Machine Learning Methods for Chinese Web Page Categorization

Ji He[1], Ah-Hwee Tan[2] and Chew-Lim Tan[1]
[1]School of Computing, National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
{heji,tancl}@comp.nus.edu.sg
[2]Kent Ridge Digital Labs
21 Heng Mui Keng Terrace, Singapore 119613
ahhwee@krdl.org.sg

Abstract

This paper reports our evaluation of k Nearest Neighbor (kNN), Support Vector Machines (SVM), and Adaptive Resonance Associative Map (ARAM) on Chinese web page classification. Benchmark experiments based on a Chinese web corpus showed that their predictive performance were roughly comparable although ARAM and kNN slightly outperformed SVM in small categories. In addition, inserting rules into ARAM helped to improve performance, especially for small well-defined categories.

1 Introduction

Text categorization refers to the task of automatically assigning one or multiple predefined category labels to free text documents. Whereas an extensive range of methods has been applied to English text categorization, relatively few have been benchmarked for Chinese text categorization. Typical approaches to Chinese text categorization, such as Naive Bayes (NB) (Zhu, 1987), Vector Space Model (VSM) (Zou et al., 1998; Zou et al., 1999) and Linear List Square Fit (LLSF) (Cao et al., 1999; Yang, 1994), have well studied theoretical basis derived from the information retrieval research, but are not known to be the best classifiers (Yang and Liu, 1999; Yang, 1999). In addition, there is a lack of publicly available Chinese corpus for evaluating Chinese text categorization systems.

This paper reports our applications of three statistical machine learning methods, namely k Nearest Neighbor system (kNN) (Dasarathy, 1991), Support Vector Machines (SVM) (Cortes and Vapnik, 1995), and Adaptive Resonance Associative Map (ARAM) (Tan, 1995) to Chinese web page categorization. kNN and SVM have been reported as the top performing methods for English text categorization (Yang and Liu, 1999). ARAM belongs to a popularly known family of predictive self-organizing neural networks which until recently has not been used for document classification. The trio has been evaluated based on a Chinese corpus consisting of news articles extracted from People's Daily (He et al., 2000). This article reports the experiments of a much more challenging task in classifying Chinese web pages. The Chinese web corpus was created by downloading from various Chinese web sites covering a wide variety of topics. There is a great diversity among the web pages in terms of document length, style, and content. The objectives of our experiments are two-folded. First, we examine and compare the capabilities of these methods in learning categorization knowledge from real-life web documents. Second, we investigate if incorporating domain knowledge derived from the category description can enhance ARAM's predictive performance.

The rest of this article is organized as follows. Section 2 describes our choice of the feature selection and extraction methods. Section 3 gives a summary of the kNN and SVM, and presents the less familiar ARAM algorithm in more details. Section 4 presents our evaluation paradigm and reports the experi-

mental results.

2 Feature Selection and Extraction

A pre-requisite of text categorization is to extract a suitable feature representation of the documents. Typically, word stems are suggested as the representation units by information retrieval research. However, unlike English and other Indo-European languages, Chinese text does not have a natural delimiter between words. As a consequence, word segmentation is a major issue in Chinese document processing. Chinese word segmentation methods have been extensively discussed in the literature. Unfortunately perfect precision and disambiguation cannot be reached. As a result, the inherent errors caused by word segmentation always remains as a problem in Chinese information processing.

In our experiments, a word-class bi-gram model is adopted to segment each training document into a set of tokens. The lexicon used by the segmentation model contains 64,000 words in 1,006 classes. High precision segmentation is not the focus of our work. Instead we aim to compare different classifier's performance on noisy document set as long as the errors caused by word segmentation are reasonably low.

To select keyword features for classification, χ (CHI) statistics is adopted as the ranking metric in our experiments. A prior study on several well-known corpora including Reuters-21578 and OHSUMED has proven that CHI statistics generally outperforms other feature ranking measures, such as term strength (TS), document frequency (DF), mutual information (MI), and information gain (IG) (Yang and J.P, 1997).

During keyword extraction, the document is first segmented and converted into a keyword frequency vector $(tf_1, tf_2, \ldots, tf_M)$, where tf_i is the in-document term frequency of keyword w_i, and M is the number of the keyword features selected. A term weighting method based on *inverse document frequency* (IDF) (Salton, 1988) and the L1-normalization are then applied on the frequency vector to produce the keyword feature vector

$$\mathbf{x} = \frac{(x_1, x_2, \ldots, x_M)}{\max\{x_i\}}, \quad (1)$$

in which x_i is computed by

$$x_i = (1 + \log_2 tf_i) \log_2 \frac{n}{n_i}, \quad (2)$$

where n is the number of documents in the whole training set, and n_i is the number of training documents in which the keyword w_i occurs at least once.

