Semi-supervised Self-training for Sentence Subjectivity Classification

Bin Wang[1], Bruce Spencer[2], Charles X. Ling[3], and Harry Zhang[1]

[1] Faculty of Computer Science, University of New Brunswick
P.O. Box 4400, Fredericton, NB, Canada E3B 5A3
[2] National Research Council of Canada,
Fredericton, NB, Canada E3B 9W4
[3] Department of Computer Science, The University of Western Ontario
London, Ontario, Canada N6A 5B7
{bin.wang,hzhang}@unb.ca

Abstract. Recent natural language processing (NLP) research shows that identifying and extracting subjective information from texts can benefit many NLP applications. In this paper, we address a semi-supervised learning approach, self-training, for sentence subjectivity classification. In self-training, the confidence degree that depends on the ranking of class membership probabilities is commonly used as the selection metric that ranks and selects the unlabeled instances for next training of underlying classifier. Naive Bayes (NB) is often used as the underlying classifier because its class membership probability estimates have good ranking performance. The first contribution of this paper is to study the performance of self-training using decision tree models, such as C4.5, C4.4, and naive Bayes tree (NBTree), as the underlying classifiers. The second contribution is that we propose an adapted Value Difference Metric (VDM) as the selection metric in self-training, which does not depend on class membership probabilities. Based on the Multi-Perspective Question Answering (MPQA) corpus, a set of experiments have been designed to compare the performance of self-training with different underlying classifiers using different selection metrics under various conditions. The experimental results show that the performance of self-training is improved by using VDM instead of the confidence degree, and self-training with NBTree and VDM outperforms self-training with other combinations of underlying classifiers and selection metrics. The results also show that the self-training approach can achieve comparable performance to the supervised learning models.

1 Introduction

Many natural language processing (NLP) applications can benefit from identifying and extracting subjective information that expresses opinions and emotions from texts. For example, information extraction (IE) systems aim to extract facts related to a particular domain from natural language texts. Suppose we are looking for information about bombings and physical assaults in some news articles. From the sentence: "*The parliament exploded into fury against the government*

S. Bergler (Ed.): Canadian AI 2008, LNAI 5032, pp. 344–355, 2008.

when word leak out...", the IE system may report that a bombing took place and "The parliament" was the target of the bombing. But it is not correct because the verb "exploded" is used metaphorically. If we first do the subjectivity analysis, IE systems are not easily misled by the language that contains metaphors or hyperboles. For instance, the above sentence that describes negative emotions will be considered as a subjective sentence which frequently contains metaphors and hyperboles. The observations in [11] show that many incorrect extractions can be prevented by identifying subjective information and filtering extractions from them.

Machine learning models have attracted much attention when applied to subjectivity analysis. However, the subjective language can be expressed by various words and phrases, and many subjective terms occur infrequently. Subjectivity learning systems using supervised machine learning models must be trained on extremely large corpora that contain a broad and comprehensive subjective vocabulary. It is very time-consuming and expensive to collect and manually annotate a great amount of texts in corpora. Semi-supervised learning is a useful approach for reducing the effort devoted to obtaining the expensive training data. It initially builds a model with a small number of fully labeled instances and utilizes a large number of unlabeled instances to improve the model. Previous contributions to subjectivity analysis mainly focus on supervised machine learning models at the document-level. It is worthwhile to penetrate subjectivity study at the sentence-level using semi-supervised learning.

In this paper, we use self-training, a semi-supervised learning approache, to classify sentences as subjective or objective. Initially, an underlying classifier is trained using a small number of labeled sentences with all the features. Then the classifier classifies unlabeled sentences, and a selection metric is used to rank these classified sentences and to select some sentences that have high rankings to update the labeled training set. The procedure iterates until all the unlabeled sentences have been included into the training set or the maximum number of iterations is reached. The selection metric is crucial to the performance of self-training. The confidence degree is a popular selection metric, which depends on class membership probability estimates. Traditionally, naive Bayes (NB) is often used as the underlying classifier because the class membership probabilities produced from NB have the good ranking performance. In this paper, we study the performance of self-training using decision tree models, such as C4.5, C4.4, and naive Bayes tree (NBTree), as underlying classifiers. However, the class membership probabilities produced by decision tree classifiers do not have good ranking performance [4]. Therefore, we propose an adapted Value Difference Metric (VDM) [21] as the selection metric that does not depend on class membership probabilities. Based on Multi-Perspective Question Answering (MPQA) [19] corpus, a set of experiments have been designed to evaluate the performance of self-training with different underlying classifiers using different selection metrics under various conditions. The experimental results show that the performance of self-training is improved by using VDM as the selection metric instead of the confidence degree, and self-training with NBTree and VDM outperforms self-training with other combinations of underlying classifiers and selection metrics. The results also show that self-training can achieve comparable performance to the supervised learning models for sentence subjectivity classification. Although the study of self-training in this paper concentrates on sentence subjectivity classification, the approach can also be used for other applications of machine learning.

