Algorithm for Layered Sorting and Merging of P2P Information Retrieval

Yi-Hong Tan[†1, 2] and Ya-Ping Lin[††],

[†1]Department of Information and Computing Science, Changsha University, Changsha Hunan 410003,China
[†2]School of Computer and Communication, Hunan University, Changsha, 410073, China
[††]School of Software, Hunan University, Changsha, 410073, China

Summary
Due to the intrinsic characteristics, P2P information retrieval has faced quite a few challenges, and it is one of the urgent problems how to sort and merge the retrieval results from multiple nodes. For this reason, we propose a novel algorithm for layered sorting and merging of P2P information retrieval. In the proposed approach, the query submitting node submits the query string to the nodes that more probably acquires retrieval results. These nodes accept query request, sort the retrieval results in the descending order of similarities with traditional sorting method and select top-k documents to send to the query requesting node, and then the query submitting node execute sorting and merging. The final experimental results suggest that our proposed approach is feasible, correct and valid.

Key words:
Information Retrieval, Sorting, Merging, Peer-to-Peer

1. Introduction

With the development of P2P technology, great breakthrough has been achieved in the P2P information retrieval technology and many searching mechanisms have been brought forward. However, due to the intrinsic characteristics of P2P such as the dynamics, the decentralization of document storage and the node self-organization characteristic etc., P2P information retrieval has faced quite a few challenges, of which, it is one of the urgent problems how to sort and merge the retrieval results from multiple nodes.

There have been many sorting algorithms in the centralized information retrieval and traditional distributed information retrieval systems, but all these algorithms can not be directly applied in the P2P information retrieval, since the data of both centralized information retrieval and traditional distributed information retrieval are centralized stored in the specified database and it is easy to acquire the global information of the document, furthermore, specified query server is included, so it is easy to sort the retrieval results. However, the data are decentralized stored at each node in P2P information retrieval, the document information stored in the whole network can not be known through query of single node, and the query is executed by each node, it is a difficulty how to collectively sort the returned results from each query node.

At present, some scholars have made researches on the relevant questions and achieved some progress. However, the current system only takes into account to return high-quality results, even if the retrieval results have more relativity, it does not take into account the repetition problem of these results. But in P2P information retrieval, various methods are often adopted to copy the data of the node in order to improve the searching efficiency; furthermore, each node itself collects data same with the other nodes, so there probably are a large amount of repeating data at each node. If the submitted retrieval results include these repeating data, the retrieval quality will apparently be affected, because the users always expect to download more different documents from fewer and faster nodes.

The method of layered sorting and weeding out repeating data is adopted in this paper to select high-quality and non-repeating retrieval results as the returned results to improve the retrieval efficiency and quality. The basic steps of this method is shown as follows: Firstly TOP-K query results return from the query node, then the query submitting node sorts all query results according to the relativity and chooses the best one as the retrieval results included at the node with high coverage ratio and few response delay time. If the retrieval result includes repeating data, the same data in other nodes will be deleted to realize to submit high-quality and non-repeating retrieval results to the users.

2. Related Work

Many sorting methods have been brought forward in the traditional distributed information retrieval. Such as, CORI adopts the framework of decision theory [1-2] brings forward GIOSS method and adopts statistical model method [3]. In principle, these methods can be applied in P2P system, but the core concept and current mechanism of

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

P2P information retrieval system are apparently different from the traditional distributed information retrieval system mainly representing in different node number, different document and statistical information management method and different retrieval processing mode. And these methods have many fatal defects, for example, these methods are mainly based on statistical model and can not be expanded to large-scale, high-dynamics node system. Furthermore, these methods do not take the repeating data as a fatal problem. Therefore, the achievements obtained in the traditional distributed information retrieval field can not be directly applied to solve the problems faced by P2P networks.

