A Multi-agent Approach for Peer-to-Peer-based Information Retrieval Systems

Haizheng Zhang, W. Bruce Croft, Brian Levine, Victor Lesser

Computer Science Department
University of Massachusetts, Amherst
{hzhang,croft,brian,lesser}@cs.umass.edu

Abstract

This paper develops and analyzes distributed search techniques for use in a peer-to-peer (P2P) network-based Information Retrieval (IR) system. In the absence of a centralized mediator with global knowledge that directs requests to appropriate agents, agents must cooperate to forward the queries among themselves so as to find appropriate agents, and return and merge the results in order to fulfill the information retrieval task in a distributed environment. In our approach, the agent society is connected through an agent-view structure maintained by each agent. Initially, the agent-view structures are formed by agents connecting to each other randomly. However, we show that such an approach can be significantly enhanced by dynamically reorganizing the underlying agent-view topology and deploying context-sensitive distributed search algorithms. Experimental results indicate that appropriate organizational structures and distributed search mechanisms can have a positive influence on system performance.

1. Introduction

Over the past few years, peer-to-peer (P2P) networks have revolutionized the way we effectively exploit and share distributed resources. In contrast to the traditional client-server architecture, P2P systems are application level, collaborative systems where agents work together to perform certain tasks; thus multi-agent system technology seems very relevant for implementing these types of systems.

In this paper, we explore the use of distributed search techniques for use in a P2P-based Information Retrieval (IR) system. Unlike file-sharing systems that rely on exact-match, IR systems aim to find semantically relevant documents that might not contain all of the keywords in the queries. This search involves locating and retrieving relevant documents distributed among one or more data-bases. We assume each data-base is associated with an intelligent agent that is cooperating with other agents in this distributed search process. Partially decentralized IR systems over P2P networks have been studied extensively over the last decade [1]. In this type of architecture, it is assumed that there is a central mediator that can access the resource descriptions of all the client sub-collections. However, this centralized infrastructure has a number of drawbacks that motivate a more decentralized approach. First, in such a system the central mediator tends to be overwhelmed by a vast number of incoming requests, thus becoming a bottleneck and a single-point of failure. Limited by the capacity of the central mediator, the centralized system does not scale well. Second, short session lengths make a Web-like solution (crawling web pages, indexing, retrieving) impracticable due to the frequent ups and downs of agent that affects the central mediator's ability to track collection updates. We take a different approach here, a mediator-free P2P architecture where no agent in the system has a global view of the collection. Given the incomplete nature of the local views of individual agents about the collections held by other agents, the information retrieval problem in P2P networks is naturally structured as a distributed search process. We investigate how the agent society organization and the distributed search protocols can affect retrieval performance. The evaluation metric will be based on the goodness of the recall ratio (which is the percentage of all relevant documents in the network that are found as a result of the search) when only a small fraction of the network is visited during the search process.

We make the following assumptions in this paper. First, each agent maintains an independent index and IR search engine for its local document collection. However, we do not introduce any further restrictions on the local search engines and thus the network can be populated by agents having very different local search engines. Second, the experimental results presented are based on local search engines that are "perfect" in that they return all relevant documents in the collection for a given query. Third, we assume there is a third-party protocol in place to merge the returned results. Thus, our protocol does not have to deal

with the merging of the returned lists. Lastly, we assume that agents are cooperative in that they all agree to use the same protocols for propagating resource descriptions among each other, accepting queries from peers and finally returning search results to the originators of the queries.

The main contributions of this paper are as follows:

(1) A mediator-free multi-agent information retrieval system for P2P networks that incorporates language models as a resource descriptor into the agent-view structure. The incorporation of a language-model based framework as a resource descriptor allows us to calculate the similarity between collections, or between collections and queries based on a more solid theoretical ground. (2) An agent-view reorganization protocol to dynamically adapt the agent society topology that places semantically similar agents together to form loose content clusters in a distributed and implicit manner. The impact of topological factors on IR efficiency is then analyzed. (3) A set of cooperation mechanisms, based on the character of the agent organization and topology, that take full advantage of the language model and the underlying topology to fulfill information retrieval tasks.

The remainder of the paper is structured as follows: Section 2 introduces the mediator-free framework including the agent-view adaptation mechanisms and distributed search protocol/algorithms. Section 3 presents the methodology and results of the experimental evaluation of the framework. Section 4 discusses these findings in more detail, as well as possible future work. Section 5 further extends this framework into a multi-mediator framework. Section 6 describes related work and then Section 7 summarizes the main conclusions of the paper.

