Semi-supervised Text Categorization by Considering Sufficiency and Diversity

Shoushan Li[1,2], Sophia Yat Mei Lee[2], Wei Gao[1], and Chu-Ren Huang[2]

[1] Natural Language Processing Lab, School of Computer Science and Technology, Soochow University, China
[2] CBS, The Hong Kong Polytechnic University, Hong Kong
{shoushan.li,sophiaym,wei.gao512,churenhuang}@gmail.com

Abstract. In text categorization (TC), labeled data is often limited while unlabeled data is ample. This motivates semi-supervised learning for TC to improve the performance by exploring the knowledge in both labeled and unlabeled data. In this paper, we propose a novel bootstrapping approach to semi-supervised TC. First of all, we give two basic preferences, i.e., *sufficiency* and *diversity* for a possibly successful bootstrapping. After carefully considering the *diversity* preference, we modify the traditional bootstrapping algorithm by training the involved classifiers with random feature subspaces instead of the whole feature space. Moreover, we further improve the random feature subspace-based bootstrapping with some constraints on the subspace generation to better satisfy the *diversity* preference. Experimental evaluation shows the effectiveness of our modified bootstrapping approach in both topic and sentiment-based TC tasks.

Keywords: Sentiment Classification, Semi-supervised Learning, Bootstrapping.

1 Introduction

Text categorization (TC) aims to automatically assign category labels to natural language text (Sebastiani, 2002) and this task can be grouped into two major categories: topic-based text classification (Yang and Liu, 1997) (referred to as topic classification in the following) and sentiment classification (Pang et al., 2002). While the former classifies a document according to some objective topics, such as *education*, *finance*, and *politics*, the latter classifies a document according to some subjective semantic orientations, such as *positive* and *negative*. Nowadays, the most popular approach to both categories of TC tasks is based on supervised learning methods which employ large amounts of labeled data to train a classifier for automatic classification. However, it is often expensive and time-consuming to obtain labeled data. To overcome this difficulty, various semi-supervised learning methods have been proposed to improve the performance by exploiting unlabeled data that are readily available for most TC tasks (Blum and Mitchell, 1998).

In principle, an unlabeled document could be helpful for classification. Consider the following review from a corpus for sentiment classification:

G. Zhou et al. (Eds.): NLPCC 2013, CCIS 400, pp. 105–115, 2013.

Example 1: *This brand is the worst quality that I have purchased. I would avoid this brand.*

Each sentence in this review provides a strong indicator, i.e., "*worst quality*" and "*avoid this brand*", for predicting the review as a *negative* one. Assume that a trained classifier has already possessed the classification knowledge for predicting "*worst quality*" but got no idea about "*avoid this brand*". Once the review is correctly predicted and added into the labeled data for further learning, the classifier is then likely to contain the classification knowledge for predicting "*avoid this brand*". Therefore, when we iteratively label the unlabeled documents and use them to retrain the classifier, it is possible to introduce helpful knowledge in the unlabeled documents. This process is exactly a typical implementation of the semi-supervised learning approach named bootstrapping. Intuitively, this approach should be effective for semi-supervised TC since the information of many documents are often redundant for predicting categories, that is, there are usually more than one indicator for predicting the category label. Unfortunately, bootstrapping has been reported to be a poorly-performed approach for semi-supervised TC in previous studies. For example, as reported by Li et al. (2010), bootstrapping (called self-training therein) is one of the worst approaches for semi-supervised sentiment classification and fails to improve the performance across almost all the eight domains.

In this paper, we will change the awkward situation of bootstrapping. First of all, we give two basic preferences for a possibly successful bootstrapping, namely *sufficiency* and *diversity*. While *sufficiency* indicates the ability of the classifier for correctly predicting the class to enable a successful bootstrapping, *diversity* indicates the preference of adding unlabeled samples which better represent the natural data distribution. Specifically, to better satisfy the *diversity* preference, we use several feature subspace classifiers to automatically label and select samples instead of using a single classifier over the whole feature space. In this way, selected samples tend to be more different from the existing labeled data in terms of the whole feature space. Empirical studies demonstrate a great success of our novel bootstrapping approach by using feature subspace classifiers.