At present, some initial progresses have been achieved in the research on the sorting of P2P information retrieval system, for example, Planet P. [4] system adopts resource location based on Bloom filter and query result sorting mechanism. Each node adopts Bloom filter to induce the index of local file and distribute it in the network with gossiping algorithm. Each node queries the shared file according to the Bloom filter collected by it and these retrieved answers are permuted by the query submitting node. Psearch [5] is the structural P2 information retrieval system developed on the basis of Can [6], which has put forward a sorting strategy with determinacy. Through the mapping mechanism provided by Can, the index items including the specified key words are released to the specified node. When searching the document, the query is forwarded to theses nodes including key words, the latter returns the relevant index items to the query node, which computes and queries the similarity with the document and sorts the results [7]. A fully distributed sorting and merging strategy of the query results is brought forward based on the characteristics of P2P information retrieval system, every query node executes sorting processing through acquiring approximate global information [8]. A layered top-k query algorithm based on histogram is brought forward. Firstly, the distributed top-k query is realized by adopting layered method and the merging and sorting of the results are decentralized to all nodes in P2P network. The resources in the network are fully used. Secondly, the histogram is built for the node according to the returned results of the node and the possible upper limit of the fraction of the node is estimated with the histogram to select the node. The query efficiency is enhanced. The experiments have proved that top-k query increases the query effect and the histogram improves the query efficiency. However, the algorithm only adapts to the sort in pure P2P networks and does not adapt to super-node network and semantic overlay network. These methods adopt some effective sorting algorithms but none of them takes into account the repeatability problem of the retrieval.

At present, the evaluation researches on the repetition rate and novelty ratio are only involved in a few works [9]. The centralized information filtering system has mentioned the redundancy check processing is mentioned in, but does not involve how to expand it to the high-distributed networks [10-11]. A statistical method of evaluating coverage ratio and repetition rate is brought forward, which extracts the characteristics of the data set with query classification and exploring technology. With very large amount of computation, this method does not adapt to P2P query routing setting, which requires the evaluation must be on-line query and the response time is very short [12]. The evaluation method on the repeatability of P2P data set is brought forward. The bloom filter and simple decision model are adopted to execute query routing. However, since the bloom filter is adopted to represent the document information of the node, complicated processing will be need for the joining and leaving of node as well as adding and deletion of the document, which is not described in detail in the paper. Furthermore, the code length of the bloom filter is fixed, which does not adapt to the heterogeneous of the node [10]. A method of evaluating the coverage ratio and repeatability of the results with association rules mining technology is brought forward to sort the query results. The fatal shortcoming of this method is that the data set for association rules mining must be fixed in advance, i.e. it is difficult to realize to acquire the correlation set between the query words and document through pre-query in P2P networks.

Also, several merging strategies have been proposed in order to obtain a single list of relevant documents [13]. However, a large decrease of precision is generated in the process (depending on the collection, between 20-40%).

(1) Traditional merging strategies.

1) Round-robin fashion. The documents are interleaved according to ranking obtained for each document by means of monolingual information retrieval processing. In this case, the hypothesis is the homogeneous distribution of relevant documents across the collections [14].

2) Raw score. This method produces a final list sorted by document score computed independently for each monolingual collection. This method works well both when each collection is searched by the same or a very similar search engine, and when query terms are distributed homogeneously over all the monolingual collections [15].

3) Raw score normalized. An attempt to make document scores comparable is by normalizing in some way the document score reached by each document [16].

(2) Machine learning strategies.

1) Logistic regression. This approach is a statistical methodology for predicting the probability of a binary outcome variable according to a set of independent explanatory variables [17-18].

2) Bayesian logistic regression. This approach is a special case of logistic regression. It is a very efficient method during fitting and at the time of prediction [19].

3) Support vector machine. It is a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalization theory and exploiting optimization theory [20].

The sorting steps in this work are basically similar to the abovementioned sorting mechanism. The query submitting node submits the query to the node which more probably acquires the query results according to the specified searching mechanism and the query node return the top-k query results and relevant auxiliary information to the query submitting node. What is different is that it ensures the high quality and novelty of retrieval results to select repeating data based on the principle of node priority.

