Towards a Fully Distributed P2P Web Search Engine

Jin Zhou, Kai Li and Li Tang
Department of Automation
Tsinghua University, Beijing, China
{zhoujin00,li-k02,tangli03}@mails.tsinghua.edu.cn

Abstract

Most centralized web search engines currently become harder to catch up with the growing step of people's information need. Here, we present a fully distributed, collaborative peer-to-peer web search engine named Coopeer. The goal of the work is to complement centralized search engines to provide more humanized and personalized results by utilizing users' collaboration. Towards this goal, three main ideas are introduced: (a)PeerRank to use cooperation among users for evaluation, (b)query-based representation to obtain more humanized description about documents, and (c)semantic routing algorithm to obtain user-customized results.

1 Introduction

It has become increasingly difficult to search for useful information on the web, due to its large size and unstructured nature. Researchers have developed many different techniques to address this challenging problem of locating relevant web information efficiently. The most conventional example is Centralized Search Engine (CSE).

One major problem with CSEs is that they do not facilitate human user collaboration, which has potential for greatly improving web search quality and efficiency. Without collaboration, users 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 CSEs is that they ignores 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 that may be used to find suitable results for a special user.

Developing from a centralized paradigm towards a distributed one, brings in several advantages that cannot be exploited earlier. 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 paper, we propose a peer-to-peer (P2P) approach for web searching, implemented in a system named Coopeer. In Coopeer, information about web pages and user searching experiences is shared in a peer-to-peer way. Our proposal attempts to create a highly distributed system where each user computer stores a part of the web model used for indexing and retrieving web resources in response to queries. All users share these partial models that globally create a consistent model for the web resource that is equivalent to its centralized counterpart. The nodes interact in a peer-to-peer fashion in order to create a real distributed search engine.

The main features of Coopeer are: (a)Collaboration. One may look for interesting web pages in the P2P knowledge repository consisted with shared web pages. A novel collaborative filtering technique called PeerRank is presented to rank pages proportional to the votes from relevant peers; (b)Humanization. We use a query-based representation for documents, of which the relevant words are not directly extracted from page content but introduced by human users with a high proficiency in their expertise domains. (c)Personalization. Similar users are self-organized according to their semantic content of search session. Thus, requestor peer can extend routing pathes along its neighbors, rather than just take a blind shot. Further, user-customized results can be obtained along personal routing paths in contrast with CSEs.

The usage of a combination of the individual and social dimensions of users interests has been proposed for cen-

tralized and distributed knowledge sharing environments [2][11]. They usually pose two crucial problems which have not been solved in a satisfactory fashion. Firstly, privacy is a concern. Knowing that query and even actions is used to build a personal profile, people refrain from using the system. And the overall performance degrades, since it depends on people collaboration. Secondly, storage is a problem given the potentially large number of users. Repositories become intractable both for indexing and recovery. Users' information in Coopeer is storage in their own computers that is completely distributed, so that two problems above are avoided.

Some ways for taking advantage of personal information have been attempted but not much in the area of web searching engines[5][6]. This is partly due to the highly centralized nature of the indexing structure of search engines. In Coopeer, with the help of personal semantic indices, peers can customize their personal routing path. Another advantage of Coopeer is that users' routing action are fully anonymous for both requests and results, because interactions are processed in gossip-like manner, not going through any type of server.

2 Design Overview of Coopeer

The main weaknesses of present web searching system involve the machine-made representation, retrieval and evaluation to information items. In Coopeer, we design three novel methods to address the issues, respectively. In this section, we will present an overview of the concepts of these algorithms in order to set the stage for a description of our system.

In following paragraphs, we give a description to the whole workflow of Coopeer.

Launching a searching run, the requestor forwards the query based on the semantically routing. In [3], it is observed that peers in a P2P network are in general interested in a subset of the total available content on the network. Thus, one may maintain a local index about the semantic content of remote peers. And in term of the semantic clue, effective and efficient routing is achieved.

