

## Towards an Effective Personalized Information Filter for P2P Based Focused Web Crawling

Fu Xiang-hua and Feng Bo-qin

Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, 710049, China

---

**Abstract:** Information access is one of the hottest topics of information society, which has become even more important since the advent of the Web, but nowadays the general Web search engines still have no ability to find correct and timely information for individuals. In this paper, we propose a Peer-to-Peer (P2P) based decentralized focused Web crawling system called PeerBridge to provide user-centered, content-sensitive and personalized information search service from Web. The PeerBridge is built on the foundation of our previous work about WebBridge, which is a focused crawling system to crawl Web according several specified topic. The most important function of PeerBridge is to identify interesting information. So we furthermore present an efficient personalized information filter in detail, which combines several component neural networks to accomplish the filtering task. Performance evaluation in the experiments showed that PeerBridge is effective to crawl relevant information for specific topics and the information filter is efficient, which precision is better than that of support vector machine, naïve bayesian and individual neural network.

**Key words:** PeerBridge, web crawling system, P2P based, artificial neural network

---

### INTRODUCTION

Information access is one of the hottest topics of information society and it has become even more important since the advent of the Web. On one side, our society depends more and more on information. Knowing the right information, at the right moment, as soon as it is available is an essential for all of us. On the other side, the amount of available information, especially on the Web, is increasing tremendously over time and we are witnessing an information overload. The process of extract relevant information from Web is still very difficult, time-consuming and in many cases practically is unfeasible, since it requires huge cognitive processing. Researchers try their best to address the challenging problem of locating correct information from Web efficiently. They have developed many different techniques, such as centralized search engines, Meta search engines, personalized web search system and topic driven search systems<sup>[1,2]</sup>. The most conventional example is the centralized search engines (CESs).

There are some problems of CESs. One major problem with CESs is that they do not facilitate human user collaboration, which has potential for greatly improving Web search quality and efficiency. Without Collaboration, user must start from scratch every time they perform a search task, even if other users have done similar or relevant searches. Another major problem with CSE is that they ignore completely the interests and preferences of users. For a same query, different users will be answered with a same list of

results. But actually, a substantial amount of personal information could be obtained during user's searching process which may be used to find suitable results for a special user.

With the emergence of successful application like Gnutella, Kazaa, Freenet, peer-to-peer (P2P) technology has received significant visibility over the past few years. P2P systems are massively distributed computing system in which peer (node) communication directly with one another to distribute task or exchange information or accomplish tasks. Also there are a few projects such as Apoidea<sup>[3]</sup>, Edutella<sup>[4]</sup>, ODISSEA<sup>[5]</sup> attempt to build a P2P based Web search or crawling system. Developing a P2P-based distributed paradigm will bring in several advantages that cannot be exploited in a centralized paradigm. Basically, they are ascribed to the fact that information has been collected, refined, and stored among users according to their interests. The active contributions of users provide multiple advantages. In effect, the creation of a special user profile allows filtering search results depending on the user interests, introducing a certain degree of personalization in search. Further, if one considers users not only as isolated individuals but also as a community then this social dimension could be exploited in order to access the expertise of people with similar interests. The social dimension of the community allows clustering users according to their interests and expertise and so focus on interesting information by reducing the domain of interest.

In this study, we present a P2P based focused Web crawling system called PeerBridge, which is developed

based on our WebBridge<sup>[4]</sup>. In PeerBridge, each node only search and store a part of the Web model that the user is interested in and the nodes interact in a peer-to-peer fashion in order to create a real distributed search engine. All users share these partial models that globally create a consistent model for the web resource that is equivalent to its centralized counterpart. One key problem we must to solve in PeerBridge is to search information that is relevant to special node. To avoid get irrelevant information, PeerBridge would try to 'guess' exactly what kind of document the user desires, basing that guess not only on the key words provided by the user, but also on a profile of the user's background and interests and on evaluations of how the system satisfied or failed to satisfy the user's requests in the past. Moreover, it would retrieve only the specific kind of documents defined by the user model component. A personalized information filter based on heterogeneous neural network ensemble classifier (HNNE)<sup>[5]</sup> is used as the content filter to model the peer's preference and filter irrelevant information. Furthermore, Topic Overlay Network Search algorithm (TONS) is developed to support complex queries on top of the existing structured network<sup>[6]</sup>.

