Movie Review Classification Based on a Multiple Classifier*

Kimitaka Tsutsumi[a], Kazutaka Shimada [a] and Tsutomu Endo[a]

[a]Department of Artificial Intelligence,
Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502 Japan
{k_tsutsumi, shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract. In this paper, we propose a method to classify movie review documents into positive or negative opinions. There are several approaches to classify documents. The previous studies, however, used only a single classifier for the classification task. We describe a multiple classifier for the review document classification task. The method consists of three classifiers based on SVMs, ME and score calculation. We apply two voting methods and SVMs to the integration process of single classifiers. The integrated methods improved the accuracy as compared with the three single classifiers. The experimental results show the effectiveness of our method.

Keywords: Sentiment analysis, p/n classification, Integration, Movie reviews, WWW.

1. Introduction

The World Wide Web contains a huge number of on-line documents that are easily accessible. Finding information relevant to user needs has become increasingly important. The most important information on the Web is usually contained in the text. We obtain a huge number of review documents that include user's opinions for products. For example, buying products, users usually survey the product reviews. Movie reviews are likewise one of the most important information for users who go to a movie. More precise and effective methods for evaluating the products are useful for users.

Many researchers have recently studied extraction of evaluative expressions and classification of opinions (Kobayashi et al., 2005; Osajima et al., 2005; Pang et al., 2002; Turney, 2002). Studies of opinion classification are generally classified into three groups: (1) classifying documents into the positive (p) or negative (n) opinions, (2) classifying sentences into the positive or negative opinions, and (3) classifying words into the positive or negative expressions. In this paper, we focus on the classification of movie review documents. Pang et al. (2002) have reported the effectiveness of applying machine learning techniques to the p/n classification. They compared three machine learning methods: Naive Bayes, Maximum Entropy and Support Vector Machines. In their experiment, SVMs produced the best performance. Osajima et al. (2005) have proposed a method for polarity classification of sentences in review documents. The method is based on a score calculation process of word polarity and outperformed SVMs in the sentence classification task.

The previous studies, however, used only a single classifier for the classification task. We (Tsutsumi 2006 et al.) have proposed a method consisting of two classifiers: SVMs and the scoring method by Osajima et al. (2005). The method identified the class (p/n) of a document on

 This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 18500115, 2007.

the basis of the distances that were measured from the hyperplane of each classifier. It obtained the better accuracy as compared with the single classifiers. This method, however, contained a problem for the determination of the final output, namely positive or negative. We needed to normalize the classifier's outputs manually because the scale of the scoring method was different from that of SVMs.

To solve this problem, we apply the 3rd machine learning method (Maximum Entropy) into the method based on the scoring and SVMs. Figure 1 shows the outline of the proposed method. In this paper, we compare three processes for the method: naive voting, weighted voting and determination with SVMs.