2. A mediator-free framework

In the absence of a mediator, agents must cooperate to forward the queries among themselves so as to locate appropriate agents, rank the collections, and finally return and merge the results in order to fulfill the information retrieval task in a distributed environment. Figure 1 illustrates part of a pure P2P agent society that could comprise thousands of agents. Each agent is composed of five components: a collection, a collection descriptor, a search engine, an agent-view structure and a control center. The collection is a set of documents to share with other peers. The collection descriptor can be considered as the "signature" of the collection. By distributing collection descriptors around, agents can have better knowledge about how content is distributed in the agent society. Specifically, in this system we use collection models as collection descriptors. A collection model is the language model built for a particular collection. It characterizes the distribution of the vocabulary in the collection. The

language model concept was originally introduced in information retrieval research [10] and has proven effective in the distributed IR field [2][4]. It has many interesting properties which are easily exploitable in the peer-to-peer network system: first, a collection model is lightweight since it significantly condenses the description of the content of the collection and thus is much smaller in size compared to the collection. Additionally, the size of the collection model grows minimally with the size of the document collection. Secondly, the collection model is a relatively accurate indicator of the content of the collection. The agent control center is the unit that accepts user queries and is also responsible for performing the distributed search algorithm. The local search engine allows each agent to conduct a local search on its document collection so as to determine whether there are any documents that meet the criteria of a specific user query and then return relevant documents. The agent-view structure, also called the local view of each agent, contains information about the existence and structure of other agents in the network and thus defines the underlying topology of the agent society. The functionality of an agent-view is analogous to the routing table of a network router. In practice, the agent-view structure contains the collection model of the collections as well as the address and other related information about these agents.
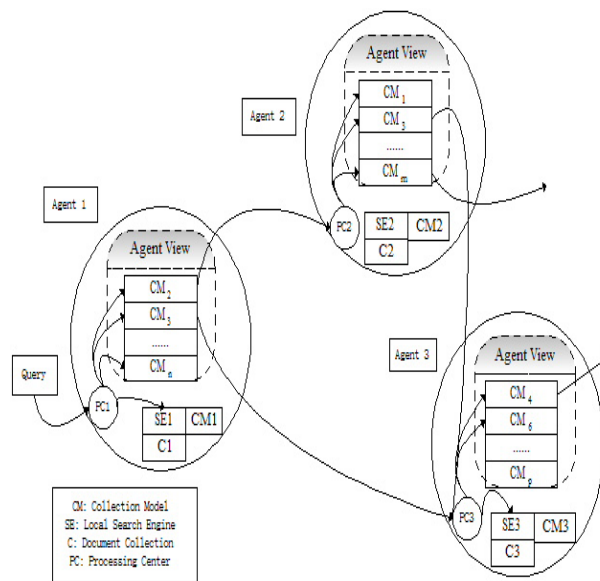

Figure 1. A Mediator-free Multi-agent System

2.1. Agent-view algorithm

A common approach to forming an initial agent-view, as used in the Gnutella system, is for agents when first joining the system to initiate a discovery protocol by sending out ping and pong packets with their IP addresses. In our system, we modify this approach slightly by also

transmitting the document collection model of the agent. This agent discovery procedure results in a random-graph like topology being constructed, where each agent in the network establishes an agent-view structure with the collection models and IP addresses of its neighboring agents.

However, one obvious drawback of such a topology is the lack of search efficiency since there is no connection between the agent-view and how to effectively search for agents that have relevant documents. Therefore, we propose an agent-view reorganization algorithm (AVRA) based on the initial agent-view. The goal of AVRA is to create an agent-view that contains agents whose content is similar, implicitly creating semantically close agent clusters. For example, we can have a "sports" cluster, an "economics" cluster, and so on. These clusters are not disjoint which means that an agent can belong to many of these clusters. For example, an agent can belong to both a "basketball" cluster and a "college" cluster based on its content. Clusters are connected to each other, so if a query is issued to an agent without many relevant documents it can be routed swiftly to the appropriate clusters. To this end, agents exchange their local agent-views to expand the scope of their local agent-view so that each agent is more informed about the content distribution over the entire network. Specifically, each agent decides locally which agent in its agent-view to interact with so as to construct an expanded view in a directed way. The decision to expand along a particular direction results in the sending of an *Expand?* message to appropriate agents. The *Expand?* message includes the address of the sending agent so that the target agents can send back the answer. Upon receiving the *Expand?* message, each agent sends its own agent-view to the requesting agent. Such communication augments each agent's local view with further information about the content distribution in the agent society, and allows the agent to make a more informed decision about whom to forward queries. The agent-view reorganization algorithm then prunes the topology implied by the agent-view so that the agent-view does not become unreasonably large leading to the same scalability issues found in a centralized mediator architecture.

