Scalability of findability: effective and efficient IR operations in large information networks

Conference Paper · July 2010

DOI: 10.1145/1835449.1835465 · Source: DBLP

2 authors, including:

Weimao Ke
Drexel University
69 PUBLICATIONS **1,357** CITATIONS

SEE PROFILE

Scalability of Findability: Effective and Efficient IR Operations in Large Information Networks

Weimao Ke
School of Information and Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3360, USA
wke@unc.edu

Javed Mostafa
School of Information and Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3360, USA
jm@unc.edu

ABSTRACT

It is crucial to study basic principles that support adaptive and scalable retrieval functions in large networked environments such as the Web, where information is distributed among dynamic systems. We conducted experiments on decentralized IR operations on various scales of information networks and analyzed effectiveness, efficiency, and scalability of various search methods. Results showed network structure, i.e., how distributed systems connect to one another, is crucial for retrieval performance. Relying on partial indexes of distributed systems, some level of network clustering enabled very efficient and effective discovery of relevant information in large scale networks. For a given network clustering level, search time was well explained by a polylogarithmic relation to network size (i.e., the number of distributed systems), indicating a high scalability potential for searching in a growing information space. In addition, network clustering only involved local self-organization and required no global control – clustering time remained roughly constant across the various scales of networks.

Categories and Subject Descriptors

H.3.4 [**Information storage and retrieval**]: Systems and Software—*Distributed systems, Information networks*

General Terms

Algorithms, Performance, Experimentation

Keywords

distributed IR, scalability, network clustering, decentralized search, weak tie, strong tie, clustering paradox, connectivity

1. INTRODUCTION

In today's digital environments, there exist a variety of information networks where information is distributed among dynamic systems. On the Web, for example, individual web sites host diverse information topics and form a network by means of hyperlinks. Likewise, digital libraries interoperate with one another and serve information distributed across collections in a network. For reasons such as copyright and privacy, lots of information cannot be fully collected and indexed in advance for retrieval purposes. In addition to this is the dynamics of many environments such as the *deep web* and peer-to-peer networks, in which it is not only difficult to gather information but also challenging to keep an index up to date.

Centralized IR solutions can hardly survive the continued growth of today's information spaces – they are vulnerable to scalability demands [3]. A distributed architecture is desirable and, due to many constraints, is often the only choice. Distributed (federated) IR research is a response to the challenge of retrieving information from distributed sources. Recent distributed IR research has focused on intra-system retrieval fusion/federation, cross-system communication, and distributed information storage and retrieval algorithms [9, 23].

Classic distributed information retrieval has shown some potential of efficiently and effectively bringing distributed information together. However, the reliance on centralization of a metasearch server will continue to suffer from critical problems such as scalability, single point failure, and fault tolerance. Further decentralization of meta search models will involve issues beyond the main focus of federated IR research.

Research has been done under the theme of peer-to-peer information retrieval (P2P-IR) and, more recently, large scale distributed systems for IR (LSDS-IR) [5, 10, 16, 11]. While classic distributed IR often focuses on tens, if not hundreds, of distributed collections, P2P- or LSDS-IR usually envisions an IR problem situated in thousands and even millions of distributed, dynamic systems. The magnitude, distribution, and dynamics of information in such an environment remain a great challenge in IR. Applications of this research include not only search in peer-to-peer environments but also information retrieval in digital libraries, intelligent information discovery on the deep web, distributed desktop search, and agent-assisted web surfing etc.

Finding relevant information in distributed networked environments transforms into a problem concerning information retrieval and complex networks. In this study, we focus on how relevant information can be effectively and efficiently found in large scale information networks, where no centralized index can possibly be built. We investigate the impact of network structure/topology on the effectiveness and effi-

SIGIR'10 July 19–23, 2010. Geneva, Switzerland.

ciency of decentralized IR operations relying on distributed indexes. We test the proposed retrieval methods in a growing information space and examine the scalability potential.

2. RELATED WORK

While traditional IR and distributed IR research provides basic tools for attacking decentralized search problems, the evolving dynamics and heterogeneity of today's networked environments challenge the sufficiency of classic methods and call for new innovations [3]. Whereas peer-to-peer offers a new type of architecture for application-level questions and techniques to be tested, research on complex networks studies related questions in their basic forms [2, 23].

2.1 P2P Information Retrieval

In an open, dynamic information space such as a peer-to-peer network, people, information, and technologies are all mobile and changing entities. Identifying where relevant collections are for the retrieval of information is essential. Without global information, decentralized IR methods have to rely on individual indexes in distributed nodes and their limited local intelligence to collectively construct paths to desired information.

Recent years have seen growing popularity of peer-to-peer (P2P) networks for large scale information sharing and retrieval [17]. There have been ongoing discussions on the applicability of existing P2P search models for IR, the efficiency and scalability challenges, and the effectiveness of traditional IR models in such environments [23]. Some researchers applied Distributed Hashing Tables (DHTs) techniques to *structured* P2P environments for distributed retrieval and focused on building an efficient indexing structure over peers [7, 18, 21].