The rest of this paper is organized as follows. Section 2 reviews related work on semi-supervised TC. Section 3 describes the two preferences for a successful bootstrapping. Section 4 proposes some novel alternatives of bootstrapping with a focus on the *diversity* preference. Section 5 presents experimental results on both topic and sentiment classification. Finally, Section 6 gives the conclusion and future work.

2 Related Work

2.1 Topic Classification

Generally, two major groups of methods are exploited in topic classification: the first is Expectation Maximization (EM) which estimates maximum posteriori parameters of a generative model (Dempster et al., 1977; McCallum and Nigam, 1998; Nigam et al., 2000; Cong et al., 2004) and the second one is Co-training which employs two or multiple disjoint views to train a committee of classifiers to collectively select

automatically labeled data (Blum and Mitchell, 1998; Braga et al., 2009). Both of them have achieved great success on topic classification. To compare EM and Co-training, Nigam and Ghani (2000) present an extensive empirical study on two benchmark corpus: WebKB and 20News. The results show that EM performs slightly better than Co-training on WebKB while Co-training significantly outperforms EM on 20News. The general better performance of Co-training is due to its more robustness to the violated assumptions.

Except the two main groups, some studies propose other semi-supervised learning approaches for topic classification, such as transductive learning (Joachims, 1999) and SemiBoost (Mallapragada et al., 2009). All these studies confirm that using unlabeled data can significantly decrease classification error in topic classification.

2.2 Sentiment Classification

While supervised learning methods for sentiment classification have been extensively studied since the pioneer work by Pang et al. (2002), the studies on semi-supervised sentiment classification are relatively rare.

Dasgupta and Ng (2009) integrate several technologies, such as spectral clustering, active learning, transductive learning, and ensemble learning, to conduct semi-supervised sentiment classification. However, the obtained performance remains very low (the accuracies on Book and DVD domains are about 60% when using 100 labeled samples).

More recently, Li et al. (2010) propose a Co-training algorithm with personal/impersonal views for semi-supervised sentiment classification. Their experiments show that both self-training and tranductive learning completely fail and even in their co-training approach, incorporating unlabeled data is rather harmful on DVD domain.

Unlike both studies mentioned above, our bootstrapping approach is much more successful for semi-supervised sentiment classification and impressively improves the performance on Book and DVD domains when using 100 labeled samples.

3 Two Basic Preferences for Successful Bootstrapping

Bootstrapping is a commonly used approach for semi-supervised learning (Yarowsky, 1995; Abney, 2002). In bootstrapping, a classifier is first trained with a small amount of labeled data and then iteratively retained by adding most confident unlabeled samples as new labeled data.

To guarantee successful bootstrapping, two basic preferences should be reinforced. On one side, the classifier C in bootstrapping should be good enough to correctly predict the newly-added samples in each iteration as many as possible. Otherwise, many wrongly predicted samples would make bootstrapping fail completely. For clarity, we refer to this preference as *sufficiency*.

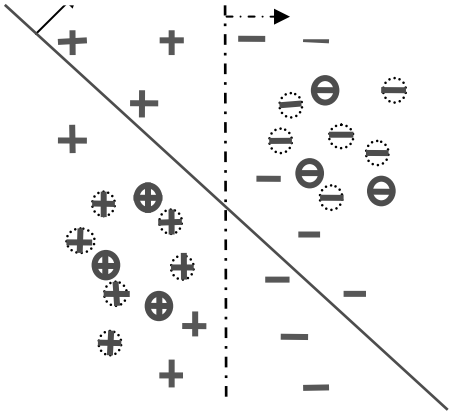

Fig. 1. Possible hyperplane (the solid red line) when the labeled samples are more concentrated

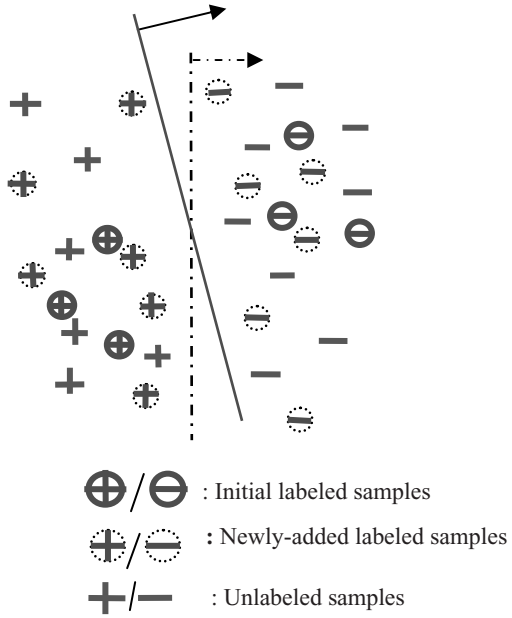

Fig. 2. Possible hyperplane (the solid red line) when the labeled samples are less concentrated