3 The Classifiers

3.1 k Nearest Neighbor

k Nearest Neighbor (kNN) is a traditional statistical pattern recognition algorithm (Dasarathy, 1991). It has been studied extensively for text categorization (Yang and Liu, 1999). In essence, kNN makes the prediction based on the k training patterns that are closest to the unseen (test) pattern, according to a distance metric. The distance metric that measures the similarity between two normalized patterns can be either a simple L1-distance function or a L2-distance function, such as the plain Euclidean distance defined by

$$D(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_i (a_i - b_i)^2}. \quad (3)$$

The class assignment to the test pattern is based on the class assignment of the closest k training patterns. A commonly used method is to label the test pattern with the class that has the most instances among the k nearest neighbors. Specifically, the class index $y(x)$ assigned to the test pattern $\mathbf{x}$ is given by

$$y(x) = \text{argmax}_i\{n(\mathbf{d}_j, c_i) | \mathbf{d}_j \in kNN\}, \quad (4)$$

where $n(\mathbf{d}_j, c_i)$ is the number of training pattern $\mathbf{d}_j$ in the k nearest neighbor set that are associated with class c_i.

The drawback of kNN is the difficulty in deciding a optimal k value. Typically it has to be determined through conducting a series of experiments using different values.

3.2 Support Vector Machines

Support Vector Machines (SVM) is a relatively new class of machine learning techniques first introduced by Vapnik (Cortes and Vapnik, 1995). Based on the *structural risk minimization* principle from the computational learning theory, SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the *support vectors* that are selected as the only effective elements from the training set.

Given a set of linearly separable points $S = \{\mathbf{x}_i \in \mathbf{R}^n | i = 1, 2, \ldots, N\}$, each point $\mathbf{x}_i$ belongs to one of the two classes, labeled as $y_i \in \{-1, +1\}$. A *separating hyper-plane* divides S into two sides, each side containing points with the same class label only. The *separating hyper-plane* can be identified by the pair $(\mathbf{w}, b)$ that satisfies

$$\begin{aligned} &\mathbf{w} \cdot \mathbf{x} + b = 0 \\ \text{and} \quad &y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \end{aligned} \tag{5}$$

for $i = 1, 2, \ldots, N$; where the dot product operation $\cdot$ is defined by

$$\mathbf{w} \cdot \mathbf{x} = \sum_i w_i x_i \tag{6}$$

for vectors $\mathbf{w}$ and $\mathbf{x}$. Thus the goal of the SVM learning is to find the *optimal separating hyper-plane* (*OSH*) that has the maximal margin to both sides. This can be formularized as:

$$\begin{aligned} \text{minimize} \quad &\tfrac{1}{2}\mathbf{w} \cdot \mathbf{w} \\ \text{subject to} \quad &y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \end{aligned} \tag{7}$$

The points that are closest to the *OSH* are termed *support vectors* (Fig. 1).

The SVM problem can be extended to linearly non-separable case and non-linear case. Various quadratic programming algorithms have been proposed and extensively studied to solve the SVM problem (Cortes and Vapnik, 1995; Joachims, 1998; Joachims, 1999).

During classification, SVM makes decision based on the *OSH* instead of the whole training set. It simply finds out on which side of the *OSH* the test pattern is located.

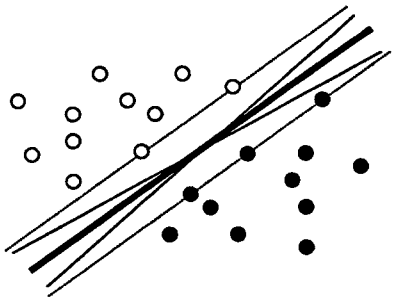

Figure 1: *Separating hyperplanes* (the set of solid lines), *optimal separating hyperplane* (the bold solid line), and *support vectors* (data points on the dashed lines). The dashed lines identify the max margin.

This property makes SVM highly competitive, compared with other traditional pattern recognition methods, in terms of computational efficiency and predictive accuracy (Yang and Liu, 1999).

In recent years, Joachims has done much research on the application of SVM to text categorization (Joachims, 1998). His SVM^{light} system published via http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html is used in our benchmark experiments.

