An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks[*] [†]

Iraklis A Klampanos
Department of Computing Science
University of Glasgow, Scotland
iraklis@dcs.gla.ac.uk

Joemon M Jose
Department of Computing Science
University of Glasgow, Scotland
jj@dcs.gla.ac.uk

ABSTRACT

Peer-to-Peer (P2P) networking is aimed at exploiting the potential of widely distributed information pools and its effortless access and retrieval irrespectively of underlying networking protocols, operating systems or devices. However, prohibiting limitations have been identified and perhaps the most important one is the successful location of relevant information sources and the efficient query routing in large, highly distributed P2P networks. In this paper, a novel, cluster-based architecture for IR over P2P networks is presented and its evaluation is focused on retrieval effectiveness. We reason in favour of using clustering for P2P IR, by considering two fundamental hypotheses drawn from current P2P file-sharing systems. We also study the potential usefulness of a simplified version of Dempster-Shafer (D-S) theory of evidence combination for results fusion in the network. We simulated the IR behaviour of the system by using the TREC 6 and 7 ad-hoc track. The proposed architecture bears very promising results in terms of precision and recall.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous—*Information Retrieval*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

1. INTRODUCTION

P2P networking is one of the most rapidly developing areas of modern computing. By utilising the exponentially increasing number of Internet nodes, which can be anything from powerful servers to mobile devices, the P2P paradigm attempts to create open and collaborative networks of the most diverse functionality nature. Such functionality extends from the most popular file-sharing protocols, like Gnutella, up to P2P instant messengers and chat applications, not to mention dependable P2P networks for Business-to-Business (B2B) commerce, deployment of Web-Services, E-Education and E-Government [5, 2].

However, regardless of its various applications, P2P networking always involves the discovery of relevant information from within a potentially extremely large pool of peers and the subsequent retrieval of relevant data. This means that P2P computing needs effective and capable IR methods for its successful application. So far, most P2P approaches deal with structured information and so they tend to draw solutions from the databases field. However, if information is unstructured and free-text search is desirable, the role of proper IR methodology is important. In this work, we are interested in free, full-text searching and retrieval of documents and we care to explore standard IR approaches in a P2P environment.

In this paper we propose an architecture for IR over large semi-collaborating P2P networks. By the term "semi-collaborating" we define networks where peers have to cooperate in order to perform information retrieval. However, they do not have to share any detailed information with the rest of the network, nor do they have to be consistent with respect to the IR systems they use. We reason toward the usefulness of clustering in open P2P networks by relying on two fundamental hypotheses, presented in Section 3.2. Finally, we argue in favour of the use of a fusion technique for improving the overall retrieval.

The paper is organised as follows: In the following section we briefly present other ongoing research efforts on P2P IR. Subsequently, we present our proposed architecture for P2P IR and we state and discuss the two key assumptions that led us to the adoption of clustering as a resource participation technique. In Section 4, we present our experimental setup and discuss the results obtained in terms of the IR-standard precision and recall measures. Finally, we conclude by discussing future research areas and issues in Section 5.

2. RELATED WORK

P2P technologies and their exploitation in various fields experience an increasing popularity. During the last four years, many ongoing efforts have been reported, attempting to produce satisfactory solutions for the field of highly distributed P2P IR. Issues related to retrieval, among other things, which arise in networks of "equals" had been conceptually identified long before P2P became popular, in [14].

Arguably, the first project to demonstrate the potential of P2P IR was the Infrasearch project [11, page 100]. Infrasearch was a Gnutella [13]-based meta-search engine which demonstrated the potential of P2P networking in highly diverse information environments. Subsequently, and after the Sun Inc. initiated JXTA [4]

[*] This work is being jointly funded by EPSRC and Sharp Laboratories of Europe Ltd. (Project number: GR/P02653/01).

[†] The authors would like to thank Dr. Victor Poznanski of the Sharp Laboratories of Europe Ltd. and Dr. Peter Dickman and Prof. C. J. "Keith" van Rijsbergen of the University of Glasgow for their invaluable help and guidance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC '04, March 14-17, 2004, Nicosia, Cyprus
Copyright 2004 ACM 1-58113-812-1/03/04 ...$5.00.

