Comparison of Extreme Learning Machine with Support Vector Machine for Text Classification

Ying Liu[1], Han Tong Loh[1], and Shu Beng Tor[2]

[1] Singapore-MIT Alliance, National University of Singapore, Singapore 117576
{G0200921, mpelht}@nus.edu.sg
[2] Singapore-MIT Alliance, Nanyang Technological University, Singapore 639798
{msbtor}@ntu.edu.sg

Abstract. Extreme Learning Machine, ELM, is a recently available learning algorithm for single layer feedforward neural network. Compared with classical learning algorithms in neural network, e.g. Back Propagation, ELM can achieve better performance with much shorter learning time. In the existing literature, its better performance and comparison with Support Vector Machine, SVM, over regression and general classification problems catch the attention of many researchers. In this paper, the comparison between ELM and SVM over a particular area of classification, i.e. text classification, is conducted. The results of benchmarking experiments with SVM show that for many categories SVM still outperforms ELM. It also suggests that other than accuracy, the indicator combining precision and recall, i.e. F_1 value, is a better performance indicator.

1 Introduction

Automated text classification aims to classify text documents into a set of predefined categories without human intervention. It has generated interests among researchers in the last decade partly due to the dramatically increased availability of digital documents on the World Wide Web, digital libraries and documents warehouses [20].

Text classification (TC) is an area with roots in the disciplines of machine learning (ML) and information retrieval (IR) [1], [15]. Text mining has become a terminology very frequently used to describe tasks whose major concerns are to analyze high volumes of texts, detect interesting patterns and reveal useful information. TC has become one of the most important pillars of text mining.

In order to accomplish the TC tasks, one or more classifiers are needed. Most of current popular classifiers, i.e. support vector machine (SVM), neural network (NN), kNN, decision tree and decision rule, Naïve Bayes and so on, are built in an inductive learning way. Among them, SVM is acclaimed by many researchers for its leading performance [20]. Therefore, it has been widely used for TC purpose.

Most recently, a new learning algorithm, extreme learning machine (ELM), is available for the training of single layer feedforward neural network. The inventors of ELM have done a set of comprehensive experiments in regression and general classification to compare its performance with SVM [7]. The experimental results show that compared with classical learning algorithms in neural network, e.g. Back Propagation, ELM can achieve better performance with much shorter learning time [7].

© Springer-Verlag Berlin Heidelberg 2005

Compared with SVM, ELM is sometimes better than SVM in terms of accuracy, though not always. But as the number of neurons available for each ELM machine is the only parameter to be determined, ELM is much simpler for parameter tuning compared with SVMs whose kernel functions are nonlinear, e.g. RBF functions, thus saving tremendous time in searching for optimal parameters. Currently, SVMs, even for those with linear kernel function only, have gained wide acceptance by researchers as the leading performer for TC tasks. Our interest in this research is to benchmark ELM and SVM with linear kernel function for TC tasks and see whether ELM can serve as an alternative to SVM in TC tasks.

Having described the motivation of comparison between ELM and SVM, the rest of this paper is organized as follows. Some previous work in TC field by using neural network and SVM is reviewed in section 2. A brief introduction to ELM is given in section 3. We explain the experiment details and discuss the results in section 4. Finally, conclusions are drawn in section 5.

2 Related Work

Since several years ago, Neural network (NN) has been applied to TC tasks as a classifier. A NN is composed of many computing units (neurons) interconnected with each other with different weights in a network. In TC domain, the inputs to NN are the weights of features, i.e. terms, in a text document. And the output is the desired category or categories of the text document [2], [20], [23], [24].

Perceptron, the simplest type of NN classifier, is a linear classifier and has been extensively researched. Combined with effective means of feature selection, perceptron has achieved a very good performance and remains as the most popular choice of NN [16]. A non-linear NN, on the other hand, is a network with one or more additional "layers" of neurons, which in TC usually represent higher-order interactions between terms that the network is able to learn [17], [18], [23], [24], [26]. The literature on comparative experiments relating non-linear NNs to their linear counterparts show that the former has yielded either no improvement or very small improvements [23]. With their flexible architectures, NNs are well suited for applications of hierarchy text classification also [24].