**Design overview of PeerBridge:** PeerBridge have five main components: a content filter which makes relevance judgments on pages crawled from Web and query results searched from other nodes, a distiller which determines a measure of centrality of crawled pages to determine visit priorities, a crawler with dynamically reconfigurable priority controls which is governed by the content filter and distiller, a P2P infrastructure which supports to construct a P2P overlay network with other nodes to share and search information each others and an user interface with which user can edit training samples, select category taxonomy to training classifier and query information from the personalized data resource base and other nodes. A block diagram of the general architecture of PeerBridge is shown in Fig. 1. Here we briefly outline the basic processes of each component.

**The content filter:** The content filter is a document classifier implemented by a heterogeneous neural networks ensemble to determine whether the downloaded documents are useful. It is the central component to guarantee the quality of the search results. The representative features of the sample Web pages are extracted as inputs to train the HNNE content filter. Training's objective is to let the HNNE configure itself and adjust its weight parameters according to the training examples, to facilitate generalization beyond the training samples. In our system, the training sample includes a selected canonical taxonomy (such as Yahoo!, the Open Directory Project) and the examples specified by the user. All of the training samples define

what topics the user is interest in. We use vector model of documents to represent the user model and compute the similarity between documents and interests. A pre-trained HNNE classifier can be used to filter irrelevant information.

**The distiller:** The distiller is used to analyze the link structures of the downloaded Web pages and identify pages containing large numbers of links to relevant pages, called hubs. Since the citations signify deliberate judgment by the page author, most citations are to semantically related material. Intermittently, the system runs a topic distillation algorithm to identify hubs. The visit priorities of these pages and immediate neighbors are raised. All of the page links distilled by the distiller will be place into the search list orderly according their priorities.

**The crawler:** The function of the crawler is simple. It gets page links from the search list and then seeks and acquires the corresponding Web pages from the Web. Integrating with the distiller and the content filter, the crawler runs as a focused crawling to access only a narrow segment of the Web. We have presented a focused crawler with online-incremental adaptive learning ability in<sup>[6]</sup>. It entails a very small investment in hardware and network resources and yet achieves respectable coverage at a rapid rate. In PeerBridge, there are several crawling threads to crawl Web page synchronously during the working process.

**The P2P infrastructure:** With the P2P infrastructure, the instances of PeerBridge run on many user computers form a P2P overlay networks to share their information resource. DHT based distributed lookup and information-exchange protocols<sup>[7]</sup> are used to exchange vital information between the peers. Each peer maintains a small routing table. Given a key, these techniques guarantee the location of its value in a bounded number of hops within the network. Bloom filter<sup>[8]</sup> is used to store the list of URLs already crawled by a peer. TONS is used to support complex queries<sup>[9]</sup>. Thus Web content is managed by a distributed team of peers, each of which specializing in one or a few topics. When a query is required, each peer will not only look for it in the local host but also publish it to the overlay network. With our effective P2P search algorithm, the relevant query results in the whole overlay network will be return to the user.

**The user interface:** The user interface mainly provides a convenient operation interface to the user. User can use it to select category taxonomy, edit and judge examples, query information and display query result with rank and so on. In our prototype, it still has not been implemented completely now.