The rest of the paper is organized as follows. In Section 2, we present the related works on subjectivity study and semi-supervised learning methods. In Section 3, we introduce the self-training algorithm, underlying classifiers, and selection metrics. In Section 4, the experiments and results of sentence subjectivity classification are presented and analyzed. We summarize and propose the future research in Section 5.

2 Related Works on Subjectivity Analysis

Much research has appeared recently in the areas of opinion extraction, sentiment analysis, polarity classification, and subjectivity recognition. The work of subjectivity recognition mainly focuses on document-level classification. Turney et al. [15] propose methods for classifying reviews as positive or negative. Some research in genre classification has included recognition of subjective genres, for example, editorials and objective genres of business or news [18]. Subjectivity classification at sentence-level is more useful than at document-level. Most documents consist of a mixture of subjective and objective sentences. For example, newspaper articles are generally considered as relatively objective documents, but 44% of sentences in a news collection are found to be subjective [18]. Moreover, subjectivity classification at the sentence-level assists when identifying and extracting more complex subjectivity information, for example, the opinion expression, holder extraction, and opinion relationship.

Most previous methods of sentence-level subjectivity classification are developed by supervised learning approaches [12] [1]. One of the main obstacles for supervised learning methods is the lack of fully labeled training data. It is much more difficult to obtain collections of individual sentences that can be easily identified as subjective or objective. Previous work on sentence-level subjectivity classification [16] uses training corpora that had been manually annotated for subjectivity. Manually annotations are expensive and time-consuming so that only a relatively small amount of annotated sentences are available. This situation gives researchers motivation to explore the semi-supervised learning way to solve the task.

Ellen Riloff et al. have developed a boostrapping method to learn patterns for extracting subjective sentences [10]. They build two separated high-precision subjective and objective rule-based classifiers that utilize subjectivity clues to assign subjective or objective labels to sentences. The labeled sentences from two classifiers are represented by extraction patterns. The extraction patterns are learned by a fully automatic process similar to AutoSlog [8]. The subjective patterns generated by the pattern learner further label more unannoatated texts.

In a recent paper, Wiebe et al. extend the above work by replacing two rule-based classifiers with one NB classifier [17]. The procedure is similar to our method. But their work was restricted by using the confidence degree and applying NB model as the underlying classifier. Because the confidence degree is based on the differences among class membership probability estimates, other classifiers whose produced class membership probabilities have poor ranking performance are not suitable for this setting.

3 Self-training with Various Underlying Classifiers and Selection Metrics

3.1 General Algorithm of Self-training

Self-training, as a single-view semi-supervised learning method, has been widely used in NLP research [7] [13]. In self-training, an underlying classifier is first trained with a small number of labeled data which is also called the initial training set. The underlying classifier is used to classify the unlabeled data. The most confident unlabeled instances with their predicted labels are added to the training set. The underlying classifier is then re-trained and the procedure repeats. The following is the general procedure of self-training algorithm.

Algorithm Self-training

Input: L is labeled instance set, U is unlabeled instance set, C is underlying classifier, t is the number of times of iteration, θ is the number of selected unlabeled instances for next iteration, M is the selection metric, $S(U_t, \theta, C, M)$ is the selection function, and $maxIteration$ is the maximum number of iterations

Initial: $t = 0$, $L_t = L$, $U_t = U$, where L_t and U_t are the labeled and unlabeled instance set at the tth iteration

Repeat:

train C on L_t;
$S_t = S(U_t, \theta, C, M)$, where S_t is the selected unlabeled instance set;
$U_{t+1} = U_t - S_t$; $L_{t+1} = L_t + S_t$;
$t = t + 1$;

Until: (U_t is empty) $\vee$ ($maxIterations$ reached)

Note that the selection function is used to rank the unlabeled instances and select a certain number of unlabeled instances to update the training instance set for the next iteration. The function is not only influenced by the underlying classifier that should have good ranking performance, but also affected by the selection metric.