3.3 Adaptive Resonance Associative Map

Adaptive Resonance Associative Map (ARAM) is a class of predictive self-organizing neural networks that performs incremental supervised learning of recognition categories (pattern classes) and multidimensional maps of patterns. An ARAM system can be visualized as two overlapping Adaptive Resonance Theory (ART) modules consisting of two input fields F_1^a and F_1^b with an F_2 category field (Tan, 1995; Tan, 1997) (Fig. 2). For classification problems, the F_1^a field serves as the input field containing the document feature vector and the F_1^b field serves as the output field containing the class prediction vector. The F_2 field contains the activities of the recognition categories that are used to encode the patterns.

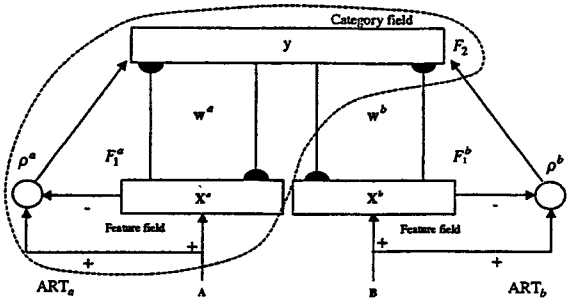

Figure 2: The Adaptive Resonance Associative Map architecture

When performing classification tasks, ARAM formulates recognition categories of input patterns, and associates each category with its respective prediction. During learning, given an input pattern (document feature) presented at the F_1^a input layer and an output pattern (known class label) presented at the F_1^b output field, the category field F_2 selects a winner that receives the largest overall input. The winning node selected in F_2 then triggers a top-down priming on F_1^a and F_1^b, monitored by separate reset mechanisms. Code stabilization is ensured by restricting encoding to states where resonance are reached in both modules. By synchronizing the unsupervised categorization of two pattern sets, ARAM learns supervised mapping between the pattern sets. Due to the code stabilization mechanism, fast learning in a real-time environment is feasible.

The knowledge that ARAM discovers during learning is compatible with IF-THEN rule-based presentation. Specifically, each node in the F_2 field represents a recognition category associating the F_1^a patterns with the F_1^b output vectors. Learned weight vectors, one for each F_2 node, constitute a set of rules that link antecedents to consequences. At any point during the incremental learning process, the system architecture can be translated into a compact set of rules. Similarly, domain knowledge in the form of IF-THEN rules can be inserted into ARAM architecture.

The ART modules used in ARAM can be ART 1, which categorizes binary patterns, or analog ART modules such as ART 2, ART 2-A, and fuzzy ART, which categorize both binary and analog patterns. The fuzzy ARAM (Tan, 1995) algorithm based on fuzzy ART (Carpenter et al., 1991) is introduced below.

Parameters: Fuzzy ARAM dynamics are determined by the choice parameters $\alpha_a > 0$ and $\alpha_b > 0$; the learning rates $\beta_a \in [0,1]$ and $\beta_b \in [0,1]$; the vigilance parameters $\rho_a \in [0,1]$ and $\rho_b \in [0,1]$; and the contribution parameter $\gamma \in [0,1]$.

Weight vectors: Each F_2 category node j is associated with two adaptive weight templates $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$. Initially, all category nodes are uncommitted and all weights equal ones. After a category node is selected for encoding, it becomes *committed.*

Category choice: Given the F_1^a and F_1^b input vectors $\mathbf{A}$ and $\mathbf{B}$, for each F_2 node j, the choice function T_j is defined by

$$T_j = \gamma \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{\alpha_a + |\mathbf{w}_j^a|} + (1-\gamma)\frac{|\mathbf{B} \wedge \mathbf{w}_j^b|}{\alpha_b + |\mathbf{w}_j^b|}, \quad (8)$$

where the fuzzy AND operation $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i), \quad (9)$$

and where the norm $|.|$ is defined by

$$|\mathbf{p}| \equiv \sum_i p_i \quad (10)$$

for vectors $\mathbf{p}$ and $\mathbf{q}$.

The system is said to make a choice when at most one F_2 node can become active. The choice is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \quad (11)$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$.

Resonance or reset: Resonance occurs if the *match functions*, m_J^a and m_J^b, meet the vigilance criteria in their respective modules:

$$m_J^a = \frac{|\mathbf{A} \wedge \mathbf{w}_J^a|}{|\mathbf{A}|} \geq \rho_a \quad (12)$$

and

$$m_J^b = \frac{|\mathbf{B} \wedge \mathbf{w}_J^b|}{|\mathbf{B}|} \geq \rho_b. \quad (13)$$

Learning then ensues, as defined below. If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_J is set to 0 for the duration of the input presentation. The search process repeats to select another new index J until resonance is achieved.

