An Experimental Study on Large-Scale Web Categorization

Tie-Yan LIU[1], Yiming YANG[2], Hao WAN[3*], Qian ZHOU[3*], Bin GAO[4*],
Hua-Jun ZENG[1], Zheng CHEN[1], and Wei-Ying MA[1]

[1]Microsoft Research Asia, 5F, Sigma Center, No. 49, Zhichun Road, Beijing, 100080, P. R. China
[2] School of Computer Science, Carnegie Mellon University, PA, USA
[3]Dept. Electronic Engineering, Tsinghua University, Beijing, 100084, P. R. China
[4] Dept. Mathematics, Peking University, Beijing, 100084, P. R. China

{t-tyliu, hjzeng, zhengc, wyma}@microsoft.com, yiming@cs.cmu.edu

ABSTRACT

Taxonomies of the Web typically have hundreds of thousands of categories and skewed category distribution over documents. It is not clear whether existing text classification technologies can perform well on and scale up to such large-scale applications. To understand this, we conducted the evaluation of several representative methods (Support Vector Machines, *k*-Nearest Neighbor and Naive Bayes) with Yahoo! taxonomies. In particular, we evaluated the effectiveness/efficiency tradeoff in classifiers with hierarchical setting compared to conventional (flat) setting, and tested popular threshold tuning strategies for their scalability and accuracy in large-scale classification problems.

Categories and Subject Descriptors

F.2 [Analysis of Algorithms and Problem Complexity]: Miscellaneous; I.5.4 [Pattern Recognition]: Applications – Text processing.

General Terms

Technology Assessment, Performance and Scalability Analysis, Empirical Validation.

Keywords

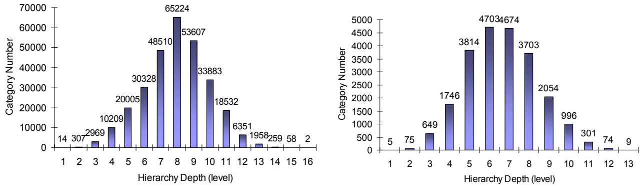
Text categorization, very large Web taxonomies, parameter tuning strategies and algorithm complexity

1. INTRODUCTION

With the fast development of the World Wide Web, there has emerged a great need to manage the massive information on the Web. As compared to manually labeling, automated text categorization (TC) might be more desirable for this purpose. Recently, many machine learning algorithms [5][6] have been developed or adopted to text categorization, including Support Vector Machines (SVM), *k*-Nearest Neighbor (*k*-NN), Linear Regression, Naïve Bayes (NB) and so on. Although researchers have achieved great progress in TC, empirical studies in the literature have not yet provided us with an answer whether existing methods can successfully solve the problem of large-scale Web categorization. The major challenge is that Web taxonomies (such as Yahoo! Directory) often have hundreds of thousands of categories and skewed category distribution over documents, which is quite different from widely-used benchmark data sets (such as RCV1 [3]) in the literature of TC. To tackle this challenge, we conducted an experimental study on the effectiveness and efficiency of popular TC methods (SVM, *k*-NN and NB) directly over a specific sub set of Yahoo! Directory that has very similar statistics to the full domain of Yahoo! taxonomy.

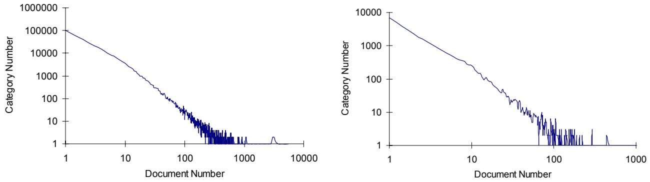
2. DATA CORPUS

Yahoo! Directory is a famous Web taxonomy maintained by the human editors of Yahoo.com, and has been used in many previous works on text categorization [1][4]. Considering the large scale of this corpus (in June 2004, it contained 292,216 categories and 792,601 documents which were organized into a 16-level hierarchy), data sampling was conducted in these works. However, their sampling strategies, i.e. only using the top few levels or selective common categories, could not preserve the original characteristics of Yahoo! Directory. To tackle this problem, we manually chose a specific sub set of Yahoo! Directory which has very similar category distribution to the full set. We named this sub set by MERG, which consists of five sub trees "News and **M**edia", "**E**ntertainment", "**R**eference", "**G**overnment" and "Regional" (documents in "Regional" are selected if they also belong to one of the other four categories). MERG has totally 22,803 categories and 54,542 documents that are organized into a 13-level hierarchy. As can be seen from Figure 1 and 2, both MERG and Yahoo! Directory take similar statistical characteristics: spindle category distribution over levels, and power law distribution of category size (which means that most categories are rare categories with very few positive examples). In this regard, we believe that the experiments on this subset can better reflect the true situation of Web directory classification than any previous works.



(a) Yahoo! Directory (b) MERG

Figure 1. Spindle category distribution over levels.



(a) Yahoo! Directory (b) MERG

Figure 2. Power law distribution of the category size.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

*The works of Hao WAN, Qian ZHOU and Bin GAO were performed at Microsoft Research Asia.

3. EXPERIMENTAL RESULTS

When conducting our experimental study, we followed the settings listed as below. We divided MERG into a training set and a test set with a ratio of 7:3. Note that during this process, we removed those categories containing only one document for ease of evaluation. As a result, 8050 categories of MERG were used while the training and the test sets contained 33,689 and 19,964 documents respectively. For SVM, we selected 4000 features using CHI algorithm [2]; for *k*-NN, we set *k* to 100 according to [3]; for NB, we used the logarithm of the probabilities for convenience of computation. For each classifier, we conducted three runs: "flat classification with SCut [7]", "flat classification with RCut [7]" and "hierarchical classification with SCut". For each run, we logged the classification performance as well as the time complexity (a workstation with 3G-Hz CPU and 2GB memory was used in our experiments).