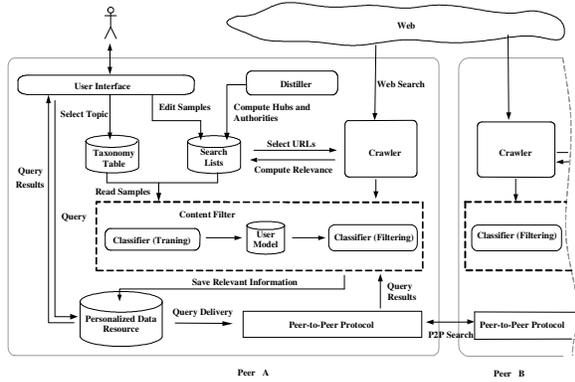


Fig. 1: The general architecture of PeerBridge

**Adaptive content filtering model:** An information filtering system can use intelligent content analysis to automatically classify documents. If a document is judged not belonging to a user specific class, it is an irrelevant document should be discarded. Such methods include k-nearest neighbor classification, linear least square fit, linear discriminant analysis and naïve Bayesian probabilistic classification<sup>[1,2,10]</sup>. However, because real-world data such as we're using tend to be noisy and are not clearly defined, linear or low-order statistical models cannot always describe them. We use artificial neural networks because they are robust enough to fit a wide range of distributions accurately and can model any high-degree exponential models. Neural networks are chosen also for computational reasons since, once trained, they operate very fast. Moreover, such a learning and adaptation process can give semantic meaning to context-dependent words.

User Model

To filter information for specific users according to their preference and interests, user model is created as an image of what users need. We define a user model as:

$$UM := (MID, FD, FT, UI, UIV) \tag{1}$$

Where, UMID is an user model identifier,  $FD := \{d_1, d_2, \dots, d_N\}$  is a set of sample documents,  $FT := \{t_1, t_2, \dots, t_M\}$  is a lexicon comprise all feature terms of FD,  $UI := \{u_1, u_2, \dots, u_T\}$  is a set of interests specified by users and  $UIV := \{UIV_1, UIV_2, \dots, UIV_T\}$  is a set of interest vectors of a special user, of which every element responds to an interest  $u_k$  ( $1 \leq k \leq T$ ) and is defined as  $UIV_k := \langle (t_1, w_{1k}), (t_2, w_{2k}), \dots, (t_M, w_{Mk}) \rangle$ , where  $w_{ik}$  is the frequency of term  $t_i$  ( $1 \leq i \leq M$ ) in  $UIV_k$ .

According vector space model (VSM), FD constitutes a term by document matrix  $X := (d_1, d_2, \dots, d_N)$ , where a column  $d_j := \langle (t_1, x_{1j}), (t_2, x_{2j}), \dots, (t_M, x_{Mj}) \rangle$  is a document vector of the document  $d_j$  and every element  $x_{ij}$  is the frequency of the term  $t_i$  in document  $d_j$ . TDFIF frequency is used, which is defined as:

$$X_{ij} = tf_{ij} \cdot \log(N/df_i) \tag{2}$$

Where,  $tf_{ij}$  is the number of the term  $t_i$  that occurs in the document  $d_j$  and  $df_i$  is the number of documents where the word  $t_i$  occurs. The similarity between document vectors is defined as:

$$Sim(d_i, d_j) = d_i^T d_j = \frac{\sum_{k=1}^N x_{ki} x_{kj}}{\sqrt{\sum_{k=1}^N x_{ki}^2 \sum_{k=1}^N x_{kj}^2}} \tag{3}$$

Equation (3) also can be used to compute the similarity between document vector and interest vector.

**Neural networks-based content filtering:** The neural networks-based adaptive content filter comprises two major processes: training and classification. During training, the filter learns from sample documents to form a knowledge base. And then it classifies incoming documents according to their content. Before training or classification, a preprocessing procedure is needed to extract from the documents words and phrases with the use of specific feature selection algorithm.

The Neural networks contain an input layer, with as many elements as there are feature terms needed to describe the documents to be classified as well as a middle layer, which organizes the training document set so that an individual processing element represents each input vector. Finally, they have an output layer also called a summation layer, which has as many processing elements there are interests of user to be recognized. Each element in this layer is combined via processing elements within the middle layer, which relate to the same class and prepare that category for output. Figure 2 illustrates the form of a content filter based on a three-layer feedforward artificial neural network.

