A Meta-Learning Approach for Text Categorization

Wai Lam*
Department of Systems Engineering and Engineering Management
Ho Sin Hang Engineering Building
The Chinese University of Hong Kong
Shatin, Hong Kong
wlam@se.cuhk.edu.hk

Kwok-Yin Lai
Department of Systems Engineering and Engineering Management
Ho Sin Hang Engineering Building
The Chinese University of Hong Kong
Shatin, Hong Kong
kylai@se.cuhk.edu.hk

ABSTRACT

We investigate a meta-model approach, called Meta-learning Using Document Feature characteristics (MUDOF), for the task of automatic textual document categorization. It employs a meta-learning phase using document feature characteristics. Document feature characteristics, derived from the training document set, capture some inherent category-specific properties of a particular category. Different from existing categorization methods, MUDOF can automatically recommend a suitable algorithm for each category based on the category-specific statistical characteristics. Hence, different algorithms may be employed for different categories. Experiments have been conducted on a real-world document collection demonstrating the effectiveness of our approach. The results confirm that our meta-model approach can exploit the advantage of its component algorithms, and demonstrate a better performance than existing algorithms.

Keywords

Text Categorization, Text Mining, Meta-Learning

1. INTRODUCTION

Textual document categorization aims to assign a number of appropriate categories to a document. The goal of automatic text categorization is to learn a classification scheme from training examples. Once a classification scheme is learned, it can be used to categorize documents automatically. Automatic categorization has many applications such as document routing, document management, or document dissemination.

There has been some research conducted for automatic text categorization. Apte et al. [1] adopted a decision tree learning technique to learn a classifier. Yang and Chute [20] proposed a statistical approach known as Linear Least Squares Fit (LLSF) which estimates the likelihood of the associations between document terms and categories via a linear parametric model. Lewis et al. [12] explored linear classifiers for the text categorization problem. Cohen and Singer [3] developed the sleeping experts algorithm which is based on a multiplicative weight update technique. Yang [17] developed an algorithm known as ExpNet which derives from the k-nearest neighbor technique. ExpNet achieves good categorization performance on large document corpora such as the Reuters collection and the OHSUMED collection [22]. Recently, Yang et al. [19] also employed a modified KNN text categorization method for event tracking. Lam and Ho [10] proposed the generalized instance set approach for text categorization. Joachims [8], as well as, Yang and Liu [21] recently compared support vector machines with KNN. Dumais et al. [4] compared support vector machines, decision trees and Bayesian approaches on the Reuters collection. All the above approaches developed a single paradigm to solve the categorization problem.

In machine learning community, several methods on multi-strategy learning or combination of classifiers have been proposed. Chan and Stolfo [2] presented their evaluation of simple voting and meta-learning on partitioned data, through inductive learning. Ting and Witten [15] demonstrated the effectiveness of stacked generalization for combining different types of learning algorithms. By combining a high-level model with low-level models, a better predictive accuracy was found. Kumar et al. [9] proposed a hierarchical multi-classifier system to perform hyperspectral data analysis. Ho [5] analyzed the complexity of classification problems using decision trees and nearest neighbour, and showed that dependences of classifiers' behaviour on data characteristics exist. All these multi-strategy techniques have not been tested on text categorization. Recently, several meta-model methods have been proposed for text domains. Yang et al. [18] proposed the Best Overall Results Generator (BORG) system which combined classification methods linearly, using simple equal weight for each classifier in the Topic Detection and Tracking (TDT) domain. Classification methods employed in BORG are Rocchio, kNN and Language Modeling. Larkey et al. [11] reported improved performance, by using new query formulation and weighting methods, in the context of text categorization by combining three classifiers, namely KNN, relevance feedback and Bayesian inde-

*This work was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK 4385/99E).

SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA..

pendence classifiers. Instead of applying method combination on text categorization, Hull at al. [6] examined various combination strategies in the context of document filtering. Learning algorithms included Rocchio, nearest neighbor, linear discriminant analysis and neural net.