Others, however, questioned the sufficiency of DHTs for dealing with high dimensionality of IR (e.g., a large number of terms for document representation) in dynamic P2P environments [5, 17, 16]. For retrieval with a large feature space, which often requires frequent updates to cope with a transient population, it is challenging for distributed hashing to work in a traffic- and space-efficient manner. *Unstructured* overlay systems work in an nondeterministic manner and have received increased popularity for being fault tolerant and adaptive to evolving system dynamics [17].

2.2 Decentralized Search in Networks

Research on complex networks provides valuable principles for searching/navigation in distributed systems. Not only do many information networks such as the Web share the common phenomenon of *small world* but they also appear to be searchable [2]. Particularly, studies showed that without global information about where targets are, members of a very large network are able to collectively construct short paths (if not the shortest) to destinations [15, 22, 8].

The implication in IR is that relevant information, in various networked environments, is very likely a small number of connections/links away from the one who needs it and is potentially findable. This indicates potentials for decentralized retrieval algorithms to traverse an information network to find relevant information efficiently. However, this is not an easy task because not only relevant information is a few degrees/connects away but so is all information.

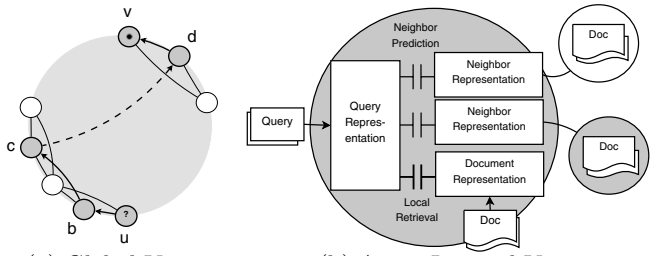
To find relevance in a densely-packed "small world" network remains very challenging. Nonetheless, research has demonstrated how nodes connect to one another and the structure of the network they thus form have critical impacts on how searches function. Network clustering, sometimes by means of semantic overlay, can significantly improve effectiveness and efficiency of IR operations in an information network.

Clustering, the process of bringing similar entities together, is useful for information retrieval. Traditional IR research utilized document-level clustering to support exploratory searching and to improve retrieval effectiveness. In large scale distributed IR, topical clustering techniques such as semantic overlay networks (*SONs*) have been widely used, in which systems containing similar information form semantic groups for efficient searches [5, 10, 16].

Research indicated that a proper degree of network clustering with some presence of remote connections has to be maintained for efficient searches [15, 20]. Clustering reduces the number of "irrelevant" links and aids in creating topical segments useful for orienting searches. With very strong clustering, however, a network tends to be fragmented into local communities with abundant *strong ties* but few *weak ties* to bridge remote parts [12]. Although searches might be able to move gradually toward targets, necessary "hops" become unavailable. We refer to this phenomenon as the *Clustering Paradox*, in which neither strong clustering nor weak clustering is desirable. The *Clustering Paradox* has received attention in complex network research and requires further scrutiny in a decentralized IR context [15, 14].

3. EXPERIMENTAL SYSTEM

We have developed a multi-agent decentralized search architecture named *TranSeen* for finding relevant information distributed in networked environments. We illustrate the conceptual model in Figure 1 (a) and major components in Figure 1 (b). The *TranSeen* system is an implementation in Java, based on two well-known open-source platforms: 1) JADE, a multi-agent system/middle-ware that complies with the FIPA (the Foundation for Intelligent Physical Agents) specifications [6], and 2) Lucene, a high-performance library for full-text search [13].



(a) Global View (b) Agent Internal View

Figure 1: Conceptual Framework. (a) Global View of agents working together to route a query in the network space. (b) Agent Internal View of how components function within an agent.

Assume that agents, representatives of distributed information systems, reside in an n dimensional (hypersphere) space. An agent's location in the space represents its information topicality. Therefore, finding relevant sources for an information need is to route the query to agents in the *relevant* topical space. To simplify the discussion, assume all

agents can be characterized using a two-dimensional space. Figure 1 (a) visualizes a 2D circle (1-sphere) representation of the information space. Let agent A_u be the system that receives a query from the user whereas agent A_v has the relevant information. The problem becomes how agents in the connected society, without global information, can collectively construct a short path to A_v so that relevant information can be retrieved from there. In Figure 1 (a), the query traverses a search path $A_u \rightarrow A_b \rightarrow A_c \rightarrow A_d \rightarrow A_v$ to reach the target. While agents A_b and A_d help move the query toward the target gradually (through strong ties), agent A_c has a remote connection (weak tie) for the query to "jump."