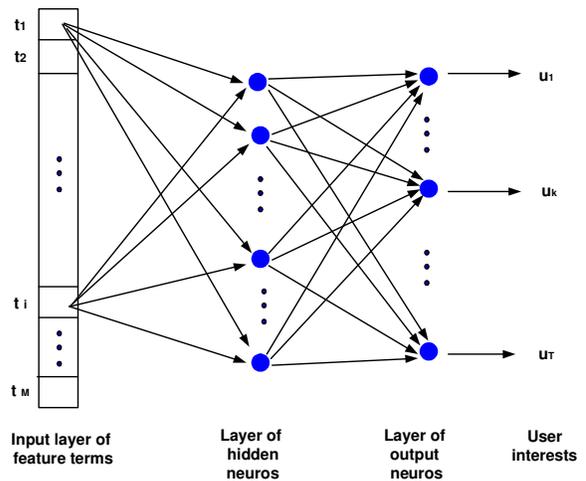


Fig. 2: Adaptive content filter based on three layer feedforward artificial neural network

In our content filter, the numerical input obtained from each document is a vector containing the frequency of appearance of terms. Owing to the possible appearance of thousands of terms, the dimension of the vectors can be reduced by singular value decomposition (SVD), Principal Component Analysis, Information Entropy Loss and word frequency threshold<sup>[10]</sup>, etc.

**Heterogeneous neural networks ensemble classifier:** Neural Network ensemble (NNE) is a learning paradigm where many neural networks are jointly used to solve a problem<sup>[11]</sup>. It originates from Hansen and Salamon's work<sup>[12]</sup>, which shows that the generalization performance of a neural network system can be significantly improved through combining several individual networks on the same task. The creation of a neural network ensemble is constructed in two steps, the first being the judicious creation of the individual ensemble members and the second their appropriate combination to produce the ensemble output.

There has been much work in training NN ensembles<sup>[11-16]</sup>. However, all these methods are used to change weights in an ensemble. The structure of the ensemble, e.g., the number of NNs in the ensemble and the structure of individual NNs, e.g., the number of hidden nodes, are all designed manually and fixed during the training process. While manual design of NNs and ensembles might be appropriate for problems where rich prior knowledge and an experienced NN expert exist, it often involves a tedious trial-and-error process for many real-world problems because rich prior knowledge and experience human experts are hard to get in practice.

In<sup>[17]</sup>, we propose a new method to construct heterogeneous neural network ensemble (HNNE) with negative correlation. It combines ensemble's architecture design with cooperative training of individual NNs in an ensemble. It determines automatically not only the number of NNs in an ensemble, but also the number of hidden nodes in individual NNs. It uses incremental training based on negative correlation learning<sup>[10,13]</sup> in training individual NNs. The main advantage of negative learning is that it encourages different individual NNs to learn different aspects of the training data so that the ensemble can learn the whole training data better. It does not require any manual division of the training data to produce different training sets for different individual NNs in an ensemble.

#### Theory Foundation of Neural Network Ensemble

Suppose a data set  $D := \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , where  $x_p$  is the input sample and  $y_p$  is the output result ( $1 \leq p \leq N$ ). An ensemble comprising  $H$  component neural network and every component network is trained to approximate a function  $f: \mathbb{R}^N \rightarrow \mathbb{C}$  where  $\mathbb{C}$  is the set of class labels. Suppose the weight of the  $i$ th component network is  $w_i$  ( $1 \leq i \leq H$ ) and all the weights satisfies

$w_i \geq 0, \sum_{i=1}^H w_i = 1$ . When the input sample is  $x_p$ , the output of the  $i$ th component network is  $f_i(x_p)$  and the output of the ensemble is:  $f(x_p) = \sum_{j=1}^H w_j f_j(x_p)$ . Thus the generalization error of the ensemble in the whole data set is:

$$E = \sum_{p=1}^N (y_p - f(x_p))^2 \quad (4)$$

The generalization error of the  $i$ th component network in the whole data set is:

$$E_i = \sum_{p=1}^N (y_p - f_i(x_p))^2 \quad (5)$$

The weighted generalization of the ensemble is:

$$\bar{E} = \sum_{i=1}^H w_i E_i \quad (6)$$

The diversity of the ensemble is:  $\bar{A} = \sum_{i=1}^H w_i \sum_{p=1}^N (f_i(x_p) - f(x_p))^2$ . So the generalization of the ensemble satisfies:

$$E = \bar{E} - \bar{A} \quad (7)$$

Combining the outputs is clearly only relevant when they disagree on some or several of the inputs. This insight was formalized by<sup>[15]</sup>, who showed that squared error of the ensemble when predicting a single target is equal to the average squared error of the individual networks, minus the diversity define as the variance of the individual network output. Thus, to reduce the ensemble error, one tries to increase the diversity without increasing the individual network errors too much.