Receiving a query message from remote peer, current peer check it against the local store. In order to facilitate this work, a novel query-based representation about documents is introduced. This is due to two thoughts that (a)the human users' queries are more accurate to describe the retrieved documents and (b)people tend to use the same subset of words very frequently. Based on query representation, cosine similarity between new query and documents can be computed. And we think the documents are relevant enough, if the similarity exceeds a certain threshold. Then these results are returned to the requestor.

Receiving the returned results, the requestor peer need to rank them in term of preference of its human owner. Hence, a novel P2P collaborative filtering algorithm named PeerRank is given. The rationale of PeerRank is that, rather than machine-based methods, evaluation from human users is more important to the resource quality, especially, on the basis that results have adequate relevance.

2.1 PeerRank

Enlightened by collaborative filtering and social voting, we develop a novel PeerRank algorithm working in P2P network. In fact, when a user first time runs a search, it is likely that the same request has been ever raised by a lot of other users for many times. The searching experience, such as ever used query terms and the evaluation for the results, would be much helpful for the new requestor. However, neither term-frequency-based methods[10] nor linkage-based methods[7] utilizes the human searching experience. In part this is due to that the highly centralized CSEs prefer those machine-based methods. By contrast, in the Coopeer network, all the users are taken as a "Referrer Network". PeerRank determines page's relevance by examining a radiating network of "referrers". Documents with more referrers gain higher ranks. In this way, PeerRank has potential to obtain better rank order, as collaborative evaluation of human users is much more precise than description of term frequency or link amount. Moreover, PeerRank makes a great improvement to prevent spam, since it is difficult to pretend evaluation from human users.

2.2 Query-based Representation

CSEs usually use "important" terms extracted in documents to describe documents seems reasonable. However, some of the "important" terms often mislead people. This is due to the limit of technique of nature language processing. By contrast, human comprehend documents much easier. If a user issues a query and expresses her satisfaction at the returned documents (for example, adding a page to her favorites), we deduce that the query reflects the content of the documents at certain aspects.

Here, Coopeer uses a novel type of query-based representation based on the relevant words introduced by human users with a high proficiency in their expertise domains. Query-based representation is efficient on the P2P platform, the user's evaluation can be utilized easily through the client application. However, CSEs have too many difficulties in using query to represent documents. Gaining the user's evaluation through a web browser seems inefficient for a CSE server, and it is impractical to store and index the documents based on every user's query whose amount will infinitely increase.

In practice, resembling conventional content-based inverted index[1], a novel type of query-based inverted index can be constructed.

2.3 Semantic Routing Algorithm

For the query routing in Coopeer network, we present a directed BFS(Breadth First Search) based on semantic clue which has twofold advantages:

(a) Semantic routing. Coopeer client forwards queries according to the content of neighbor peers.

(b) Adaptive Index Updating. Response to query is used to update local indices of requester and intermediator which record the content of remote peers.

In each Coopeer client, there is a Topic Neighbor Index to describe the content of other peers. Only those topics may be interested to the local user is hold in index. A peer forwards a query to the promising peers which probably have similar content to the query. Requestor's and intermediator's index are updated according to responses which is similar to at least one local topic. Our design for routing algorithm is on the premise that past performance is a good indication of future performance. In practice, peers providing more interested resource would move to the top of one's local index, while others would drop off. Thus, the Coopeer member would be developing an referrer group of companions who have similar interest profile and expertise knowledge.

Other P2P systems, such as CAN[8] and Pastry[9], consider similarity between files in terms of a key space generated by a cryptographic hash. Users must know a file's key in order to retrieve it. Files are inserted into particular locations with aggressive caching activity. By contrast, Coopeer is there for those situations in which users don't know exactly which file they want. And Coopeer does not move files around, rather it gradually learns the existing location of remote content.