The approaches mentioned above combine several classifiers, which are learned independently from different classification algorithms. Boosting method, one of the meta-learning strategies proposed recently, however, combines classifiers, called the weak hypotheses, which are sequentially learned by the same learning method, called the weak learner. At each iteration of a boosting method, a weak hypotheses is learned by taking into account how the weak hypotheses, that are learned in the previous iterations, perform on the training documents. After a specific number of iterations, a final hypothesis is obtained by a linear combination of all the weak hypotheses. The final hypothesis is then used to classify the unseen documents. Based on the boosting method, Schapire and Singer [13] proposed a new family of boosting algorithms for text and speech categorization. Sebastiani et al. [14] recently proposed an improved boosting algorithm based on AdaBoost for text categorization by generating a set of, rather than only one single, weak hypotheses at each iteration of the boosting process. Iyer et al. [7] investigated the behavior of RankBoost on different ranking functions for the weak hypotheses in the context of document routing.

Most existing meta-model approaches for text categorization are based on linear combination of several basic algorithms. The linear combination approach makes use of limited knowledge in the training document set. To address this limitation, we propose a meta-model approach, called Meta-learning Using Document Feature characteristics (MUDOF), which employs a meta-learning phase using document feature characteristics. Document feature characteristics, derived from the training document set, capture some inherent category-specific properties of a particular category. This approach aims at recommending a suitable algorithm automatically for each category. Hence, different algorithms may be employed for different categories. Specifically, the relationship between the document feature characteristics and the predicted classification error is learned by using the technique of multivariate regression analysis. Based on the relationship, it can make automatic recommendation of algorithms for different categories.

We have conducted extensive experiments on a real-world document collection known as Reuters-21578. This collection contains news articles from Reuters in 1987. Each article is assigned to none or several pre-defined categories. There are 90 categories used in our experiments. The results demonstrate that our new approach of meta-learning model for text categorization shows better performance than existing component algorithms.

2. AUTOMATIC TEXT CATEGORIZATION BACKGROUND

Textual document categorization aims to assign none or any number of appropriate categories to a document. The goal of automatic text categorization is to construct a classification scheme, or called the classifier, from a training set containing sample documents and their corresponding categories. Since there are quite a number of categories and a document can be assigned to more than one category, we tackle the problem by decomposing it into individual category level. Specifically, there is a classification scheme for each category. During the training phase, documents in training set are used to learn a classification scheme for each category by using a learning algorithm. This learning process is repeated, using training examples derived from the training set for each category. Eventually, after completing the whole training phase, each category will have a different learned classification scheme.

After the training phase, the learned classification scheme for each category will be used to categorize unseen documents. Given a document to be categorized, a score can be computed by the classification scheme of a category, indicating the degree of confidence assigning that category to the document. In particular, the document is assigned to that category if the calculated score is greater than a certain threshold value, while a score smaller than the threshold declines the assignment.

Similar calculation is performed on the classification schemes of other categories. By collecting the decisions made of all the classification schemes, a certain number of categories are assigned to the document.

Instead of using only one algorithm, meta-model learning involves more than one categorization algorithm. During the training phase, each learning algorithm constructs its classification scheme for a given category as described above. Under the meta-model approach, classification schemes that have been separately learned by different algorithms for a category, are combined together in a certain way, to yield one single meta-model classification scheme. Given a document to be categorized, the meta-model classification scheme can be used for deciding the document membership for the category. As a result, each meta-model classifier for a category is the combined contributions of all the involved algorithms.

Our MUDOF approach combines the evidence of predicted classification errors of different algorithms by regression analysis on document feature characteristics.

3. META-LEARNING USING DOCUMENT FEATURE CHARACTERISTICS (MUDOF)

3.1 Overview

While the improvements reported by most of the previous studies of different approaches on text categorization were based on several single overall performance scores calculated by different utility measures, however, performance comparisons, on category-by-category basis, between different algorithms are seldom investigated. Our preliminary results show that, though a particular algorithm may obtain a better overall performance in different single performance scores, it is not guaranteed that its performance is the best for particular categories, when compared with other algorithms. This can be attributed to the fact that algorithms perform differently for a certain category, which exhibits specific nature or different characteristics from other remaining categories. Regarding this, we observe that if an algorithm, of less capable in performance, for a particular category, can be replaced by another better algorithm, a better classification performance can be further increased.