Learning: Once the search ends, the weight vectors $\mathbf{w}_J^a$ and $\mathbf{w}_J^b$ are updated according to the equations

$$\mathbf{w}_J^{a(\text{new})} = (1-\beta_a)\mathbf{w}_J^{a(\text{old})} + \beta_a(\mathbf{A} \wedge \mathbf{w}_J^{a(\text{old})}) \tag{14}$$

and

$$\mathbf{w}_J^{b(\text{new})} = (1-\beta_b)\mathbf{w}_J^{b(\text{old})} + \beta_b(\mathbf{B} \wedge \mathbf{w}_J^{b(\text{old})}) \tag{15}$$

respectively. *Fast learning* corresponds to setting $\beta_a = \beta_b = 1$ for committed nodes.

Classification: During classification, using the choice rule, only the F_2 node J that receives maximal $F_1^a \rightarrow F_2$ input T_j predicts ART_b output. In simulations,

$$y_j = \begin{cases} 1 & \text{if } j = J \text{ where } T_J > T_k \\ & \text{for all } k \neq J \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The F_1^b activity vector $\mathbf{x}^b$ is given by

$$\mathbf{x}^b = \sum_j \mathbf{w}_j^b y_j = \mathbf{w}_J^b. \tag{17}$$

The output prediction vector $\mathbf{B}$ is then given by

$$\mathbf{B} \equiv (b_1, b_2, \ldots b_N) = \mathbf{x}^b \tag{18}$$

where b_i indicates the likelihood or confidence of assigning a pattern to category i.

Rule insertion: Rule insertion proceeds in two phases. The first phase parses the rules for keyword features. When a new keyword is encountered, it is added to a keyword feature table containing keywords obtained through automatic feature selection from training documents. Based on the keyword feature table, the second phase of rule insertion translates each rule into a M-dimensional vector $\mathbf{a}$ and a N-dimensional vector $\mathbf{b}$, where M is the total number of features in the keyword feature table and N is the number of categories. Given a rule of the following format,

IF $x_1, x_2, \ldots, x_m$
THEN $y_1, y_2, \ldots, y_n$

where $x_1, \ldots, x_m$ are antecedents and $y_1, \ldots, y_n$ are consequences, the algorithm derives a pair of vectors $\mathbf{a}$ and $\mathbf{b}$ such that for each index $i = 1, \ldots, M$,

$$a_i = \begin{cases} 1 & \text{if } w_i = x_j \text{ for some } j \in \{1, \ldots, m\} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where w_i is the i^{th} entry in the keyword feature table; and for each index $i = 1, \ldots, N$,

$$b_i = \begin{cases} 1 & \text{if } w_i = y_j \text{ for some } j \in \{1, \ldots, n\} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where w_i is the class label of the category i.

The vector pairs derived from the rules are then used as training patterns to initialize a ARAM network. During rule insertion, the vigilance parameters ρ_a and ρ_b are each set to 1 to ensure that only identical attribute vectors are grouped into one recognition category. Contradictory symbolic rules are detected during rule insertion when identical input attribute vectors are associated with distinct output attribute vectors.

4 Empirical Evaluation

4.1 The Chinese Web Corpus

The Chinese web corpus, collected in-house, consists of web pages downloaded from various Chinese web sites covering a wide variety of topics. Our experiments are based on a subset of the corpus consisting of 8 top-level categories and over 6,000 documents. For each category, we conduct binary classification experiments in which we tag the current category as the positive category and the other seven categories as the negative categories. The corpus is further partitioned into training and testing data such that the number of the training documents is at least 2.5 times of that of the testing documents (Table 1).

Table 1: The eight top-level categories in the Chinese web corpus, and the training and test samples by category.

Category	Description	Train	Test
Art	Topic regarding literature, art	325	102
Belief	Philosophy and religious beliefs	131	40
Biz	Business	2647	727
Edu	Education	205	77
IT	Computer and internet informatics	1085	309
Joy	Online fresh, interesting info	636	216
Med	Medical care related web sites	155	57
Sci	Various kinds of science	119	39

Table 2: A sample set of 19 rules generated based on the accompanied description of the Chinese web categories.