On the other side, traditional bootstrapping is prone to label the samples very similar to the initial labeled data in the initial several iterations because these samples could be predicted with much more confidence due to the small scale of the labeled data. However, labeling similar samples might be dangerous because the labeled data including the initial and the newly-added ones would violate the data distribution and

fail to obtain a good classification hyperplane. Figure 1 shows the trained hyperplane (the solid line) under the situation when the labeled data are concentrated. We can see that when the newly-added data is too close to the initial labeled data, the trained hyperplane might be far away from the optimal one (the dotted line). One possible way to overcome the concentration drawback is to make the added data more different from the initial data and better reflect the natural data distribution. Figure 2 shows the situation when the labeled data are less concentrated. In this case, the trained hyperplane would be much better. For clarity, we refer to this preference of letting newly-labeled data more different from existing labeled data as *diversity*.

4 Subspace-Based Bootstrapping for Semi-supervised TC

4.1 Feature Subspace in TC

A document is represented as a set of features $F = \{f_1,...,f_m\}$ in a machine learning-based method for TC. Assume $X = (X_1, X_2,..., X_n)$ the training data containing *n* documents and a document X_i is denoted as $X_i = (x_{i1}, x_{i2},..., x_{im})$ where x_{ij} is some statistic information of the feature f_j, e.g., tf, $tf \cdot idf$.

When a feature subset, i.e., $F^S = \{f_1^S,..., f_r^S\}$ ($r < m$), is used to generate the feature vectors of the documents, the original *m*-dimensional feature space becomes an *r*-dimensional feature subspace. In this way, the modified training data $X^S = (X_1^S, X_2^S,..., X_n^S)$, denoted as subspace data, consists of *r*-dimensional samples $X^S = (x_{i1}^S, x_{i2}^S,..., x_{ir}^S)$ $(i = 1,...,n)$. A classifier trained with the subspace training data is called a subspace classifier.

4.2 Bootstrapping with Random Subspace

In bootstrapping, the classifier for choosing the samples with high confidences is usually trained over the whole feature space. This type of classifier tends to choose the samples much similar to the initial labeled data in terms of the whole feature space. As pointed in Section 3, this might cause the labeled data too concentrated to form a reasonable classification hyperplane. Instead, when a subspace classifier is applied, the added data is only similar to the existing labeled data in terms of the feature subspace and thus could be possibly more different in terms of the whole feature space. Generally, the extent of the differences between each two subspace classifiers largely depends on the differences of the features they used. One straight way to obtain different subspace classifiers is to randomly select *r* features from the whole feature set in each iteration in bootstrapping. Figure 3 illustrates the bootstrapping algorithm with random subspace classifiers.

Input:
Labeled data L
Unlabeled data U
Output:
New classifier C
Procedure:
For k=1 to N
(1). Randomly select a feature subset F_k^S of size r from F
(2). Generate a subspace data L_k^S with F_k^S and L
(3). Learn a subspace classifier C_k^S with L_k^S
(4). Use C_k^S to predict samples from U_{k-1}
(5). Choose n most confidently predicted samples A_k
(6). Add them into L_k, i.e., $L_k = L_k \bigcup A_k$
(7). Remove A_k from U_k, i.e., $U_k = U_{k-1} - A_k$
Use the updated data L_N to train a classifier C

Fig. 3. Bootstrapping algorithm with random subspace classifiers

The size of the feature subset r is an important parameter in this algorithm. The smaller r, the more different subspace classifiers are from each other. However, the value of r should not be too small because a classifier trained with too few features is not capable of correctly predicting samples.