**Construct neural network ensemble with negative correlation:** Because all the component networks are trained with the samples of the same data set  $D$  to approximate the same function, the output of the component networks are high correlated potentially leading to severe colinearity and reducing the robustness of the ensemble network<sup>[16]</sup>. Define the correlation of the  $i$ th component network with the others is:

$$C_i = \sum_{p=1}^N (f_i(x_p) - f(x_p)) \sum_{j=1, j \neq i}^N (f_j(x_p) - f(x_p)) \quad (8)$$

To mitigate this potential colinearity problem, Equation (5) is modified by adding a decorrelation

penalty to it. The new error function for an individual network  $i$  is:

$$E_i = \sum_{i=p}^N (y_p - f_i(x_p))^2 + \lambda C_i \quad (9)$$

Where  $\lambda$  ( $\lambda \geq 0$ ) is an adjustable parameter, which is used to adjust the strength of the penalty. So the individual networks attempt to not only minimize the error between the target and their output, but also to decorrelate their error with those from previously trained networks.

When the simple average weight is used to combine the component networks, namely  $w_i = 1/H$ , then Equation (9) can be modified as:

$$E_i = \sum_{p=1}^N \left( \frac{1}{2} (y_p - f_i(x_p))^2 - \lambda (f(x_p) - f_i(x_p))^2 \right) \quad (10)$$

The average value of all the component error is:

$$E_{sum} = \frac{1}{H} \sum_{i=1}^H \sum_{p=1}^N \left( \frac{1}{2} (y_p - f_i(x_p))^2 - \lambda (f(x_p) - f_i(x_p))^2 \right) \quad (11)$$

The partial derivative of Equation (10), with respect to the output of network  $i$  on the  $p$ th training sample, is

$$E_{sum} = \frac{1}{H} \sum_{i=1}^H \sum_{p=1}^N \left( \frac{1}{2} (y_p - f_i(x_p))^2 - \lambda (f(x_p) - f_i(x_p))^2 \right) \quad (12)$$

When  $\lambda = 1/2$ ,  $E = E_{sum}$ , so we get

$$\frac{\partial E_i(x_p)}{\partial f_i(x_p)} \propto \frac{\partial E(x_p)}{\partial f(x_p)} \quad (13)$$

According Equation (13), the minimization of the empirical risk function of the ensemble is achieved by minimizing the error functions of the individual networks. From this view, negative correlation learning provides a novel way to decompose the learning task of the ensemble into a number of subtasks for different individual networks.

In literature<sup>[17]</sup>, we provide a new method to incremental construct heterogeneous neural network ensemble with negative correlation. The new method includes two processes: at first the Cascor<sup>[18]</sup> is modified to construct optimal individual heterogeneous networks with negative correlation learning, during this process, what are consider is: (1) constructing all the individual networks with the same data set sequent; (2) Equation (10), (12) are used to guarantee all of the individual networks are negative correlation; and then the optimal individual heterogeneous networks are

selected to combine a heterogeneous neural network ensemble.

**Performance evaluation:** As one of the most important work of our adaptive content filtering, we have implemented a P2P-based information search and discovery system called PeerBridge for user-centered timely information search and extract from Web and other peers incrementally. The infrastructure tools of the PeerBridge include Full-text Indexing and Retrieval Engine, Metadata Manager, User Mode Manager, HNNE based Content Filter, Web Crawler, P2P Protocol, P2P Search Engine. The PeerBridge currently built on Windows platform. Figure 3 shows the snapshots of the WebBridge, and Figure 4 shows a snapshot of the PeerBridge.