3.1 Decentralized Search

When an agent receives a query, it first conducts local search operations to retrieve relevant information from its individual document collection. If local results are unsatisfactory, e.g., relevance/similarity scores do not reach a predefined threshold, the agent will contact its neighbors for help. Therefore, there requires a mechanism for matching query representation with potential *good* neighbors – either the neighboring agent is more likely to have relevant information to answer the query directly or more likely to connected with relevant targets. Agents explore their neighborhoods through interactions (e.g., query-based sampling), develop some knowledge about neighbors' topicality and connectivity, and serve as local decision makers in the search process. They are essentially metasearch systems for one another.

3.2 Network Structure & Local Clustering

As discussed earlier, network structure plays an important role in decentralized search. We used a parameter called the clustering exponent α to guide network clustering for decentralized search: the probability p_r of two nodes being connected/linked is proportional to $r^{-\alpha}$, where r is the pairwise topical distance and α the *clustering exponent*.

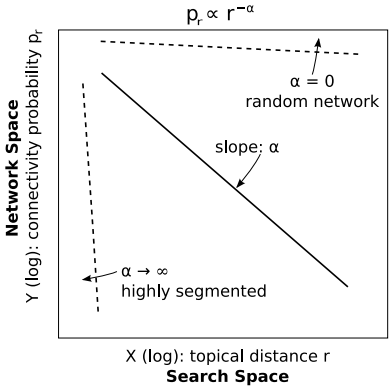
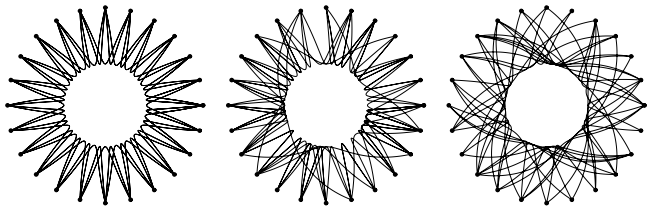


Figure 2: Function of Clustering Exponent α

The *clustering exponent* α, as shown in Figure 2, describes a correlation between the network (topological) space and the search (topical) space [15, 8]. When α is small, connectivity has little dependence on topical closeness – local segments become less visible as the network is built on increased randomness. As shown in Figure 3 (c), the network is a random graph given a uniform connectivity distribution at $\alpha = 0$. When α is large, weak ties (long-distance connections) are rare and strong ties dominate [12]. The network becomes highly segmented. As shown in Figure 3 (a), when $\alpha \rightarrow \infty$, the network is very regular (highly clustered) given that it is extremely unlikely for remote pairs to connect. Given a moderate α value, as shown in Figure 3 (b), the network becomes a narrowly defined *small world*, in which both local and remote connections present.



$\alpha \rightarrow \infty$	$\alpha = 2.5$	$\alpha = 0$
(a) Segmented	(b) Small World	(c) Random

Figure 3: Network Clustering: Impact of Clustering Exponent α.

The *clustering exponent* α influences the emergence of local segments and overall network clustering. In complex network research, it has been shown that only with some particular value of α, search time (i.e., search path length) is optimal and bounded by a poly-logarithmic function of network size [15]. One important aspect of this research is to study the impact of network structure on decentralized IR effectiveness and efficiency.

4. ALGORITHMS

This section elaborates on specific algorithms used in the research. Section 4.1 presents the basic functions for information representation, neighbor representation, and similarity measurement. Section 4.2 describes four search (neighbor selection) algorithms based on neighbor similarity and/or connectivity. Section 4.3 elaborates on the function for agent rewiring (clustering) based on the *clustering exponent* α.

4.1 Basic Functions

*4.1.1 TF*IDF Information Representation*

We used the Vector-Space Model (VSM) for information (document and query) representation [4]. Given that information was highly distributed, a global term space was not assumed. Instead, each agent processed information it individually had and produced a local term space, which was used to represent each information item using the TF*IDF (Term Frequency * Inverse Document Frequency) weighting scheme. An information item was then converted to a numerical vector where a item t was computed by:

$$W(t) = tf(t) \cdot log(\frac{N}{df(t)}) \quad (1)$$

where $tf(t)$ is the frequency of the term t of the term space in the information item, N is the total number of information items (e.g., documents) in an agent's local collection, and $df(t)$ is the number of information items in the set containing the term t of the term space. We refer to $log(\frac{N}{df(t)})$ as IDF. IDF values were computed within the information space of an agent given no global information.

4.1.2 DF*INF Agent Representation

Following a simple federated IR model, we allowed agents to collect document frequency (DF) information from neighors (distributed systems) and to use it to create metadocuments for neighbor representation [19]. Treating each metadocument as a normal document, it was then straightforward to calculate *neighbor frequency* (NF) values of terms, i.e., the number of metadocuments (neighbors) containing a particular term. A metadocument (neighbor) was then represented as a vector where a term t was computed by:

$$W'(t) = df'(t) \cdot log(\frac{N'}{nf'(t)}) \quad (2)$$

where $df'(t)$ is the frequency of the term t of the term space in the metadocument, N' is the total number of an agent's neighbors (metadocuments), and $nf'(t)$ is the number of neighbors containing the term t. We refer to this function as *DF*INF*, or document frequency * inverse neighbor frequency.

