A Comparative Study on Chinese Text Categorization Methods

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 comparative evaluation of three machine learning methods on Chinese text categorization. Whereas a wide range of methods have been applied to English text categorization, relatively few studies have been done on Chinese text categorization. Based on a People's Daily news corpus, a series of controlled experiments evaluate three machine learning methods, namely k Nearest Neighbor (kNN) algorithm, Support Vector Machines (SVM), and Adaptive Resonance Associative Map (ARAM), in terms of their capabilities in mining categorization knowledge from high dimensional, sparse, and relatively noisy document feature vectors. Experiments reveal that all three methods produce satisfactory performance on the test corpus while ARAM exhibits a marginally better generalization capability, especially from relatively small and noisy training sets.

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 have 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) [21], Vector Space Model (VSM) [22][23] and Linear List Square Fit (LLSF) [2][17], have well studied theoretical basis derived from the information retrieval research, but are not known to be the best classifiers [16][18]. 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) [19], Support Vector Machines (SVM) [15], and Adaptive Resonance Associative Map (ARAM) [12][13] to Chinese text categorization. kNN and SVM have been reported as the top performing methods for English text categorization [16]. ARAM belongs to a popularly known family of predictive self-organizing neural networks but until recently, has not been used for document classification. A People's Daily news

A.-H. Tan and P. Yu (Eds): PRICAI 2000 Workshop on Text and Web Mining, Melbourne, pp. 24-35, August 2000.

corpus is employed to evaluate the three machine learning methods. Through our benchmark experiments, we examine and compare the capabilities of these methods in mining categorization knowledge from high dimensional, sparse, and relatively noisy document feature vectors.

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 brief description of kNN and SVM and presents the relatively less familiar ARAM algorithm in more details. Section 4 presents our evaluation paradigm and reports the experimental results. The final section summarizes our findings and concludes.

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 [7].

In our experiments, a word-class bi-gram model is adopted to segment each 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 the keyword features for classification, the standard 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) [20].

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 seleted. A term weighting method based on *inverse document frequency* (IDF) [11] and the L2-normalization are then applied on the frequency vector to produce the keyword feature vector

$$\mathbf{x} = \frac{(x_1, x_2, \ldots, x_M)}{||\mathbf{x}||}, \tag{1}$$

where the L2-norm function $||\cdot||$ is defined by

$$||\mathbf{x}|| = \sqrt{\sum_i x_i^2} \tag{2}$$

for vector $\mathbf{x}$, and x_i is computed by

$$x_i = (1 + \log_2 tf_i) \log_2 \frac{n}{n_i}, \tag{3}$$

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 Classifiers

3.1 k Nearest Neighbor

k Nearest Neighbor (kNN) is a traditional statistical pattern recognition algorithm [1]. It has been studied extensively for text categorization applications [16]. In essence, kNN makes a prediction based on the k training patterns 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

$$D(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_i (a_i - b_i)^2}. \tag{4}$$

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 $y(\mathbf{x})$ assigned to the test pattern $\mathbf{x}$ is given by

$$y(\mathbf{x}) = \mathrm{argmax}_i \{n(\mathbf{d}_j, c_i) | \mathbf{d}_j \in kNN\}, \tag{5}$$

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 k values.

3.2 Support Vector Machines

Support Vector Machines (SVM) is a relatively new class of machine learning techniques first introduced by Vapnik [15]. 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 in the training set.