3. Traditional Sorting Approach

3.1 Extraction of Text Characteristic

The characteristic representation of text is to preprocess the text and extract the metadata representing its characteristic to save it in structural form as the intermediate representation form of the document. The Vector Space Model (VSM) [21] is one of representation methods of text characteristics applied more in recent years. In this model, the information resource (document) of node is represented with VSM, and each document d represents a normalized characteristic vector,

$$V(d) = (t_1, w_1(d); \ldots; t_j, w_j(d); \ldots; t_m, w_m(d)) \quad (1)$$

Where, t_j is the characteristic word, $w_j(d)$ is weight value of t_j in d, which is generally defined as function of occurrence frequency $tf_j(d)$ of t_j in d, for example,

$$w_j(d) = \Psi(tf_j(d)) \quad (2)$$

$$\Psi = tf_i(d) \times \log\left[\frac{N}{n_j}\right] \quad (3)$$

Where, N is the number of all documents, n_j is the number of document including characteristic word t_j. Similar to the vector space model of document, the submitted query string is represented as:

$$V(q) = (t_1, w_1(q); \ldots; t_j, w_j(q); \ldots; t_n, w_n(q)) \quad (4)$$

Where, t_j is the characteristic word, $w_j(q)$ is the weight value of t_j in q. The concrete weight value can be set by the users themselves.

3.2 Similarity Computation

The similarity computation between query and document is completed through the cosine distance computation as shown in Formula (1):

$$\mathrm{sim}(q, d_i) = \frac{\sum_{j=1}^{n} w_j(q) \times w_j(d_i)_j}{\sqrt{\sum_{j=1}^{n} (w_j(q))^2 \times \sum_{j=1}^{n} (w_j(d_i))^2}} \quad (5)$$

The similarities between all documents and the submitted query strings are computed with Formula (1) and are sorted in descending order.

4. Layered Sorting and Merging

The basic steps of this method are shown as follows: The query submitting node submits the query string to the nodes that more probably acquires retrieval results. These nodes accept query request, sort the retrieval results in the descending order of similarities with traditional sorting method and select top-k documents to send to the query requesting node, and then the query submitting node execute sorting and merging. The query node sorting method is similar to [22] and is not discussed any more here. Following that, the sorting and merging mechanism of query submitting nodes will be mainly discussed.

4.1 Query node sorting

In order to facilitate query sorting and merging, the query submitting node needs to acquire the information of the node returning retrieval results. Node

$$p_i = (d_{i,1}, d_{i,2}, \ldots d_{i,h}; \Delta t) \quad (6)$$

Representing p_i includes k retrieval results; the response delay time is Δt, where d_{ij} is the j^{th} document in node p_i, which is represented with characteristic vector as,

$$V(d_{i,j}) = (t_1, freq_1; \ldots; t_2, freq_2; \ldots; t_m, freq_m) \quad (7)$$

Where, t_m is the characteristic word of d_{ij} and $freq_m$ is occurrence times of tm in d_{ij}.

The query submitting node receives the returned retrieval results from all query nodes and takes all returned retrieval results as a result set, which is recorded as D. The relativity of the document and query string is computed with Formula (5). In the formula, $tf_i(d)$ represents the occurrence frequency of characteristic word t in the whole retrieval result set D, N represents the total document number included in D, n_j represents the number of document including characteristic word t. $sim(q, d_{ij})$ is computed for all documents and the documents are sorted in descending order. In order to reduce the amount of computation and improve the retrieval efficiency, we only select the documents with similarity exceeding some one set threshold value as the candidate set to be submitted to the users.