4.1.3 Similarity Scoring Function

Based on the *TF*IDF* (or *DF*INF*) values obtained above, pair-wise similarity values can be computed. Given a query q, the similarity score of a document d matching the query was computed by :

$$\sum_{t \in q} tf(t) \cdot idf^2(t) \cdot coord(q,d) \cdot queryNorm(q) \quad (3)$$

where $tf(t)$ is term frequency of term t in document d, idf(t) the inverse document frequency of t, $coord(q,d)$ a coordination factor based on the number of terms shared by q and d, and $queryNorm(q)$ a normalization value for query q given the sum of squared weights of query terms. The function is a variation of the well-known cosine similarity measure. Additional details can be found in [13, 4].

4.2 Search Methods

When an agent found no sufficiently relevant information from its local collection, it forwarded the query to another agent. We proposed the following four neighbor selection strategies, i.e., search methods, to be tested and compared in experiments.

4.2.1 RW: Random Walk

The *Random Walk* (RW) strategy ignores knowledge about neighbors and simply forwards a query to a random neighbor. Without any learning module, *Random Walk* is presumably neither efficient nor effective. Hence, the *Random Walk* served as the search performance lower-bound.

4.2.2 SIM: Similarity-based Search

Let k be the number of neighbors an agent has and $S = [s_1, .., s_k]$ be the vector about neighbors' similarity scores to a query. The *SIM* method sorts the vector and forwards the query to the neighbor with the highest score. We assumed that agents were cooperative – that is, they shared with one another document frequency (DF) values of key terms in their collections, based on which a meta document were created as representative of a neighbor's topical area. A query was then compared with each meta document, represented by *DF*INF* (see Equation 2), to generate the similarity vector S.

4.2.3 DEG: Degree-based Search

In the degree-based strategy, information about neighbors' degrees, i.e., their numbers of neighbors, was known to the current agent. Let $D = [d_1, .., d_k]$ denote degrees of an agent's neighbors. The *DEG* method sorts the D vector and forwards the query to the neighbor with the highest degree, regardless of what a query is about [1].

4.2.4 SimDeg: Similarity*Degree Search

The *SimDeg* method combines information about neighbors' relevance to a query and their degrees. [20] reasoned that a navigation decision relies on the estimate of a neighbor's distance from the target, or the probability that the neighbor links to the target directly, and proposed a measure based on the product of a degree term (d) and a similarity term (s) to approximate the expected distance. Following the same formulation, the *SimDeg* method used a combined measure $SD = [s_1 \cdot d_1, .., s_k \cdot d_k]$ to rank neighbors, given neighbor relevance vector $S = [s_1, .., s_k]$ and neighbor degree vector $D = [d_1, .., d_k]$. A query were forwarded to the neighbor with the highest sd value.

4.3 Agent Rewiring and Network Clustering

We used the clustering exponent α to guide agent self-organization and network clustering. For each agent, the first step was to determine how many neighbors it should have. Given the web collection (Section 5.1) used in this study, we obtained each agent's (i.e., a web domain) indegree based on hyperlink analysis and normalized the degree to a value $d \in [30, 60]$. Once agent u determined its degree d_u, a number of random agents were selected for u such that the total number of random neighbors $d_T \gg d_u$ ($d_T \approx 150$ in this study). Then, the current agent (u) used its metadocument to query each of the d_T neighbors (v) to determine their topical distance r_{uv}. Finally, the following probability function was used by the agent to decide who should remain as neighbors (overlay): $p_{uv} \propto r_{uv}^{-\alpha}$, where α is the *clustering exponent* and r_{uv} pairwise topical distance.

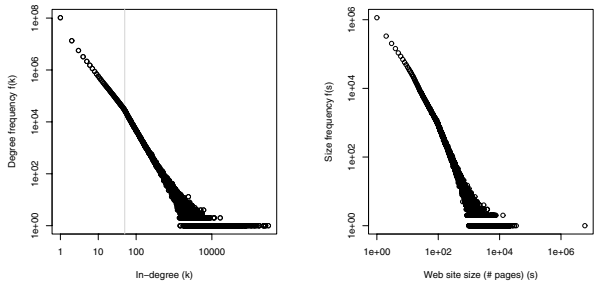
5. EXPERIMENTAL SETUP

5.1 Data Collection

We used the ClueWeb09 Category B collection created by the Language Technologies Institute at CMU for IR experiments, which contains a crawl of 50 million English pages during Jan - Feb 2009. Analysis of the hyperlink graph produced Figures 4 (a) in-degree frequency distribution and (b) Site size (#pages per site) distribution based on $50,221,776$ pages extracted from $2,777,321$ unique domains (treated as sites) (on log/log coordinates).

5.2 Task and Queries

Given the large size of the data collection, it is nearly impossible to manually judge the relevance of every document and to establish a complete relevance base. While previous research on large scale distributed information retrieval mainly relied on similarity thresholds to do automatic relevance judgment, such an approach was rather arbitrary and was biased by the centralized IR system that served as the gold standard [5, 16].