Given a set of N 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 2 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

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{6}$$

$$\text{and } \begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ if } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1 \end{cases} \tag{7}$$

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

$$\mathbf{w} \cdot \mathbf{x} = \sum_i w_i x_i \tag{8}$$

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:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{w} \cdot \mathbf{w} \\ \text{subject to} & \begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ if } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1 \end{cases} \text{for } i = 1, 2, \ldots, N. \end{array} \tag{9}$$

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

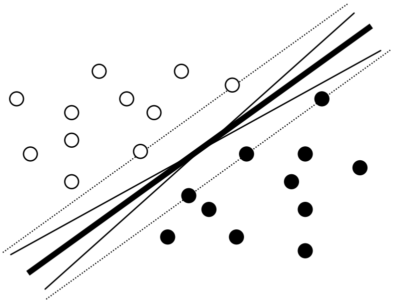

Fig. 1. *Separating hyperplanes* (the set of solid lines), *optimal separating hyperplane* (the bold solid line), and *support vectors* (data points on the dashed lines). The dashed lines identify the max margin.

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 [8][10][15].

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. This property makes SVM highly competitive, compared with other traditional pattern recognition methods, in terms of computational efficiency and predictive accuracy [16].

In recent years, Joachims has done much research on the application of SVM to text categorization [9]. 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[12][13] (Fig. 2). For classification problems, the F_1^a field serves as the input field containing the input activity vector and the F_1^b field servers as the output field containing the output class vector. The F_2 field contains the activities of the recognition categories that are used to encode the patterns. When performing classification tasks, ARAM formulates recognition categories of input patterns, and associates each category with its respective prediction. Given an pair of training patterns presented at the feature fields F_1^a and F_1^b, 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.

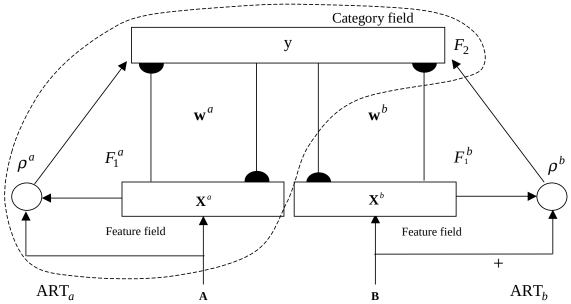

Fig. 2. The Adaptive Resonance Associative Map architecture

The ART modules used in ARAM can be ART 1 [3], which categorizes binary patterns, or analog ART modules such as ART 2 [4], ART 2-A [5], and fuzzy ART [6][14], which categorize both binary and analog patterns. The ARAM 2-A algorithm based on ART 2-A is introduced below.

Parameters: ARAM 2-A dynamics are determined by the learning rates $\beta_a \in [0,1]$ and $\beta_b \in [0,1]$; the vigilance parameters $\rho_a \in [0,1]$ and $\rho_b \in [0,1]$; the contribution parameter $\gamma \in [0,1]$; and the k-max decision parameter k.
Weight vectors: Each category node F_2 is associated with two adaptive weight template vectors $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$. Initially, all category nodes are uncommitted. 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

$$T_j = \gamma \frac{\mathbf{A} \cdot \mathbf{w}_j^a}{||\mathbf{A}||||\mathbf{w}_j^a||} + (1-\gamma) \frac{\mathbf{B} \cdot \mathbf{w}_j^b}{||\mathbf{B}||||\mathbf{w}_j^b||}. \tag{10}$$

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

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

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:

$$m_J^a = \frac{\mathbf{A} \cdot \mathbf{w}_J^a}{||\mathbf{A}||||\mathbf{w}_J^a||} \geq \rho_a \text{ and } m_J^b = \frac{\mathbf{B} \cdot \mathbf{w}_J^b}{||\mathbf{B}||||\mathbf{w}_J^b||} \geq \rho_b. \tag{12}$$