4.2 Query node merging

Many repeating data (redundant data) are probably included in D', so the redundant data must be weeded out, otherwise the retrieval results submitted to the users will include some high-quality data but with many repetitions, which will affect the retrieval quality. For the redundant data, it needs to determine the retrieval results included in which node shall be selected and the other repeating results will be deleted, i.e. if there are multiple nodes including same one document, it is a difficulty to select the document in which node and make the users be able to acquire information easily and rapidly. The strategy we adopt is based on the principle of node priority. This strategy is to preferentially select the node including more number of relevant documents with high quality and few response delay time after comprehensive consideration of the quality of relevant document included in these nodes and response delay time.

Definition 1 (coverage ratio of node): The quantity and quality of the documents related to query string included in the node. The computation formula is:

$$\text{coverage}_q(p_i) = \frac{\sum_{k-1}^{m} sim(q, d_{i,k})}{\sum_{j=1}^{n}\sum_{k=1}^{m} sim(q, d_{j,k})} \quad (8)$$

Where, m is the document quantity included in node p_i, n is the node quantity includes the documents in D, $sim(q, d_{i,k})$ is computed according to Formula (5). The larger the value is, it shows the coverage ratio of the node is higher.

Definition 2 (repeating document): If the characteristic vectors of document d_i and d_j are same, which is recorded as $V(d_i) = V(d_j)$, d_i and d_j are called as repeating documents.

The coverage ratios of all nodes in D are computed and are sorted in descending order. The documents respectively included in all nodes still are sorted in the descending order of similarity. Thus, sorting is made firstly in the descending order of the coverage ratios of nodes in D, and then is made in the descending order of similarity in each node. The merging steps are: Firstly, all documents in the node with highest coverage ratio are extracted as submitted result set, which is recorded as D', and then next node is extract in turn to determine whether the document in the node is the repeating data. If yes, the determination shall be made whether the repeating data of this document has been in D', if it is not included, this document will be added to D', otherwise, it will not be added. The purpose of weeding out redundant data is realized with this method.

4.3 Algorithm for sorting and merging

The variety of computational environment is an important challenge to the sorting and merging of P2P information retrieval. There are three solutions for solving this problem. (1) The system provides one or many metadata for all the nodes. Obviously, this mode is not fit to the P2P environment because of the huge load between server and each node. (2) The interrelated metadata is maintained by each node. Because the huge spending of the access space and coherence of metadata maintained, this solution is not satisfied too. The third solution is to design a completely distributed strategy. That is, each node can only obtain the system global value of relative metadata for the local files. There are three advantages for this approach. (1) Each node only needs to access a few loading. (2) The global value of relative metadata is maintained in the local computer, the sorting computation of each query can be executed in the local computer, and it can largely improve the response speed. (3) This strategy can guarantee the global consistency and timely updating of metadata. Taking one with another, the third solution is appropriate to the sorting and merging of P2P information retrieval.

Suppose each node p has an exclusive identifier PID; k_i denotes a local index item of one node, and n_i^p denotes the number of documents, which includes the key word k_i in the node p; N^p denotes the total number of local documents.

Define a mapping R, which mapped the index item k_i to the corresponding node X, and X denotes the target node of key word k_i. We use *Target* to denote the resultant node X. *Uploaders* denotes an upload node. Each node p can map every index item k_i to its target node with following form

$$\langle PID(p), n_i^p, N^p, TimeStamp \rangle .$$

Based on the approximate global value, when obtain a query command, each node can computer the similar index of query file and local file.

Algorithm 1: Algorithm for Computing the Approximate global value n_i and N for each index item k_i

Inputs: R, PID, n_i^p, N^p

Outputs: n_i, N

For peer P (Initial Peer):

```
While (True) {
If (bExit) //user want to exit
    exit(0);
If (Isinitiated()|FileModify()|FileAdd()|FileDelete())
    { GetStatistics(n_i^p N^p);
    PID= R(k_i);
    SendToTarget( PID,< PID, PID(P), n_i^p, N^p >);
    GetFromTarget( PID, n_i, N); }
If (RunTime()>24 hours) {
    GetFromTarget( PID, n_i, N);
    SetRunTimeToZero(); }
sleep(InternalTime); }
```