Art	:-	国画 (Chinese painting)
Belief	:-	祷告 (pray) 法师 (rabbi)
Biz	:-	促销 (promotion) 房地产 (real estate) 客户 (client)
Edu	:-	本科 (undergraduate) 导师 (supervisor) 校园 (campus)
IT	:-	版本 (version) 病毒 (virus) 防火墙 (firewall) 程式 (program)
Joy	:-	灯谜 (lantern riddle)
Med	:-	保健 (health care) 处方 (prescription) 法医学 (medical jurisprudence)
Sci	:-	超自然 (supernaturalism) 高技术 (high technology)

4.2 Experiment Paradigm

kNN experiments used the plain Euclidean distance defined by equation (3) as the similarity measure. On each pattern set containing a varying number of documents, different values of k ranging from 1 to 29 were tested and the best results were recorded. Only odd k were used to ensure that a prediction can always be made.

SVM experiments used the default built-in inductive SVM parameter set in SVM^{light}, which is described in detail on the web site and elsewhere (Joachims, 1999).

ARAM experiments employed a standard set of parameter values of fuzzy ARAM. In addition, using a voting strategy, 5 ARAM systems were trained using the same set of patterns in different orders of presentation and were combined to yield a final prediction vector.

To derive domain theory on web page classification, a varying number (ranging from 10 to 30) of training documents from each category were reviewed. A set of domain knowledge consists of 56 rules with about one to 10 rules for each category was generated. Only positive rules that link keyword antecedents to positive category consequences were included (Table 2).

4.3 Performance Measures

Our experiments adopt the most commonly used performance measures, including the *recall*, *precision*, and F_1 measures. *Recall* (R) is the percentage of the documents for a given category that are classified correctly. *Precision* (P) is the percentage of the predicted documents for a given category that are classified correctly. F_1 rating is one of the commonly used measures to combine R and P into a single rating, defined as

$$F_1 = \frac{2RP}{(R+P)}. \tag{21}$$

These scores are calculated for a series of binary classification experiments, one for each category. Micro-averaged scores and macro-averaged scores on the whole corpus are then produced across the experiments. With micro-averaging, the performance measures are produced across the documents by adding up all the documents counts across the different tests, and calculating using these summed values. With macro-averaging, each category is assigned with the same weight and performance measures are calculated across the categories. It is understandable that micro-averaged scores and macro-averaged scores reflect a classifier's performance on large categories and small categories respectively (Yang and Liu, 1999).

Table 3: Predictive performance of the four classifiers on the Chinese web corpus.

	kNN			SVM		
Category	P	R	F_1	P	R	F_1
Art	.795	.304	.440	.398	.402	.400
Belief	.773	.425	.548	.556	.500	.526
Biz	.724	.689	.706	.692	.703	.698
Edu	.380	.351	.365	.602	.074	.180
IT	.309	.333	.321	.394	.307	.345
Joy	.381	.236	.291	.462	.255	.328
Med	.833	.351	.494	.330	.544	.411
Sci	.625	.128	.213	.137	.179	.156
Micro-ave.	.584	.482	.528	.523	.521	.522
Macro-ave.	.600	.352	.422	.384	.454	.380
	ARAM			ARAM w/rules		
Category	P	R	F_1	P	R	F_1
Art	.653	.461	.540	.706	.471	**.565**
Belief	.750	.750	.750	.714	.750	.732
Biz	.742	.622	.677	.745	.604	.667
Edu	.421	.312	.358	.420	.273	.331
IT	.444	.259	.327	.437	.291	**.350**
Joy	.600	.208	.309	.618	.194	.296
Med	.421	.421	.421	.448	.456	**.452**
Sci	.292	.179	.222	.409	.231	**.295**
Micro-ave.	.619	.453	.523	.628	.450	**.524**
Macro-ave.	.540	.402	.451	.562	.409	**.461**

4.4 Results and Discussions

Table 3 summarizes the three classifier's performances on the test corpus in terms of *precision*, *recall*, and F_1 measures. The micro-averaged scores produced by the trio, which were predominantly determined by the classifiers' performance on the large categories (such as *Biz*, *IT*, and *Joy*), were roughly comparable. Among the three, kNN seemed to be marginally better than SVM and ARAM. Inserting rules into ARAM did not have a significant impact. This showed that domain knowledge was not very useful for categories that already have a large number of training examples. The differences in the macro-averaged scores produced by the three classifiers, however, were much more significant. The macro-averaged F_1 score obtained by ARAM was noticeably better than that of kNN, which in turn was higher than that of SVM. This indicates that ARAM (and kNN) tends to outperform SVM in small categories that have a smaller number of training patterns.

