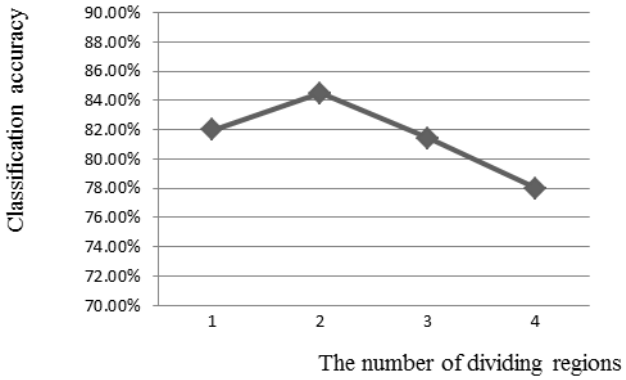

Fig. 3. The highest classification accuracy rates of different demarcation regions.

Fig. 3 shows the highest classification accuracy rate of different demarcation regions. K=1 is the result of the baseline classifier, whose accuracy rate reaches 82.0%. The accuracy rate of K=2 is the highest: 84.5%. The region set of the training sets is $\{(w_{1_l}=0, w_{1_r}=20),(w_{2_l}=7, w_{2_r}=13)\}$ and the region set of the testing sets is $\{(w'_{1_l}=0, w'_{1_r}=20),(w'_{2_l}=7, w'_{2_r}=13)\}$ when K=2.

In order to verify this conclusion further, we will calculate the average classification accuracy in any possible scene by using 3-fold cross-validation when K=2, W=20.

In Fig. 4, the X axis represents the position of training demarcation point, Y axis represents the position of testing demarcation point and Z axis represents the classification accuracy rate. X=0 is the experiment result of the baseline classifier. According to the discussion in 4.2, the case that testing set region is bigger than testing set region is unreasonable, it would be discarded.

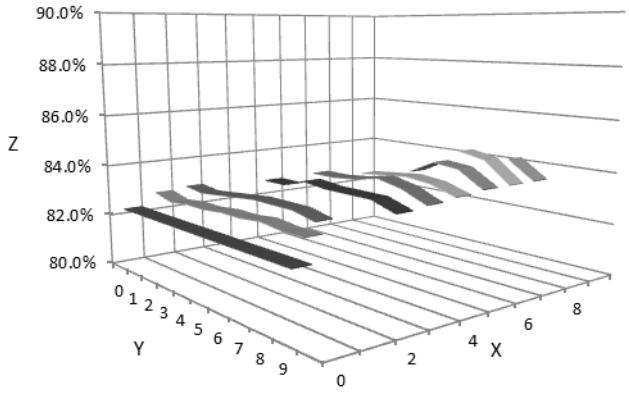

Fig. 4. Compare the MNB with the baseline classifier when K=2.

In Fig. 4, the performance of multi-level Bayes classifier is better than the baseline classifier. When X=7，Y=7, its accuracy gets peak: 84.7% and 2.5 percentage point higher than the baseline classifier's accuracy which is 82.3%. Multi-level Bayes classifier improves the classification accuracy of movie reviews.

The priori knowledge got from the outboard 'direct-style' mixture model disturbs the classification of texts generated by inboard 'tactful type' mixture model can be the explanation in a certain extent. For example, some unigram feature acts totally different roles in different mixture models, but whose value is obviously affected by outboard mixture models.

Conclusion

This paper proposed a definition that multi-level mixture model corresponding to the same category in the classification problem and the mixture models could be distinguished by the orderly posterior probability. According to that definition, we put forward a classifier named multi-level naïve Bayes

(MNB) to promote the performance of classification. The MNB's performance is significantly better than the comparative baseline naïve Bayes classifier in the task of movie reviews classification. In [3], the accuracy of classification on the same corpus reached up to 86.5% by means of the technique of subjective sentence extraction, however, the method this paper introduces promotes the accuracy from different angles and need no subjective/objective sentence training set. In the future, we will consider the fusion of the two methods.

This work was supported by National Natural Science Foundation of China (Grand no. 61105103)

References

[1] B.Pang, L.Lee. Opinion mining and sentiment analysis. Foundation and Trends in Information Retrieval. 2008.2(1-2).1-135

[2] B.Pang, L.Lee, and S.Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques, Proceedings of EMNLP Philadelphia, USA: ACL, 2002:79-86.

[3] B.Pang, L.Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the Association for Computational Linguistics. Barcelona, Spain.2004.Morristown, USA. Association for Computational Linguistics.2004.271-278

[4] Zhang Yan-bo. Research of Text Sentiment Classification [Dissertation]. Beijing. Beijing Jiaotong University, 2010

[5] FAN Xing-Hua, SUN Mao-Song. High Performance Two-Class Chinese Text Categorization Method. CHINESE JOURNAL OF COMPUTERS, 2006,(01) .124-131

[6] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning, Vol. 39, and pp.:103-134

[7] John Blitzer, Ryan McDonald, Fernando Percira. Domain adaptation with structural correspondence learning[C]. Processings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, July 2006, PP.120-128.

[8] Dai W. Y, GR Xue, Q. Yang, and Y. Yu. Transferring naive Bayes classifiers for text classification. In Proceedings of the 22nd Conference on Artificial Intelligence (AAAI), pages 540–545, 2007.