For Peer X (Target Peer)

```
While (True) {
If (bExit) //user want to exit
    exit(0);
If(!QueueEmpty()) {
    ReceiveData( PID, n_i^p, N^p);
    n_i = Σ_p n_i^p + Σ_p (n_it^p − n_i(t−1)^p);
    N = Σ_p N^p + Σ_p (N_t^p − n_(t−1)^p);
    SendBackData( PID, n_i, N); }
sleep(InternalTime); }
```

Algorithm 2: Algorithm for sorting and merging

Input: $D = \{D_1, D_2, \cdots, D_n\}$ is the returned results set from all nodes, D_n is the returned result set from the nth node and n is the node quantity returning the result set.

Output: D' is the result set submitted to the users.

Please notes:

(1) $Sortdoc(D)$ is to sort the computed values of D according to Formula (5) in reverse order;

(2) $delete\min doc(D)$ is to delete the document with document relativity of less than one set threshold value ε in D;

(3) $compute\_coverage(D_i, D)$ is to separately compute the coverage ratio of each node including the document in D according to the computation formula of coverage ratio of node;

(4) $sortpeer(P)$ is to sort the nodes in reverse order according to W_i value;

(5) $sortinvert(D)$ is to re-sort the documents in D, firstly the sorting is made in reverse order of the nodes, then the documents in the node are sorted in descending order;

(6) $redundancy(d_{ij})$ is to determine whether d_{ij} has repeating data according to definition of repeating document, if yes, it will return true, otherwise, it will return false;

(7) $isExist(d_{ij})$ is to determine whether there is repeating data of dij included in D.

1) D=*sortdoc*(D)
2) D=*deletemindoc*(D);
3) For all nodes P do
   Coverage(p_i)=*compute_coverage*(D_i, D)
   W_i=α*Coverage*(p_i)+$\beta\triangle t$
4) P=*sortpeer*(P)
5) D=*sortinvert*(D)
6) For D do
   $D' = D' \cup D_1$
7) While i<=n do
   While j<=$|D_i|$ do
   If not is *Redundancy*(d_{ij})
   Then $D' = D' \cup \{d_{ij}\}$
   Else if not *isExist*(d_{ij})
   Then $D' = D' \cup \{d_{ij}\}$
   End
   End
8) Return D'.

5. Experiment

5.1 Experimental Surroundings

The purpose of this experiment is to measure the quality of retrieval results based on layered sorting and merging and

analyze the influence of threshold value θ against the network load and searching efficiency. In the experiment, network topology generation module adopts GT-ITM [9] software package of American University of Georgia, the reason is that GT-ITM can generate similar mixed networking mode of multiple LANs (local area network) and MANs (metropolitan area network) existing in the actual network as the famous network topology generation system with open source code. The experiment is completed on a PC computer with configuration of CPU P3.0 GHz, memory of 1GB and operating system of Windows XP. The document set used for the experiment adopts the testing document set CACM used in SMART [10] system, which includes about 30,000 computer document summaries and about 200 queries. The document set and query are distributed stochastically at each node in the network. 10,000 nodes are generated in the experiment and each node joins in the network and submits query stochastically.

5.2 Performance Indexes

In this paper, three performance indexes are defined to evaluate our proposed approach. These three performance indexes are listed as follows.

(1) Recall, its computational formula is,

$$\mathrm{Re}call = \frac{\sum_{\mathrm{Re}trieved} Answer}{\sum_{CurrentAvailable} Answer} \quad (9)$$

Where, $\sum_{\mathrm{Re}trieved} Answer$ denotes the amount of resultant answer obtained by the algorithm, and $\sum_{CurrentAvailable} Answer$ denotes all the available answer in the current system.