One major concern of the AVRA algorithm is connectivity. Pruning the agent-view without caution can result in very poor connectivity, which in turn reduces the number of agents that can be reached and thus decreases overall system performance. Additionally, agents in P2P networks often vary in their capacity in terms of outgoing connections they can maintain. Therefore, throughout the reorganization process, we keep the agents' out-degree unchanged. Meanwhile, the in-degree of each agent is checked from time to time to ensure that it can be reached by other agents. The algorithm works as follows:

For each agent A_i in the system, we calculate its similarity $W_{cc}(A_i, A_j)$ with the neighbors A_j. After ranking,

agent A_i probes its most similar K neighbor agents with *Expand?* messages. In our experiment, we use $K=1$ to reduce communication. Upon receiving *Expand?* messages, each agent responds with its current agent-view. To prevent the same agents from being chosen repeatedly and thereby slowing convergence, we specify that no agent can be picked more than twice in three consecutive rounds. This heuristic helps an agent construct a more encompassing view so that the reorganization process proceeds smoothly.

After expanding its view as a result of interacting with its neighboring agents, agent A_i prunes its agent-view according to the following rules:

(1) M% of its degree are designated as its most similar neighbors while the rest (1-M%) neighbors are randomly chosen from the agent-views it has collected. This randomization has the effect of maintaining connectivity in the agent society. As experiments show, if all the neighbors are chosen from the most similar agents (M is 100%), the resultant network suffers from poor connectivity. Specifically, it contains many separate "clusters" though these clusters are quite semantically close. After testing different values, we set M to 80. (2) If the number of incoming connections (in-degree) of Agent A_i falls below 2, an empirical threshold which indicates whether this agent is easily reached by the outside world, then the agent contacts its neighbors to request that they add it as one of their neighbors in their local agent-views.

2.2 Distributed search algorithms

The distributed search process is initiated when an agent receives a query. The agent then needs to make a number of local decisions such as whether it should perform a local search to see if it can satisfy the query locally, whether it should forward this query to other agents and to whom, or whether it should drop the query. During this process, if an agent receives a query that it previously processed, it simply skips this message as Gnutella does. Otherwise, the search continues in the network until all the agents receiving the query drop the message. There is no explicit recognition by individual agents that the query is no longer being processed by any agent in the network.

Section 2.2.1 and 2.2.2 propose two search algorithms, namely, k Nearest Neighbors (kNN) collection model-based approach and Gradient Search Scheme (GS), to take advantage of collection models and the reorganized topology. We define agents with local documents that are relevant to the query as "relevant agents", or "irrelevant agents" otherwise. As the agent reorganization algorithm aims to cluster the agents by content, the key to the search algorithms performance is to direct the queries to the relevant agent clusters swiftly.

We introduce two concepts to facilitate the description of the search algorithms.

Definition 1: Covered Agents Level (CAL) is defined as the number of agents visited during a query search divided by the size of the agent society.

$$CAL_n = n / N$$

Definition 2: Cumulative Recall Ratio (CRR) for a query after n agents are searched is defined as

$$CRR_{qi,n} = \sum_{j=1}^{n} r_j / R_{qi}$$

Here R_{qi} is defined as the total number of relevant documents located in the entire network for the query q_i, and r_j is the number of relevant documents located at agent j. CRR is used as a metric to measure the performance of a distributed search algorithm in relationship to its CAL.