3 Implementation

3.1 System Description

The system architecture is shown in Figure 1. The Coopeer client consists of four main software agents: User Agent, Web-searcher Agent, Collaborator Agent, and Manager Agent. The User Agent is responsible for interacting with the users. It provides a friendly user interface, so that users can conveniently manage and manipulate the whole search sessions. The Web-searcher Agent is the resource of P2P knowledge repository. It performs the user's individual searching with several search engines from the Internet. The Collaborator Agent is the key component for performing users' real-time collaborative searching. It facilitates maintaining the P2P knowledge repository, such as information sharing, searching, and fusion. The Manager Agent is the key component of Coopeer, which coordinates and manage the other types of agents. It is also responsible for updating and maintaining data. Once a search is issued, the webpages in WWW will be fetched by the Web-searcher Agent and passed to the User Agent to display. Meanwhile, the Collaborator Agent will share the user's new results to the whole networks.

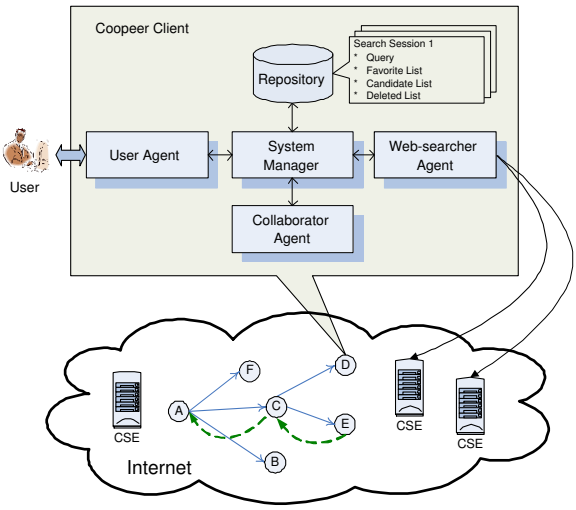

Figure 1. Architecture of Cooper.

Search session is the unit to record all information of a search task. It contains four types of data: query, favorite list, candidate list and deleted list. Favorite list contains the satisfied pages selected by the user; Candidate list contains returned pages but not browsed; Deleted list contains those deleted pages in order to avoid they being sent to the user again.

During the work process of Coopeer client, there are three key operations:

a) Search session configuration

Manipulations about search sessions contain creation, modification and deletion. Launching a search session, the user merely needs to designate the keywords, set expected result number and CSEs. She can select one of the three searching modes: individual search (towards pages in WWW), collaborative search (towards pages in Coopeer network), and hybrid search(towards pages in both WWW and Coopeer network).

b) Results Management

Result panel (Figure 2) consists of favorite list, candidate list, and entry properties. Entry properties area shows the details of selected result entry. In Table 1, a piece of result entry consists of several items, such as Rate, URL, Referrers and so on. A user can customize the favorite list,

such as changing entry's order, or moving the satisfied entry in candidate list to the favorite list.

Table 1. Result list.

Rate	URL	Referrer
2.25	www.cep.org/tvviolence.html	ID718, ID903, ID974
1.55	www.cep.org/studies.html	ID718, ID974
0.80	www.limitv.org	ID521,ID718, ID903

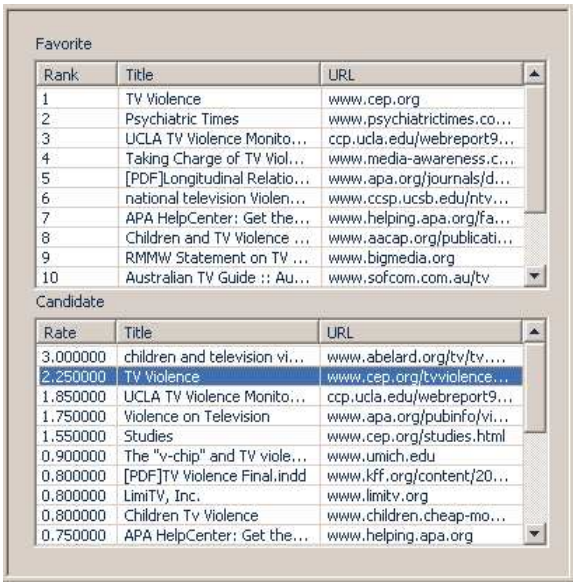

Figure 2. Favorite and candidate lists.

c) Results fusion

If some peer receives new result entries about a search session from peer j, then peer j is regarded as referrer of these entries. For each entry, receiver need to judge whether entry's URL already existed in current search session. So the following three cases are considered:

(i) If the URL exists in deleted list, it will be deleted automatically.