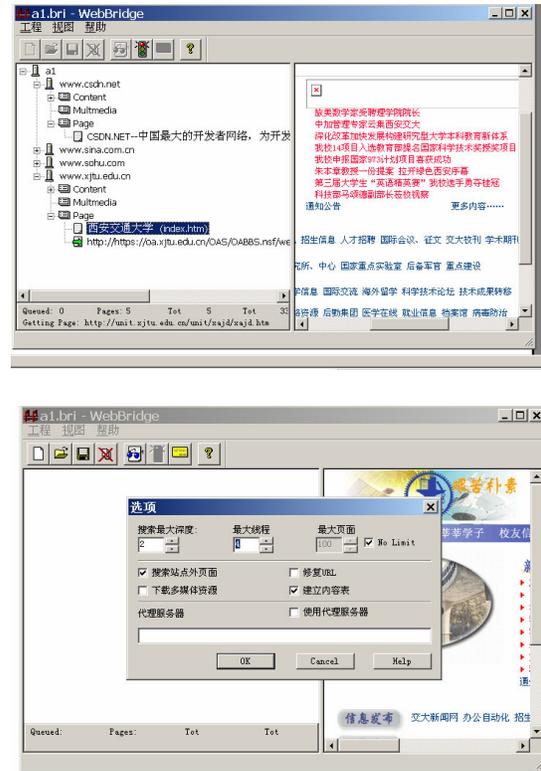


Fig. 3: A snapshot of the WebBridge

Based on PeerBridge we have evaluated the filtering performance of the Chinese Web pages content filter with variant number of component neural network in Web search task. We find the heterogeneous neural network ensemble classifier is efficient and feasible for adaptive information filter in distributed heterogeneous neural network environment. In our experiments six different heterogeneous neural network ensembles are tested, the number of component neural network of which are respectively 1,5,10,15,20,25 and are notated as NNE1, NNE5, NNE10, NNE15, NNE20 and NNE25. With above different content filters trained by the same

interest documents, PeerBridge search relevant web documents from Yahoo China (<http://cn.yahoo.com>). The evaluation results are shown in Fig. 5.



Fig. 4: A snapshot of PeerBridge

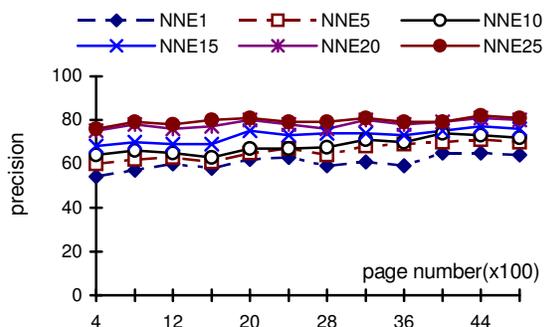


Fig. 5: Precision of content filter with different number of component neural networks

Table 1: The document number of the training set and test set in six categories

	Earn	acq	money-fx	crude	grain	trade
Training set	2709	1488	460	349	394	337
Test set	1014	630	133	160	130	106

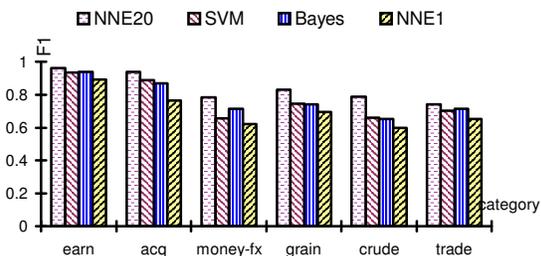


Fig. 6: Comparison with NNE20, SVM, Bayes, NNE1 in Reuters-21578 collection

The measurement  $F_1 = \frac{2R_p R_r}{R_p + R_r}$  is used to evaluate the performance of the classifiers, where if  $a$  is the number of documents correctly assigned to this

category,  $b$  is the number of documents incorrectly assigned to this category and  $c$  is the number of documents incorrectly rejected from this category, then precision  $R_p = \frac{a}{a+b}$  and recall  $R_r = \frac{a}{a+c}$ . The experiment results are shown in Fig. 6.

Figure 5 manifested combining many component neural networks improved the content filtering precision of the Web search system. It is also obviously that increasing the number of the component neural network can improve the precision largely at the beginning, but when the number is sufficiently large, the improvement became small.

Figure 6 showed that the heterogeneous neural network ensemble based classification algorithm was better than other classification algorithm. Once trained, neural network ensemble operates very fast. Moreover, the assumptions on the problem's distribution model of neural network classifier are much less than that of Naïve Bayes classifier, so it is has less independence on the problem and they are robust enough to fit a wide range of distributions accurately and can model any high-degree exponential models.