2.2.1. kNN collection model-based approach. The insight that the collection model is a stable representation for a collection leads to an intuitive distributed search scheme. An agent first determines if the cluster it belongs to is a "relevant agent zone" or "bad agent zone" by comparing the similarity of the collection it hosts with the query q_i (i.e, $W_{cq}(A_i, q_i)$) and a threshold T_{sim}. Specifically, the algorithm works as follows:

(0) If an agent A_i receives a query q_i, then A_i would drop the query q_i if it has been processed previously, otherwise calculate the similarity $W_{cq}(c_i, q_i)$

(1) If $W_{cq}(A_j, q_i)$ is above threshold T_{sim}, A_j is likely to be located in a "good agent zone." In this situation, the agent computes the similarity of its neighbors A_j and the query q_i (i.e $W_{cq}(A_j, q_i)$) and selects the k agents with the highest $W_{cq}(A_j, q_i)$ value to forward the query. However, in practice, we face the same situation as in the agent-view reorganization process. If we forward all the queries to agents highly-similar with the initiator, then the most similar agents tend to receive the query repeatedly since they form a clique; thereby lowering the chance that other agents are examined as part of the search process. This leads to a low CAL value when the search is completed which motivates an alternative strategy. Thus, instead of forwarding queries solely to highly-similar agents, we also forward queries to some high-degree agents since researchers [12] have found that the high-degree agents are of special importance in the distributed search algorithm. After testing different parameters, we use the top 20% highest-degree neighbors and the top 40% most-similar agents for forwarding the queries.

(2) If $W_{cq}(A_j, q_i)$ is below T_{sim} [1], then the agent considers itself as part of an "irrelevant agent zone." It then tries to expand the search so as to get out of the "irrelevant agent zone" by forwarding the query to high-degree agents rather than highly-similar agents.

2.2.2. Gradient Search Scheme. The Gradient Search Scheme (GSS) differs from the kNN approach in how it deals with the situation when the initiator is in a "bad agent zone." As we will see in Section 3.2, most of the agents in the society are irrelevant to any one query. Therefore, the strategy of how to deal with queries starting from irrelevant agent zones is critical to the search performance. Though the kNN collection model algorithm is designed to direct queries out of "bad agents zone" as soon as possible, experimental results show that the system performance is still very sensitive to where the initial search is originated; the kNN approach suffers from a drastic decrease in performance when it is started from irrelevant agents since it still takes a long time for the query to reach relevant agents. The GS addresses this issue by first trying to locate an appropriate agent for initiating the search for the given query by distinguishing between good and bad starting agents based on the similarity value between the query and agent as the kNN approach does. If the initial agent is good, the Gradient algorithm simply follows the kNN collection model algorithm. Otherwise, the algorithm starts a gradient search process to find a new originator for this query. The detailed protocol works as follows:

(0) If an agent A_i receives a query q_i, then A_i would drop the query q_i if it has been processed previously, otherwise calculate the similarity $W_{cq}(c_i, q_i)$ of the collection on A_i and the query q_i.

(1) If $W_{cq}(c_i, q_i)$ is above a certain threshold T_{sim}, it indicates that the agent is a good candidate for the query. The algorithm then proceeds following the kNN collection model algorithm.

(2) Otherwise, for each neighbor agent A_i, pick the neighbor B which satisfies $\arg\max_j W_{cq}(c_j, q_i)$. A message is then sent from A_i to B with the value $max\ W_{cq}(c_j, q_i)$.

(3) Step (2) is repeated N times. At each round, the old $\max W_{cq}(c_j, q_i)$ values and the new one are accumulated in the message that is then forwarded to the next node. For example, let us assume that agent P is selected after N rounds, the message P receives will contain N maximum similarity values generated as the result of previous rounds. P will then pick the agent with the highest similarity value as the new originator to restart the search using the kNN algorithm. There is a trade-off involved in determining the value for N; the bigger the value of N the more likely that a good originator will be found while the smaller the value of N the quicker the search for relevant documents will begin. Considering these two factors, we used a value for N of three in our experiments.

3. Experimental methodology and results

3.1. Similarity computation

---

[1] After testing different values, we set T_{sim} as 0.15.

Similarity measures are heavily used in both the AVRA and the distributed search algorithms. In our framework, both collection models and query models are treated as language models, and therefore, distributions. We use Kullback-Leibler (KL) divergence to measure the distance between collection models or collection models and query models [6]. The formula is:

$$D(p \| q) = \sum_i p(i) \log \frac{p(i)}{q(i)}$$

Unfortunately, this formula counts each possible word, so it is very time-consuming to compute. To speed the process, an approximate formula is used [7]:

$$D(p \| q) = -\sum_{w \in D \cap Q} p(w|\theta_Q) \log \frac{p_s(w|\theta_D)}{\alpha_D p(w|\theta_C)} - \sum_{w \in Q} p(w|\theta_D) \log \alpha_D p(w|\theta_C)$$