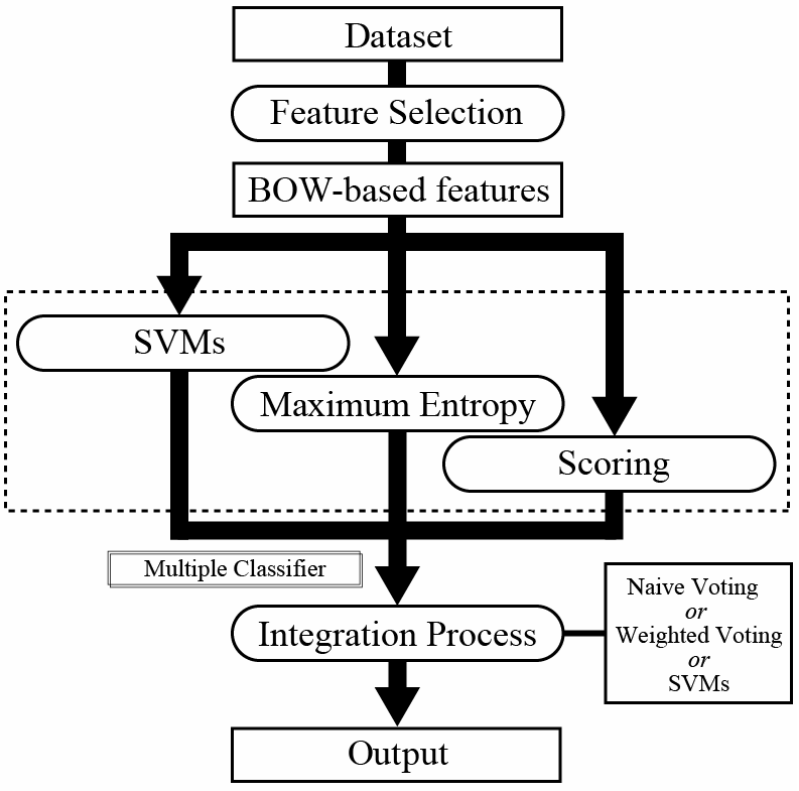

Fig. 1.The outline of our method.

2. Classifiers

In this section, we explain classifiers for a multiple classifier that will be proposed in this paper. The 1st and 2nd classifiers are SVMs and Maximum Entropy respectively. These classifiers have been used in related work by Pang et al. (2002). In their paper, they reported that SVMs produced the best performance. The 3rd classifier is based on polarity scores of words in documents. The method is an expansion of Osajima et al. (2005).

2.1.SVMs

SVMs are a machine learning algorithm that was introduced by Vapnik (1999). They have been applied to tasks such as face recognition and text classification. An SVM is a binary classifier that finds a maximal margin separating hyperplane between two classes. The hyperplane can be written as:

$$y_i = w \cdot x + b \quad (1)$$

where x is an arbitrary data point, i.e., feature vectors, w and b are decided by optimization, and $y_i \in \{+1,-1\}$. The instances that lie closest to the hyperplane are called support vectors. We use SVM^{light} package[1] for training and testing, with all parameters set to their default values (Joachims, 1999).

2.2.Maximum Entropy

Maximum entropy modeling (ME) is one of the best techniques for natural language processing (Berger et al., 1996). The principle of the ME is expressed as follows:

$$P_\Lambda(c \mid d) = \frac{1}{Z_\Lambda(d)} \exp\left(\sum_i \lambda_{i,c} f_{i,c}(d,c) \right) \quad (2)$$

$$Z_\Lambda(d) = \sum_{d,c} \exp\left(\sum_i \lambda_{i,c} f_{i,c}(d,c) \right) \quad (3)$$

where $Z_\Lambda(d)$ is a normalization function. $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ are parameters for the model. These parameters denote weights and significance of each feature. The parameter values are a set that maximizes the entropy concerning the classifier. $f_{i,c}(d,c)$ is a feature function that is defined as follows:

$$f_{i,c}(d,c') = \begin{cases} 1 & \textit{if } exist(d,i) > 0 \textit{ and } c' = c \\ 0 & \textit{otherwise} \end{cases} \quad (4)$$

where $exist(d,i)$ is a indicator function. The value is 1 in the case that a feature i exists in a document d.

In this paper we use Amis, which is a parameter estimator for maximum entropy models[2]. We estimate parameters by using the generalized iterative scaling algorithm.

2.3.Scoring

Osajima et al. (2005) have proposed a method for polarity classification of sentences in review documents. The method is based on a score calculation process of word polarity. In this paper we apply some adjustments to the feature selection of the method.

First we explain score calculation of each word. The score of a word w_i is computed as follows:

$$Score_W(w_i) = \log\left(\frac{pos(w_i)+1}{\sum pos} \times \frac{\sum neg}{neg(w_i)+1} \right) \quad (5)$$

where $pos(w_i)$ and $neg(w_i)$ are the frequency of a word w_i in the positive opinions and the negative opinions respectively. $\sum pos$ and $\sum neg$ are the number of words in the positive and negative opinions respectively.