3.2 NB vs. Decision Tree as Underlying Classifiers

NB and decision tree classifiers have been commonly used in many machine learning applications. NB classifier is very fast for induction, and robust to irrelevant attributes. However, the strong conditional independence assumption often influences the performance of NB classifier. Decision tree classifiers are comprehensible and fast. The trees grow by choosing a split attribute recursively using some criterion from the root to leaves. In decision tree algorithms, C4.5 [5] executes a pruning step to reduce the tree size after a full tree is built. C4.4 [4] turns off the pruning and uses Laplace correction when producing the class membership probabilities. But as the underlying classifiers of self-training, decision tree classifiers face two obstacles to producing good ranking of instances: one is that the sample size on a leaf is small, and the other is that the instances falling into the same leaf are assigned to the same class membership probability.

Kohavi proposed the hybrid approach, NBTree [3]. NBTree is similar to the classical decision tree algorithms except that a NB classifier is deployed on the leaf nodes. NBTree combines the advantages of both NB and decision tree classifiers. Moreover, it deals with the above obstacles of decision tree classifiers. In the NBTree algorithm, a threshold is chosen that prevents the sample size on a leaf from being too small. A NB classifier is deployed on a leaf, which assigns the different class membership probabilities. Among the decision tree algorithms, NBTree is suitable to be used as the underlying classifier in self-training.

3.3 Confidence Degree vs. VDM as Selection Metrics

The selection metric used to rank and select classified unlabeled instances for the next iteration is crucial to the performance of self-training. Traditionally, the confidence degree is often used as the selection metric in self-training. The confidence degree ranks a classified unlabeled instance by the differences among its class membership probability estimates. Most previous works on self-training choose NB as the underlying classifier because the class membership probabilities produced from NB classifier have good ranking performance [2]. However, it has constrained the capability of self-training to apply other machine learning models whose class membership probabilities do not have good ranking performance.

In order to overcome the constraint, we propose adapting Value Difference Metric (VDM) [21] as the selection metric in self-training. The original idea of VDM is to evaluate the distance between instances from the differences among feature conditional probability estimates. Given two instances x and y, the VDM distance between them is defined as

$$VDM(x, y) = \sum_{i=1}^{C} \sum_{j=1}^{N} |P(c_i|a_j(x)) - P(c_i|a_j(y))|, \tag{1}$$

where C is the number of class labels, N is the number of features in instances, and $P(c_i|a_j(x))$ is the feature conditional probability of class i given the feature a_j's value in instance x.

Since self-training is an iterative procedure, it is very time-consuming to compute the VDM distance between a classified unlabeled instance and each labeled instance in the training set. We adapt VDM to compute the average VDM distance between an unlabeled instance and the training set. Given a feature value of an instance, the feature's conditional probability is compared with the probabilities of the corresponding feature for all possible values in the training set. The adapted distance function of VDM for a classified unlabeled instance x is defined as

$$VDM(x) = \sum_{i=1}^{C} \sum_{j=1}^{N} \sum_{k=1}^{M} |P(c_i|a_j(x)) - w_j^k P(c_i|a_j^k)|. \quad (2)$$

where M is the number of possible values of feature a_j in the training set, $P(c_i|a_j^k)$ is the feature conditional probability of class i given the kth value of feature a_j, and w_j^k is the proportion of the kth value in all the possible values of feature a_j within the training set. We rank an unlabeled instance higher if it has a smaller VDM distance.

Using VDM as the selection metric relaxes the constraint of the underlying classifier to depend on class membership probabilities, and provides the opportunity to apply decision tree classifiers in self-training. For example, C4.5 algorithm assigns the same class membership probabilities for the instances that fall into the same leaf node, which makes it difficult to rank and select such instances using the confidence degree. VDM is based on feature conditional probabilities that are different for instances even in the same leaf node. NBTree deploys NB models in leaf nodes and produces the different class membership probabilities. However, using VDM makes NBTree not restrict to the leaf nodes any more because the feature conditional probabilities are estimated in the whole training set.