(ii) If the URL exists in favorite list or candidate list, its rate value will be recalculated. The detail of computation on entry's rate will be given in next subsection. If peer j presents in referrer set (means this recommendation is back-call), nothing will happen; Otherwise, peer j will affect the entry's rate value.

(iii) If the URL does not exist in the search session (namely a new web page), the entry's rate will be calculated firstly.

3.2 PeerRank Algorithm

From the view of PeerRank, we determine relevance of a certain web page by examining all its referrers. The Re-

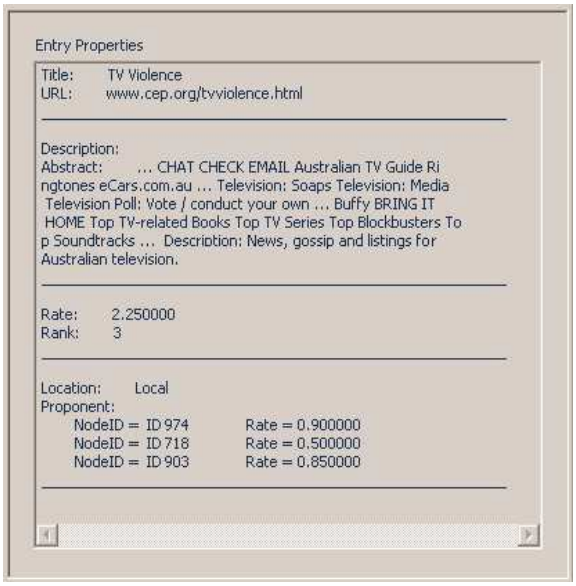

Figure 3. Result entry properties.

ferrers are those peers which have recommended the page to the requestor. It is easy to understand the ranking should bias the referrers more relevant to the requestor. So, for a given search session, we firstly compute the similarity between requestor's favorite lists and referrer's, then the similarity is used as the baseline of recommending degree of the referrer. To compute lists' similarity, a Kendall measure[4] is introduced.

A web page is uniquely denoted by its canonicalized URL. Ranking computation of a certain URL is given as following:

$$R(e) = \sum_{\forall p_i \in C(e)} [Z^{1-K^{(r)}(L_p, L_{p_i})} \times S_{L_{P_i}}(e)] \quad (1)$$

$$S_{L_{p_i}}(e) = \frac{R_{Max} - R_e + 1}{R_{Max}} \quad (2)$$

In equation (1), $R(e)$ represents the weight of URL e. $C(e)$ is the set constituted by e's referrers. Z is a constant larger than 1. p is local peer and p_i represents a remote peer, L_p and L_{p_i} represents their list respectively. $K^{(r)}(L_p, L_{p_i})$ denotes the Kendall function to measure the distance of the local list and the recommended list, where r is the decay factor. $S_{L_{p_i}}(e)$ is the score of e in the recommended list.

In equation (2), R_e is the rank of e and R_{Max} is the highest rank of list p_i, which equals to the length of the list.

Firstly, as shown on the left of product sign in equation (1), the similarity of local list and recommended list is given by the Kendall measure. Secondly, we convert the rank of a given URL in its recommended list to a moderate score,

as shown on the right of product sign in equation (1) and extended expression is given by equation (2). Thus, we can regard the product of lists' similarity and URL's score as recommended degree of one referrer. Finally, from the view of PeerRank, the total of the recommended degrees of all referrers is computed to describe the URL's ranking.