(*http://www.jxta.org*) project began, Infrasearch was acquired by Sun and was transformed into the JXTASearch [18] project.

Those projects, despite demonstrating the potential of IR over P2P networks, they practically unveiled some of the most serious problems involved. The most immediate of those issues is that of proper resource discovery and query routing. Gnutella (v0.4), for instance, has serious scalability limitations because it follows a query *flooding* strategy by propagating any query to all reachable peers up to a specific number of hops from the initiator.

This naive approach, apart from creating scalability problems due to the excessive abuse of network resources, it also limits the quality of the retrieved results[1]. In such networks the precision and recall of the result sets cannot be argued at all (let alone guaranteed) since every node in the network is a potential source of results for every given query, regardless of whether it actually has relevant content or not. JXTASearch attempted to address this problem by differentiating among different kinds of peers based on some XML-based meta-data. However, this poses problems for full-text retrieval since it is based on the structure of queries rather than of the content shared.

Currently, a major research front is adopting distributed hash table (DHT) techniques in order to overcome the problem of resource discovery and effective query routing. A DHT is a distributed data structure that maintains information about each peer's content. The main characteristic of this scheme is that the peers get addressed according to the keywords of their corresponding shared documents. Query routing is then reduced into the problem of computing an address, by hashing, according to the keywords of the query at hand. Such approaches include but are not limited to CAN [12] and Chord [15]. The major problem with DHT-based approaches is the same as with JXTASearch: hashing and routing is being done upon extracted data and therefore those approaches are more suitable for structured and semi-structured data where various fields exist and their respective domains can be well defined. This contrasts full-featured text retrieval emphasised in this work.

An interesting hybrid approach named PeerSearch [16] exists, which attempts to combine the functionality and advantages of a DHT with the very successful Vector-Space Model (VSM) for IR. In this system, each peer is responsible for storing indices containing specific terms. For each document, the important terms are extracted and its index is published to all the peers that are responsible for those terms. During retrieval, each query gets propagated to the peers responsible for the keywords of the query. Those peers, finally, search and return matching indices using VSM. Although this approach could work extremely well in controlled environments, it restricts the peers in terms of the IR systems they could use.

A more IR-oriented (rather than networking-oriented) approach is presented within the PlanetP [3] project. Each peer content is expressed by a binary vector whose components represent the presence or absence of individual keywords from each peer collection. These vectors, after being compressed into Bloom filters [1], are diffused in the network so that other peers can be informed. Then, a peer, upon receiving a query, has to compute similarities between the incoming query and the other peers' Bloom filters and then route the query correspondingly. The main drawback of this approach is that every peer has to maintain the corresponding bloom filters of every other peer in the network.

Finally, clustering-based approaches have been reported. In [8], Krishnamurthy et. al. deals primarily with file-sharing problems by describing each file as, mainly, a set of its filename's keywords. This architecture uses a centralised server that performs clustering administrative operations. Each query is routed through this server to nearby peers that hold files which satisfy the query. Additionally, [7] refers to communities of peers seen as interest groups based on sets of attributes. These attributes, which are used to describe each peer's content, are either to be set manually by the user or to be derived from past queries. Finally, [10] applies IR clustering techniques in order to get a phone-directory -like networking structure mainly targeted at multimedia retrieval. This approach organises the network into clusters depending on document description vectors. Queries are then compared to the peer-cluster vectors and routed accordingly. In general, these studies, although using different terminologies, attempt to organise the globally shared content, by clustering or categorising the peers.

3. AN ARCHITECTURE FOR P2P IR

In this section we present our architecture for P2P IR. We also discuss two key assumptions that led to the adoption of clustering as an appropriate content division technique. Finally we propose our content advertisement and query routing procedures as well as the results fusion technique adopted within this architecture.

3.1 Networking Components

In our model peers may choose to implement one or more of the following services, which can be viewed as concurrent threads of execution within each peer process:

Client Service This provides the end-user's interface with the network. Through that service, a user can issue queries to the network, view and retrieve documents and also perform some local administrative tasks.