We are particularly interested in the classifier's learning ability on small categories. In certain applications, such as personalized content delivery, a large pre-labeled training corpus may not be available. Therefore, a classifier's ability of learning from a small training pattern set is a major concern. The different approaches adopted by these three classifiers in learning categorization knowledge are best seen in the light of the distinct learning peculiarities they exhibit on the small training sets.

kNN is a lazy learning method in the sense that it does not carry out any off-line learning to generate a particular category knowledge representation. Instead, kNN performs on-line scoring to find the training patterns that are nearest to a test pattern and makes the decision based on the statistical presumption that patterns in the same category have similar feature representations. The presumption is basically true to most pattern instances. Thus kNN exhibits a relatively stable performance across small and large categories.

SVM identifies *optimal separating hyperplane* (*OSH*) across the training data points and makes classification decisions based on the representative data instances (known as *support vectors*). Compared with kNN, SVM is more computationally efficient during classification for large-scale training sets. However, the *OSH* generated using small training sets may not be very representative, especially when the training patterns are sparsely distributed and there is a relatively narrow margin between the positive and negative patterns. In our experiments on small training sets including *Art*, *Belief*, *Edu*, and *Sci*, SVM's performance were generally lower than those of kNN and ARAM.

ARAM generates recognition categories from the input training patterns. The incrementally learned rules abstract the major representations of the training patterns and eliminate minor inconsistencies in the data patterns. During classifying, it works in a similar fashion as kNN. The major difference is that ARAM uses the learned recognition categories as the similarity-scoring unit whereas kNN uses the raw in-processed training patterns as the distance-scoring unit. It follows

that ARAM is notably more scalable than kNN by its pattern abstraction capability and therefore is more suitable for handling very large data sets.

The overall improvement in predictive performance obtained by inserting rules into ARAM is also of particular interest to us. ARAM's performance was more likely to be improved by rule insertion in categories that are well defined and have relatively fewer numbers of training patterns. As long as a user is able to abstract the category knowledge into certain specific rule representation, domain knowledge could complement the limited knowledge acquired through a small training set quite effectively.

Acknowledgements

We would like to thank our colleagues, Jian Su and Guo-Dong Zhou, for providing the Chinese segmentation software and Fon-Lin Lai for his valuable suggestions in designing the experiment system. In addition, we thank T. Joachims at the University of Dortmund for making SVM^{light} available.

References

Suqing Cao, Fuhu Zeng, and Huanguang Cao. 1999. A mathematical model for automatic Chinese text categorization. *Journal of the China Society for Scientific and Technical Information [in Chinese]*, 1999(1).

G.A. Carpenter, S. Grossberg, and D.B. Rosen. 1991. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771.

C. Cortes and V. Vapnik. 1995. Support vector networks. *Machine learning*, 20:273–297.

Belur V. Dasarathy. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, Las Alamitos, California.

Ji He, A.-H. Tan, and Chew-Lim Tan. 2000. A comparative study on Chinese text categorization methods. In *PRICAI'2000 International Workshop on Text and Web Mining*, Melbourne, August.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, Springer.

T. Joachims. 1999. *Making large-Scales SVM learning Pracical. Advances in Kernel Methods - Support Vector Learning.* B. Scholkopf, C. Burges and A. Smola (ed.), MIT Press.

Salton. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

A.-H. Tan. 1995. Adaptive resonance associative map. *Neural Networks*, 8(3):437–446.

A.-H. Tan. 1997. Cascade ARTMAP: Integrating neural computation and symbolic knowledge processing. *IEEE Transactions on Neural Networks*, 8(2):237–235.

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

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 42–49.

Y. Yang. 1994. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*.

Y. Yang. 1999. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88.

Lanjuan Zhu. 1987. The theory and experiments on automatic Chinese documents classification. *Journal of the China Society for Scientific and Technical Information [in Chinese]*, 1987(6).

Tao Zou, Ji-Cheng Wang, Yuan Huang, and Fu-Yan Zhang. 1998. The design and implementation of an automatic Chinese documents classification system. *Journal for Chinese Information [in Chinese]*, 1998(2).

Tao Zou, Yuan Huang, and Fuyan Zhang. 1999. Technology of information mining on WWW. *Journal of the China Society for Scientific and Technical Information [in Chinese]*, 1999(4).