4 Sentence Subjectivity Classification

4.1 Data

The benchmark data set for sentence subjectivity classification is hard to achieve because the sentences should be manually annotated and the annotation process is time-consuming and expensive. Recent research on subjectivity analysis for the English language uses the Multi-Perspective Question Answering (MPQA) corpus [1] as the benchmark [14] [17] [11]. The MPQA corpus consists of 535 news articles. These articles that are collected from a variety of countries and publications have been manually annotated. The sentences used in the following experiments also come from the MPQA corpus, where there are 11,112 sentences.

Private state is a concept that generally covers the components of subjectivity, such as opinions, beliefs, thoughts, feelings, emotions, goals, evaluations, and judgments [6]. Analysis of sentence subjectivity recognizes and characterizes expressions of private states in a sentence. Wiebe et al. put forward an annotation scheme for evaluating private state expressions [19]. Two kinds of private state frames have been proposed: one is the expressive subjective element frame, and the other one is the direct subjective frame. Both frames consist of several attributes. For example, the attribute *Intensity* indicates the intensity of

[1] http://www.cs.pitt.edu/mpqa/

the private state expressed in sentences, and the attribute *Insubstantial* denotes whether the private state is not real or not significant.

The gold-standard classes that are used to label the sentences as subjective or objective in this paper are the same as in other subjectivity research [17] [10] [12]. The gold-standard classes are defined as follows: a sentence is considered as subjective if (1) the sentence contains a direct subjective frame with the value of attribute *Intensity* NOT low or neutral, and NOT with an attribute *Insubstantial*; or (2) the sentence contains an expressive subjectivity frame with the value of attribute *Intensity* NOT low. Otherwise, the sentence is an objective sentence.

4.2 Structure of Sentence Subjectivity Classification

In the procedure of self-training for sentence subjectivity classification, the entire set of sentences are first put into the part of pre-processing, where OpenNLP [2] is used to tokenize sentences into a set of words, and assign part-of-speech (POS) tags to each word. In this part, the Abney stemmer of SCOL [3] is also used to stem words.

After pre-processing, sentences go through the part called feature maker where the features of sentences are built in terms of subjectivity clues and subjectivity patterns. The subjectivity clues are those which have been published with OpinionFinder [20]. They are divided into strongly subjective clues and weakly subjective clues. A strongly subjective clue is one that is always used with a subjective meaning, whereas a weakly subjective clue is one that commonly has both subjective and objective meanings. The subjectivity clues are matched with sentences according to the stemmed words and their POS tags. The subjectivity patterns consist of subjective patterns and objective patterns that are extracted by the Sundance information extraction system [9]. Using the extraction patterns defined in [10], Sundance searches and extracts the subjective patterns and objective patterns from sentences.

Next, we build the instances of sentences with features made from feature maker. Each instance representing a sentence includes several features: the strong subjective clues, the weak subjective clues, the subjective patterns, and the objective patterns. The following POS tags are also added into the feature set: pronouns, modal verbs (excluding "will"), adjectives, cardinal numbers, and adverbs (excluding "not"). In addition, the above features in the previous and next sentences are taken into account in the current sentence's feature set in order to incorporate the contextual information. All the features have three possible values (0, 1, $\geq$2) which are based on the presence of features in the corresponding sentence. Finally, the gold-standard classes are assigned to corresponding sentence instances as class labels (subjective or objective).