In the classification process, we compute the sum of scores of which words appear in a document d.

$$Score(d) = \sum_{ALL w_i \in d} Score_W(w_i) \quad (6)$$

[1] http:/svmlight.joachims.org
[2] http:/www-tsujii.is.s.u-tokyo.ac.jp/amis/index.html

where d denotes a document. The $Score_W(w_i)$ are given by Eq. 5. Finally, we evaluate the score as follows:

$$d = \begin{cases} Positive & Score(d) > 0 \\ Negative & Score(d) \le 0 \end{cases} \quad (7)$$

As expansions, we apply two conditions to the feature selection of the classifier.

- Use of POS tags
  We use weighted scores for adjective words.
- Selection with χ^2 -test

  We select features for the classifier by using the result of χ^2 -test. We reject words that possess low reliability. We set 20% on the significant level for the experiment.

3. Integration of classifiers

We examined the outputs of three classifiers, i.e. an error analysis. As a result, we obtained knowledge that the misclassifications of each classifier are often different, that is exclusive misclassifications. In other words, a single classifier could classify a document correctly even though other single classifiers could not classify it correctly. This result shows the significance of integration of single classifiers.

In this section we explain three methods for combining the classifiers. In this paper we adopt two voting processes, naive voting and weighted voting, and integration with SVMs.

Naive voting As the final output, we use the majority vote from three classifiers, namely SVMs, ME and Scoring methods.

Weighted voting This method uses each distance from hyperplanes of each classifier as weights (confidence) of the outputs. This is based on a supposition that an output that is close to the hyperplane of a classifier contains low reliability. However ranges of each distance computed from each classifier are not equivalent. We normalize each distance as follows.

Scoring: The actual value of the output from the classifier

SVM: $dist(d) \times l$

ME: $(p(postive, d) - p(negative, d)) \times m$

where $dist(d)$ is the distance from the hyperplane. $p(postive, d)$ and $p(negative, d)$ are the probabilities of a document d as positive and negative opinions. *l* and *m* are constant numbers for the normalization. The values are computed beforehand from training data. In other words, the values are determined from the results obtained from training data by using each classifier constructed from it. *l* and *m* in this paper are based on the average of the distance from the hyperplane in the classification results.

SVMs We use SVMs again for the integration process of single classifiers. Here the features for SVMs are the three outputs of each single classifier, namely distances from hyperplanes. We do not normalize each output. First, we need to construct training data for SVMs in this integration process. In this paper we use the same training data for learning both the single classifier and the integration process. In other words, this SVM obtains a hyperplane for the integration from the training data that is used in the learning process of the single classifiers. Figure 2 shows the outline of the process.

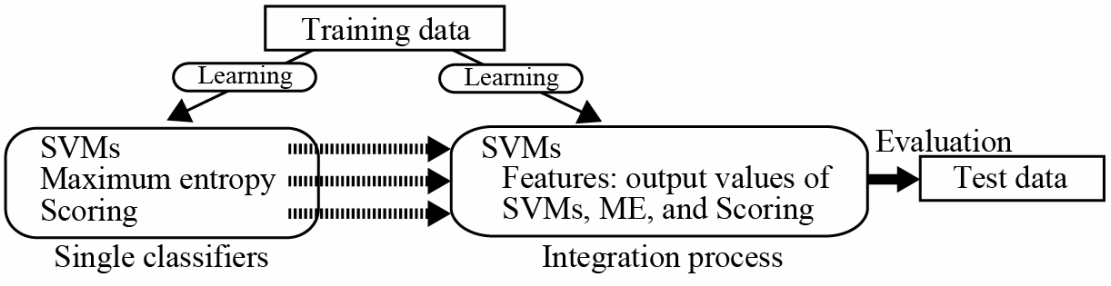

Fig. 2.The integration process with SVMs.