Information Provider Service By implementing this service, the peer agrees to expose its local document collection to the community so that meta-information can be drawn as well as peer organisation, query routing and retrieval can be accomplished. This denotes the willingness and ability of the peer to share documents with the network (for our experimental purposes we consider only full-text documents).

Hub Service *Hub*-enabled peers form the message routing layer of the network. These are the only peers that are allowed to interconnect with each-other as well as with other kinds of peers, thus forming network topologies. They are responsible for handling meta-information, routing query requests and delivering of results.

Fusion Service This handles the fusion of retrieved results, on behalf of weaker (usually in terms of bandwidth and processing power) peers, before presenting them to the end-user. For our experiments, we restricted fusion to occur only on Hub-enabled peers.

For example, in order for a user to issue a query to the system, the user's peer must implement the *Client Service* and also be able to discover a remote *hub*-enabled peer, already part of the network. This *hub*-enabled peer will, at least initially, be the mediator between the end-user's peer and the rest of the P2P community.

Despite these components being the building blocks of our architecture, they will not be discussed in great detail in this study, as our main focus is the behaviour of such a network with respect to its information retrieval prospects and capabilities.

3.2 Information-Sharing Hypotheses

By taking into consideration existing, widely used P2P file-sharing applications, we base our work on the following hypotheses. It has

[1] This problem has been addressed in Gnutella v0.6, which attempts to solve it by deploying directory nodes in a hybrid environment[9].

to be noted that the following hypotheses do not hold in general but they are expected to hold in a multitude of situations such as generic Internet-based document-sharing.

1. *Individual peers will tend to hold information relevant to a small number of topics. That is, each peer's information provision area will not be unlimited nor random.*

2. *Retrievable items (e.g. documents) that are outliers to some peers will have a high probability to also reside in other peers, where they will be part of the information bulk.*

We argue that, if these hypotheses hold in open information-sharing environments, then perhaps the most natural way to pursue the issue is by applying clustering methods adapted for IR purposes. Following from assumption 1, each peer's content can be described to the rest of the network by a finite number of in-peer cluster descriptors that, together, characterise whole peer collections.

Moreover, assumption 2 is derived from the fact that the retrieval of documents by a peer implies content-dependent replication of documents across the network. Replication is an important factor in P2P environments since it can potentially enhance IR effectiveness if exploited properly. We describe how our clustering techniques take replication into account in Section 3.3.

3.3 Content-Aware Clustering

In order to organise the shared information within our network, we apply a two-stage clustering procedure. Firstly, the individual topics addressed by each peer's collection are identified by clustering (Section 3.3.1). The contents of the peers are subsequently described, to the network, by the corresponding clusters' descriptors. Finally, those clusters get organised further into global, conceptual groups (Section 3.3.2).

3.3.1 In-Peer Document Clustering

Within the individual peers' collections we cluster the documents using a simple form of hierarchic clustering [17]. The descriptor of each document is simply its term frequency (tf) vector. The clustering process stops when there are no single-document clusters left. We calculate the distances between document vectors by using the Cosine measure. The exact clustering algorithm follows:

```
DOC_CLUSTERING(List of Documents$\{D_i\}$) $\rightarrow$ List of Clusters
Place each $D_i$ in its own cluster $C_i$;
Compute a cluster-to-cluster similarity matrix $\mathcal{M}$;
**while** ($\exists C_k.\, |C_k| = 1$) **loop**
    Find MAX($\mathcal{M}(C_m, C_n)$);
    Create new cluster $C_{NEW} = C_m \circledast C_n$;
    Remove $C_m$ and $C_n$ entries from $\mathcal{M}$;
    Add $C_{NEW}$ to $\mathcal{M}$;
    Update $\mathcal{M}$;
**end loop**;
```

where $x \circledast y$ denotes the merging of clusters x and y and $|C_i|$ denotes the size of C_i, *i.e.* the number of documents within the cluster C_i.

However, a cluster's quality cannot be taken for granted and so, for the subsequent clustering of peers (discussed in the next section), we devised two additional characteristics in order to safeguard against ill-formed clusters.