Compared with NN, support vector machine (SVM) is relatively new to researchers in the fields of machine learning and information retrieval. However, it has quickly become the most popular algorithm mainly due to its leading performance. It is invented by Vapnik [22] and first introduced into the TC area by Joachims [8], [9]. His SVM implementation, i.e. SVM Light, has become one of the most popular packages of SVM application and has been widely used for TC [5], [11], [20], [26]. According to Joachims [8], SVM is very suitable for TC purpose, because SVM is not very sensitive to the high dimensionality of the feature space and most of TC jobs can be linearly separated. Yang and Liu's experiments [26] over a benchmarking TC corpus show that compared with the assumption of non-linear separation, the linear separation case can lead to a slightly better performance and save much effort on parameter tuning.

Invented by Huang Guangbin, extreme learning machine (ELM) is a newly available learning algorithm for a single layer feedforward neural network [7]. ELM randomly chooses the input weights and analytically determines the output weights of the network. In theory, this algorithm tends to provide the good generalization performance at extremely fast learning speed. The regression and classification experiments conducted by the inventors have shown that compared with BP and SVM, ELM is easier to use, faster to learn and has the higher generalization performance [7].

3 Extreme Learning Machine

A standard single layer feedforward neural network with n hidden neurons and activation function $g(x)$ can be mathematically modeled as:

$$\sum_{i=1}^{n} \boldsymbol{\beta}_i g(\boldsymbol{w}_i \boldsymbol{x}_j + b_i) = d_j, j = 1,\ldots,N \tag{1}$$

where $\boldsymbol{w}_i$ is the weight vector connecting inputs and the ith hidden neurons, $\boldsymbol{\beta}_i$ is the weight vector connecting the ith hidden neurons and output neurons, d_j is the output from ELM for data point j.

With N data points in a pair as $(\boldsymbol{x}_j, \boldsymbol{t}_j)$, $\boldsymbol{x}_i \in R^n$ and $\boldsymbol{t}_i \in R^m$ where $\boldsymbol{t}_j$ is the corresponding output for data point $\boldsymbol{x}_j$, the ideal case is training with zero errors, which can be represented as:

$$\sum_{i=1}^{n} \boldsymbol{\beta}_i g(\boldsymbol{w}_i \boldsymbol{x}_j + b_i) = t_j, j = 1,\ldots,N \tag{2}$$

The above equations can be written compactly as:

$$\mathbf{H}\beta = \mathbf{T} \tag{3}$$

where

$$\mathbf{H} = \begin{bmatrix} g(\boldsymbol{w}_1\boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{w}_n\boldsymbol{x}_1 + b_n) \\ \vdots & \cdots & \vdots \\ g(\boldsymbol{w}_1\boldsymbol{x}_N + b_1) & \cdots & g(\boldsymbol{w}_n\boldsymbol{x}_N + b_n) \end{bmatrix}_{N\times n} \tag{4}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_n^T \end{bmatrix}_{n\times m} \text{ and } \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m} \tag{5}$$

So the solution is:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$ is called Moore-Penrose generalized inverse [7].

The most important properties of this solution as claimed by the authors [7] are:

1. Minimum training error
2. Smallest norm of weights and best generalization performance
3. The minimum norm least-square solution of $\mathbf{H}\beta = \mathbf{T}$ is unique, which is $\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

So finally, the ELM algorithm is [7]:

Given a training set $\left\{(\boldsymbol{x}_i, \boldsymbol{t}_i) \middle| \boldsymbol{x}_i \in R^n, \boldsymbol{t}_i \in R^m, i = 1,\ldots,N\right\}$, activation function $g(x)$, and N hidden neurons, Step 1: Assign arbitrary input weights $\mathbf{w}_i$ and bias b_i , $i = 1, \dots, n$.

Step 2: Calculate the hidden layer output matrix $\mathbf{H}$.

Step 3: Calculate the output weights β :

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{7}$$

where $\mathbf{H}$, β and are as defined before.