In the part of sentence set separation, all the sentence instances are separated into the train set and test set. The test set is held for evaluation. In the train

[2] http://opennlp.sourceforge.net/

[3] http://www.ivnartus.net/spa/

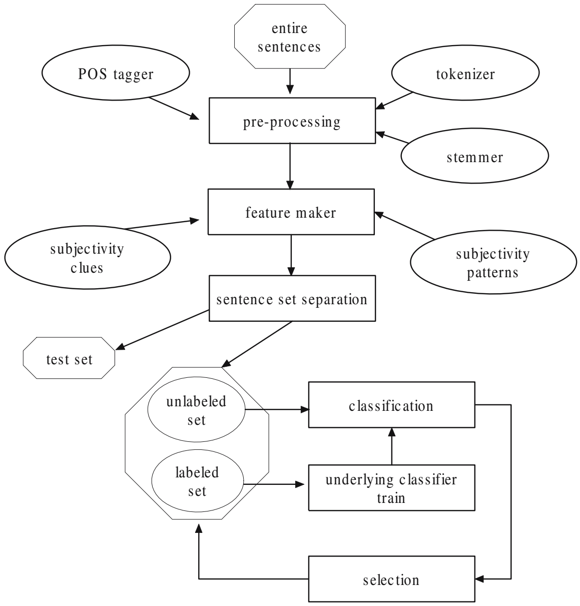

Fig. 1. Self-training process for sentence subjectivity classification

set, we hide the labels of most sentence instances to make the unlabeled instance set, and keep the remaining portion as the labeled instance set which is initially used to train the underlying classifier.

The underlying classifier is trained by the labeled instance set, and classifies sentences in the unlabeled instance set. Then, the unlabeled instances with predicted labels are ranked and a user-defined number of instances with top rankings are selected. The underlying classifier is trained again by the selected instances together with the original labeled instances. This iterative procedure is repeated until it runs out of all the unlabeled sentence instances or the maximum number of iterations is reached. The overall process is depicted in Figure 1.

4.3 Experiments

The experiments conducted on WEKA [22] machine learning environment compare the performance of self-training using different underlying classifiers and different selection metrics under various conditions. We implement the self-training structure and selection metric methods in WEKA, and utilize the implementations of NB, C4.5, C4.4 and NBTree in WEKA as the underlying classifiers. The experiments are evaluated by three kinds of measures: accuracy, the area under the ROC curve (AUC), and F-measure (F-M) which combines precision and recall. All the results of the evaluation measures are averages of 100 runs (10 runs of ten-fold cross validation) for the focused algorithm. Runs with the various algorithms are carried out on the same train sets and evaluated on the same test sets.

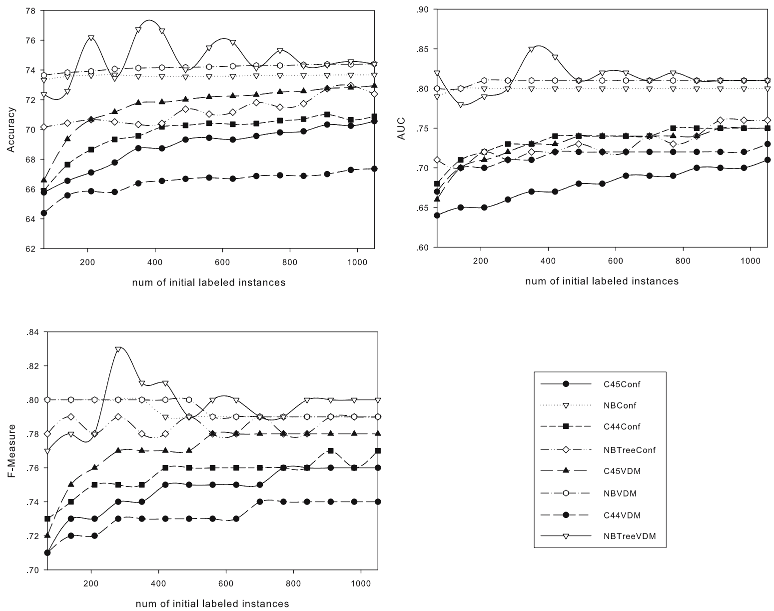

Fig. 2. The experimental results of self-training with different sizes of initial labeled instance sets

There are two factors which may influence the performance of self-training: one is the size of the initial labeled instance set, and the other one is the number of classified unlabeled instances selected for the next iteration. First, a set of experiments is developed for self-training with different sizes of initial labeled instance sets. Then, we design the experiments of self-training with different numbers of selected unlabeled instances for the next iteration. The experimental results are plotted in Figure 2 and Figure 3. Each curve in the two figures represents self-training with a kind of combination. For example, "C4.5Conf" represents self-training using C4.5 as the underlying classifier and the confidence degree as the selection metric. From Figure 2 and Figure 3, we can see that using VDM as the selection metric improves the performance of self-training with various underlying classifiers except C4.4. Self-training with NBTree and VDM outperforms self-training with other combinations of underlying classifiers and selection metrics, especially when the size of the initial labeled instance set is small.