(a) In-degree distribution (b) Site size distribution

Figure 4: ClueWeb09 Category B Statistics

5.2.1 Documents as Queries

Serving diverse users in an open, dynamic environment, implies that some queries are likely to be narrowly defined. We reasoned that relevant information is rare when a query is very specific. In this study, we used documents (web pages with title and content) as queries to simulate decentralized searches. We obtained a set of query documents by sampling documents from the 100 most popular web domains. Removing queries that were too broad or vague resulted in 85 queries.

5.2.2 Task: Exact/Rare Item Search

To make searches more realistic/challenging and automatic evaluation more objective, we considered extreme rarity of relevant documents given very specific information needs. We decided that, given each query, there was only one relevant document among all documents distributed in the network and the task was to find *that exact* document. When a query document was issued to a random system/site in the network, the task involved finding the system who hosted it. The strength of this task is that relevance judgment was more objective provided the relative unambiguity of a "hosting" relationship. The extreme rarity, however, posed a great challenge on the proposed decentralized search methods.

5.3 Evaluation Metrics

This study focused on effectiveness and efficiency of IR operations in networks and scalability of decentralized search. We emphasized the finding of exact/rare information in large distributed environments and proposed the use of the following evaluation metrics.

5.3.1 Effectiveness

Of various evaluation metrics used in TREC and IR, *precision* and *recall* are the basic forms. Whereas precision P measures the fraction of retrieved documents being relevant, recall R evaluates the fraction of relevant documents being retrieved. The harmonic mean of precision and recall, known as F_1, is computed by $F_1 = \frac{2 \cdot P \cdot R}{P+R}$ [4].

5.3.2 Efficiency

In experiments, we measured the search path length L (i.e., the number of agents involved) and actual time τ taken to find relevant information for each query. The average search length $\bar{L}$ of all queries was calculated to measure efficiency. When fewer agents are involved, the entire distributed system is considered to be more efficient. Likewise, average search time $\bar{\tau}$ was calculated to evaluate efficiency.

5.3.3 Scalability

One important objective of this research was to learn how decentralized IR systems can function and scale in very large information network. For scalability, we ran experiments on different network size scales $N \in [10^2, 10^3, 10^4]$. First, we used the 100 most highly linked web domains to form a 100-agent network and conducted experiments on it. Then, we extended the network to $1,000$ and $10,000$ systems/sites for additional experiments. Table 1 shows the total number of documents on each network scale. After experiments, we analyzed the functional relationships of effectiveness and efficiency to network size.

Network Size N	10^2	10^3	10^4
# Documents	$0.5M$	$1.7M$	$4.4M$

Table 1: Network Size and Total # Docs

5.4 Simulation Procedures and Setup

Pseudo code in Algorithm 1 illustrates how different experimental parameters were combined for the simulations. Experiments were conducted on a Linux cluster of 10 PC nodes, each having Dual Intel Xeon e5405 (2.0 Ghz) Quad Core Processors (8 processors), 8 GB fully buffered system memory, and a Fedora 7 installation. The computer nodes were connected internally through a dedicated 1Gb network switch. Agents were distributed among the 80 processors. The Java Runtime Environment version was 1.6.0_07.

Algorithm 1 Simulation Experiments

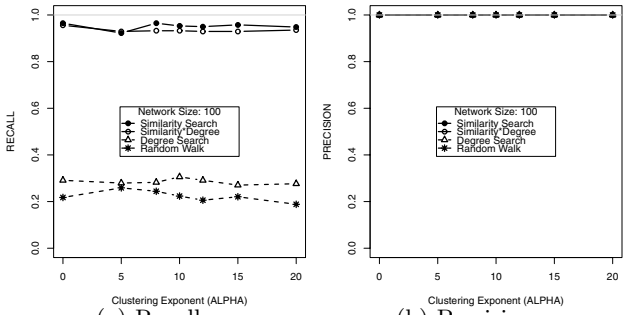
```
1: for each Network Size ∈ [10^2, 10^3, 10^4] do
2:   for each α ∈ [0, .., 15] do
3:     rewire network with the α value
4:     for each Search Method do
5:       for each Query do
6:         assign query to a random agent
7:         repeat
8:           forward query from one another
9:         until relevant agent found OR search path L ≥
           L_max
10:        if sufficient relevant information found then
11:          send the results back
12:        else
13:          send failure message back
14:        end if
15:      end for
16:      measure effectiveness P, R, and F1
17:      measure efficiency τ̄ and L̄
18:    end for
19:  end for
20: end for
```

6. RESULTS

We conducted experiments on networks of 10^2, 10^3, and 10^4 systems. We set the max search length length L_{max} to 20% of network population so that even less effective/efficient methods will be able to persist in searches. Figures 5 and

6 present results on IR effectiveness (recall, precision, and F_1) while Figures 7, 8, and 9 report on efficiency (search path length and time) against different network clustering conditions (guided by clustering exponent α).