4 Experiments

4.1 Data Set – MCV1

Manufacturing Corpus Version 1 (MCV1) is an archive of 1434 English language manufacturing related engineering papers. It combines all engineering technical

Table 1. The 18 major categories of MCV1

C01. Assembly & Joining	C07. Machining & Material Removal Processes	C13. Product Design Management
C02. Composites Manufacturing	C08. Manufacturing Engineering & Management	C14. Quality
C03. Electronics Manufacturing	C09. Manufacturing Systems, Automation & IT	C15. Rapid Prototyping
C04. Finishing & Coating	C10. Materials	C16. Research & Development / New Technologies
C05. Forming & Fabricating	C11. Measurement, Inspection & Testing	C17. Robotics & Machine Vision
C06. Lean Manufacturing & Supply Chain Management	C12. Plastics Molding & Manufacturing	C18. Welding

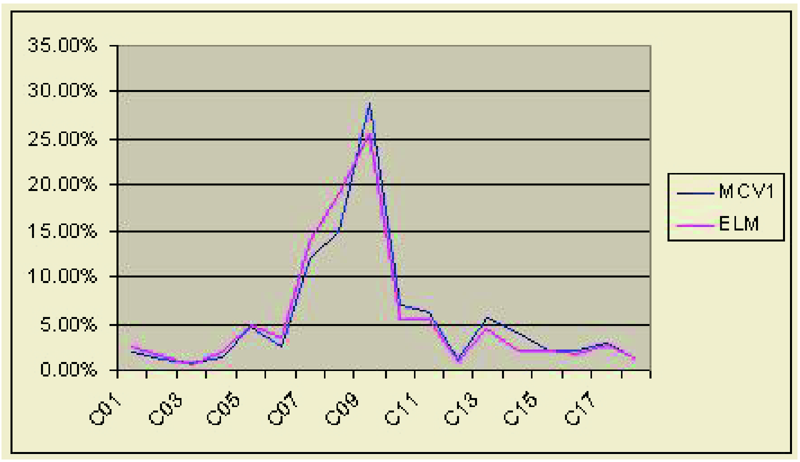

Fig. 1. Documents frequency distribution of MCV1 and ELM data set

papers from Society of Manufacturing Engineers (SME) from year 1998 to year 2000 [12]. There are 18 major categories of documents and two levels of subcategories below them. The 18 major categories are shown in Table 1:

Each document in MCV1 is labeled with one to nine category labels. For the purpose of this research, only one label is associated with each document. It is mainly because the current version of ELM only takes the highest value from output neurons as the prediction; it cannot handle the problem of multiclass classification using a single ELM machine.

Figure 1 shows that the documents frequency distribution in ELM data set matches very well with the original distribution in MCV1.

Table 2 shows the detailed distribution of 1434 documents from different categories.

Table 2. Percentage of documents of 18 categories in ELM data set

C01	**C02**	**C03**	**C04**	**C05**	**C06**
2.58%	1.47%	0.70%	1.81%	4.95%	3.63%
C07	**C08**	**C09**	**C10**	**C11**	**C12**
13.96%	19.12%	25.40%	5.51%	5.44%	1.05%
C13	**C14**	**C15**	**C16**	**C17**	**C18**
4.47%	2.30%	2.02%	1.74%	2.65%	1.19%

4.2 Experimental Setting

In the experiments, only the abstract of each paper is used. All standard text processing procedures are applied in the experiments, including stop words removal, stemming. By using the general *tfidf* weighting scheme, the documents are represented in vector format. Chi-square five fold cross validation is used to evaluate the features for ELM dataset.