The performance of SVM classification was reported in Figure 3. From it, we found that the performance decreased with the increasing hierarchy depth. For upper levels, SCut outperformed RCut. However, RCut finally went beyond SCut for deep levels. We also found that hierarchical SVM had higher accuracy than flat SVM. From the time complexities of SVM classification listed in Table 1, we could see that 1) SCut had the dominant complexity in the training process; 2) hierarchical classification could save the computations greatly.

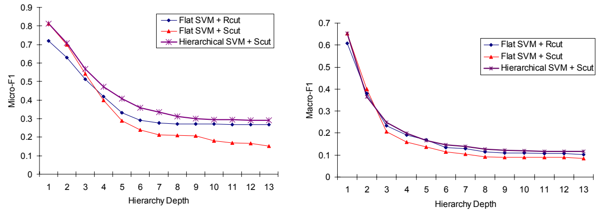


Figure 3: Classification performance of SVM over MERG

The classification performance for *k*-NN was shown in Figure 4, from which we could see similar trend to SVM for the performance with respect to hierarchy depth. The difference is that this time hierarchical *k*-NN performed poorer than flat *k*-NN. Our explanation to this is as follow. Since many categories in Yahoo! Directory only have few documents, for the corresponding local classification tasks of hierarchical *k*-NN, the number of training examples might be even less than *k*, thus not enough for reliable instance-based learning. And from Table 1, we found that unlike SVM, hierarchical classification increased the complexity of *k*-NN. Actually, this is true for all *m*-way classifiers since only the training of the top-level classifiers will already take the same complexity with flat classification.

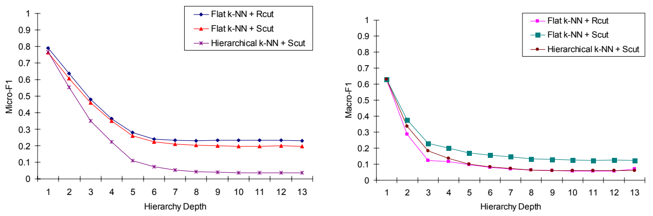


Figure 4: Classification performance of *k*-NN over MERG

The performance for NB was shown in Figure 5. From this figure we found that unlike SVM and *k*-NN, no matter with which setting, NB always gave very poor classification accuracy. Our explanation is that NB, as a generative classifier, suffers more from the data sparseness in Yahoo! Directory than SVM and *k*-NN. That is, the lack of positive examples simply can not provide enough information to learn a reliable NB classifier. Due to the low classification accuracy, no matter how fast NB could be, we can not use it on real-world Yahoo-like corpora.

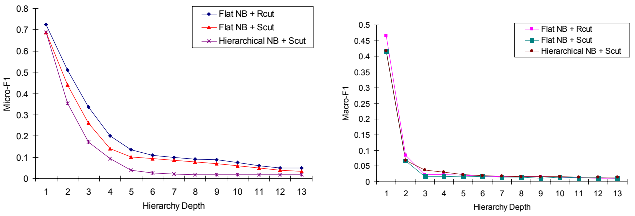


Figure 5: Classification performance of NB over MERG

Table 1. Time complexity of different classifiers (hour)

Algorithms	Training		Test
	Without SCut	With SCut	
Flat SVM	14.15	62.51	8.41
Hierarchical SVM	1.09	4.54	0.167
Flat *k*-NN	0.01	1.85	1.09
Hierarchical *k*-NN	0.23	1.30	171.06
Flat NB	0.46	1.85	0.28
Hierarchical NB	0.14	76.73	47.98

To summarize, hierarchical SVM with SCut performed best among all the settings. However, even this best one can not offer satisfactory classification accuracy for real-world applications. For the 3rd and deeper levels, the Macro-F1 has gone below 25%. This indicates that automated text categorization with very-large taxonomies still poses unsolved challenges.

4. CONCLUSIONS

In this paper, we conducted the evaluation of representative text categorization methods (Support Vector Machines, *k*-Nearest Neighbor and Naive Bayes) with the Yahoo! web-page taxonomy. Based on our experiments, we got the following conclusions:

1) Hierarchical setting saved computations and improved classification accuracy for SVM, but did harm to *k*-NN and NB in sense of both effectiveness and efficiency.
2) Threshold tuning (SCut in our paper) could be a dominant part of the offline training.
3) Hierarchical SVM with SCut had the best tradeoff between efficiency and effectiveness. However, its classification accuracy was still rather low, indicating that automated text categorization with very-large taxonomies still poses unsolved challenges.

5. REFERENCES

[1] Attardi, G., Gulli, A., Sebastiani, F., Automatic Web Page Categorization by Link and Context Analysis, *THAI* 1999.

[2] Forman, G. An extensive experimental study of feature selection metrics for text classification. *Journal of Machine Learning Research*, Vol.3, 1289-1305, 2003.

[3] Lewis, D. D., Yang, Y., Rose, T. G., Li, F. RCV1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, Vol.5, 361-397, 2004.

[4] Mladenic, D., Grobelnik, M., Word sequences as features in text-learning. *ERK* 1998, 145-148.

[5] Sebastiani, F. Machine learning in automated text categorization. *ACM Computing Surveys*, Vol.34, No.1, 1-47, 2002.

[6] Yang, Y., and Liu, X. A re-examination of text categorization methods, *SIGIR* 1999, 42-49.

[7] Yang, Y. A study of thresholding strategies for text categorization, *SIGIR* 2001, 137-145.