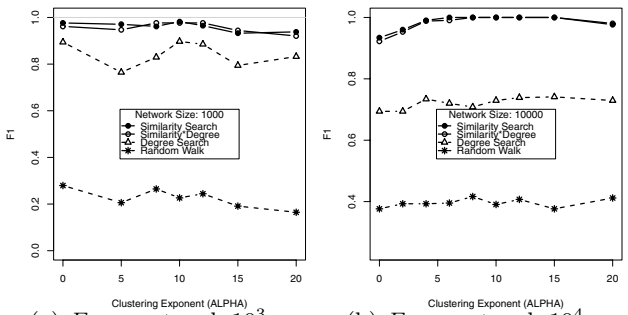
6.1 Effectiveness



(a) Recall (b) Precision

Figure 5: Effectiveness on Network 100

As shown in Figures 5 and 6, the similarity-based search (SIM) and similarity*degree (SimDeg) method performed very well in terms of effectiveness, showing a very large advantage in recall over the degree-based (DEG) and random-walk (RW) methods. When the network was under some proper clustering conditions (e.g., with $\alpha \approx 10$ for network 10,000), the SIM and Sim*Deg methods achieved nearly 100% recall. Precision was 1.0 for all conditions because a document was retrieved only when it exactly matched the query.



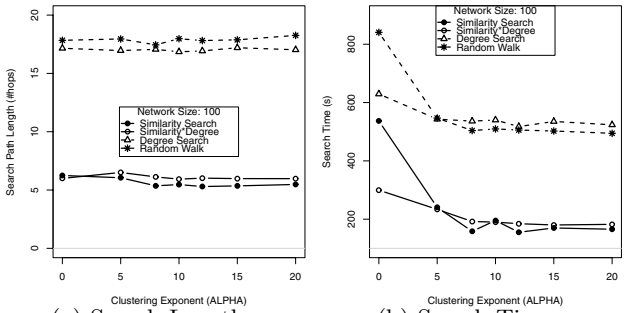
(a) F_1 on network 10^3 (b) F_1 on network 10^4

Figure 6: Effectiveness on Larger Networks

The *DEG* search method, biased toward highly linked (popular) sites in the searches, achieved moderate performance between *SIM* and *RW* methods and had improved performance in larger networks, e.g., a roughly 0.7 recall in the 10,000-system network. Random walk (RW) consistently performed below a 0.4 recall across all network sizes and α conditions.

6.2 Efficiency

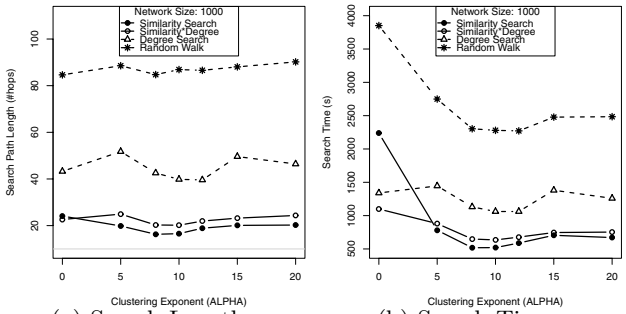
Figures 7, 8, and 9 show very high efficiency of the SIM and SimDeg search methods across the network sizes, especially under stronger clustering conditions. The efficiency gap between the SIM/SimDeg and RW/DEG methods increased dramatically as network size increased. For example, in the 100-node network, while SIM searched roughly



(a) Search Length (b) Search Time

Figure 7: Efficiency on Network 100.

5 hops and 150 milliseconds to find exact match for each query, it took RW more than 15 hops and 400 milliseconds to reach 20% of targets (a 3-time difference in efficiency). When the network size increased to $10,000$, RW search took 50 seconds and traversed about $1,500$ nodes on average to reach a < 0.4 recall whereas SIM search took less than 4 seconds and roughly 110 nodes to achieve a 1.0 recall – a more than 10-time difference in efficiency.



(a) Search Length (b) Search Time

Figure 8: Efficiency on Network 1,000.

Figures 7 - 9 demonstrate that network structure had a great impact on decentralized IR performance, particularly on efficiency in larger networks. While search efficiency (in terms of search path length and search time) under different clustering conditions only differed slightly in the 100-agent network, the difference was much larger in the $10,000$-agent network (Figure 9). For example, the average search path length for the SIM method decreased from 6 to 5 (a 20% difference) when the clustering exponent was changed from 0 (random network) to 10 (strong clustering) in the 100 network. In the $10,000$-agent network, however, the same degree of change in network clustering led to a roughly 200% difference in search efficiency. Statistical tests indicated that SIM search achieved significantly better results with a balanced level of network clustering (i.e., at $\alpha = 10$) than with over- or weak-clustering networks. The significant differences not only appeared in the $10,000$-system network but also in the 100- and 1000-system networks.