To calculate $p_s(w|\theta_D)$, Dirichlet prior is used [7]

$$p_s(w|\theta_D) = \frac{c(w,D) + \mu * p(w|\theta_C)}{\mu + \sum_w c(w,D)}$$

$$p(w|\theta_Q) = \frac{c(w,Q)}{\sum_{w' \in Q} c(w',Q)}$$

$$p(w|\theta_C) = \frac{c(w,C)}{\sum_{w' \in C} c(w',C)}$$

Theoretically, KL divergence is always positive and ranges from 0 to infinity. However, the above approximate formula can end in negative results. We use a conversion formula $W_{KL}(p,q) = 10^{-\beta D(p\|q)}$ to transform the dissimilarity measure into the similarity measure [6]. Here an empirical parameter β maps the original distance value to domain (0,1). After testing different values, we set β value as 10.

3.2. Experiment metrics

An ideal search algorithm would have a 100% CRR when all the agents are searched, i.e, the CAL is 100%. However, if the network topology is not strongly connected or if the strategy is not designed properly, the search process might only be able to reach a portion of the entire agent network. In reality, considering the network scale and search latency, we are not interested in the situation when the entire network is searched. Rather, we aim to improve the CRR value when the CAL is small.

To simulate a real P2P document-sharing system, we first created a P2P network topology, and randomly distributed hundreds of collections to the nodes in the network. In our experiments, we use the topology generation algorithm discussed in [10] to generate topologies that satisfy power law and small world properties.

In distributing collections to agents, we use TREC 123 and TREC VLC1 collections as both were previously used in the distributed IR research [2,4]. TREC123 is split into

100 small sub-collections based on source and publication date while TREC VLC1 is split to 921 sub-collections largely by source. We denote the two datasets as TREC-123-100 and TREC-VLC-921 respectively. The statistics on the two datasets are shown in Table 1 [2]. The query set 001-050 runs on top of TREC123 and query set 301-350 runs on VLC921. The two collections along with corresponding queries and judgments come from the TREC conference. Figure 2 shows the relevant document distribution. For each query, all agents are ranked according to the number of relevant documents hosted. Each point in Figure 2 is an average of the relevant documents portion for all 50 queries at a certain rank. To make the different collections comparable, the horizontal axis shows the rank over the network size. This figure indicates that most agents contain few or no relevant documents, while a small portion of the agents account for most of the relevant documents.

Table 1. Collection statistics

Sub-collection Number	Source	Number of Docs			Megabytes		
		Min	Avg	Max	Min	Avg	Max
100	TREC 1,2,3	752	10,782	39,723	28	33	42
921	TREC VLC	12	8,157	31,703	1	23	31

As observed from Table 1 and Figure 2, dataset TREC-VLC-921 is more heterogeneous than TREC-123-100 in terms of source, document length, and relevant document distribution. Hence, TREC-VLC-921 is much closer to real document distributions in P2P environments.
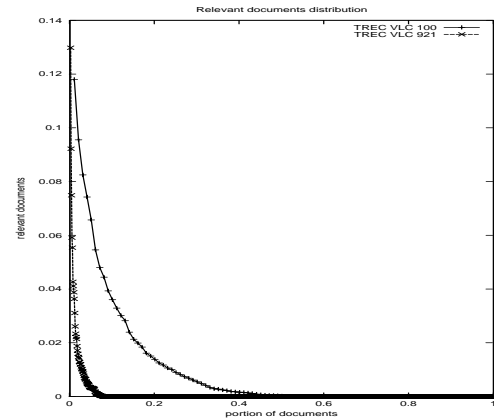

Figure 2. Relevant documents distribution on two datasets

After the underlying topology and collections are distributed to the agents, we reorganize the topology with parameter $K = \{0,3,10\}$ respectively. To perform the search algorithm, we randomly pick agents as originators. In this experiment, we examined the performance of 50 queries for each combination. Each query was repeated 50

times. By averaging 50 results for each query, there was a greater than 95 percent confidence interval.