Motivated by such observation, we propose MUDOF, a novel approach of the meta-learning framework for text categorization, based on multivariate regression analysis, by capturing category specific feature characteristics. In MUDOF, there is a separate meta-learning phase using document feature characteristics. Document feature characteristics, derived from the training set of a particular category, can capture some inherent properties of that category. Different from existing categorization methods, instead of applying a single method for all categories during classification, this new meta-learning approach can automatically recommend a suitable algorithm during training, from an algorithm pool, for each category based on the category specific statistical characteristics and multivariate regression analysis. To achieve this task, we employ a learning approach by learning the relationship between the feature characteristics and the classification errors by conducting multivariate regression analysis for each algorithm on each category. The learned relationship is expressed by sets of parameter estimates, based on which, suitable classification algorithms are recommended for that category. Document feature characteristics, on category basis, are statistics which can be regarded as the descriptive summary for each category. Normalized document feature statistics are fitted into our meta-model as independent variables. The problem of predicting the expected classification error of an algorithm for a category, therefore, can be interpreted as a function of these feature characteristics. ust improve

3.2 The MUDOF Algorithm

In MUDOF, we make use of categorical feature characteristics and classification errors. In particular, we wish to predict the classification error for a category based on the feature characteristics. This is achieved by a learning approach based on regression model, in which, the document feature characteristics are the independent variables, while the classification error of an algorithm is the dependent variable. Feature characteristics are derived from the categories. We further divide the training collection into two sets, namely the training set and the tuning set. Two sets of feature characteristics are collected separately from these two data sets. Statistics from the training set are for parameter estimations. Together with the estimated parameters, the statistics from the tuning set are used for predicting the classification error of an algorithm for a category. The algorithm with the minimum estimated classification error for a category will be recommended for that category during the on-line classification, or validation, phase.

Consider the ith category. Suppose we have several component classification algorithms. Let e_{ij} be the classification error of the training set on the jth algorithm. Classification errors will first undergo a logistic transformation to yield the response variable, or the dependent variable, for the meta-model. Precisely, the transformation is given in Equation 1.

$$y_{ij} = \ln \frac{e_{ij}}{1 - e_{ij}} \tag{1}$$

where y_{ij} is the response variable. This transformation ensures that the response variable is in the range of 0 and 1. The response variable, y_{ij} is related to the feature characteristics by the regression model, as shown in Equation 2.

$$y_{ij} = \beta_j^0 + \sum_{k=1}^{p} \beta_j^k * F_i^k + \epsilon_{ij}, \tag{2}$$

where F_i^k is the kth feature characteristic in the ith category. The number of document feature characteristics used in the meta-model is p. β_j^k is the parameter estimate for the kth feature, by using the algorithm j. ϵ_{ij} is assumed to follow a Gaussian distribution $N(0, var(\epsilon_{ij}))$. Based on the regression model above, the outline of meta-model for text categorization is given in Figure 1.

Step 1 to 9, in Figure 1, aims to estimate a set of betas ($\widehat{\beta}_j^k$), the parameter est imates of the feature characteristics in the regression model, for each individual algorithm. In Step 2, an algorithm, with optimized parameter settings, is picked from the algorithm pool. By repeating Step 3 to 7, the algorithm is applied on training and tuning examples to yield classification errors of the classifier for all categories. Documents in tuning set, as shown in Step 5, are used for obtaining the classification performance, and so the classification error, of a trained classifier for each category. A set of betas, belonging to the algorithm being considered, can be obtained by fitting all classification errors of the categories, and their corresponding feature characteristics in the training set, into the regression model. After Step 9, there will be j sets of estimated parameters, the betas, which are then used for the subsequent steps.

The predictions on the classification errors of the involved algorithms are made from Step 10 to Step 16. In Step 12, one algorithm with the same optimized parameter settings as in Step 2, is picked from the algorithm pool. The corresponding set of betas of the selected algorithm, together with the feature characteristics of a category in the tuning set, will be fitted into the regression model, in Step 13, to give the estimated classification errors of the algorithm on the category. Decisions, about which algorithm will be applied on the category, are based on the predicted minimum classification errors in Step 14. After Step 16, classification algorithms are recommended for categories, and the recommended algorithm will be applied to each category during the on-line classification, or validation, step.

The robustness of the meta-model approach rests on its fully automatic estimations. The whole process, from parameter estimation to recommending algorithms for categories, is fully automatic. The operation of our meta-model approach is carried out as usual, except that different algorithms will be applied to the categories, instead of applying a single algorithm on all categories as what is done in other approaches.

3.3 Document Feature Characteristics

In MUDOF, eight document feature characteristics are used in our regression model as independent variables:

1. ***PosTr***: The number of positive training examples of a category.
2. ***PosTu***: The number of positive tuning examples of a category.
3. ***AvgDocLen***: The average document length of a category. Document length refers to the number of indexed terms within a document. The average is taken across all the positive examples of a category.
4. ***AvgTermVal***: The average term weight of documents across a category. Average term weight is taken for individual documents first. Then, the average is taken across all the positive examples of a category.
5. ***AvgMaxTermVal***: The average maximum term weight of documents across a category. Maximum term weight of individual documents are summed, and the average is taken across all the positive examples of a category.