In order to compare with SVM strictly, one ELM machine is built over each of 18 major categories. Document vectors sharing the same category label will be set as positive and all other vectors are set as negative. This way of building data set is generically the same as the one for SVM. In this paper, we call this "one-against-all". One-against-all is different from purely binary classification in the sense that the negative part is composed by many different categories, instead of from a single opposite category. Therefore, there are totally 18 datasets. For each of them, five fold cross validation is assessed. SVM Light is chosen as the SVM package with linear function as the kernel function. For ELM, all data points have been normalized to $(-1,1)$ and sigmoid has been chosen as the activation function. The way to search for the optimal size of neurons is suggested by the authors in [7]. With the starting size of 20, the number of neurons increases with a step of 20. Based on the output performance, the optimal size of neurons will be decided. Finally based on the optimal sizes of neurons, 50 more trials are performed in order to collect the best output.

4.3 Performance Indicator

Accuracy has been used as the performance indicator for classification comparison with SVM in [7]. However, if the datasets are formed as one-against-all, accuracy is not always a good indicator. A very obvious example for this argument is a dataset that might have some categories with very few documents. If the system predicts all data points as negative, it can still generate a very high accuracy value since the negative portion of this data set, which is composed by many different categories, occupies the large percentage of this data set. With the negative prediction for a document, it is still unclear which category it belongs to. The building of our dataset rightly fits into this case. In order to avoid this problem and show the real performance of both algorithms, the classic F_1 value which is defined as $F_1 = \frac{2pr}{p+r}$ is adopted, where p represents precision and r represents recall [1], [15], [20]. This performance indicator combines the effects of precision and recall, and it has been widely used in TC domain.

4.4 Results and Discussion

Figure 2 shows the relationship between the size of neurons and its performance for ELM machines built over major categories in MCV1. Obviously, with the increase of neurons, ELM machines achieve the best performance very quickly and remain stable

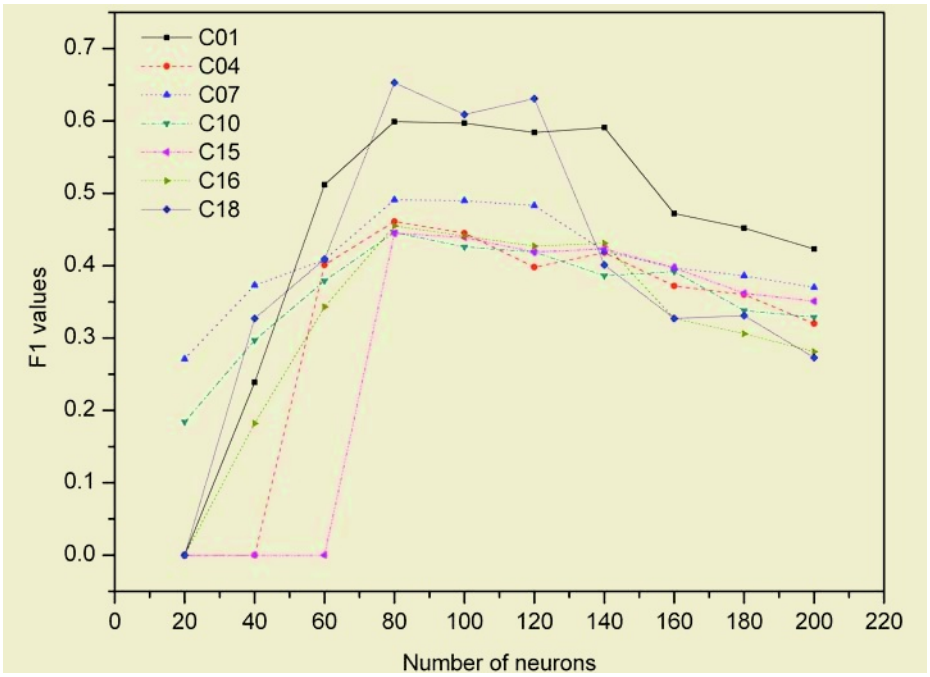

Fig. 2. Number of neurons vs. F_1 performance

for a wide range of neuron sizes. The broad spectrum of neuron size implies that ELM is robust to this critical parameter setting. It is also noted that for MCV1 dataset, 60-120 neurons can provide most categories with good performance in a few trials.