6.3 Scalability

For each network size, we identified network clustering conditions under which superior performance was observed (i.e., at $\alpha = 10$) and plotted average search path length (efficiency) against network size in Figure 10. As discussed ear-

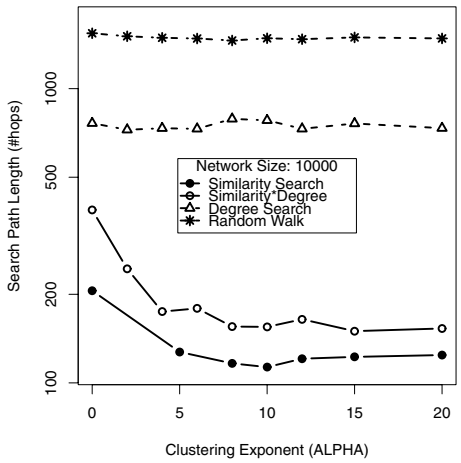


Figure 9: Efficiency on Network $10,000$. Y is log transformed.

lier, SIM and SimDeg searches consistently achieved nearly 1.0 recall and precision across the various network sizes, much better than DEG and RW methods. DEG search tended to perform slightly better in larger networks than in smaller ones. However, as shown in Figure 10, search path length for RW and DEG dramatically increased in larger networks, while the increases for SIM and SimDeg were relatively moderate.

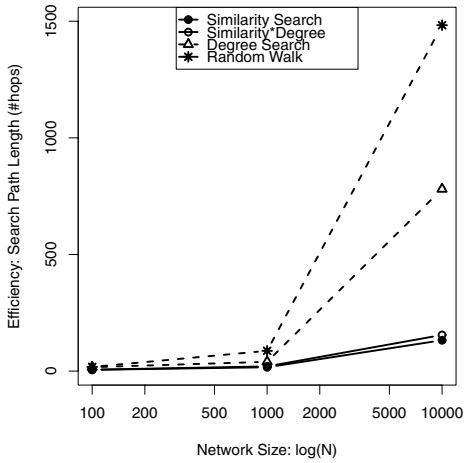


Figure 10: Scalability of all search methods with $\alpha = 10$. X denotes network size and is log transformed.

6.3.1 Scalability of SIM Search

SIM and SimDeg methods appeared to be much more scalable than RW and DEG methods. To better understand the scalability of SIM search and to predict how it could perform in even larger networks (e.g., a network of millions of nodes), we conducted further analysis on the relationship of its efficiency to network size.

Previous research on complex networks suggested that optimal network clustering supports scalable searches, in which search time is a poly-logarithmic function of network size [15]. We relied on a generalized regression model that modeled search path length L (and search time τ) against log-transformed network size N. The model was specified to reach the origin $(0, 0)$ because, when $log(N) = 0$ (i.e., $N = 1$), there is only one node and no effort is needed to search a network. The best fit for search path length L was produced by the model in Table 2, in which $L = 0.0275 \cdot log_{10}^{6}(N)$ with a nearly perfect $R^2 = 0.997$[1].

Search Length: $L \sim 0 + \beta \log_{10}^{6}(N)$, where N is network size.				
	Estimate	Standard Error	t	$Pr(>\|t\|)$
β	0.0275	0.0042	65.73	$< 2E^{-16}$ ***
$R^2 = 0.997$ (adj. 0.9968), $F = 4320$ on 1 and 13 DF				

Table 2: Search path length vs. Network size

The same model was also applied to identify a poly-logarithmic function of search time τ and network size N with a smaller $R^2 = 0.752$. Apparently, search time involves other factors such as machine load fluctuation and is less predictable than search path length.

Overall, the scalability analysis supports search time as a poly-logarithmic function of network size – so that when an information network continues to grow in magnitude, it is still promising to conduct effective IR and search operations within a manageable time limit. Although we found the order of the poly-logarithmic relationship to be roughly 6 in this study, a smaller exponent can be expected when other factors on network structure and search methods can be optimized.

6.3.2 Scalability of Network Clustering

Our search methods relied on local indexes and a structure self-organized by distributed systems in the network. Without global information and centralized control, network clustering was performed locally – distributed systems formed the network structure in terms of their limited opportunities to interact and individual preferences and constraints on building indexes for others. This local mechanism for clustering demonstrated a high level of scalability. As shown in Figure 11, average clustering time τ_c remained relatively constant, < 1 sec, across all network size scales $N \in [10^2, 10^3, 10^4, 10^5]$.

7. CONCLUSION

We conducted experiments on decentralized IR operations on various scales of information networks and analyzed effectiveness, efficiency, and scalability of proposed search methods. Results showed network structure, i.e., how distributed systems connect to one another, is crucial for retrieval performance. With a balanced level of network clustering under local topical guidance, similarity-based search functions (i.e., SIM and SimDeg) were found to perform very efficiently while maintaining a high level of effectiveness even in very large networks. For example, in searches for single unique documents among the 4.4 million documents distributed among $10,000$ agents/systems, selectively involving only 110 agents within 4 seconds yielded 100% precision and 100% recall with a guiding clustering exponent $\alpha = 10$. Under these conditions, more importantly, search time was well

[1]Each of the three X levels has multiple data points. Future work will integrate whether the relationship can be used to predict search efficiency on larger scales.