Input: The training set TR and tuning set TU
An algorithm pool A and categories set C

1) Repeat
2) Pick one algorithm ALG_j from A.
3) For each category C_i in C
4) Apply ALG_j on TR for C_i to yield a classifier CF_{ij}.
5) Apply CF_{ij} on TU for C_i to yield classification error e_{ij}.
6) Take logistic transformation on e_{ij} to yield y_{ij} for later parameter estimation.
7) End For
8) Estimate $\widehat{\beta}_j^k$ (k=0,1,2,...,p) for ALG_j by fitting y_{ij} and F_i^k (in TR) into the regression model.
9) Until no more algorithms in A.
10) For each category C_i in C
11) Repeat
12) Pick one algorithm ALG_j from A.
13) Estimate the classification error $\widehat{e}_{ij}$ by fitting $\widehat{\beta}_j^k$ and corresponding F_i^k (in TU) into the regression model.
14) If $\widehat{e}_{ij}$ is minimum, recommend ALG_j for C_i as the output.
15) Until no more algorithms in A.
16) End For

Figure 1: The Meta-Model algorithm

6. ***AvgMinTermVal***: The average minimum term weight of documents across a category. Minimum term weight of individual documents are summed, and the average is taken across all the positive examples of a category.
7. ***AvgTermThre***: The average number of terms above a term weight threshold. The term weight threshold is optimized and set globally. Based on the preset threshold, the number of terms with term weight above the threshold within a category are summed. The average is then taken across all the positive examples of the category.
8. ***AvgTopInfoGain***: The average information gain of the top m terms of a category. The information gain of each individual term is calculated for each category and ranked. The average is then taken across the top m terms with highest information gain.

$$\begin{aligned} G(t) = & -\sum_{i=1}^{m} P_r(c_i)\log P_r(c_i) + \\ & P_r(t)\sum_{i=1}^{m} P_r(c_i|t)\log P_r(c_i|t) + \\ & P_r(\bar{t})\sum_{i=1}^{m} P_r(c_i|\bar{t})\log P_r(c_i|\bar{t}) \end{aligned} \quad (3)$$

9. ***NumInfoGainThres***: The number of terms above an information gain threshold. The threshold is set globally. Based on the preset threshold, the number of terms with information gain above the threshold within a category are counted.

Two sets of normalized feature characteristics are collected separately from the training set and the tuning set. As illustrated in Step 8 and Step 13 in Figure 1, the feature characteristics from these two data sets serve different purposes in the meta-model: feature characteristics from training set are combined for parameter estimation, while feature characteristics from tuning set are used for predicting classification errors, base on which algorithms are recommended.

4. EXPERIMENTS AND EMPIRICAL RESULTS

4.1 Experimental Setup

Extensive experiments have been conducted on the Reuters-21578 corpus. 90 categories are used in our experiments. We divided the 21,578 documents in the Reuters-21578 document collection according to the "ModApte" split into one training collection of 9603 documents, and one testing collection of 3299 documents [1]. The remaining 8,676 documents are not used in the experiments as the documents are not classified by human indexer. For those meta-models requiring a tuning set, we further divided the training collection into training set of 6000 documents and 3603 tuning documents. For each category, we used the training document collection to learn a classification scheme. The testing collection is used for evaluating the classification performance.

All documents are pre-processed and converted into internal representation before conducting the experiments. The major steps involve stop-word removal, stemming and calculating term weights of all the stemmed words by $tf \cdot idf$ for each document vector.

Six component classification algorithms have been used in our meta-model approaches. They are Rocchio, WH, KNN, SVM, GISR and GISW, with optimized parameter settings. These are six recent algorithms, each of which exhibits certain distinctive nature: Rocchio and WH are linear classi-

[1] Other studies may refine the Reuters-21578 corpus by further eliminating those documents that do not belong to those 90 categories, resulting in 7,769 training documents and 3,019 testing documents.