4. Results and analysis

Various combinations of topologies (K=0,3,10) and search algorithms (random, kNN, Gradient) are tested. For the sake of clarity, Figures 3 and 4 highlight the cumulative recall ratio compared to the covered agent level ranging from 0 to 35% for three search strategies combined with the initial topology and reorganized topology (K=3) respectively. Table 2 presents the results of the random strategy on dataset TREC-VLC-921 and the results of central KL approach. Tables 3 and 4 show the performance of kNN and GSS strategies. More results are available at [15]. Centralized KL divergence-based approach is used as an upper bound on performance, which is common in the distributed IR literature. This approach assumes the network is a fully interconnected graph and visits agents in the order of decreasing similarity values between collections and each query.
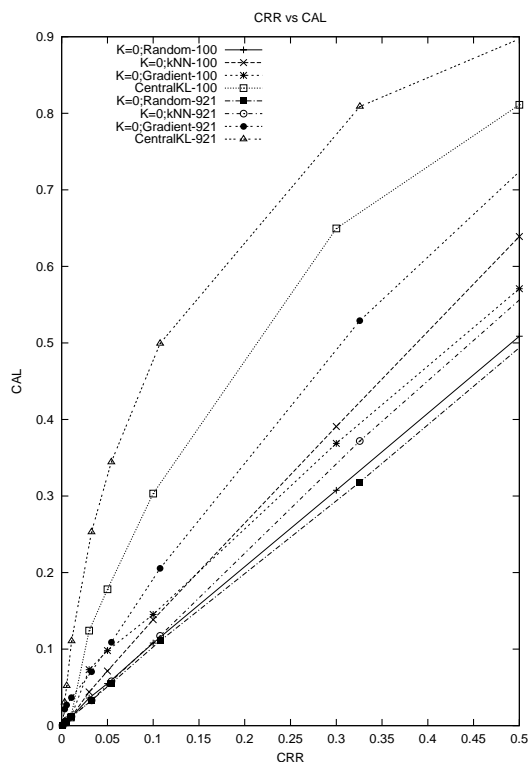

Figure 3. Results of various searching strategies on initial topology (K = 0)

A number of important insights can be gained from our results:

(1) The specifics of the collection properties significantly affect performance. Not surprisingly, IR performance of a random search remains stable when using various collections, as long as the distribution of relevant documents is random. Indeed, as little is known about the content distribution, the number of relevant documents retrieved by random algorithm is proportional to the covered agent level. On the other hand, since kNN, GSS and central KL approaches use the similarity measurement, the results vary between collections. For example, intuitively, in a heterogeneous collection, KL divergence can better reflect the distance between documents than in a homogenous collection. Despite the performance difference, we observe that the impact of search strategies and topologies is consistent on both collections. Hence, without loss of generality, we use the TREC-VLC-921 results to analyze the impacts of searched algorithms and topology algorithms as demonstrated in Tables 2, 3, and 4.
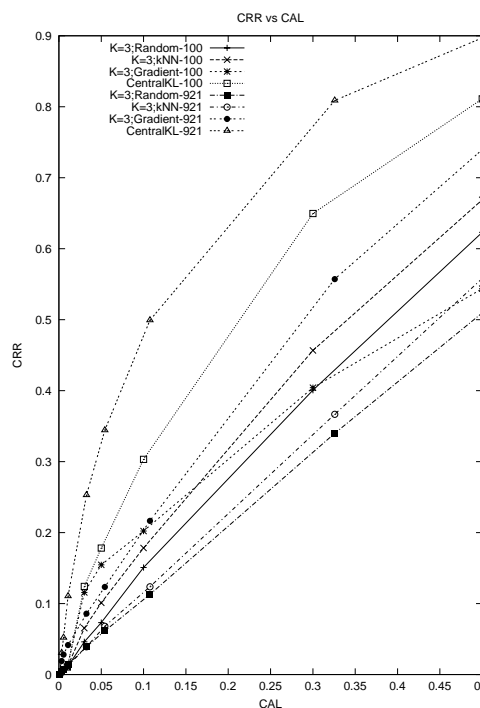

Figure 4. Results of various searching strategies on reorganized topology (K =3)

(2) Tables 2, 3, and 4 show that, as expected, the central KL approach with its global view consistently outperforms the other three approaches. The results are consistent with the conclusion that the collection model is a stable indicator for the collection from distributed information retrieval experiments. Correspondingly, both the GSS and kNN algorithms, which take advantage of collection models, significantly outperform the random search scheme when using the same underlying topology. Tables 2, 3, and 4 show that when a topology with $K = 0$ is used, the cumulative recall ratio of kNN and GSS are significantly better than the random approach when CAL

is below 50%. Another observation is that when CAL is low, which would be the expected case in real networks, GSS further improves the performance of kNN. However, with the increase of CAL, this gain diminishes. For example, when CAL is above 30%, the difference between kNN and GSS is indistinguishable. Therefore, we conclude that although the GSS benefits from a good starting agent, this impact fades as more and more agents are reached.