4. Experiment

In this section, we explain a dataset and basic feature selection for this experiment first. Then we evaluate our methods with the dataset and compare them with single classifiers.

4.1.Dataset and features for the classifiers

First we describe the data set for the experiment. We evaluated our method with movie review documents that were used in Pang et al. (2002). The dataset consists of 700 positive reviews and 700 negative reviews. We divided the data into three equal-sized folds in the same manner as the previous work (Pang et al., 2002). In other words, we tested our method with three-fold cross-validation.

We extracted basic features for classifiers in the same way the previous work did. The conditions of the previous work for the feature selection were as follows: (1) no stemming or stopword lists were used, (2) use of the negation tag, and (3) limitation of the frequency of words. For the 2nd condition, we added the tag "NOT_" to every word between a negation word ("not", "isn't", "didn't", etc.) and the first punctuation mark following the negation word[3]. For the 3rd condition, we used words appearing at least four times in our 1400-document corpus.

We constructed the feature space for each classifier on the basis of the basic features. For the scoring method, we used features extracted with the conditions described in Section 2.3 (use of POS tags and χ 2-test). For the tagging, we used the Brill's Tagger[4]. However, we used basic features for SVMs and ME[5].

4.2.Experimental results

First, we compared six methods in this experiment: single classifiers (SVMs, ME and Scoring) and the proposed method based on naive voting, weighted voting and SVMs. Table 1 shows the experimental result[6].

As a result, our methods, namely multiple classifiers, outperformed single classifiers. Even the naive voting produced higher accuracy than these methods. This result shows the effectiveness of our method consisting of some classifiers.

---

[3] For example, the sentence "It is not good" is converted to "It is not NOT_good". As a result, we can treat a polarity reversal correctly.

[4] http://www.cs.jhu.edu/ brill/

[5] The reason is that there was no significant difference between the methods with the conditions and without the conditions. The same tendency was reported in the previous work.

[6] Note that the results of SVM and ME in this paper differ from the accuracy described in Pang et al. (2002). These accuracies were based on our implementation. We think that the reason is that (1) negation words that we used might be not completely same as the previous work and (2) the tool and the iterative scaling method for ME also differed from the method of the previous work.

Table 1. The experimental result

The method		Accuracy
Single	SVM	82.2%
	ME	80.5%
	Scoring	83.4%
Multi	Naive Voting	85.8%
	Weighted Voting	86.4%
	SVM	87.1%

In this experiment, the integration method with SVMs produced the best performance. For the weighted voting, we used the values computed beforehand from training data: *l* and *m*. If we tuned up these values optimally, the accuracy was 87.5%. This result denotes that the weighting process was one of the most important processes in the multiple classifier. The weighting process in this paper was a very simple weighting method. It was based on the average of distance obtained from a tentative dataset.

One approach to improve the accuracy is refinements to the weighting for the voting process. Boosting is one of the most famous techniques in ensemble learning methods (Freund and Schapier, 1996). We applied the boosting algorithm to the integration process. In the method, SVMs, Scoring and ME are weak learners for the boosting. However the accuracy was lower than SVMs and the weighted voting method. To improve the accuracy, we need to add other machine learning methods as weak learners. Furthermore, we need to consider other ensemble learning methods, such as Bagging (Breiman, 1996) and Random Forests (Breiman, 2001).

In this paper we used BOWs for the feature set of the classifiers. Matsumoto et al. (2005) have argued the significance of syntactic relations between words, that is frequent word sub-sequences and dependency sub-trees. For their experiment with the same dataset, using dependency sub-trees led to the improvement of the accuracy: the accuracy was 87.3%. Bai et al. (2004) have proposed a new method based on a two-stage Bayesian algorithm that is able to capture the dependencies among words. The accuracy of this method was 87.5%. Although the feature set for our method was very simple and naive (BOWs), the accuracy was equivalent as compared with these methods that used relations between words. Furthermore, since our method can incorporate other single classifiers flexibly, applying them as single classifiers boosts up the accuracy rate of our method.