4.3 Bootstrapping with Excluded Subspace

Although random feature selection is able to make the subspaces in different bootstrapping iterations differ from each other to some extent, the degree is still limited. To better satisfy the *diversity* preference, we improve the random subspace generation strategy with an constraint which restricts that every two adjacent subspace classifiers do not share any feature, i.e., $F_k^S \bigcap F_{k-1}^S = \varnothing$ where F_k^S represents the feature subset used in k-th iteration. This can be done by selecting a feature subset F_k^S from $F - F_{k-1}^S$ instead of F. We refer to this feature generation strategy as subspace excluding strategy. Figure 4 illustrates the bootstrapping algorithm with excluded subspace classifiers.

Input:
Labeled data L
Unlabeled data U
Output:
New classifier C
Procedure:
For k=1 to N
(1). Select a feature subset F_k^S of size r from $F - F_{k-1}^S$
(2). Generate a subspace data L_k^S with F_k^S and L
(3). Learn a subspace classifier C_k^S with L_k^S
(4). Use C_k^S to predict samples from U_{k-1}
(5). Choose n most confidently predicted samples A_k
(6). Add them into L_k, i.e., $L_k = L_k \bigcup A_k$
(7). Remove A_k from U_k, i.e., $U_k = U_{k-1} - A_k$
Use the updated data L_N to train a classifier C

Fig. 4. Bootstrapping algorithm with excluded subspace classifiers

4.4 Diversity Consideration among Different Types of Features

TC tasks, especially sentiment classification, often involve many types of features, such as word anagrams, word diagrams, or even syntactic features from dependency parsing (Xia et al., 2011). Although different types of the features may differ in morphology, some are sharing similar knowledge. Take *excellent* and *is_excellent* as examples of word unigram and bigram features respectively. Obviously, these two features share similar classification ability and are very likely to select similar samples. Therefore, it is necessary to consider the diversity among different types of features for real diversity between each two adjacent subspaces F_{k-1}^S and F_k^S.

Therefore, we introduce another constraint which restricts that every two adjacent subspace classifiers do not share any similar features. Here, two features are considered similar when they contain the same informative unigram. For example, *is_excellent* and *very_excellent* are considered similar because they both contain the informative unigram 'excellent'. In this study, we perform a standard feature selection method, mutual information (MI), on the labeled data to select top-*N* unigrams as the informative unigrams (Yang and Pedersen, 1997).

To satisfy this constraint, we first select a set of unigram features, denoted as F^{S-Uni}, from $F - F_{k-1}^S$; Then, we collect all the other-type features that contain any informative feature in F^{S-Uni} and put them into the feature subset. For example, assume that *excellent* is an informative feature. Once it is selected in F^{S-Uni}, we collect all bigrams like *is_excellent*, *very_excellent*, *not_excellent*, etc., and put them into the feature subset. It is important to note that the total number of the features for generating subspace is not guaranteed to a fixed value such as *r*. Instead, we make that size of

the unigram feature set fixed, which equals $\left|F^{Uni}\right| \cdot \theta$ where F^{Uni} is the feature set of word unigrams and θ ($\theta = r/m$) is the proportion of the selected features and all features.

5 Experimentation

5.1 Experimental Setting

In topic classification, we use two common benchmark corpora: 20News and WebKB, where the former consists of 20017 articles divided almost evenly into twenty different categories (Joachims, 1999) and the latter contains 4199 web pages from four popular categories (Craven et al., 1998). In sentiment classification, we use the product reviews from four domains: book, DVD, electronic, and kitchen appliances (Blitzer et al., 2007). Each of the four domains contains 1000 positive and 1000 negative reviews. In the experiments, 200 documents in each category are served as testing data and the remaining data are served as initial labeled data and unlabeled data.

Maximum Entropy (ME) is adopted as the classification algorithm with the help of Mallet[1] tool. All parameters are set to their default values. In particular, we employ both word unigrams and bigrams as the features. Our experimental results show that combining both word unigram and bigram features achieves similar results to only using unigrams in topic classification but apparently more preferable in sentiment classification. Nevertheless, our feature subspace-based bootstrapping approach is effective in both cases. To highlight the importance of diversity consideration of bigram features, we focus on the results of using both unigram and bigram features.