Table 2. Results for Dataset TREC-VLC-921 with random strategy; The percentage numbers in the columns "K=3" and "K=10" demonstrate the performance gain over (K=0, Random)

CAL	Random Strategy			Central KL
	K=0	K=3	K=10	
0.01	0.01	0.01 21.8%	0.01 24.3%	0.11
0.03	0.03	0.04 12.4%	0.04 17.8%	0.25
0.05	0.05	0.06 7.0%	0.06 11.9%	0.34
0.11	0.11	0.11 -0.9%	0.11 -0.6%	0.50
0.33	0.32	0.33 1.4%	0.34 5.7%	0.81
0.54	0.54	0.53 -1.2%	0.56 4.3%	0.92
0.76	0.76	0.76 1.1%	0.79 4.6%	0.97
1	1.00	1.00 0.0%	1.00 0.0%	1.00

Table 3. Results for Dataset TREC-VLC-921 with kNN Strategy; The percentage in the column "K=0" indicates the performance gain over combination (K=0, random); The percentage numbers in the columns "K=3" and "K=10" demonstrate the performance gain over (K=0, kNN)

CAL	Random Strategy			Central KL
	K=0	K=3	K=10	
0.01	0.02 84.5%	0.02 14.7%	0.02 15.8%	0.11
0.03	0.06 99.7%	0.07 19.6%	0.08 32.6%	0.25
0.05	0.11 101%	0.12 11.4%	0.13 23.1%	0.34
0.11	0.21 90.4%	0.21 4.7%	0.23 13.1%	0.50
0.33	0.58 79.5%	0.60 3.8%	0.62 7.3%	0.81
0.54	0.86 59.0%	0.85 -1.0%	0.85 -0.9%	0.92
0.76	0.89 18.3%	0.92 2.7%	0.93 4.0%	0.97
1	0.89 -10.8%	0.92 2.7%	0.93 4.0%	1.00

Table 4: Results for Dataset TREC-VLC-921 with GSS; The percentage in the column "K=0" indicates the performance gain over combination (K=0, random); The percentage numbers in the columns "K=3" and "K=10" demonstrate the performance gain over combination (K=0, kNN)

CAL	Random Strategy			Centr al KL
	K=0	K=3	K=10	
0.01	0.04 283.5%	0.04 8.9%	0.06 41.8%	0.11
0.03	0.08 158.6%	0.10 20.0%	0.13 59.6%	0.25
0.05	0.12 124.3%	0.15 30.1%	0.19 65.2%	0.34
0.11	0.23 110.1%	0.26 16.9%	0.30 34.4%	0.50
0.33	0.58 80.3%	0.63 9.3%	0.64 10.4%	0.81
0.54	0.85 57.6%	0.85 -0.6%	0.85 -0.4%	0.92
0.76	0.89 17.7%	0.91 2.3%	0.92 3.4%	0.97
1	0.89 -11.2%	0.91 2.3%	0.92 3.4%	1.00

(3) The importance of topology reorganization is dependent on the specific search algorithms used. Table 2 demonstrates that there are no obvious gains from the AVRA algorithm for the random search strategy as the latter does not take advantage explicitly of collection models. On the other hand, there is considerable performance improvement using the GSS when K increases, ranging from *10%-30%* when K=3. When K further increases, the performance benefits are more obvious. The fact that AVRA brings more benefits to the GSS than kNN search is based on GSS's ability to relocate to good originators sooner and the new originators are often surrounded by many other good agents. Performance results stabilize when K>3. Therefore, we conclude that the local agent-view reorganization process tends to converge after three rounds. Of course, this number may differ with various applications and network sizes.

5. Discussion and future work

Notice that in the mediator-free framework, both the agent-view reorganization process and the search process have significant communication costs. In this section we propose a content based multi-mediator framework to reduce the communication costs without sacrificing the positive features of the reorganized topology in mediator-free framework. The essential goal of such a protocol is to split the agents into a number of non-disjoint content-based clusters explicitly. Each cluster focuses on one topic. The intersecting set of two clusters consists of the agents whose document collections fall into many topics. For each cluster a mediator is selected as a "centroid" of the cluster. With new agents joining the system over time, the mediator could change accordingly.