We also need to handle the confidence of each classifier's output appropriately. In this paper we regarded the confidence computed from the output of a classifier as a linear function. However, Platt (1999) has reported a method to convert the output of SVMs into probability by using the sigmoid function and showed the effectiveness as a probability function. We need to consider a method for converting the output to the best suited confidence.

Next we explain a coverage rate of the proposed method. Our method can not classify a document into positive or negative correctly if every single classifier for the multiple classifier classifies it incorrectly. In other words, our method contains the possibility that it can classify a document correctly if a single classifier identifies the class label (p/n) of a document correctly. Therefore the coverage of our method for this dataset is computed as follows:

$$Coverage = \frac{CorrectNum}{N}$$

where N is the number of documents in test data. $CorrectNum$ is the number of documents that are classified correctly by at least one classifier. The result is shown in Table 2. The coverage with three classifiers was 94.9% although the best accuracy in this experiment was

87.1% (See Table 1). This result shows that our method, namely a multiple classifier, can improve the classification accuracy essentially.

Table 2. The Coverage

Combination	Coverage
SVM + Scoring	91.6%
ME + Scoring	91.8%
SVM + ME	92.2%
SVM+ME+Scoring	94.9%

5. Conclusion

In this paper, we proposed a method to classify movie review documents into positive or negative opinions. The method consisted of three classifiers based on SVMs, ME and score calculation. We compared two voting process, naive voting and weighted voting, and the integration with SVMs for the method. Our methods consisting of three classifiers outperformed single classifiers. Even the naive voting produced higher accuracy than these classifiers. In this experiment, SVMs in the integration process produced the best performance. If we tuned up the weights for the voting method optimally, it obtained the best accuracy. Our future work includes (1) evaluation with the ensemble learning methods, (2) addition of other single classifiers and (3) use of other feature sets, such as dependency trees, for classifiers

References

Bai, X., R. Padman, and E. Airoldi. 2004. Sentiment Extraction from Unstructured Text Using Tabu Search-enhanced Markov Blanket. In *Proceedings of the International Workshop on Mining for and from the Semantic Web (MSW2004).*

Berger, A. L., S. A. Della Pietra, and V. J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. 22(1):39–71.

Breiman, L. 1996. Bagging Predictors. *Machine Learning*, 24:123–140.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45:5–23.

Freund, Y. and R. E. Schapier. 1996. Experiments with a New Boosting Algorithm. In *Proceedings of ICML*, pages 148–156.

Joachims, T. 1999. Transductive Inference for Text Classification Using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning,* pages 200–209.

Kobayashi, N., R. Iida, K. Inui, and Y. Matsumoto. 2005. Opinion Extraction Using a Learning-based Anaphora Resolution Technique. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP- 05),* pages 175–180.

Matsumoto, S., H. Takamura, and M. Okumura. 2005. Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In *Proceedings of the 9th Pacific-Asia International Conference on Knowledge Discovery and Data Mining (PAKDD-05),* pages 301–310.

Osajima, I., K. Shimada, and T. Endo. 2005. Classification of Evaluative Sentences Using Sequential Patterns. In *Proceedings of the 11nd Annual Meeting of The Association for Natural Language Processing (in Japanese)*.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* pages 79–86.

Platt, J. 1999. Probabilistic Outputs for Support Vector Machines and Comparison to

Regularized Likelihood Methods. In B. Schoelkopf D. Schuurmans A.J. Smola, P. Bartlett, editor, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Tsutsumi, K., K. Shimada, and T. Endo. 2006. Sentiment Classification Using Reliability of Multiple Classification Results (in Japanese). In *SIG-FPAI-A601*, pages 27–32.

Turney, P. D. 2002. Thumbs up? or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Vapnik, V. N. 1999. *Statistical Learning Theory*. Wiley.