(2) Precision, its computational formula is,

$$\mathrm{Pr}ecision = \frac{\sum_{Qualified} Answer}{\sum_{\mathrm{Re}trived} Answer} \quad (10)$$

Where, $\sum_{\mathrm{Re}trieved} Answer$ denotes the amount of resultant answer obtained by the algorithm, and $\sum_{Qualified} Answer$ denotes the number of correct results among the resultant results.

(3) Correctness, its computational formula is,

$$Correctness = \frac{\bigcup_{Quieried\_peer} \{a_1, a_2, \cdots, a_M\} \cap \{TopK_{GlobalAnswers}\}}{\{TopK_{GlobalAnswers}\}} \quad (11)$$

Where, suppose it will obtain M best answers after each query, and its set is $\{a_1, a_2, \cdots, a_M\}$; then the answer set of all queries is $\bigcup_{Quieried\_peer} \{a_1, a_2, \cdots, a_M\}$; also, support all the sharing documents is maintained by a node, then to each same query, the set of best answers obtained by this node is $\{GlobalAnswers\}$, its best K answer is $\{TopK_{GlobalAnswers}\}$.

5.3 Experimental Results

We executed 50 queries randomly in an experiment. The average query results of these 50 queries will be taken as the final results of one experiment. In order to avoid the randomness of experiments, we repeat each experiment for 20 times. The experimental results are listed as Table 1.

Table 1. The Final Experimental Results

Runs	Recall	Precision	Correctness
1	94.05	73.04	85.15
2	80.39	78.35	70.75
3	87.53	86.64	82.39
4	85.23	72.18	77.08
5	92.93	74.50	85.04
6	90.48	75.65	79.25
7	84.67	75.58	82.89
8	76.35	82.87	77.95
9	91.61	76.90	75.76
10	84.45	75.58	73.74
11	87.69	72.27	73.80
12	91.05	85.44	82.41
13	93.51	80.01	75.73
14	90.03	88.77	79.93
15	79.35	80.39	73.06
16	83.71	79.54	82.68
17	93.77	87.23	77.06
18	93.42	81.45	85.54
19	83.80	75.65	85.42
20	92.98	84.10	80.85
Average	87.85	79.31	79.32

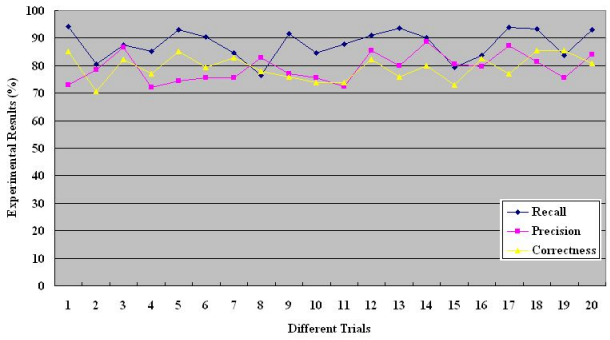

Fig. 1. The final experimental results

The final experimental results of Table 1 can be displayed as Fig. 1. From this figure, we can see that the experimental results of recall, precision and correctness are all satisfied. The experimental results suggest that our proposed approach is feasible, correct and valid.

6. Conclusions

The contribution of this paper can be summarized as follows. It proposed a novel algorithm for layered sorting and merging of P2P information retrieval. The final experimental results suggest that our proposed approach is feasible, correct and valid.

Reference

[1] N. Fuhr. A decision-theoretic approach to database selection in networked IR. ACM Transactions on Information Systems, 17(3): 229-249, 1999.

[2] L. Gravano, H. Garcia-Molina, and A. Tomasic. Gloss: text-source discovery over the internet. ACM Transactions on Database System, 24(2): 229-264, 1999.

[3] L. Si, R. Jin, J. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In Proceedings of the eleventh international conference on Information and knowledge management, ACM Press, 391-397, 2002.

[4] Cuenca-Acuna FM, Nguyen TD. Text-Based content search and retrieval in ad hoc P2P communities. Department of Computer Science, Rutgers University: Technical Report DCS-TR-483, 2002.