In the experiments, authors are curious whether feature selection still contributes towards the performance of ELM. Chi-Square five fold cross validation has been applied to select the salient features. With feature selection, the dimension has been dramatically reduced from over five thousand to less than one hundred. Table 3 shows the performance difference before and after feature selection. It is now clear that feature selection still has a critical role in ELM computation.

Table 3. Performance difference before and after feature selection

Category	No. of Documents	Percentage	F_1 ELM Before Feature Selection	F_1 ELM After Feature Selection
C01	37	2.58%	0.231	0.599
C02	21	1.47%	N/A	N/A
C03	10	0.70%	N/A	N/A
C04	26	1.81%	0.145	0.461
C05	71	4.95%	N/A	0.370
C06	52	3.63%	N/A	0.369
C07	200	13.96%	0.247	0.491
C08	274	19.12%	0.213	0.346
C09	364	25.40%	N/A	0.330
C10	79	5.51%	0.183	0.446
C11	78	5.44%	N/A	0.338
C12	15	1.05%	N/A	N/A
C13	64	4.47%	N/A	N/A
C14	33	2.30%	N/A	N/A
C15	29	2.02%	N/A	0.445
C16	25	1.74%	N/A	0.455
C17	38	2.65%	N/A	N/A
C18	17	1.19%	0.236	0.653

The most important results are F_1 values and accuracy values of SVM and ELM over 18 categories as shown in Table 4.

Note that SVM still outperforms ELM for the majority of categories. In some cases, the algorithms yield no results due to the lack of training samples or probably noise. In category C02, C03, and C14, when SVM does not work, ELM does not work as well. There are three categories, i.e. C12, C13, and C17, ELM does not work, while SVM still gives results. In two categories, i.e. C04 and C06, ELM slightly outperforms SVM and in two more categories, the performance from both are close to each other. It is also noted that the performance of both algorithms, evaluated by F_1

values, does not necessarily link to the values of accuracy. In many instances, even where the ELM has higher accuracy values, SVM still outperforms ELM in terms of F_1 values.

Table 4. F_1 values and accuracy values of SVM and ELM over 18 categories

Category	No. of Documents	Per	F_1 SVM	F_1 ELM	Accuracy SVM	Accuracy ELM	F_1 Difference (SVM-ELM)	Accuracy Difference (SVM-ELM)
C01	37	2.58%	0.699	0.599	0.980	0.984	0.099	-0.004
C02	21	1.47%	N/A	N/A	0.970	0.985	N/A	-0.014
C03	10	0.70%	N/A	N/A	0.986	0.994	N/A	-0.007
C04	26	1.81%	0.459	0.461	0.978	0.986	-0.002	-0.008
C05	71	4.95%	0.486	0.370	0.932	0.930	0.116	0.002
C06	52	3.63%	0.361	0.369	0.934	0.961	-0.007	-0.026
C07	200	13.96%	0.624	0.491	0.864	0.866	0.134	-0.003
C08	274	19.12%	0.548	0.346	0.684	0.800	0.202	-0.116
C09	364	25.40%	0.491	0.330	0.534	0.687	0.161	-0.153
C10	79	5.51%	0.485	0.446	0.927	0.944	0.039	-0.018
C11	78	5.44%	0.521	0.338	0.922	0.933	0.183	-0.011
C12	15	1.05%	0.511	N/A	0.977	0.988	>>	-0.011
C13	64	4.47%	0.225	N/A	0.884	0.953	>>	-0.069
C14	33	2.30%	N/A	N/A	0.959	0.976	N/A	-0.017
C15	29	2.02%	0.566	0.445	0.969	0.977	0.121	-0.008
C16	25	1.74%	0.558	0.455	0.987	0.986	0.104	0.001
C17	38	2.65%	0.267	N/A	0.953	0.970	>>	-0.018
C18	17	1.19%	0.709	0.653	0.988	0.990	0.056	-0.002

In our experiments, the CPU time spent by both ELM and SVM are trivial. As mentioned before in section 2, in TC tasks, many documents can be linearly classified in high dimensional space [8]. It is well known that with the sigmoid or RBFs as the kernel functions, SVM suffers from its tedious parameter tuning. So in TC tasks it is ideal for SVM to adopt a linear function as the kernel function to save much time on parameter tuning. By comparison, even with a single parameter to be tuned, the arbitrary assignment of initial weights requires ELM to search for the optimal size of neuron and run many times to get the average value [7]. In this case, ELM loses its edge over SVM.