Instead of assigning the same probability to the same leaf node, C4.5 with VDM needs only to consider the feature conditional probabilities that are different even within the same leaf node, which makes ranking on unlabeled instances perform better. NBTree is no longer required to generate class membership probabilities on leaf nodes. The larger instance space can be used to improve the ranking using VDM as the selection metric. However, the performance of C4.4 with VDM is not better than the one using the confidence degree. The original

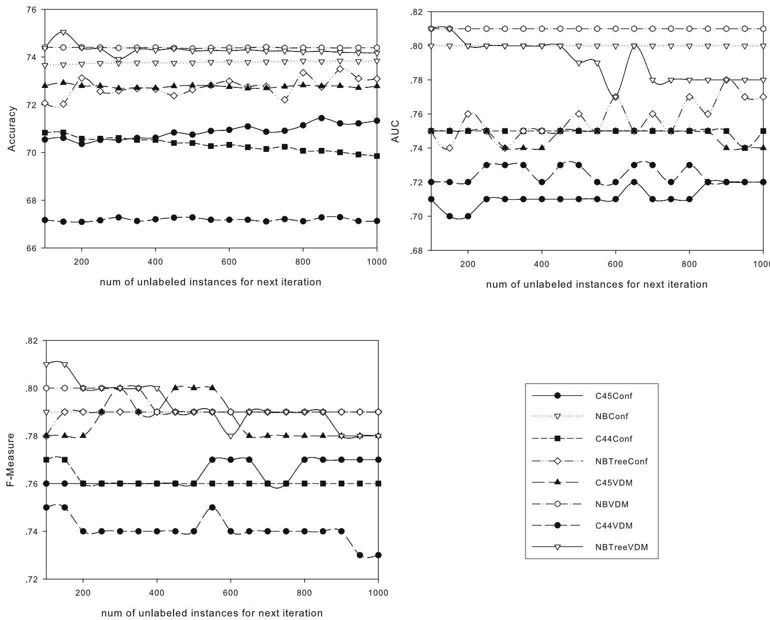

Fig. 3. The experimental results of self-training with different numbers of unlabeled instances for next iteration

Table 1. The results of baseline, self-training, and supervised learning classifiers

	C4.5			NB			C4.4			NBTree		
%	Acc	AUC	F-M	Acc	AUC	F-M	Acc	AUC	F-M	Acc	AUC	F-M
Base	63.88	62.65	69.10	69.47	75.00	74.86	61.32	64.24	68.26	66.97	70.81	72.91
Conf	68.74	67.87	74.56	73.59	80.00	78.10	67.64	73.26	74.69	70.34	72.93	78.46
VDM	71.79	73.00	77.05	74.13	80.96	78.68	66.38	71.24	73.09	74.74	81.42	78.79
Super	74.13	78.52	79.19	75.05	82.00	79.00	68.29	74.19	75.63	75.05	82.26	79.13

purpose of C4.4 is to improve the ranking performance of C4.5 by turning off pruning and using Laplace correction. But turning off pruning results in a large tree so that unlabeled instances with the same class label in the same leaf node share more identical feature values along the path from root to leaf. As a result, the large tree hurts the ranking performance of VDM that is based on feature conditional probability estimates. We also observe that, self-training with NB as the underlying classifier achieves better performance when using VDM as the selection metric.

In Table 1, the results of various evaluation measures on self-training are compared with the results from baseline and supervised learning classifiers. The results from baseline are obtained from the corresponding supervised classifiers that are trained by initial labeled set whose size is 350. The results from self-training show averages of 100 runs when the number of selected unlabeled instances for next iteration is 100 and the size of the initial labeled instance set is 350. From the results, we can see that the performance of self-training is better than the baseline and comparable with the performances of corresponding supervised learning classifiers for sentence subjectivity classification, especially when VDM is used as the selection metric.