5.2 Experimental Results on Bootstrapping

In this section, we systematically evaluate the performance of our feature subspace-based bootstrapping and compare it with the supervised baseline:

1) Baseline: training a classifier with the initial labeled data (no unlabeled data is employed);
2) Bootstrapping-T: the traditional bootstrapping algorithm as shown in Figure 1;
3) Bootstrapping-RS: the bootstrapping algorithm with random subspace classifiers as shown in Figure 3;
4) Bootstrapping-ES: the bootstrapping algorithm with excluded subspace classifiers as shown in Figure 4;
5) Bootstrapping-ES+: the Bootstrapping-ES implementation with a feature excluding strategy as described in Section 4.4 to guarantee the difference between different types of features, i.e., word unigrams and bigrams in this study.

Performance of Different Bootstrapping Approaches

Figure 5 illustrates the results of the baseline and different bootstrapping approaches in topic classification and sentiment classification. For those approaches involving random selection of features, we run 5 times for them and report the average results.

[1] http://mallet.cs.umass.edu/

Figure 5 shows that:

- Semi-supervised learning in sentiment classification is much more difficult than that in topic classification. While the traditional bootstrapping, i.e., Bootstrapping-T could dramatically outperforms the baseline in both datasets of topic classification, it performs much worse than baseline in all four domains of sentiment classification.
- Bootstrapping-RS significantly outperforms Bootstrapping-T (p-value<0.001) except in the dataset of WebKB. This may be due to the fact that topic classification on WebKB has reached its performance ceiling via traditional bootstrapping and thus become difficult to make further improvement.
- Bootstrapping-ES is more effective than Bootstrapping-RS across four datasets but fails to improve Bootstrapping-RS in two datasets: Book and Electronic. This failure is due to the fact that using bigrams makes each two adjacent subspaces similar to each other to some extent. In fact, if only unigrams is used, Bootstrapping-ES always outperforms Bootstrapping-RS, increasing the accuracy from 0.62 to 0.67 in Book and from 0.71 to 0.73 in Electronic.

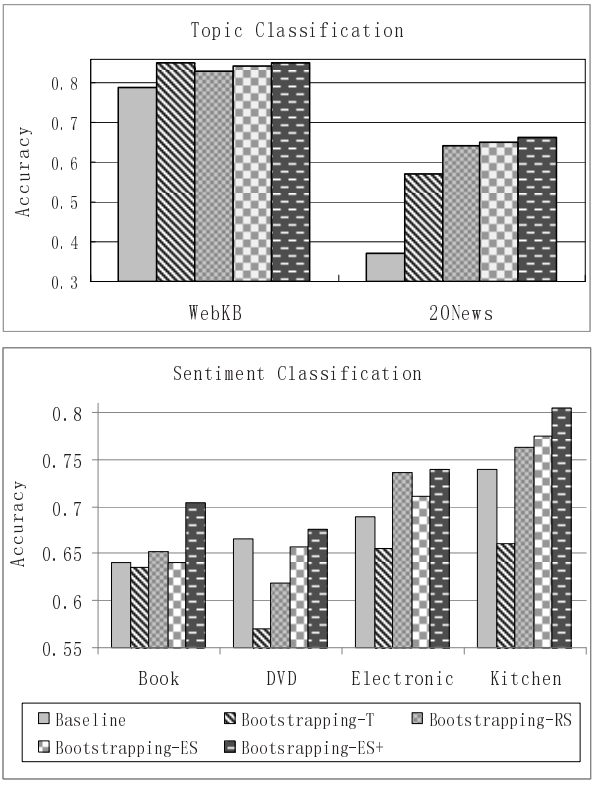

Fig. 5. Comparison of different bootstrapping approaches in topic classification (10 labeled samples per category) and sentiment classification (50 labeled samples per category)

- Bootstrapping-ES+ performs best among the four types of bootstrapping approaches and it almost outperforms both Bootstrapping-ES and Bootstrapping-RS in all datasets. Especially, it performs much better than Bootstrapping-ES in Book and Electronic, which verifies the importance of considering the diversity among different types of features.

Sensitiveness of the Parameter θ (r/m)

The size of the feature subspace is an important parameter in our approach. Figure 6 shows the performance of Bootstrapping-ES+ with varying sizes of the feature subspace. From Figure 6, we can see that a choice of the proportion between 1/3 and 1/6 is recommended. The size of the feature subspace should not be too small because a small amount of features would prevent a subspace well representing the samples and violate the *sufficiency* preference.