5 Conclusion

In this paper, we have studied the performance of SVM and the newly available ELM algorithm for TC tasks. F_1 has been used to evaluate the performance because of its better suitability than accuracy as an indicator. While the ELM is easy to tune with a single parameter and is robust to the parameter settings, it is shown that SVM still outperforms ELM for the majority of categories in terms of F_1 values. Furthermore, accuracy does not have clear links with the performance evaluated by F_1. Compared to SVM with linear function as kernel function, the advantage of fast training of ELM is not significant in TC tasks.

References

1. Baeza-Yates, R. & Ribeiro-Neto, B.: Modern information retrieval. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, (1999)
2. Bishop, C. M.: Neural Networks for Pattern Recognition. Oxford University Press, (1996)
3. Burges, C. J. C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, Vol. 2, (1998) 121-167
4. Cristianini, N. & Shawe-Taylor, J.: An introduction to Support Vector Machines: and other kernel-based learning methods. Cambridge University Press, (2000)
5. Dumais, S. & Chen, H.: Hierarchical classification of Web content. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR2000), (2000)
6. Flach, P. A.: On the state of the art in machine learning: a personal review. Artificial Intelligence, Vol. 13, (2001) 199-222
7. Huang, G. B., Zhu, Q. Y. & Siew, C. K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. International Joint Conference on Neural Networks (IJCNN'2004), (2004)
8. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. Machine Learning: ECML-98, Tenth European Conference on Machine Learning, (1998)
9. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. Proceedings of the 16th International Conference on Machine Learning (ICML), (1999)
10. Kasabov, N. Data mining and knowledge discovery using neural networks. 2002
11. Leopold, E. & Kindermann, J.: Text Categorization with Support Vector Machines - How to Represent Texts in Input Space. Machine Learning, Vol. 46, (2002) 423-444
12. Liu, Y., Loh, H. T. & Tor, S. B.: Building a Document Corpus for Manufacturing Knowledge Retrieval. Singapore MIT Alliance Symposium 2004, (2004)
13. Mangasarian, O. L.: Data Mining via Support Vector Machines. 20th International Federation for Information Processing (IFIP) TC7 Conference on System Modeling and Optimization, (2001)
14. Manning, C. D. & Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, (1999)
15. Mitchell, T. M.: Machine Learning. The McGraw-Hill Companies, Inc., (1997)
16. Ng, H. T., Goh, W. B. & Low, K. L.: Feature selection, perception learning, and a usability case study for text categorization. ACM SIGIR Forum , Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, (1997)
17. Ruiz, M. E. & Srinivasan, P.: Hierarchical Neural Networks for Text Categorization. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, (1999)

18. Ruiz, M. E. & Srinivasan, P.: Hierarchical Text Categorization Using Neural Networks. Information Retrieval, Vol. 5, (2002) 87-118
19. Schölkopf, B. & Smola, A. J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. 1st edn. MIT Press, (2001)
20. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR), Vol. 34, (2002) 1-47
21. Sun, A. & Lim, E.-P.: Hierarchical Text Classification and Evaluation. Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001), (2001)
22. Vapnik, V. N.: The Nature of Statistical Learning Theory. 2nd edn. Springer-Verlag, New York (1999)
23. Wiener, E. D., Pedersen, J. O. & Weigend, A. S.: A neural network approach to topic spotting. Proceedings of {SDAIR}-95, 4th Annual Symposium on Document Analysis and Information Retrieval, (1995)
24. Weigend, A. S., Wiener, E. D. & Pedersen, J. O.: Exploiting hierarchy in text categorization. Information Retrieval, Vol. 1, (1999) 193-216
25. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval, Vol. 1, (1999) 69-90
26. Yang, Y. & Liu, X.: A re-examination of text categorization methods. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, (1999)