5 Conclusion and Future Work

In this paper, we introduce a semi-supervised learning method, self-training, to solve the task of sentence subjectivity classification. Instead of focusing only on NB classifier, we bring decision tree classifiers into self-training as the underlying classifiers. However, the class membership probabilities produced by decision tree classifiers do not have good ranking performance. The traditional selection metric, the confidence degree, is not suitable when using decision tree classifiers as underlying classifiers. We adapt VDM as the selection metric in self-training, which does not depend on class membership probabilities. Based on MPQA corpus, a set of experiments have been designed to compare the performance of self-training with different underlying classifiers using different selection metrics under various conditions. The experimental results show that self-training with NBTree and VDM outperforms self-training with other combinations of underlying classifiers and selection metrics, and VDM improves the performance of self-training that uses NB, C4.5 and NBTree as underlying classifiers. The results also show that self-training can achieve comparable performance to the supervised learning models for sentence subjective classification.

In future work, we will extend our study of self-training to other applications of machine learning. The experiments on other benchmark data sets of machine learning will be done to see whether self-training with NBTree and VDM is better than other combinations of underlying classifiers and selection metrics.

References

1. Dave, K., Lawrence, S., Pennock, D.M.: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Produce Reviews. In: Proceedings of the 12th International World Wide Web Conference (WWW 2003) (2003)
2. Huang, J., Lu, J., Ling, C.X.: Comparing Naive Bayes, Decision Trees, and SVM using Accuracy and AUC. In: Proceedings of The Third IEEE International Conference on Data Mining (ICDM 2003) (2003)
3. Kohavi, R.: Scaling Up the Accuracy of Naive-bayes Classifiers: a decision-tree Hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (1996)
4. Provost, F., Domingos, P.: Tree Induction for Probability-based Ranking. Machine Learning 52, 199–216 (2003)
5. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., Los Altos, California (1993)
6. Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: A comprehensive Grammar of the Enligsh Language. Longman, New York (1985)

7. Reichart, R., Rappoport, A.: Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007) (2007)
8. Riloff, E.: Automatically Generating Extraction Patterns from Untagged Text. In: Proceedings of the AAAI 1996 (1996)
9. Riloff, E., Phillops, W.: An Introduction to the Sundance and AutoSlog Systems. Technical Report (2004)
10. Riloff, E., Wiebe, J.: Learning Extraction Patterns for Subjective Expressions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2003)
11. Riloff, E., Wiebe, J., Phillips, W.: Exploiting Subjectivity Classification to Improve Information Extraction. In: Proceedings of the 20th National Conference on Artificial Intelligence, AAAI Press, Menlo Park (2005)
12. Riloff, E., Wiebe, J., Wilson, T.: Learning Subjective Nouns Using Extraction Pattern Bootstrapping. In: Proceedings of the 7th conference on Natural Language Learning (CoNLL 2003), pp. 25–32 (2003)
13. Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S., Crim, J.: Bootstrapping statistical parsers from small datasets. In: Proceedings of the 11th conference of the European Association for Computational Linguistics (EACL 2003) (2003)
14. Seki, Y., Evans, D.K., Ku, K., Chen, H., Kando, N., Lin, C.: Overview of Opinion Analysis Pilot Task at NTCIR-6. In: Proceedings of NTCIR-6 Workshop Meeting (2007)
15. Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceedings of the ACL 2002 (2002)
16. Wiebe, J., Bruce, R., O'Hara, T.: Development and Use of a Gold Standard Data Set for Subjectivity Classifications. In: Proceedings of the ACL 1999 (1999)
17. Wiebe, J., Riloff, E.: Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, Springer, Heidelberg (2005)
18. Wiebe, J., Wilson, T., Bell, M.: Identifying Collocations for Recognizing Opinions. In: Proceedings of the ACL 2001 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation, pp. 24–31 (2001)
19. Wiebe, J., Wilson, T., Cardie, C.: Annotating Expressions of Opinions and Emotions in Language. Language Resources and Evaluation 39, 165–210 (2005)
20. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: OpinionFinder: Asystem for subjectivity analysis. In: Proceedings of HLT/EMNLP Interactive Demonstrations (2005)
21. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
22. Witten, I.H., Frank, E.: Data Mining-practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, San Mateo (2000)