The first metric is the average standard deviation $\overline{\sigma}$ of the tf components among the respective documents within each cluster. This metric is used to protect the clustering and the consequent retrieval processes from low-quality in-peer clusterings by measuring the scale of randomness (sparseness) of the cluster's document collection. Peers whose contents are more consistent (*i.e.* whose $\overline{\sigma}$ is small) are to be preferred over others.

The second statistic is the participation level of a cluster $\mathcal{P}$, which is calculated as $\mathcal{P} = \frac{\text{\#docs within cluster}}{\text{\#docs within peer}}$. This metric expresses the level of expertise of a peer concerning a particular topic characterised by the cluster. Peers whose level of expertise is higher (*i.e.* their bulk of information is consistent with a particular topic) are to be preferred, for query routing, over others since they are more likely to contain more relevant documents (following from assumption 2 of Section 3.2).

Therefore, each document cluster within a peer is expressed in terms of its centroid document D^*, $\overline{\sigma}$ and $\mathcal{P}$.

3.3.2 Peer Clustering

At the networking level, peers get clustered into what we will, for clarity reasons, refer to as *Content-Aware Groups* (CAGs). CAGs are conceptual representations of different topics in the network. For this clustering procedure we use a simple one-pass algorithm, but we also take into consideration the two metrics described in the previous section; peers get clustered according to their content differences as well as in terms of $\overline{\sigma}$ and $\mathcal{P}$.

Peers can belong to more than one CAG depending on their internal clusters. Their internal clusters are thought to represent the topics that characterise the peers. The network, effectively the *Hub* layer, gets informed about groups of peers and their content and $\overline{\sigma}$ and $\mathcal{P}$ characteristics (which for a CAG are calculated by averaging its members' corresponding values).

Bearing in mind that each peer P is advertised as a finite set of document clusters $\{C_{P_1}, C_{P_2} \ldots C_{P_n}\}$, the exact algorithm for the organisation of peers into CAGs is as follows:

```
PEER_CLUSTERING(List of Peers $\{P_i\}$) $\rightarrow$ List of CAGs
**for** (each peer $P$ entering the network) **loop**
    **for** (each document cluster $C_{P_i}$ of $P$) **loop**
        /* t is a threshold angular separation */
        **if**(cosDiff$(C_{P_i}.D^*, CAG_C) < t$ **and**
          $C_{P_i}.\overline{\sigma} \leq CAG_C.\overline{\sigma}$ **and**
          $C_{P_i}.\mathcal{P} \geq CAG_C.\mathcal{P})$
        **then**
          Merge $CAG_C$ and $C_{P_i}$;
        **else**
          Create new $CAG_{New}$ to accommodate $C_{P_i}$;
        **end if**;
    **end loop**;
**end loop**;
```

where "$x.y$" denotes that y is a member attribute of x and cosDiff(x, y) denotes the cosine distance between tf vectors x and y..

An example overlay network illustrating the concept of CAGs can be seen in Fig. 2.

3.4 Query Routing

By using the peer clustering algorithm introduced in section 3.3, we effectively create clusters in three dimensions, namely *cosine distance*, $\overline{\sigma}$ and $\mathcal{P}$. Actually, peers of similar such properties get clustered together. This fact is being exploited during query routing.

Upon receiving a query Q, a *Hub*-enabled peer, maintaining the descriptors of all CAGs in the network, calculates a score $\mathcal{S}_i$ for each CAG_i in the network. According to the hypotheses of Section 3.2, the best candidate CAGs for answering Q are primarily those

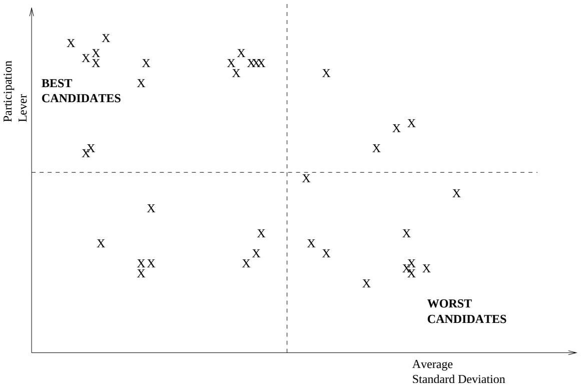


Figure 1: Illustrating the importance of $\overline{\sigma}$ and $\mathcal{P}$ for clustering and query routing.

whose centroids have the minimum angular distance from Q. In addition, the best candidate CAGs are those whose documents are more closely related to each other, measured by each CAG's $\overline{\sigma}$, and whose, related to Q, documents form the information bulk among the participating peers, measured by $\mathcal{P}$.