To form the content clusters, we use a distributed K-Means algorithm. Specifically, when an agent is about to join the system, the agent first sends a *join?* message to any mediator in the system. The mediator then forwards the message to the appropriate mediator according to the similarity of the collection model of this agent and the mediator. There can be many mediator candidates and therefore, many clusters the agent can join. When the agent joins one cluster, it could trigger a mediator reelection process. Basically this process is to choose an appropriate mediator to reflect the changed content cluster. In the search phase, the queries are first propagated among mediators. Once an appropriate mediator is located, the queries can then be either forwarded to the agents the mediator supervises or to other mediators.

6. Background and related work

This paper relates to two research areas in different ways as described below:

6.1. A distributed search algorithm

Distributed information retrieval has been formulated as distributed problem solving in the previous research. A P2P based information retrieval system is a special distributed IR problem. Gnutella was initially designed for file-sharing, but provides no support for IR applications. Many researchers have been working on improving IR efficiency in such systems [3][5]. Kalogeraki showed that Gnutella can be revised into a full-text information retrieval system [5]. Cohen investigated how to use "guided rules" to perform partial search[3]. In Kalogeraki's paper, collections are represented by keywords and some BFS (Breadth First Search) variations proposed based on Gnutella-like topology to reduce the number of messages.

6.2. Topology algorithm and multi-agent systems

Recently there have been several works addressing P2P network problems in a multi-agent approach [8][9][13][14]. Yu investigated how searching in a P2P network can benefit from a "referral system"[13][14]. Ogston proposed several clustering and group techniques for multi-agent systems, particularly, P2P networks[8][9]. Nevertheless, to our knowledge, our system is the only work to attack information retrieval in P2P network from a multi-agent perspective.

7. Conclusion

This paper develops and analyzes distributed search techniques for use in a peer-to-peer (P2P) network based Information Retrieval (IR) system. Specifically, we propose a mediator-free multi-agent framework including the initial formation and reorganization of agent-view structures as well as the context-sensitive searching algorithms based on the various topologies. The results demonstrate that a significant increase in IR performance is achieved by merely the distributed search schemes based on the collection model. More IR performance gains occur when the underlying topology is optimized in accordance with the search algorithm. Inspired by such results, we further propose a multi-mediator agent structure. In one such system, there are several distributed mediators which are in charge of a content-based group respectively. The simulation of this framework is left as future work**.**

8. Acknowledgements

9. References

[1] D. Bertolini, etc. Designing peer-to-peer applications: an agent-oriented approach. *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003.

[2] J. Callan. *Distributed information retrieval*. Kluwer Academic Publishers, Reading, Massachusetts, 2000.

[3] E. Cohen, A. Fiat, and H. Kaplan. "Associative search in peer to peer networks: Harnessing latent semantics." *Proceedings of the IEEE INFOCOMS'03 Conference*, 2003.

[4] J. French, Allison, Powell, etc. "Comparing the performance of database selection algorithms." *Proceedings of the SIGIR'99 Conference*, 1999.

[5] Kalogeraki, etc "A local search mechanism for peer-to-peer networks." *Proceedings of the ACM CIKM'03 Conference*, 2002.

[6] C. Manning and H. Schutze. Foundations of Statistical Natural Language Processing, MIT Press. Cambrige, MA: 1999

[7] P. Ogilvie and J. Callan. Experiments using the Lemur toolkit. In *Proceedings of the Tenth Text Retrieval Conference (TREC-10)*.

[8] E. Ogston, B. Overeinder, etc. "A method for decentralized clustering in large multi-agent systems." *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003

[9] E. Ogston. "Group formation among peer-to-peer agents: learning group." *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003

[10] C. Palmer and J. Steffan "Generating network topologies that obey power laws." *Proceedings of GLOBECOM '2000*, November 2000.

[11] J. Ponte and W. B. Croft. "A language modeling approach to information retrieval." *Proceedings of SIGIR*, pages 275--281, 1998.

[12] T.Walsh. Search on high degree graphs. *Proceedings of International Joint Conference on Artificial Intelligence,*2001.

[13] B. Yu and M. Singh. "Searching social networks." *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003

[14] B. Yu and M. Singh. "Incentive mechanisms for agent-based peer-to-peer systems." *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 2003

[15] H. Zhang, W. B .Croft, and B. Levine. "Efficient topologies and search algorithms for peer-to-peer content sharing." *University of Massachusetts, Amherst, CIIR Technical Report IR-314*, August 2003.