Now we give the definition of Kendall measure. Usually, Kendall[4] is used to measure the distance between two lists in the same length. We extend it to fit in with measuring two lists in different length. We give Kendall function as follow:

$$K^{(r)}(\tau_1, \tau_2) = \frac{\sum_{\{i,j\} \in U(\tau_1, \tau_2)} \overline{K}^r_{i,j}(\tau_1, \tau_2)}{C^2_{2L}} \quad (3)$$

In equation (3), τ_1 and τ_2 are two lists composed with URLs, and if they have different lengths appending new elements that differ from any element of $U(\tau_1, \tau_2)$ to the shorter list so as to make their lengthes equivalent. $K^r(\tau_1, \tau_2)$ is the distance between τ_1 and τ_2, and r is a fixed parameter with $0 \leq r \leq 1$. C^2_{2L} used for normalization is the possible maximum of the distance. $U(\tau_1, \tau_2)$ is the set consists of all the URLs in τ_1 and τ_2, and $\overline{K}^r_{i,j}(\tau_1, \tau_2)$ means the penalty of the URL pair (i, j) belonging to $U(\tau_1, \tau_2)$, for which there are four cases:

Case 1: i and j appear in both two lists. If i and j are in the same order (such as i being ahead of j in both τ_1 and τ_2), then let the penalty $\overline{K}^r_{i,j}(\tau_1, \tau_2) = 0$. Otherwise if i and j are in the opposite order, then let $\overline{K}^r_{i,j}(\tau_1, \tau_2) = 1$.

Case 2: i and j both appear in one list(say τ_1), and exactly one of i or j, say i, appears in the other list(τ_2). If i is ahead of j in τ_1, then let $\overline{K}^r_{i,j}(\tau_1, \tau_2) = 0$, otherwise let $\overline{K}^r_{i,j}(\tau_1, \tau_2) = 1$.

Case 3: i, but not j, appears in one list(say τ_1), and j, but not i, appears in the other list(τ_2). Then let $\overline{K}^r_{i,j}(\tau_1, \tau_2) = 1$.

Case 4: i and j both appear in one list(say τ_1), but neither i nor j appears in the other list(τ_2). Then we let $\overline{K}^r_{i,j}(\tau_1, \tau_2) = r$, where r as the penalty parameter represents the penal attitude. $r = 0$ means optimistic, $r = 0.5$ means middle, while $r = 1$ means pessimistic.

Based on the Kendall measure, we give the definition of Kendall Similarity function of two lists:

$$S^{(r)}_{Ken}(\tau_1, \tau_2) = Z^{1-K^{(r)}(\tau_1, \tau_2)} \quad (4)$$

where $K^{(r)}(\tau_1, \tau_2) \in [0, 1]$, $S^{(r)}_{Ken}(\tau_1, \tau_2) \in [0, 1]$. In Equation (1), the left to the product sign is namely Kendall Similarity. In the experiment in section 4, r is set to 0.

3.3 Query-based Inverted Index

Query-based representation is used to represent and organize the local documents for responding remote query. For each peer, an inverted index table is maintained, whose key is terms extracted from the previous queries and the IDs of the documents that were replied and collected to the query are recorded. For example, when peer j writes in two queries "P2P Overlay" and "P2P Routing" and obtains two set of documents, {d1, d2, d3} and {d3, d4} respectively. These retrieved documents will be updated with their corresponding query terms. Thus, a query-based inverted index can be constructed in peer j, as shown in Table 2. When any other peer issues a query about "Overlay Routing Algorithm", peer j would look up relevant documents in the inverted index by using VSM cosine similarity as ranking algorithm, and d3 would gain the highest ranking.

Table 2. Query-based inverted index.

Query Term	Doc	Freq
Overlay	d1	1
Overlay	d2	1
Overlay	d3	1
P2P	d1	1
P2P	d2	1
P2P	d3	2
P2P	d4	1
Routing	d3	1
Routing	d4	1

3.4 Semantic Routing Algorithm

In order to route semantically, each Coopeer client maintains a local Topic Neighbor Index. The index records the used performance of remote peers which has similar topics to the local peer. From the view of query-based representation, we use search sessions' queries to represent the peers' semantic content. As shown in Table 3, session 1 is the local peer which has two topics (queries), other sessions below denote the remote peers are interested in by the local peer in some aspect. Session 2 and 3 are relevant to "P2P Routing" topic of local peer, while others are about "Pattern Recognition". One topic may contain several peers (session 4,5), and one peer may own several topics (session 3,6). The peers on a same topic are in descending order of the rate.