We calculate this score $\mathcal{S}$ as the weighted average $\alpha\text{CosDiff}(Q, \text{CAG}_j) + \beta(1 - \overline{\sigma}_{\text{CAG}_j}) + \gamma\mathcal{P}_{\text{CAG}_j}$[2], where CosDiff is the cosine difference between the incoming query Q and each CAG's centroid document.

Fig. 1 illustrates the intuition behind the usefulness of $\overline{\sigma}$ and $\mathcal{P}$ at a hypothetical situation where a number of CAGs have the same angular distance from an incoming query Q.

CAGs get ranked according to that score and subsequently Q is sent to the top n ones. Building further on assumption 2 of Section 3.2, n depends on each CAG's participation level. The query, after reaching the top n CAGs, through the *Hub* layer of the network, also gets routed within each CAG into individual peers. Within each CAG, individual peers (represented by document clusters) get ranked following the same procedure, with the difference that scores get normalised over the maximum score obtained within the CAG. Finally, the top m peers get to answer the query. The exact procedures followed for our experimental evaluation are described in Section 4.1.

3.5 Fusion

For the combination of results in our system we use the Dempster-Shafer (D-S) theory of evidence combination as presented in [6]. D-S application, apart from its ability to handle uncertainties arising from the network, it also provides useful insight on how to accommodate potential relevance feedback techniques to further enhance retrieval.

Suppose that a peer is connected to k information sources and one of these sources, j, returns n_j results in total. Then, the initial score of each of j's returned results would be:

$$S_{i_j} = \frac{[n_j - (p_{i_j} - 1)]}{R_j} \tag{1}$$

where p_i is the proposed, by the information source, rank of the result i (*i.e.* 1 for the most relevant result, 2 for the next one and so on) and R_j is given by $R_j = \sum_{\iota=1}^{n_j} \iota$, which acts as a normalising factor.

Each *Information Provider* j, is assigned a positive real number β_j, an un-trust coefficient, where $0 \leq \beta_j \leq 1$ for each incoming query Q. This comes from the score $\mathcal{S}$ calculated for the peer of a CAG for a query Q and is $\beta_j = 1 - \mathcal{S}_j$. The coefficient ($\beta_j$) is assigned to the entire set of documents of the *Information Provider* and is interpreted as the uncertainty of the source of informal evidence[3]. By using β_j and Equation 1, we can evaluate the mass function for each result:

$$m_j(\{d_i\}) = S_{i_j}(1 - \beta_j) \tag{2}$$

where $1 - \beta_j = \mathcal{S}_j$. Finally, the results coming from different information sources (different evidence) can be calculated by applying the Dempster-Shafer theory of Combination of Evidence as follows. For each two information sources 1 and 2, the new mass function of each result d_i of the information provider 1 is given by:

$$\begin{aligned} m'(\{d_i\}) &= m_1(\{d_i\}) \otimes m_2(\{d_i\}) \\ &= m_1(\{d_i\})m_2(\{d_i\}) + \\ &\quad\; m_1(\{d_i\})m_2(\Theta) + \\ &\quad\; m_2(\{d_i\})m_1(\Theta) \end{aligned} \tag{3}$$

where Θ is the global set of documents.

Additionally, the new un-trust coefficient $m'(\Theta)$ of the combination can be obtained from

$$m'(\Theta) = m_1(\Theta) \times m_2(\Theta) \tag{4}$$

Any new set of results, from a third information source, can be combined further by re-using Equations 3 and 4.

3.6 Summary

In our network, the workload is divided among the peers, depending on their willingness and ability to participate, by a set of services (Section 3.1). According to their content and topology, peers are being organised into *Content-Aware Groups* or *CAG*s (Section 3.3). Figure 2, depicting a sample network, gives an illustration of CAGs.