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 w_J^a and w_J^b are updated according to the equations

$$\mathbf{w}_J^{a(\text{new})} = (1-\beta_a)\mathbf{w}_J^{a(\text{old})} + \beta_a \mathbf{A} \tag{13}$$

and

$$\mathbf{w}_J^{b(\text{new})} = (1-\beta_b)\mathbf{w}_J^{b(\text{old})} + \beta_b \mathbf{B} \tag{14}$$

respectively. For ARAM 2-A, it is typical to set $\beta_a = \beta_b = 1$ when J is an uncommitted node, and then take $\beta_a < 1$ and $\beta_b < 1$ after the category node is *committed.*
K-max decision rule: During classification, ARAM 2-A works in the spirit of kNN system. Using a k-max rule, the output is predicted by a set of k F_2 nodes with the largest $F_1^a \to F_2$ input T_j. The F_2 activity values y_j are then computed by

$$y_j = \begin{cases} T_j / \sum_{k \in \pi} T_k & \text{if } j \text{ in } \pi \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where π is the set of k category nodes with the largest input T_j. The F_1^b activity vector $\mathbf{x}^b$ is given by

$$\mathbf{x}^b = \sum_j \mathbf{w}_j^b y_j. \tag{16}$$

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

$$\mathbf{B} \equiv (b_1, b_2, \ldots, b_N) = \mathbf{x}^b, \tag{17}$$

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

4 Empirical Evaluation

4.1 TREC-5 People's Daily News Corpus

The TREC-5 People's Daily news corpus is a subset of the Mandarin News Corpus announced by the Linguistic Data Consortium (LDC) in 1995. 77,733 documents in the corpus cover a variety of topics, including international and domestic news, sports and culture. The corpus uses a labeled bracketing format expressed in the style of SGML. Since the corpus is intended for evaluating information retrieval systems, each document is pre-labeled in our experiments for the purpose of text categorization. The labeling process is based on the topic information in the header field of each document, which is manually labeled in the original newspaper and retained in the corpus as *headline*. Documents in the corpus are first automatically clustered into 101 groups under a simple rule that documents in each group contain the same *headline*. With manual review of each group, groups with too general contents or containing too few articles are discarded; small groups with similar contents are merged. Six top-level categories are then generated based on the contents of the remaining 33,047 documents. Each category is finally labeled with the respective *headline* (Table 1)[1].

Table 1. Six top-level categories in TREC-5 People's Daily news corpus

Index	Category
1	政治，法律和社会 (Politics, Law and Society)
2	文学和艺术 (Literature and Arts)
3	教育, 科技和文化 (Education, Science and Culture)
4	体育 (Sports)
5	理论学术 (Theory and Academy)
6	经济 (Economics)

4.2 Experiment Paradigm

Our experiments are conducted for each top-level category of TREC-5 People's Daily news corpus. Each experiment is a binary classification case, in which we

[1] TREC-5 People's Daily news corpus mentioned here and in the following, refers to the re-constructed document set associated with category labels.

tag a chosen category as the positive category and the other five categories as the negative categories. In each experiment, a varying number (ranges from 10 to 500) of positive documents and double amount of negative documents evenly distributed across the negative categories are randomly picked out to serve as the training set. The testing set always contains 100 positive documents and 200 negative documents generated in the same way. The training set and testing set do not overlap with each other and do not containing any repetitive document in either set.

kNN experiments use the plain Euclidean distance defined by equation (4) as the similarity measure. On each pattern set containing a varying number of documents, different values of k ranging from 1 to 29 are tested and the best results are reported. Only odd k are used to ensure that a prediction can always be made by equation (5).

SVM experiments use the default built-in inductive SVM parameter set in SVM^{light}, which are described in detail on the SVM^{light} web site and elsewhere [10].

ARAM experiments use the following parameter values: learning rates $\beta_a = 0.5$, $\beta_b = 1.0$; contribution parameter $\gamma = 1.0$; vigilance parameters $\rho_a = 0.3$, $\rho_b = 1.0$, and $k = 5$ in category prediction. Using a voting strategy, 5 ARAM systems are trained using the same set of patterns in different orders of presentation. During classification, the output vectors of multiple ARAM are combined to yield a final prediction vector

$$\mathbf{B} = \frac{\sum \mathbf{B}_v}{\mathbf{V}}, \tag{18}$$

where $\mathbf{B}_v$ is the output prediction vector produced by the vth voting-ARAM and $\mathbf{V}$ is the number of voting ARAMs. The prediction vector is then thresholded at a cut off point of 0.5 to produce a binary class prediction.