In Coopeer, each search session can be used to preform not only short term task but long term task. Long term means the result set for a topic is obtained from different runs. For a search session, user can set the starting time in a day. For a remote peer, its rate is adjusted according to equation (5) and (6).

$$\begin{cases} R_{p_i}(t+1) = \Delta R_{p_i} & if R_{p_i}(t) = 0; \\ R_{p_i}(t+1) = a * R_{p_i}(t) + (1-a) * \Delta R_{p_i} & otherwise \end{cases} \quad (5)$$

$$\Delta R_{p_i} = Z^{1-K^{(r)}(L_p, L_{p_i})} \quad (6)$$

Equation (5) shows the adjustment of rate when local peer p receives new results from remote peer p_i about an old session. Here, R_{p_i} represents the rate of peer p_i and the constant a is a decay factor. ΔR_{p_i} is p_i's new rate in the $(t+1)th$ run. As shown in equation(6), we use Kendall Similarity to describe the new rate ΔR_{p_i}. Similarly, the information about the responders will be recorded by intermediators as well.

The peers providing more interested resource would move to the top of an individual's local index, while others would drop off and new recruits would join. In the process, the Coopeer member would be developing an initial referrer group of like-minded web surfers who have similar interest profile and expertise knowledge.

Table 3. Topic Neighbor Index.

Session ID	Query of Known Session	Node ID	Rate
1	"P2P Routing", "Pattern Recognition"	Local	-
2	"P2P Network Routing"	ID718	9.64
3	"P2P Network Topology"	ID244	4.51
4	"AI Pattern Recognition"	ID974	8.50
5	"AI Pattern Recognition"	ID903	4.10
6	"Computer Vision Recognition"	ID244	2.13
7	"Face Recognition"	ID782	1.67

4 Experiments

4.1 Design

We design a experiment on Coopeer system to mainly examine the performance of PeerRank which bases on the cooperation among users in Web IR. Five of the 50 topics used in the TREC11[13] Web Ad hoc task were selected as searching objective, namely "Home buying", "U.S. / Russian relationship", "World population growth", "Mother-infants nutrition", and "Television violence". 40 undergraduates with approximate expertise in using search engines participated in the experiment as subjects. Subjects were given all the five topics with clear description and asked to choose keywords by themselves to perform search and maintain favorite list for each topic as well as possible. They were divided into four groups to perform following searching tasks respectively:

(i) *Individual Group* consists of 30 subjects. They can start Individual Search(IS) directly from CSE without using cooperation among users on the P2P network. They are equally divided into 6 subgroups. Each subgroup uses a CSE. The set of CSEs includes five famous CSEs: AOL, AllTheWeb, Hotbot, Lycos, and Google, and a MetaSearch search engine developed by ourselves consists of all above five CSEs.

(ii) *Collaborative Group* includes 5 subjects, who can carry out only Collaborative Search(CS) but not IS. The cooperations base on the repository of Individual Group.

(iii) *Hybrid Group* includes 5 subjects, who use Hybrid Search, that is using both IS and CS. Their cooperation also base on the repository of Individual Group.

(iv) *Expert Group* consists of 3 "experts", who have abundant experience for searching and well comprehend the five topics, acted by laboratory assistants. The results collected by experts are used as standard to evaluate subjects' performance. The experts run searches in CS manner and maintain elaborate result lists.

4.2 Metrics

Common evaluation measures, such as Recall-Precision Graph, Mean Average Precision and so on, all only focus on total number of relevant results, but ignore their quality. For example, for a same query, users may obtain two result lists from two search engines respectively. If pages in two lists are all relevant to the query, the mean average precision of two search engines are both 100%. But in fact, users may feel that the quality of service from two search engines are completely different. The reason for this is that the order and quality of pages in two lists are all different. These factors are greatly important for users, but cannot be reflected by those traditional measures.