## CONCLUSION

Information access is one of the most important requirements of everybody in nowadays. Facing to the information overload on the Web and CESs' problem, we provide a P2P based, content-sensitive, interest-related and personalized web crawling system. A new content filter based on HNNE classifier base is proposed to guarantee each node only crawling personalized relevant information. Performance evaluation in the experiments showed that PeerBridge is effective to crawl relevant information for specific topics. To compare with other classifiers such as SVM, naïve bayesian and individual artificial neural network, the experiment results showed that HNNE classifier is very efficient and feasible. In the future we will take into account those issues in PeerBridge such as efficiently information search, fault tolerance and access control etc.

## REFERENCES

1. Arasu, A., J. Cho, H. Garcia-Molina and S. Raghavan, 2001. Searching the Web. ACM Trans. Internet Technol., 1: 2-43.
2. Baeza-Yates, R., 2003. Information retrieval in the Web: Beyond current search engines. Intl. J. Approx. Reasoning, 34: 97-104.
3. Singh, A., M. Srivatsa, L. Liu and T. Miller, 2003. Apoidea: A decentralized peer-to-peer architecture for crawling the World Wide Web. SIGIR 2003 Workshop on Distributed Information Retrieval.

4. Nejdl, W., B. Wolf, C. Qu, S. Decker, M. Sinterk, A. Naeve, M. Nilsson, M. Palmer and T. Risch, 2003. Edutella: A p2p networking infrastructure based on RDF. Proc. 12th Intl. Conf. World Wide Web, Hawaii, USA, pp: 604-15.
5. Suel, T., C. Mathur, J.W. Wu and J. Zhang, 2003. ODISSEA: A peer-to-peer architecture for scalable web search and information retrieval. In 6th Intl. Workshop on the Web and Databases.
6. Fu, X.H., B.Q. Feng, Z.F. Ma and M. He, 2004. Focused crawling method with online-incremental adaptive learning. J. Xi'An JiaoTong Univ., 38: 599-602.
7. Stoica, I., R. Morris, D. Karger, M.F. Kaashoek and H. Balakrishnan, 2001. Chord: A scalable peer-to-peer lookup service for internet application. Proc. SIGCOMM Ann. Conf. Data Communication.
8. Bloom, B., 1970. Space/time trade-off in hash coding with allowable errors. Commun. ACM, 12: 422-426.
9. Fu, X.H. and B.Q. Feng, 2005. Distributed information search based on topic segments in structured peer-to-peer networks. J. Xi'An JiaoTong Univ. (Accepted)
10. Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comp. Surveys, 34: 1-47.
11. Zhou, Z.H., J.X. Wu and W. Tang, 2002. Ensembling neural networks: Many could be better than all. Artificial Intelligence, 137: 239-263.
12. Hansen, L.K. and P. Salamon, 1990. Neural network ensembles. IEEE Trans. Pattern Analysis and Machine Intelligence, 12: 993-1001.
13. Liu, Y. and X. Yao, 2000. Evolutionary ensembles with negative correlation learning. IEEE Trans. Evolution. Comp., 4: 380-387.
14. Dietterich, T., 2000. Ensemble methods in machine learning. First Intl. Workshop on Multiple Classifier Systems, pp: 1-15.
15. Krogh, A. and J. Vedelsby, 1995. Neural network ensembles cross validation and active learning. Advances in Neural Information Processing Systems, San Mateo, CA: Morgan Kaufman.
16. Rosen, B.E., 1996. Ensemble learning using decorrelated neural networks. Connection Sci., 8: 373-378.
17. Fu, X.H., B.Q. Feng, Z.F. Ma and M. He, 2004. Method of incremental construction of heterogeneous neural network ensemble with negative correlation. J. Xi'An JiaoTong Univ., 38: 796-799.
18. Fahlman, S.E. and C. Lebiere, 1990. The Cascade-correlation learning architecture. Advances in Neural Information Processing Systems, 2. Los Altos, USA: Morgan Kaufmann Publishers, pp: 524-532.
19. Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Proc. ECML-98, 10th Eur. Conf. Machine Learning, pp: 137-142.