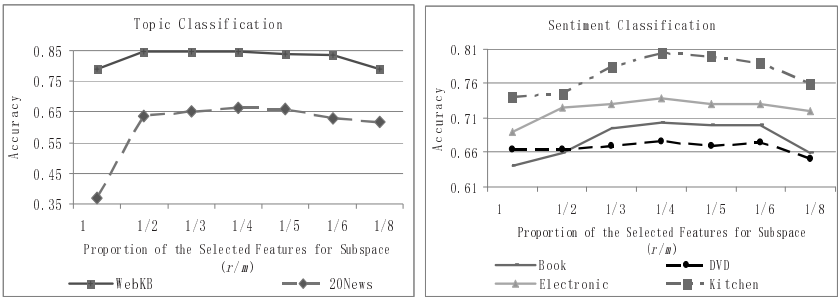

Fig. 6. Performances of Bootstrapping-ES+ over varying sizes of feature subspace

6 Conclusion and Future Work

In this paper, we first give two basic preferences for successful bootstrapping, namely *sufficiency* and *diversity*. To better satisfy the *diversity* preference, we present a novel bootstrapping approach by using feature subspace classifiers. Empirical studies show that our approach can effectively exploit unlabeled data in both topic and sentiment classification and significantly outperforms the traditional bootstrapping approach.

In our future work, we will try to develop a sound theoretical understanding to the effectiveness of our approach and propose other diversity strategies to further improve the performance on text categorization. Moreover, we will apply our feature subspace-based bootstrapping to other tasks in NLP.

Acknowledgments. This research work has been partially supported by two NSFC grants, No.61003155, and No.61273320, one National High-tech Research and Development Program of China No.2012AA011102, one General Research Fund (GRF) sponsored by the Research Grants Council of Hong Kong No.543810.

References

1. Abney, S.: Bootstrapping. In: Proceedings of ACL 2002, pp. 360–367 (2002)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In: Proceedings of ACL 2007, pp. 440–447 (2007)
3. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: Proceedings of COLT 1998, pp. 92–100 (1998)
4. Braga, I., Monard, M., Matsubara, E.: Combining Unigrams and Bigrams in Semi-supervised Text Classification. In: Proceedings of EPIA 2009: The 14th Portuguese Conference on Artificial Intelligence, pp. 489–500 (2009)
5. Cong, G., Lee, W.S., Wu, H., Liu, B.: Semi-supervised text classification using partitioned EM. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 482–493. Springer, Heidelberg (2004)
6. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to Extract Symbolic Knowledge from the World Wide Web. In: Proceedings of AAAI 1998, pp. 509–516 (1998)
7. Dasgupta, S., Ng, V.: Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In: Proceedings of ACL-IJCNLP 2009, pp. 701–709 (2009)
8. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society: Series B 39(1), 1–38 (1977)
9. Joachims, T.: Transductive Inference for Text Classification Using Support Vector Machines. In: Proceedings of ICML 1999, pp. 200–209 (1999)
10. Kullback, S., Leibler, R.: On Information and Sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
11. Li, S., Huang, C., Zhou, G., Lee, S.: Employing Personal/Impersonal Views in Supervised and Semi-supervised Sentiment Classification. In: Proceedings of ACL 2010, pp. 414–423 (2010)
12. Mallapragada, P., Jin, R., Jain, A., Liu, Y.: SemiBoost: Boosting for Semi-Supervised Learning. IEEE Transaction on Pattern Analysis and Machine Intelligence 31(11), 2000–2014 (2009)
13. McCallum, A., Nigam, K.: Employing EM and Pool-Based Active Learning for Text Classification. In: Proceedings of ICML 1998, pp. 350–358 (1998)
14. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning 39(2/3), 103–134 (2000)
15. Nigam, K., Ghani, R.: Analyzing the Effectiveness and Applicability of Co-training. In: Proceedings of CIKM 2000, pp. 86–93 (2000)
16. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of EMNLP 2002, pp. 79–86 (2002)
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
18. Xia, R., Zong, C., Li, S.: Ensemble of Feature Sets and Classification Algorithms for Sentiment Classification. Information Sciences 181, 1138–1152 (2011)
19. Yang, Y., Pedersen, J.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the 14th International Conference on Machine Learning, ICML 1997, pp. 412–420 (1997)
20. Yarowsky, D.: Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: Proceedings of ACL 2005, pp. 189–196 (1995)