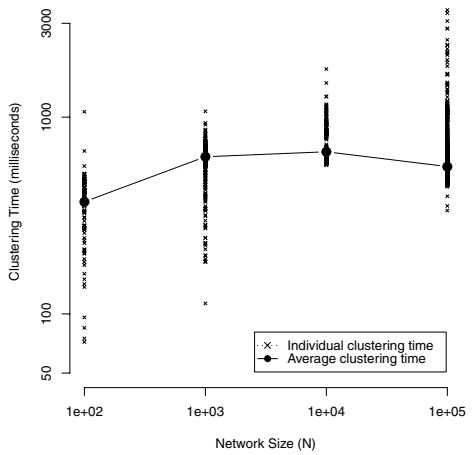


Figure 11: Scalability of Network Clustering

explained by a poly-logarithmic function of network size, suggesting high scalability of the proposed methods.

In addition, the network clustering function that supported very high effectiveness and efficiency of IR operations in large networks is itself scalable. Clustering only involved local self-organization and required no global control – clustering time remained roughly constant across the various network sizes $N \in [10^2, 10^3, 10^4, 10^5]$.

This study provides guidance on how IR operations can function and scale when today's information spaces continue to grow in magnitude. Particularly, we have found that connectivity among distributed systems, based on local network clustering, is crucial to the scalability of decentralized methods. The *clustering paradox* on decentralized search performance appears to have a scaling effect and deserves special attention for IR operations in large scale networks.

Acknowledgment

We appreciate valuable discussions with Gary Marchionini, Munindar P. Singh, Diane Kelly, Jeffrey Pomerantz, José R. Pérez-Agüera, and Simon Spero, and constructive comments from SIGIR'10 reviewers. We thank the NC Translational and Clinical Sciences (TraCS) Institute for support.

8. REFERENCES

[1] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187 – 203, 2005.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[3] R. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges on distributed web retrieval. *ICDE 2007: Data Engineering 2007.*, pages 6–20, April 2007.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman Publishing, 2004.

[5] M. Bawa, G. S. Manku, and P. Raghavan. Sets: search enhanced by topic segmentation. In *SIGIR '03*, pages 306–313, 2003.

[6] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.

[7] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in p2p search engines. In *SIGIR '05*, pages 67–74, 2005.

[8] M. Boguñá, D. Krioukov, and K. C. Claffy. Navigability of complex networks. *Nature Physics*, 5(1):74 –80, 2009.

[9] J. Callan, F. Crestani, and M. Sanderson. SIGIR 2003 workshop on distributed information retrieval. *SIGIR Forum*, 37(2):33–37, 2003.

[10] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. In *Agents and Peer-to-Peer Computing*, pages 1–13, 2005.

[11] C. Doulkeridis, K. Norvag, and M. Vazirgiannis. Peer-to-peer similarity search over widely distributed document collections. In *LSDS-IR '08*, pages 35–42, 2008.

[12] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, May 1973.

[13] E. Hatcher, O. Gospodnetić, , and M. McCandless. *Lucene in Action*. Manning Publications, second edition edition, March 2010.

[14] W. Ke and J. Mostafa. Strong ties vs. weak ties: Studying the clustering paradox for decentralized search. In *LSDS-IR'08*, pages 49–56, Boston, USA, July 23 2009.

[15] J. M. Kleinberg. Navigation in a small world. *Nature*, 406(6798), August 2000.

[16] J. Lu and J. Callan. User modeling for full-text federated search in peer-to-peer networks. In *SIGIR '06*, pages 332–339, 2006.

[17] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7:72–93, 2005.

[18] T. Luu, F. Klemm, I. Podnar, M. Rajman, and K. Aberer. Alvis peers: a scalable full-text peer-to-peer retrieval engine. In *P2PIR '06*, pages 41–48, 2006.

[19] A. L. Powell and J. C. French. Comparing the performance of collection selection algorithms. *ACM Transactions on Information Systems (TOIS)*, 21(4):412–456, October 2003.

[20] O. Simsek and D. Jensen. Navigating networks by using homophily and degree. *Proceedings of the National Academy of Sciences*, 105(35):12758–12762, 2008.

[21] G. Skobeltsyn, T. Luu, I. P. Zarko, M. Rajman, and K. Aberer. Web text retrieval with a p2p query-driven index. In *SIGIR '07*, pages 679–686, 2007.

[22] D. J. Watts, P. S. Dodds, and M. E. J. Newman. Identity and Search in Social Networks. *Science*, 296(5571):1302–1305, 2002.

[23] I. P. Zarko and F. Silvestri. The CIKM 2006 workshop on information retrieval in peer-to-peer networks. *SIGIR Forum*, 41(1):101–103, 2007.