[5] Tang Chun-Qiang, Xu Zhi-Chen et al. pSearch: Information retrieval in structured overlays. Computer Communication Review, 2003, 33(1): 89-94.

[6] Yu CT, Salton G. Precision weighting --- An effective automatic indexing method. Journal of the ACM, 1976, 23(1): 76-88.

[7] Ling Bo, Zhou Yong-Kang, and Zhou Ao-Ying. A Strategy of Query Result Ranking and Merging for P2P Information Retrieval Systems. Chinese Journal of Computers, 2007, 30(3): 405-414

[8] He Ying-Jie, Wang Shan, and Du Xiao-Yong. Efficient Top-k Query Processing in Pure Peer-to-Peer Network. Journal of Software, 2005, 16(4): 540-552.

[9] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In SIGIR, 2002.

[10] Z. Nie, S. Kambhampati, and T. Hernandez. Bibfinder/ statminer: Effectively mining and using coverage and overlap statistics in data integration. In VLDB, 1097-1100, 2003.

[11] T. Hernandez and S. Kambhampati. Improving text collection selection with coverage and overlap statistics. Pc-recommended poster. WWW 2005.

[12] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in p2p search engines. In SIGIR05.

[13] Savoy, J. Report on CLEF-2001 Experiments, In: C. Peters (ed.) Proceedings of the CLEF 2001 Cross-Language Text Retrieval System Evaluation Campaign. Lecture Notes in Computer Science. 27-43, 2002.

[14] Voorhees, E., Gupta, N. and Jhonson-Laird, B. The collection fusion problem, In: NIST (ed.), Proceedings of the 3rd Text Retrieval Conference TREC-3, Vol. 500. Gaithersburg, 95-104, 1995.

[15] Dumais, S. Latent Semantic Indexing (LSI) and TREC-2, In: NIST (ed.), Proceedings of TREC'2, Vol. 500. Gaithersburg, 105115, 1994.

[16] Powell, A.L., French, J.C., Callan, J., Connell, M. and Viles, CL. The impact of database selection on distributed searching, In: T. A. Press (ed.), Proceedings of the 23rd International Conference of the ACM-SIGIR'2000. New York, pp. 232-239, 2000.

[17] Le Calves, A. and Savoy, J. Database merging strategy based on logistic regression, Information Processing and Management 36 (2000), 341–359.

[18] Savoy, J. Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence, In: C. Peters (ed.), Proceedings of the CLEF 2002 Cross-Language Text Retrieval System Evaluation Campaign. Lecture Notes in Computer Science. 31-46, 2003.

[19] Genkin, A., Lewis, D.D. and Madigan, D. Large-Scale Bayesian Logistic Regression for Text Categorization. Technical report, 2004.

[20] Cristianini, N. and Shawe-Taylor, J. An Introduction to Support Vector Machines. CA: Cambridge University Press, 2000.

[21] J. Lv, X. Cheng. Wongoo: a pure peer-to-peer full text information retrieval system based on semantic overlay networks. NCA'04.

[22] Tan Yi-Hong, Chen Zhi-Ping, and Lin Ya-Ping. Research and implementation on searching mechanism based on interest mining in unstructured P2P systems. Computer Application, 2006, 26(5): 170-172.

Acknowledgment

This work is supported by Scientific Research Fund of Hunan Provincial Education Department (07B007). The authors thank the anonymous reviewers for their invaluable comments and constructive suggestions at first, thank the editor in chief, associate editor and editors for their diligent working and helpful suggestions at second, thank the Department of Information and Computing Science, Changsha University, China, for providing the excellent environment and the experiment facility.

Biography

Tan Yi-Hong is an Associate Professor in the Department of Information and Computing Science, Changsha University, China. He is working towards his PhD degree in the School of Computer and Communication, Hunan University, China. His research interests include peer-to-peer, Information retrieval, Data Mining.