Peers as well as CAGs advertise their content mainly by means of term-frequency vectors. However, individual peers' vectors are diffused only within the CAG they belong, while the knowledge of CAGs' descriptors is global through-out the routing layer of the network. Those advertisements are held and administered only by the corresponding hub-enabled peers.

Routing of queries into CAGs is based on the CAGs' descriptors and subsequently, when the query has reached a specific CAG, routing into peers is based on local knowledge of individual peers' descriptors (Section 3.4). Finally, the results are propagated back to the client, getting combined and re-ranked along the way, by following the path of the query (Section 3.5).

4. EVALUATION

4.1 Methodology

For our evaluation purposes we used the Ad-Hoc TREC collection and the relevance assessments from TREC 6 and 7 (100 topics in total). The collections we used comprised of 556,077 documents of various lengths. Our experimental setup simulated 1,500 peers and the IR system we used for retrieval at those peers is MG [19].

In order to approximate the hypotheses of Section 3.2 we distributed the relevant documents, drawn from the topics' relevance

[2] $(1 - \overline{\sigma})$ makes sense since it is calculated upon normalised tf vectors and therefore $\overline{\sigma} \in [0, 1)$.

[3] $\mathcal{S}_j$ reflects how certain we are concerning the validity of the results coming from the source j.

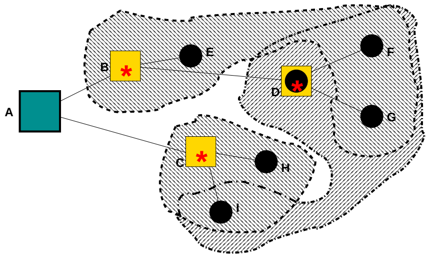


Figure 2: A sample P2P network in which the situation of peers belonging to more than one CAG is shown – e.g. F and G belong to two CAGs.

assessments, into a number of different peers. By doing so, the information bulk of those peers was relevant to specific queries. The rest of the documents were distributed randomly to the remaining *Information Provider*s, something which contradicted the first assumption and affected our evaluation results negatively. After the document distribution had finished, all peers were sharing approximately the same number of documents in total.

We followed this strategy because of the difficulty to create a realistic information environment for P2P IR. The major drawback lies in the fact that clustering can be very CPU-intense and memory expensive and in our case, we would have had to cluster the global corpus consisting of approximately half a million documents.

4.1.1 Query Routing and Retrieval

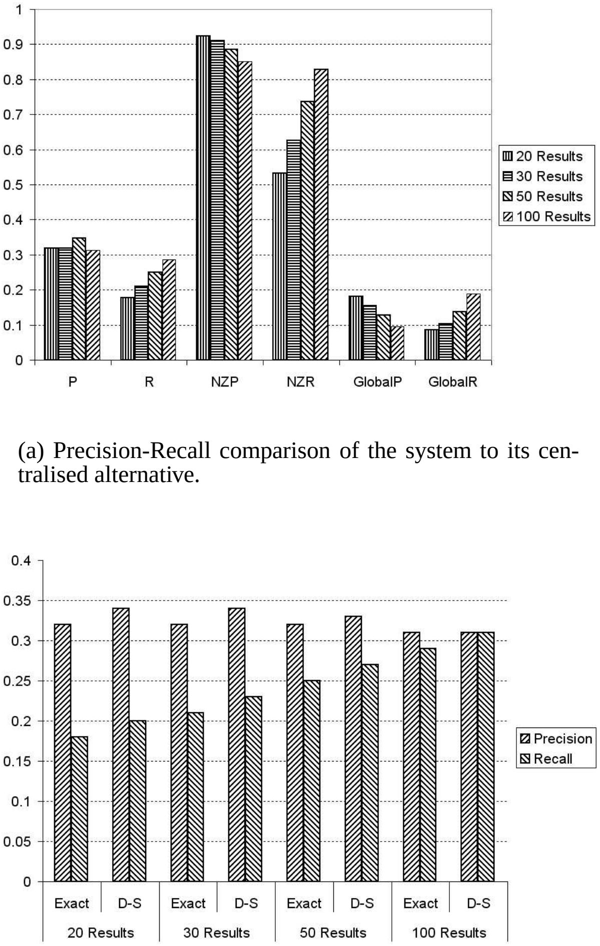
In Section 3.4 we described the formula we used in order to rank and select the most relevant CAGs as well as peers for an incoming query Q. For our experiments we used:

$$\mathcal{S} = 0.8\text{cosDiff}(Q, \text{CAG}_j) + 0.15\overline{\sigma_{\text{CAG}_j}} + 0.05\mathcal{P}_{\text{CAG}_j}$$

for ranking the CAGs. The parameter values 0.8, 0.15 and 0.05 were derived experimentally. The same parameter values and a similar formula, only normalised[4] as indicated in Section 3.4, was used for the ranking of relevant peers within individual CAGs.

In our simulation, when a query was diffused into the network, the top n CAGs were identified for the query to be routed to. The maximum number of desired results, N, was also passed onto the network within the query message. We adopted an *ad-hoc* rule, partly based on the second hypothesis of Section 3.2, to make that selection. The top CAGs were selected until the linear addition of their respective participation levels was above 1.0. Subsequently, each CAG was requested to provide a fraction of N results proportional to its participation level.

When a query reached the top ranked CAGs we followed two alternative approaches. The first approach was to retrieve exactly the required number of documents from the most relevant peers of each CAG. For clarity reasons we will, hereafter, refer to this approach as *exact retrieval*[5]. In this case we do not need to apply any fusion as the number of results returned is exactly the number of results initially requested. The second approach was to propagate the query to each peer within the CAGs and retrieve N results from each one (peer). Then, we applied the D-S rule in order to fuse the results and we calculated P-R upon the N top results.



(a) Precision-Recall comparison of the system to its centralised alternative.

(b) Precision-Recall comparison between the exact retrieval and the D-S combination alternatives.

Figure 3: Precision and Recall evaluation.

4.2 Results

In this section we present the results we obtained from our system in terms of the standard IR precision and recall measures. Those are depicted in Fig. 3(a). Of particular importance are the P-R results obtained for our P2P architecture in comparison to the centralised alternative, which demonstrate the potential P2P IR systems have over centralised sites.

For some result sets we got P-R values of 0.0, meaning that the query had not been routed to the relevant peers. However, bearing in mind that the overwhelming majority of the peers contained randomly allocated documents, their centroid vectors (which they were computed by averaging, hence insignificant) might, by pure chance, have been closer to particular queries than the centroids of the actual relevant peers. This is close to the worst case for our network and that is why we also provide the average P-R values without taking into consideration those cases (indicated by *NZ* for *Non-Zero* in Fig. 3(a)).

[4] For the case of individual peers the scores were normalised so that they could be used as mass function for the results fusion by the D-S rule.

[5] The term *exact* refers to the *number* of requested results and not to the retrieval process. Hence, *exact retrieval* as used in this paper is totally different from *exact matching* retrieval.

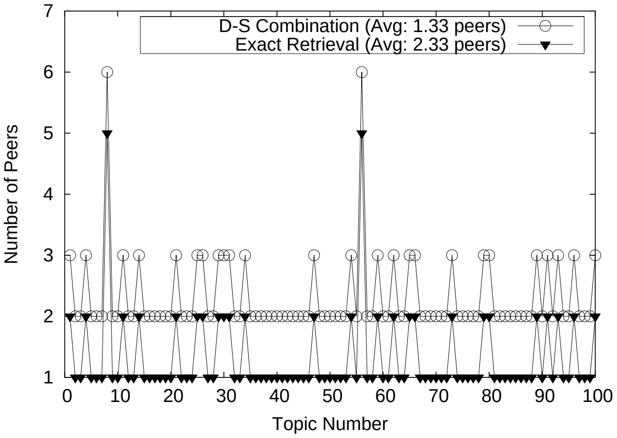


Figure 4: Comparison between exact retrieval and D-S combination approach in terms of the number of peers reached.