Algorithms	ABE	MUDOF+(%)	MBE	MUDOF+(%)
MUDOF	0.656	-	0.857	-
RO	0.578	13.495	0.776	10.438
WH	0.649	1.079	0.820	4.512
KNN	0.607	8.072	0.802	6.858
SVM	0.640	2.500	0.841	1.902
GISR	0.625	4.960	0.830	3.253
GISW	0.655	0.153	0.845	1.420

Table 1: Classification improvement by meta-model approach over component algorithms based on macro-recall and precision break-even point as well as micro-recall and precision break-even point measures over 90 categories.

fiers, KNN is an instance-based learning algorithm, SVM is based on Structural Risk Minimization Principle [16] and both GISR and GISW [10] are based on generalized instance approach. All eight document feature characteristics mentioned in Section 3.3 were adopted in MUDOF. m is chosen as 15 for the $AvgTopInfoGain$ feature characteristic.

To measure the performance, we use both micro-averaged recall and precision break-even point measure (MBE) [12], as well as the macro-averaged recall and precision break-even point measure (ABE). In micro-averaged recall and precision break-even point measure, the total number of false positive, false negative, true positive, and true negative are computed across all categories. These totals are used to compute the micro-recall and micro-precision. Then we use the interpolation to find the break-even point. In macro-averaged recall and precision break-even point measure, break-even point for individual category is calculated first, and the simple average of all those break-even points is taken across all the categories to obtain the final score.

4.2 Empirical Results

We first conducted extensive experiments for each component algorithm in order to search for the best parameters setting. Then we conducted experiments for MUDOF. Table 1 shows the classification performance improvement, based on the macro-recall and precision break-even point measure as well as the micro-recall and precision break-even point measure, obtained by our MUDOF approach. The first row depicts the performance of MUDOF. It demonstrates some improvement over component algorithms.

Table 2 shows the parameter estimates for the document feature characteristics of different component algorithms in MUDOF approach. Based on these parameter estimates and the corresponding feature characteristics, on category basis, the estimated classification errors of different algorithms on the categories can be obtained. It should be noted that, a negative parameter estimate will contribute to a smaller estimated classification error for an algorithm on a category. As a result, a feature characteristic with a large negative parameter estimate, will make itself a more distinctive feature in voting for the algorithm than others.

Besides comparing the performance of the MUDOF approach over individual component algorithms, we set up the ideal combination of algorithms as another benchmark for our MUDOF approach. The ideal combination of algorithms is set up manually and is composed of the best algorithms, which are the most appropriate algorithms that MUDOF should recommend for each category accordingly. Our results, show that the meta-model can identify the ideal algorithms for 59 categories out of the total 90 categories.

Since the ideal combination consists of the most appropriate algorithm for each category, it sets an upper bound for the amount of improvement that can be made under our meta-model. Table 3 shows the comparison of performances, under different aspects of measures, between MUDOF and the ideal combination (IDEAL). Based on the utility measures as shown in the table, we will look into how much improvement the meta-model MUDOF (M+(%)) has achieved within the improvement bound (I+(%)) set by the ideal combination in Table 4.

In Table 4, among all other algorithms, the classification improvement made by either MUDOF (M+(%)) or the ideal combination (I+(%)) over Rocchio is the largest, more than 10% on average, in all aspects. Improvement made by the meta-model over KNN is also significant, it is more than 5% in all aspects. Our meta-model can even make improvement for less frequent categories over the robust SVM.

Table 4 also reveals that the improvement made by meta-model over individual component algorithms is quite impressive, when considering the improvement bound set by the ideal combination (I+(%)). Improvement achieved by MUDOF within the improvement bound of the ideal combination $(M+/I+(\%))$ is also presented in the table. When compared with Rocchio and KNN, the meta-model has attained more than 50% of the improvement bound for all aspects. As for All 90 MBE, the meta-model can also achieve more than 50% of the improvement bound for all component algorithms except SVM and GISW, and it is more than 80% when compared with Rocchio.

5. CONCLUSIONS

We have investigated a meta-model approach for the task of automatic textual document categorization. Our meta-model approach, called Meta-learning Using Document Feature characteristics (MUDOF), employs a meta-learning phase using document feature characteristics. Different from existing categorization methods, MUDOF can automatically recommend a suitable algorithm for each category based on the category-specific statistical characteristics. Moreover, MUDOF allows flexible additions or replacement of different classification algorithms, resulting in the improved overall classification performance. Extensive experiments have been conducted on the Reuters-21578 corpus. The results confirm that our meta-model approach can exploit the advantage of its component algorithms. Besides, our meta-model approach exhibits advantages over other component algorithms by demonstrating a better classification performance.