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. It is a normal practice to combine *recall* and *precision* in some way so that classifiers can be compared in terms of a single rating. F_1 rating is one of the commonly used measures, which is defined as

$$F_1 = \frac{2RP}{(R+P)}. \tag{19}$$

These scores are calculated for a series of binary classification experiments, one for each category. Micro-averaged scores on the whole corpus are then produced across the experiments. With micro-averaging, the performance measures are produced by globally adding up all the documents counts across the different tests, and calculating using these summed values.

4.4 Results and Discussions

Figures 3, 4 and 5 depict the three classifier's performances on the test corpus, in terms of micro-averaged *precision*, *recall*, and F_1 measures respectively. With a sufficiently large training set (more than 200 positive patterns), the three classifiers exhibit similar performance, while ARAM appears to perform slightly better than kNN and SVM across all categories. In general, the performance produced by all three is rather good (with scores of 0.80 and above). These results suggest that as long as we have a sufficient number of clean training patterns, all the three learning approaches under evaluation can produce reasonably good generalization performance for Chinese text classification.

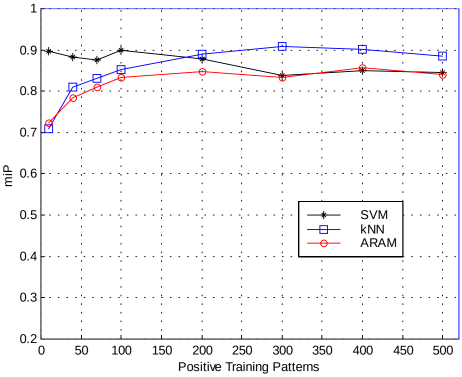

Fig. 3. Micro-averaged *precision* measures of each classifier on the document sets using 10 to 500 positive training patterns.

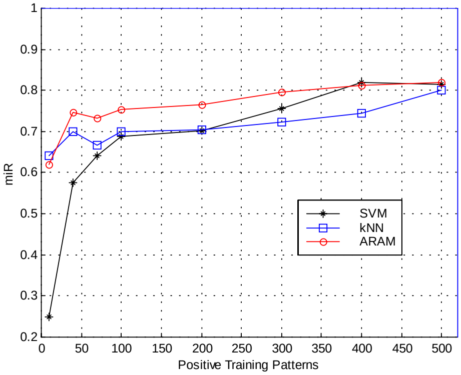

Fig. 4. Micro-averaged *recall* measures of each classifier on the document sets using 10 to 500 positive training patterns.

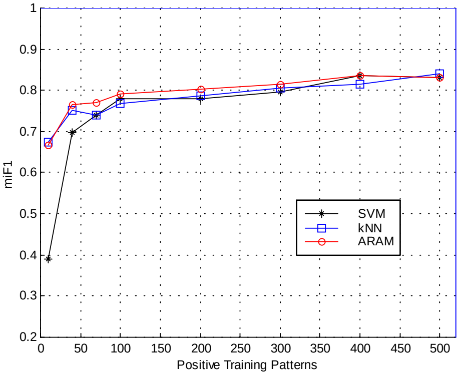

Fig. 5. Micro-averaged F_1 measures of each classifier on the document sets using 10 to 500 positive training patterns.

In our experiments, we are particularly interested in the classifier's learning ability on small training sets (that have less than 100 positive patterns). The different approaches adopted by these three classifiers in learning categorization knowledge are best seen in the light of the different learning peculiarities they exhibit on 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 satisfactory performance on small training set.

SVM identifies optimal separating hyper-plane (*OSH*) across the training data points and makes classification decisions based on 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, SVM performance is generally high in *precision* but low in *recall*. The micro-averaged F_1 measures are notably lower than those of kNN and ARAM.