In Fig. 3(b) we present a comparison between the exact retrieval process (discussed in Section 4.1) and the application of the D-S rule for combination in terms of precision and recall. The differences are marginal, however, the D-S approach performs consistently better in terms of precision and recall despite the fact that the number of retrieved documents in this case was, by many factors, greater than that of exact retrieval. We consider this to be a promising fact and an indication that P2P IR could benefit from fusion techniques.

Finally, we present the number of peers the queries had to be propagated to in Fig. 4. As it can be seen, the number of peers that were contacted, by both exact retrieval and D-S approaches, is very small compared to the total number of peers simulated. This shows that our resource discovery mechanism, although not infallible, achieves to perform comparably to its centralised counterpart by propagating the queries to a minimal number of peers.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented an IR architecture designed to work over semi-collaborating P2P networks. Our architecture supports efficient and effective resource discovery for full-text retrieval. We also presented two basic hypotheses upon which we argued in favour of the suitability of clustering as the core mechanism for the organisation of such a network. We demonstrated how a fusion technique can be successfully applied to, potentially, improve retrieval effectiveness; an aspect which, to the best of our knowledge, is neglected by most of the other published studies. Finally, we backed our claims by performing an IR-oriented evaluation, in terms of precision and recall as well as the number of peers contacted for the topics' evaluation.

IR over P2P networking has opened many interesting research areas. At the time, our current research focuses on ways to create a large, more realistic document distribution simulating P2P IR. In the near future we intend to continue researching on different aspects of P2P IR. Possible continuations include the formalisation and optimisation of the applied clustering methods, as well as the design and implementation of a full P2P IR system to aid us in future system emulations. Of particular importance and potential is the adaptability of the network and its effect on IR effectiveness. Finally we believe that fusion in P2P IR systems is important and that further research would benefit the community.

6. REFERENCES

[1] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[2] J. Brazelton and G. A. Gorry. Creating a knowledge-sharing community: If you build it, will they come? *Communications of the ACM*, 46(2):23–25, February 2003.

[3] F. M. Cuenca-Acuna and T. D. Nguyen. Text-Based Content Search and Retrieval in ad hoc P2P Communities. In *International Workshop on Peer-to-Peer Computing (co-located with Networking 2002)*. Springer-Verlag, 2002.

[4] L. Gong. Jxta: A network programming environment. *IEEE Internet Computing*, 5:88–95, May-June 2001.

[5] G. Grayston, T. Laev, and C. Macfarquar, editors. *E-Commerce and Development Report 2002*. United Nations, Geneva, 2002.

[6] J. M. Jose. *An Integrated Approach for Multimedia Information Retrieval*. PhD thesis, The Robert Gordon University, April 1998.

[7] M. Khambatti, K. Ryu, and P. Dasgupta. Peer-to-peer communities: Formation and discovery. PDCS'02, November 2002.

[8] B. Krishnamurthy, J. Wang, and Y. Xie. Early measurements of a cluster-based architecture for p2p systems. San Francisco, USA, November 2001. ACM SIGCOMM. Internet Measurement Workshop.

[9] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management CIKM 2003*, November 2003.

[10] C. H. Ng and K. C. Sia. Peer clustering and firework query model. Hawaii, May 2002. 11th World Wide Web Conference.

[11] A. Oram, editor. *PEER-TO-PEER: Harnessing the Power of Disruptive Technologies*. O'Reilly & Associates, Inc., CA 95472, USA, March 2001.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of ACM SIGCOMM 2001*, 2001.

[13] C. D. S. Services. The gnutella protocol specification v0.4. http://www.gnutella.co.uk/library/pdf/gnutella_protocol_0.4.pdf. As viewed on November 3rd 2002.

[14] P. Simpson. Query processing in a heterogeneous retrieval network. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 359–370. ACM Press, 1988.

[15] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[16] C. Tang, Z. Xu, and M. Mahalingam. Peersearch: Efficient information retrieval in peer-peer networks. Technical Report HPL-2002-198, Hewlett-Packard Labs, 2002.

[17] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[18] S. Waterhouse. Jxta search: Distributed search for distributed networks. http://search.jxta.org/JXTAsearch.pdf, May 2001. As viewed on February 25th 2003.

[19] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 2nd edition, 1999.