In the experiment, we choose a better metric, Kendall measure, which takes both number and order of the standard results into count. Given weighted standard results, Kendall has potential to precisely measure the quality of the result list. The detail of Kendall measure is given in subsection 3.2.

4.3 Results and Discussion

We computed average Kendall Similarity for each group and each subgroup of CSEs (include MetaSearch). And the comparison of them is shown in Figure 4. Compared with MetaSearch that rigidly integrates CSEs just getting an moderate position, Collaborative Group obtains the best results of all. The fact indicates that human cooperation does be of great help for improving the quality of searching, while combination of CSEs can only augment the searching database. The performance of Hybrid Group is no better than Hotbot the best one of the Individual Group, which implies the collaborative efficiency infected negatively by the individual search, and better hybrid algorithm is needed. In this experiment, we prefer web pages of relevant contents rather than those of site lists, so some CSEs whose results contain most site lists, for example, Google, obtain relevant worse places in the competition.

In the experiment, with query-based inverted index, the precision of matching results of different subjects was almost 100%, which validates the correctness of assumption that human users usually express the same request using alike query terms. Further more, semantic routing algorithm succeeded in leading the query messages to peers having semantic relevant results. In other words, wasteful messages occupy a very low percent of the total.

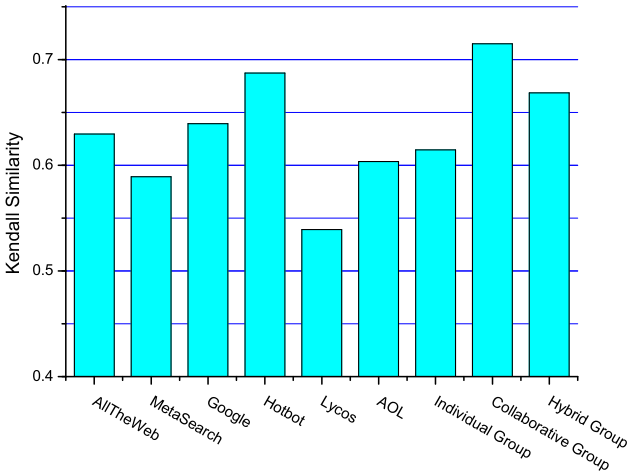

Figure 4. Average of Kendall similarity of CSEs and groups.

5 Conclusion

This paper has shown the Coopeer system, which is a multi-agent system that builds up a collaborative and fully distributed search engine. The main features in Coopeer are collaboration, humanization and personalization. Preliminary experiments encourage us to keep working on that system, since its feasibility seems to be shown.

It is worth remarking that the system uses information coming from centralized search engines, so the system is not aimed to replace CSEs, but to complement them. It does so by construct a collaborative and personalized layer. Coopeer is not able to reply to all the queries with its own information, but it does in a high percentage of cases. However, when results can be given within the system, they are of a higher quality.

In addition, the P2P-based system has other advantages, such as avoidance of the central failure problem, privacy, anonymity and so on. Without a central repository, the system's repository is constituted by thousands of personal repositories spread among the users' computers. Meanwhile, the user privacy can be kept. In Coopeer, not only routing messages but also datum (URL lists and other resource, commonly, they are no more than 100KB) are transferred in truly gossip-like manner. So none can find out the original peer, during not only process of query routing but process of results relaying. All these advantages are only possible in P2P fashion because the information is distributed among all the nodes of the network.

Several obvious extensions to the work are: (a)Balancing results from collaborative or individual search more reasonably. (b)Enhancing routing algorithm by combining path planning with relevance feedback techniques. (c)Improving knowledge acquisition from the monitoring of users' actions to evaluate the satisfaction of users implicitly.

References

[1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[2] M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.

[3] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems, 2002.

[4] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the tenth international conference on World Wide Web*, pages 613–622. ACM Press, 2001.

[5] H. A. Kautz, B. Selman, and M. A. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[6] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM Press, 2001.

[9] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[10] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[11] U. Shardanand and P. Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[12] E. M. Voorhees. Overview of TREC 2001. In *Text REtrieval Conference*, 2001.