ARAM generates recognition categories from the input training patterns. Each recognition category can be treated as an associative cluster of the training patterns that serves as a dynamic rule-based representation of the categorization knowledge. 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. There is thus little surprise that kNN and ARAM produce rather similar performance across small and large data sets. ARAM however is notably more scalable than kNN by its pattern abstraction capability and therefore is more suitable for handling very large data sets.

5 Conclusion

We have evaluated three machine learning methods, namely kNN, SVM, and ARAM for Chinese text categorization. Based on the empirical experiments conducted on the TREC-5 People's Daily news corpus, our main conclusions are as follows:

- Given a sufficient number of good quality training patterns, all three classifiers produce satisfactory generalization performance on unseen test documents. In addition, ARAM's performance is marginally better than kNN and SVM.
- For small training sets, kNN and ARAM produce similar performance whereas SVM generally yields lower F_1 scores. Compared with SVM, kNN and ARAM seem to be more suitable for representing categorization knowledge from sparse, noisy, and relatively small training sets.

Acknowledgements

We would like to thank our colleagues at Kent Ridge Digital Labs, especially 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

1. Belur V. Dasarathy. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Las Alamitos, California, 1991.
2. Cao Suqing, Zeng Fuhu and Cao Huanguang. A Mathematical Model for Automatic Chinese Text Categorization. Journal of the China Society for Scientific and Technical Information, 1999(1) (Chinese).
3. G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, 37:54-115, 1987.
4. G.A. Carpenter and S. Grossberg. ART 2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics, 26:4919-4930, 1987.
5. G.A. Carpenter, S. Grossberg, and D.B. Rosen. ART 2-A: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4:493-504, 1991. 5.
6. G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4:759-771, 1991.

7. Cheng Ying and Shi Jiulin, Research on the Automatic Classification: Present Situation and Prospects. Journal of the China Society for Scientific and Technical Information, 1999(1) (Chinese).
8. C. Cortes and V. Vapnik. Support vector networks. Machine learning, 20:273-297, 1995.
9. T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of the European Conference on Machine Learning, Springer, 1998.
10. T. Joachims. Making large-Scales SVM learning Pracical. Advances in Kernel Methods - Support Vector Learning. B. Scholkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.
11. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513-523, 1988.
12. A.-H. Tan. Adaptive Resonance Associative Map. Neural Networks, 8(3):437-446, 1995.
13. A.-H. Tan. Cascade ARTMAP: Integrating neural computation and symbolic knowledge processing. IEEE Transactions on Neural Networks, 8(2):237-250, 1997.
14. A.-H. Tan and Fon-Lin Lai. Text Categorization, Supervised Learning, and Domain Knowledge Integration. To appear, proceedings, KDD-2000 International Workshop on Text Mining, Boston, 20 August 2000.
15. Vladmimir N. Vapnik. The Nature of Statistical Learning Theory. Springer, New York, 1995.
16. Y. Yang and X. Liu. A re-examination of text categorization methods. In Proceedings, 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), 42-49, 1999.
17. Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In Proceedings, 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94), 1994.
18. Y. Yang. An Evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2):67-88, 1999.
19. Y. Yang, Jaime Carbonell, etc. Learning approaches for Detecting and Tracking News Events. IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval, 14(4):32-43, July/August 1999.
20. Y. Yang, Pedersen J.P. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), 412-420, 1997.
21. Zhu Lanjuan. The Theory and Experiments on Automatic Chinese Documents Classification. Journal of the China Society for Scientific and Technical Information, 1987(6) (Chinese).
22. Zou Tao, Huang Yuan and Zhang Fuyan. Technology of Information Mining on WWW. Journal of the China Society for Scientific and Technical Information, 1999(4) (Chinese).
23. Zou Tao, Wang Ji-Cheng, Huang Yuan and Zhang Fu-Yan. The Design and Implementation of an Automatic Chinese Documents Classification System. Journal for Chinese Information, 1998(2) (Chinese).