6. REFERENCES

[1] C. Apte, F. Damerau, and S.Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1993.

[2] P. K. Chan and S. J. Stolfo. Comparative evaluation of voting and meta-learning on partitioned data. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, pages 90–98, 1995.

Features	RO	WH	KNN	SVM	GISR	GISW
PosTr	7.580	24.087	15.433	6.280	21.466	22.436
PosTu	-5.041	-23.746	-12.237	-9.135	-20.848	-26.113
AvgDocLen	1813.020	2920.149	1214.978	2177.761	2313.910	2443.626
AvgTermVal	100.618	149.950	77.534	116.820	148.570	152.199
AvgMaxTermVal	10.750	20.807	8.549	12.303	11.539	19.061
AvgMinTermVal	-73.574	-107.843	-73.481	-95.221	-130.532	-141.879
AvgTermThre	-1791.173	-2885.762	-1196.830	-2152.466	-2280.584	-2407.613
AvgTopInfoGain	-19.577	-22.151	-25.851	-12.083	-25.755	-19.056
NumInfoGainThres	0.304	0.807	0.583	-0.255	0.019	0.601

Table 2: Parameter estimates for document feature characteristics of different algorithms.

Utility Measure	MUDOF	IDEAL
All 90 ABE	0.656	0.692
All 90 MBE	0.857	0.868

Table 3: Classification performances of meta-model (MUDOF) and the ideal combination (IDEAL).

	RO	M+(%)	I+(%)	M+/I+(%)	WH	M+(%)	I+(%)	M+/I+(%)
All 90 ABE	0.578	13.495	19.723	68.421	0.649	1.079	6.626	16.279
All 90 MBE	0.776	10.438	11.856	88.043	0.820	4.512	5.854	77.083
	KNN	M+(%)	I+(%)	M+/I+(%)	SVM	M+(%)	I+(%)	M+/I+(%)
All 90 ABE	0.607	8.072	14.003	57.647	0.640	2.500	8.125	30.769
All 90 MBE	0.802	6.858	8.229	83.333	0.841	1.902	3.210	59.259
	GISR	M+(%)	I+(%)	M+/I+(%)	GISW	M+(%)	I+(%)	M+/I+(%)
All 90 ABE	0.625	4.960	10.720	46.269	0.655	0.153	5.649	2.703
All 90 MBE	0.830	3.253	4.578	71.053	0.845	1.420	2.722	52.174

Table 4: Improvement of classification performances of meta-model (M+(%)) and the ideal combination (I+(%)) over individual algorithms, and improvement achieved by meta-model within the improvement bound set by the ideal combination (M+/I+(%)).

[3] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–315, 1996.

[4] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, pages 148–155, 1998.

[5] T. K. Ho. Complexity of classification problems and comparative advantages of combined classifiers. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS 2000)*, pages 97–106, 2000.

[6] D. A. Hull, J. O. Pedersen, and H. Schutze. Method combination for document filtering. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–287, 1996.

[7] R. D. Iyer, D. D. Lewis, R. E. Schapire, Y. Singer, and A. Singhal. Boosting for document routing. In *Proceedings of the Ninth International Conference On Information Knowledge Management (CIKM 2000)*, pages 70–77, 2000.

[8] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML'98)*, pages 137–142, 1998.

[9] S. Kumar, J. Ghosh, and M. Crawford. A hierarchical multiclassifier system for hyperspectral data analysis. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS 2000)*, pages 270–279, 2000.

[10] W. Lam and C. Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of the Twenty-First International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 81–89, 1998.

[11] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 289–297, 1996.

[12] D. D. Lewis, R. E. Schapore, J. P. Call, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

[13] R. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(213):135–168, 2000.

[14] F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to text categorization. In *Proceedings of the Ninth International Conference On Information Knowledge Management (CIKM 2000)*, pages 78–85, 2000.

[15] K. M. Ting and I. H. Witten. Stacked generalization: when does it work? In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 866–871, 1997.

[16] V. Vapnic. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[17] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, 1994.

[18] Y. Yang, T. Ault, and T. Pierce. Combining multiple learning strategies for effective cross validation. In *Proceedings of the International Conference on Machine Learning (ICML 2000)*, pages 1167–1174, 2000.

[19] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving text categorization methods for event tracking. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–72, 2000.

[20] Y. Yang and C. D. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, 1994.

[21] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the Twenty-First International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

[22] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the International Conference on Machine Learning (ICML'97)*, pages 412–420, 1997.